Arne Schuldt

Multiagent Coordination Enabling Autonomous Logistics

SCIENCE AWARD FOR LOGISTICS
2010

Springer

Multiagent Coordination Enabling Autonomous Logistics

Arne Schuldt

Multiagent Coordination Enabling Autonomous Logistics

Springer

Arne Schuldt
Universität Bremen
Technologie-Zentrum Informatik
und Informationstechnik (TZI)
Am Fallturm 1
28359 Bremen
Germany

ISBN 978-3-642-20091-5 e-ISBN 978-3-642-20092-2
DOI 10.1007/978-3-642-20092-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011934024

Cover design: eStudio Calamar S.L.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Jan and Ole

Foreword

This book is the exciting result of an extraordinary effort applied to practice and science: before the author buried himself in the scientific intricacies of Multiagent Systems communication and cooperation research questions when starting his doctoral thesis, he took the opportunity to work as an intern for several months with Tchibo Logistik GmbH. In order to get his arms around what is being done in the important field of Supply Network Management inside a logistics company which must react very fast to the turbulences of real life logistics. He then returned to the scientific work at his alma mater with a very clear understanding what his future research would be aimed at: the development of a consistent and sound concept for the necessary future decentralized decision-making in logistic supply networks. Given the fact that the author received the 2010 Science Award for Logistics by the German Logistics Association (BVL) for this doctoral work where the decisive criterion for the award is the innovative nature and that it relates to practical experience, the success of his resolution appears to be obvious.

There are four separate contributions of this work to Supply Network Management which extend to:

1. a comprehensive overview over the new field of autonomous logistics, focusing on its motivation and enabling technologies,
2. a multiagent-based development concept that specifies acting units and implements also their interaction in autonomous logistics processes,
3. an investigation of the cooperation problems of autonomous logistics entities in order to achieve the logistics objectives imposed by the cargo owners,
4. a case study of the application of the research results achieved so far in real-world logistics processes in order to derive and demonstrate the potential and the limitations of autonomous logistics.

These contributions are worked out in ten chapters in a sound scientific way underpinned by the solid requirements of the real logistic world and supported by new theoretical results. Firstly, Supply Network Management is considered which defines the basic logistic functions, the acting agents and the inherent challenges. Then the interdependencies of the service providers and the involved processes are dealt with before a theoretical basis is set by the discussion of the novel decentralised decision-making approaches including their potential and the challenges involved. This field is enabled by the latest technologies, such as RFID, GPS, microelectronic sensors, wireless communication, and the data processing facilities combining the available information for a well-founded and robust decision-making.

Having defined the logistic requirements for decentralized decision-making in logistics Supply Networks, the author now turns to implementation questions where he identifies Multiagent Systems as a well-suited vehicle to carry the load of an appropriate implementation solution. These Multiagent Systems offer the necessary approaches such as interaction, cooperation, and even team formation in order to cover the logistic requirements. This potential is discussed further on by explicitly mapping software agents to logistics providers of transport, handling, storage, and picking services and their organisational structures which typically are dynamical team structures. Therefore, the automated team formation is one of the central requirements at the concept level and is worked out very profoundly in this book, also in respect to logistics Supply Networks. Having established teams, the next step is team action, namely how to organize the teams of logistic services to achieve the required performance.

It would have been a purely theoretical work with limited practical relevance if the author had not also implemented the identified logistics functionality based on the FIPA Multiagent Systems and the PlaSMA middleware technology in order to demonstrate the feasibility and the economical value of his results. The proposed generic implementation of a Supply Network even offers the opportunity to simulate the logistics processes, provided an appropriate time model with the proper synchronisation mechanisms had been chosen. This is an additional advantage of the proposed implementation: only small changes of the simulation code are required in order to switch to a live application.

The final section of this book is concerned with a case study applying the achieved results to real-life structures, processes, and figures. The potential for cooperation and the required effort can be shown analytically which exhibits also the boundaries of a successful application of the principle of autonomy. In addition, applying the autonomy principle using a Multiagent System simulation to the real Tchibo logistics processes it can be successfully demonstrated that the transition of these actually centralized logistics functions to autonomous logistics is also of advantage economically for the onward carriage of containers and their distribution to warehouses.

This book lays a foundation for the analytical and simulated treatment of processes in Supply Networks. It offers also an example for an appropriate high-level (and thus by itself economical) implementation of the achieved results, and it even proves that the application of distributed decision-making in logistics contexts pays in the real world. There are not many works spanning this field in a most complete way. Therefore I wish this book to be read by many logistics decision-makers and I do hope that it will initiate a lively discussion on the virtues of distributed decision-making for real-world logistics processes.

Bremen, *Dr. Otthein Herzog*
January 2011

Acknowledgements

First of all, I would like to thank my advisor Prof. Dr. Otthein Herzog for motivating me to choose a challenging topic with real-world relevance and for supporting me in this research. Prof. Dr. Ingo J. Timm helped to improve this thesis by providing valuable comments from the perspective of distributed artificial intelligence. I would also like to thank my mentor PD Dr. Björn Gottfried for his collaboration and for supporting me with valuable advice.

Jens Engelmann from Tchibo gave me the possibility to investigate the field of autonomous logistics based on a real-world application. Michael Garske shared his expert knowledge about the processes that are investigated in this thesis, therewith laying the foundation for the implementation with autonomous logistics entities.

This dissertation was conducted within the Artificial Intelligence research group of the Centre for Computing and Communication Technologies (TZI), University of Bremen. I am very grateful that Deutsche Telekom AG, Freie Hansestadt Bremen, and the AI research group funded my scholarship at the International Graduate School for Dynamics in Logistics (IGS). I would like to thank Dr. Renate Klempien-Hinrichs, Dr. Ingrid Rügge, and all other colleagues from the IGS for many interdisciplinary discussions on logistics.

Many friends and colleagues have contributed to finishing this thesis. First of all, I would like to thank Ole Osterhagen and Jan D. Gehrke who accompanied me in my diploma and Ph. D. studies, respectively. By sharing their insights, they shaped my understanding of computer science and artificial intelligence. Important foundations for my thesis have also been laid by Sven Werner who introduced me into agent programming by sharing his experiences. Not less important was the support by Torsten Pickert, PD Dr. Björn Gottfried, Jan Ole Berndt, and my father Werner Schuldt who thoroughly proof-read this thesis. Florian Pantke and Dr. Hartmut Messerschmidt helped me to improve the formalisation and complexity analysis, respectively. Finally, I would like to thank all other colleagues from the AI research group for numerous discussions on this thesis.

Contents

Part I Logistics Requirements

Part II Multiagent-Based Approach

Part III Application and Evaluation

Part I

Logistics Requirements

Chapter 1
Introduction

Transport, material flow, and logistics have a long history (Jünemann, 1989, pp. 3–10). Many technical inventions enabling logistics date back up to several thousand years (Gudehus, 2007, p. 6). Transport and logistics lay the foundation for trade if producers and consumers are not located at the same place. Usually, ships have a greater capacity than land vehicles. Hence, it is an age-long practice to employ ships for transporting goods over long distances (Levinson, 2006, p. 16). However, seaborne transport does not only consume time for carrying goods. Additionally, time for loading and unloading the cargo must be taken into consideration (Levinson, 2006, p. 20). For general cargo, handling was performed manually on a piecewise basis for thousands of years (Levinson, 2006, pp. 16–17). It became, however, increasingly inefficient as demurrage correlates with ship size.

A revolution was the large-scale application of standardised shipping containers initiated by Malcom Purcell McLean (1913–2001) and his later Sea-Land Corporation. On April 26, 1956, the company loaded the first shipping containers onto the Ideal-X, a vessel converted for container transport (Levinson, 2006, p. 1). For the first years, shipping containers were mainly employed to increase efficiency of domestic transport. The North Atlantic connection to Europe was established ten years later (Levinson, 2006, p. 202). In the evening of May 5, 1966, the container vessel Fairland arrived in the Bremen Überseehafen (Schwerdtfeger, Zabern & Pölking-Eiken, 1991, pp. 17–18). The following day, the first shipping containers in Germany were unloaded. Another two years later, the vessel American Lancer was the first pure container vessel to be discharged at the Burchardkai in the port of Hamburg on May 31, 1968 (Pasdzior & Domizlaff, 2008, p. 75).

The increasing diffusion of shipping containers changed the shape of the world economy (Levinson, 2006, pp. 1–4). The standardised and highly automated handling of shipping containers even over multiple modes of transport significantly decreases transport costs. The increasingly insignificance of transport prices allows purchasing and distributing products worldwide.

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics*,
DOI 10.1007/978-3-642-20092-2_1, © Springer-Verlag Berlin Heidelberg 2011

Therewith, containerisation lays an important foundation for mass production and thus a further decrease in production costs.

In 2007, the ports of Bremen and Bremerhaven had an annual turnover of 4,892,087 TEU (Senator für Wirtschaft und Häfen, 2008, pp. 10–11). The abbreviation TEU stands for twenty feet equivalent units, a measure for counting shipping containers. At the same time, the port of Hamburg handled even 9,917,180 TEU (Statistisches Amt für Hamburg und Schleswig-Holstein, 2009, p. 10). In total, 65.0% of the general cargo and 37.1% of the total cargo handled by maritime traffic in Germany in 2007 were transported by container (Winter, 2008, p. 586). The containerisation lead to a high degree of automation in the execution of logistics. At the same time, the phenomenon referred to as globalisation, which was partly made possible by containerisation, leads to increasingly complex and dynamic processes. Apart from information systems that only support human dispatchers by providing relevant information, however, no comparable automation took place for controlling supply networks. Complexity, dynamics, and distribution of logistics processes pose an increasing challenge for the efficiency of control paradigms that incorporate all parameters centrally.

Today, about 50 years after the first journey of the Ideal-X, another revolution emerges: shipping containers become intelligent. This is enabled by an ongoing miniaturisation of the required technologies (Hellenschmidt & Wichert, 2007, p. 94), namely identification, localisation, sensors, communication, and data processing. These technologies are rather young compared to other inventions enabling logistics. Devices of the new technologies can be attached to logistics objects at low costs, thereby enhancing them with intelligent decision-making. The idea behind this intended paradigm shift towards autonomous control is that logistics objects like shipping containers can themselves achieve logistics objectives defined by the cargo owners. Decentralised decision-making is expected to decrease the computational complexity and help coping with the dynamics of logistics processes locally (Freitag, Herzog & Scholz-Reiter, 2004, pp. 23–24).

Section 1.1 lists relevant research questions regarding autonomous control in logistics that are addressed in this thesis. Subsequently, Section 1.2 presents the context of this research project and its particular contributions. An outline of how the remainder of this thesis approaches the research questions addressed is given in Section 1.3.

1.1 Research Questions

This research project aims at investigating autonomous control in logistics with respect to four major research questions. These guiding questions can be summarised as follows:

1. What constitutes autonomous control in logistics?
2. How can autonomous control in logistics be operationalised?
3. How important is cooperation for autonomous control in logistics?
4. How can autonomous control be applied to actual logistics processes?

To motivate the research conducted, it is important to start with identifying the limitations of conventional approaches to holistic supply network management. Based on this analysis, the subsequent question is how autonomous control in logistics can overcome these limitations by decentralised decision-making. A relevant aspect is the integration of all primary logistics functions, namely transport, handling, storage, and picking. Furthermore, it is particularly important to find an appropriate method for the implementation of decentralised decision-making in logistics.

Based on these foundations, the interaction of autonomous logistics entities must be operationalised. Thereby, one must answer the question what is an appropriate level for autonomous control in logistics. Possible options are components, articles, sales units, cardboard boxes, pallets, or shipping containers. Furthermore, the question of the potential for cooperation must be addressed. That is, whether and when autonomous logistics entities should act individually or in teams. And, if cooperation is asked for, how it can be implemented. On the one hand, this refers to the process of team formation. On the other hand, this also includes coordinating the joint action of teams of autonomous logistics entities.

Based on the concept developed, it is then important to investigate its applicability to real-world logistics processes. To this end, it is firstly necessary to identify a logistics process for which the efficiency of centralised control is limited. Secondly, the question must be answered how such a process can be autonomously controlled in order to reduce the computational complexity. This includes investigating the potential and the limitations for cooperation in autonomous control. Finally, the question is how the paradigm of autonomous logistics relates to established approaches from the field of operational research.

1.2 Research Context and Contributions

The thesis at hand is integrated into the research context of the Bremen Research Cluster for Dynamics in Logistics (Figure 1.1), in short LogDynamics, at the University of Bremen. Researchers of the disciplines of physics and electrical engineering, mathematics and computer science, production engineering, as well as business studies and economics contribute to the research cluster. LogDynamics approaches the field of dynamics in logistics in research, education, and application. The Collaborative Research Centre on Autonomous Logistics (SFB 637) is funded by the German Research Foundation (DFG). More than 60 researchers from the participating disciplines

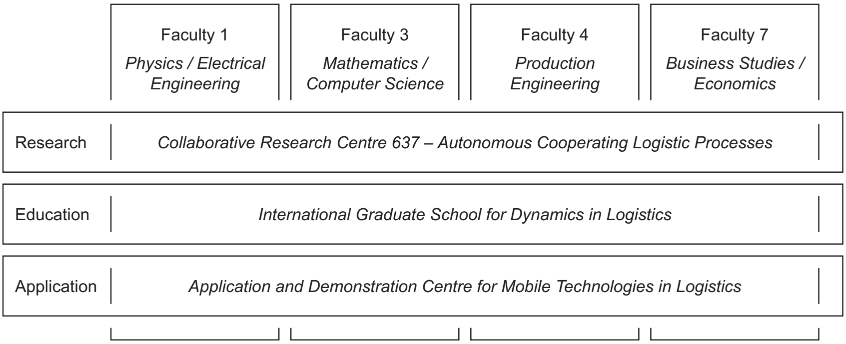

Fig. 1.1 The interdisciplinary Bremen Research Cluster for Dynamics in Logistics brings together researchers from different disciplines. The field of dynamics in logistics is approached in research, education, and application.

investigate the paradigm shift towards autonomous logistics and its limitations. The International Graduate School for Dynamics in Logistics offers a programme of structured doctoral studies on dynamics in logistics processes and networks. Its scholarships are funded by the State of Bremen and the industry, respectively. Finally, the LogDynamics Lab is a demonstration and application centre for mobile technologies in logistics.

The research presented in this thesis has been conducted within both the International Graduate School for Dynamics in Logistics and the Collaborative Research Centre 637. The thesis approaches the field of dynamics in logistics from the perspective of computer science, particularly Distributed Artificial Intelligence. Following the research questions addressed (Section 1.1), the particular contribution of this project is fourfold:

1. It gives a comprehensive overview on the new field of autonomous logistics, thereby focusing on its motivation and enabling technologies.
2. It develops a multiagent-based concept that specifies participants and operationalises their interaction in autonomous logistics processes.
3. It investigates how autonomous logistics entities can cooperatively achieve the logistics objectives imposed by the cargo owners.
4. It studies the application in a real-world logistics process, thereby deriving the potential and limitations of autonomous logistics.

1.3 Thesis Structure

To answer the research questions addressed (Section 1.1), this thesis is divided into three parts. The first part investigates requirements from logistics. The second part approaches these requirements with methods derived from Distributed Artificial Intelligence. The third part applies and evaluates these

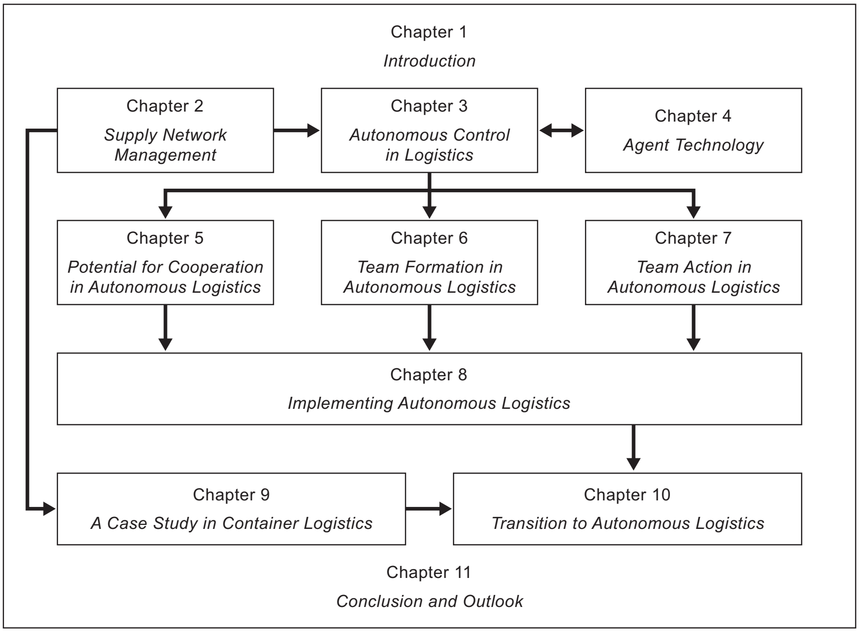

Fig. 1.2 Overview of the thesis structure. Major interrelationships between chapters are indicated by arrows and explained in the text.

findings in a real-world industrial application. The structure of the chapters is depicted in Figure 1.2. Their interrelationship is as follows.

Chapter 2 — Supply Network Management. This chapter deals with relevant foundations of supply network management. It starts with examining the objectives of logistics as well as the primary functions to achieve these objectives. A particular focus is on current and future trends in logistics. Conventionally, centralised control is applied for individual logistics functions. The efficiency of centralised control, however, is limited in complex supply networks with many participants and parameters. This chapter describes challenges for conventional centralised control in logistics. The investigation shows that the centralised perspective of former approaches has particular limitations when coping with logistics processes in increasingly complex, dynamic, and distributed supply networks.

Chapter 3 — Autonomous Control in Logistics. Taking the limitations of conventional centralised control as a starting point, this chapter presents the paradigm of autonomous control in logistics. Autonomous logistics addresses the complexity, the dynamics, and the distribution of supply networks by delegating decision-making to the logistics entities themselves. Decentralised control is expected to be capable of reacting flexibly and thus robustly on arising demands. Technologies enabling autonomous control in logistics include identification, localisation, and sensor technology, as well as communication networks. However, these technologies could also be applied in order to improve conventional logistics processes. Hence, the capability of data processing turns out to be the crucial technology for implementing control in decentralised supply network management.

Chapter 4 — Agent Technology. Following the insight from the preceding chapter, this chapter investigates how to implement local data processing. For this purpose, intelligent software agents turn out to be an appropriate means. They satisfy the requirements for autonomous logistics by representing logistics objects and acting on their behalf. Multiagent systems enable agents to interact with each other. There are many ways in which software agents can be implemented. Interoperability between software agents demands specifying interaction standards in advance. Multiagent organisations go even one step further by manifesting long-term cooperations of software agents in organisational structures.

Chapter 5 — Potential for Cooperation in Autonomous Logistics. The specification for implementing autonomous control in logistics with multiagent technology is approached in three steps. Each step will be thoroughly dealt with in a separate chapter. This chapter starts with investigating the potential for cooperation. To this end, it examines the structure of autonomous logistics networks, particularly the different tasks of the participants. On the one hand, general cargo units as service consumers request logistics services. On the other hand, logistics service providers offer their services regarding transport, handling, storage, and picking. Apart from specifying the participants, also the potential for cooperation is examined. It turns out that individual logistics entities can rarely achieve their objectives on their own. Instead, it is often beneficial to form teams.

Chapter 6 — Team Formation in Autonomous Logistics. It is an important finding of the preceding chapter that decentralised control of logistics processes requires cooperation. Firstly, cooperation helps logistics entities achieving their objectives. Secondly, cooperation is an important foundation for process optimisation. Finally, joint action significantly reduces the interaction complexity of autonomous logistics processes. Hence, this chapter introduces three interaction protocols for team formation of autonomous logistics entities. The interaction protocols differ in their properties. Therefore, this chapter provides a thorough investigation that supports agent developers in choosing the right protocol for a specific application.

Chapter 7 — Team Action in Autonomous Logistics. Based on the interaction protocols for team formation in autonomous logistics, this chapter turns the attention to joint actions of autonomous logistics entities. Firstly, it investigates how consumers and providers can individually negotiate about logistics services. This foundation is extended to inter-agent collaboration, i. e., joint allocation of logistics resources by teams of service consumers. Usually, it is necessary to combine multiple primary logistics functions in order to transform logistics objects in accordance with their objectives. To this end, also the intra-agent coordination of logistics functions is an important aspect addressed by this chapter. Finally, the new approach is compared to conventional centralised control.

Chapter 8 — Implementing Autonomous Logistics. The preceding three chapters specify the implementation of autonomous logistics with multiagent systems. Based on that foundation, this chapter presents the actual implementation. It discusses both the underlying multiagent platform and the actual software agents including their behaviour regarding team formation and team action. The overall behaviour of multiagent systems can often not be predicted at design time. However, evaluating the outcome of the new method directly in the real world might compromise the actual processes. Therefore, it is necessary to evaluate logistics strategies before implementing them in reality. This can be accomplished by applying multiagent-based simulation, another focus of this chapter.

Chapter 9 — A Case Study in Container Logistics. This chapter investigates real-world industrial logistics processes which are hitherto centrally organised. For this purpose, the procurement logistics processes of one of the major German retailers of consumer products are examined. The case study covers two aspects. On the one hand, it provides a detailed process analysis of the transport of shipping containers from East Asia into the warehouses located in Central Europe. On the other hand, the case study also focuses on the logistics entities participating in the processes as well as their parameters. Apart from shipping containers, it covers ports of discharge, warehouses, and transport relations. This case study is the foundation for evaluating the approach developed in this research.

Chapter 10 — Transition to Autonomous Logistics. Based on the logistics processes surveyed in the case study, this chapter investigates their transition to autonomous control. The chapter is divided into three sections that correspond to the three chapters specifying the implementation of autonomous control in logistics. The first two sections examine autonomous logistics analytically, the last one applies multiagent-based simulation. Firstly, the potential for cooperation is approached analytically. Secondly, also the effort for cooperation is examined analytically. Based on these foundations, insights regarding an adequate degree as well as limitations for autonomous control are derived. The last section examines how the processes from the case study can be autonomously controlled by the logistics entities themselves. In particular, the effectiveness and the efficiency of autonomous control in logistics are investigated by means of multiagent-based simulation.

Chapter 11 — Conclusion and Outlook. Finally, this chapter gives a concluding summary of the research presented. It particularly closes the circle to the introduction by returning to the initial research question.

Apart from answering the research questions addressed by the thesis at hand, this final chapter elaborates on directions for future research and thus new research questions in the field of autonomous control in logistics.

References

Freitag, M., Herzog, O. & Scholz-Reiter, B. (2004). Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen. *Industrie Management*, *20*(1), 23–27.

Gudehus, T. (2007). *Logistik 1 — Grundlagen, Verfahren und Strategien.* Heidelberg, Germany: Springer-Verlag.

Hellenschmidt, M. & Wichert, R. (2007). Selbstorganisation: Dinge in eigenverantwortlicher Kooperation – eine Systemanalyse. In H.-J. Bullinger & M. ten Hompel (Eds.), *Internet der Dinge* (pp. 91–105). Heidelberg, Germany: Springer-Verlag.

Jünemann, R. (1989). *Materialfluß und Logistik: Systemstechnische Grundlagen mit Praxisbeispielen.* Heidelberg, Germany: Springer-Verlag.

Levinson, M. (2006). *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger.* Princeton, NJ, USA: Princeton University Press.

Pasdzior, M. & Domizlaff, S. (2008). *Der Hamburger Hafen.* Hamburg, Germany: Edition Maritim.

Schwerdtfeger, H., Zabern, T. von & Pölking-Eiken, H.-J. (1991). *Bremen/Bremerhaven Container Story.* Bremen, Germany: Steintor.

Senator für Wirtschaft und Häfen. (2008). *Die bremischen Häfen* (Hafenspiegel 2008). Bremen, Germany: Freie Hansestadt Bremen.

Statistisches Amt für Hamburg und Schleswig-Holstein. (2009). *Schifffahrt und Außenhandel Hamburgs 1970 bis 2008* (Sonderbericht No. G III / H II - j/08 H). Hamburg, Germany.

Winter, H. (2008). Seeverkehr 2007. *Wirtschaft und Statistik*(7), 581–591.

Chapter 2
Supply Network Management

The objective of logistics is to provide the right quantity of the right objects in the right place at the right time in the right quality for the right price (Jünemann, 1989, p. 18). Its purpose is to provide manufacturing facilities with raw materials and to supply customers with products. Jünemann explicitly points out that minimising costs cannot be the only goal because the other mentioned goals also play an important role in satisfying elaborate logistics demands. Fleischmann (2008, p. 4) explains that the common understanding of logistics is focused on material. Several operations can be applied to transform material in order to achieve the logistics objectives listed above. The applicable transformation operations include bridging of time and space (Figure 2.1). Logistics is generally considered as planning and controlling processes rather than executing the respective operations (Fleischmann, 2008, p. 4). The research at hand particularly addresses the aspect of process control in supply chain management. Although not being an object of logistics, information plays an important role in controlling logistics processes effectively. Computer science provides the means to handle and synchronise information flows with the actual material flows (ten Hompel, Schmidt, Nagel & Jünemann, 2007, p. 1).

The aim of macro logistics is to provide an optimal infrastructure for logistics, e. g., traffic networks (Gudehus, 2007b, p. 577). By contrast, micro logistics deals with planning and controlling corporate logistics, e. g., supply chain management. A finer categorisation of micro logistics can be defined based on the steps of supply chain management (Figure 2.2). Generally, the supply chain comprises procurement logistics, production logistics, distribution logistics, and reverse logistics (Martin, 2006, p. 3). Fleischmann (2008, p. 5) explains that this distinction is helpful although it contradicts a holistic perspective on logistics at first glance. The distinction can be motivated by the fact that the structures of the respective subsystems differ significantly. Procurement logistics deals with supplying production processes with raw materials and semi-finished products. The cargo has to be collected from suppliers in large quantities. Production logistics is then concerned with delivering the

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics,*
DOI 10.1007/978-3-642-20092-2_2,

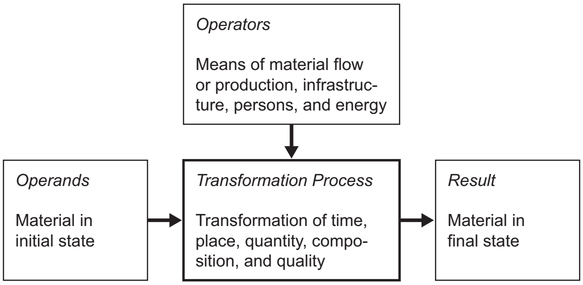

Fig. 2.1 Transformation processes in logistics. Different operators can be applied to transform objects in order to achieve logistics objectives (adapted from ten Hompel et al., 2007, p. 3).

material to workshops and workplaces within the company. Subsequent to the production step, semi-finished and finished products are distributed to consumers. Finally, disposal of waste from all steps of the supply chain is addressed by reverse logistics. Today, such prototypical linear supply chains that directly link suppliers and their customers disappeared to a large extent. Instead, relationships have evolved into more complex supply networks with a considerably high number of participants (Fleischmann, 2008, p. 5). To embrace this development, the research at hand uses the term supply network management rather than supply chain management.

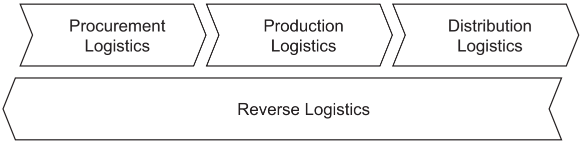

Fig. 2.2 The supply chain starts with procuring material from suppliers. Then, production logistics supports the creation of new products. Finally, products are delivered to customers by distribution logistics. Disposal of waste is addressed by reverse logistics.

The aim of this chapter is to present foundations of supply network management and to derive limitations for conventional centralised control. Therefore, Section 2.1 introduces the primary logistics functions that can be applied as operators to transform goods. Subsequently, Section 2.2 examines their combination to supply networks with a focus on logistics service providers and both current and future trends. Based on these foundations, Section 2.3 discusses challenges for conventional control caused by the complexity, the dynamics, and the distribution inherent in logistics processes.

2.1 Primary Logistics Functions

The primary objective of logistics is to ensure an optimal flow of cargo. An important question is which primary logistics functions can be applied to achieve this goal. But before turning to that question it makes sense to determine which cargo actually needs to be handled. Whenever the term logistics is used, the main focus is on material rather than on persons and information (Fleischmann, 2008, p. 4). Nevertheless, the term material still covers a broad field of potential cargo. Aberle (2003, p. 1) distinguishes between raw materials, semi-finished products, and finished products. This terminology is closely linked with the supply chain steps in Figure 2.2. Generally, procurement logistics deals with raw materials and semi-finished products. Subsequent to the production step, semi-finished and finished products are handled by distribution logistics. In the special case of trading companies without own production facilities, all cargo handled pertains to finished products. However, classifying cargo based on its appearance along the supply chain reveals scarcely anything about its physical properties.

Another classification follows the state of matter, which may be solid, liquid, or gas (Martin, 2006, p. 59). Solid cargo can be further distinguished into bulk cargo and general cargo. The term bulk cargo refers to material that is lumpy, granular, or dusty. Usually, bulk cargo is capable of flowing and changes its shape during transport. Martin (2006, p. 59) gives the following examples: ore, coal, waste, sand, cement, gravel, grain, and coffee. By contrast, general cargo does not alter its shape during transport. Pieces of general cargo can be handled as individual units (Martin, 2006, p. 62). The size of general cargo pieces may range from small to large. The size of such cargo pieces is usually less-than-carload. For the sake of efficiency it is possible to consolidate multiple pieces into larger units (Gudehus, 2007a, pp. 426). Fluids and gases can be transported through pipelines (Aberle, 2003, p. 18). However, Martin (2006, p. 59) emphasises that transporting fluids and gases through piping systems is rather in the field of process engineering than transport technology. In this research, the main focus is on general cargo logistics. This restriction, however, is relativised by the fact that also bulk cargo, fluids, and gases can be handled as general cargo by filling them into appropriate containers (Aßmann, 2008, p. 613).

To recapitulate, the initial question was which functions can be applied to achieve the logistics objectives. In this context, Gudehus (2007a, pp. 7–8) identifies the following primary logistics functions (Figure 2.3):

1. Transport
2. Handling
3. Storage
4. Picking

Fleischmann (2008, p. 3) shares this conclusion. Additionally, he points out the importance of packing as an auxiliary function in logistics processes

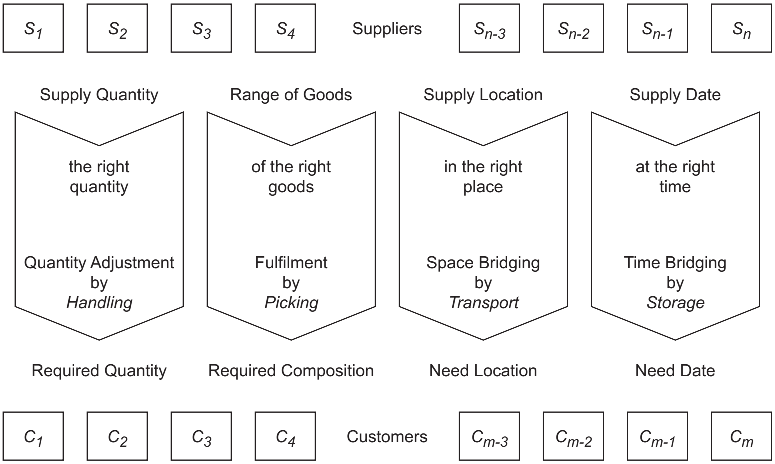

Fig. 2.3 The primary functions to provide the right quantity of the right goods in the right place at the right time: handling, picking, transport, and storage (adapted from Gudehus, 2007a, p. 8).

(Fleischmann, 2008, pp. 6–7). However, he mentions that packing is often integrated into picking processes. Therefore, it shall not be regarded separately here, but together with picking. Furthermore, information and communication technology also is necessary in order to control sophisticated logistics processes (Fleischmann, 2008, p. 7). The logistics processes controlled by information and communication technology are composed of the primary logistics functions identified before. Despite of its importance, information and communication technology is therefore still regarded as an auxiliary function here. The following sections discuss the particular task and characteristics of each primary function.

2.1.1 Transport

The purpose of transport is to bridge the spatial gap between sources and sinks. It is necessary whenever supply and demand are spatially distributed (Arnold, 2008, p. 727). Sources of transport can be stocks for material, semi-finished and finished products, as well as production facilities. Possible sinks are warehouses, shops and outlets, as well as end customers (Gudehus, 2007a, p. 7). The sink of each transport can, of course, be the source of another and vice versa. This is particularly the case for warehouses and logistics centres. Furthermore, factories, trading companies, and consumers are also sources for empties and waste which have to be disposed of by reverse logistics (Gudehus, 2007a, p. 7). Spatial distances bridged by transport range from short to long.

Distances in intralogistics are comparatively short: material is transported between workshops, or even simply between different workplaces. For this kind of intra-company transport, continuous and discontinuous conveyor systems can be applied (Martin, 2006, p. 97). Continuous conveyor systems are advantageous due to their high efficiency given a specified layout and standardised load carrying devices (ten Hompel et al., 2007, p. 122). Common examples are roller tracks, belt and circular conveyors. By contrast, the advantage of discontinuous conveyor systems is their high flexibility, which makes them adaptively applicable for different operations (ten Hompel et al., 2007, p. 122). Examples for discontinuous conveyors are forklifts, warehouse storage and retrieval systems, cranes, and electrical hanging conveyor systems.

Intralogistics is generally a part of production logistics (Figure 2.2). However, transport is not limited to single sites, but can also be conducted between different sites of one company and between sites of different companies. Particularly in procurement and distribution logistics, distances bridged are significantly longer, often on a national or even international scale. Consider a European company that has a supplier producing in East Asia. Then, the distance between continents has to be bridged in order to deliver goods from the vendor to his customer. Means of transport for external transport comprise train, road traffic, inland water navigation, air transport, and ocean navigation (Aberle, 2003, p. 18). Air transport is suitable in order to deliver cargo quickly over long distances. However, the freight capacity of airlines is generally limited and therefore expensive. Hence, air transport of cargo plays only a tangential role (Aberle, 2003, p. 20). By contrast, ocean navigation has significantly more capacity (Aberle, 2003, p. 21) and is therefore capable of transporting even large quantities of cargo at reasonable costs. Its drawback is the longer duration of transport. Transporting shipping containers by container vessel from East Asia to Europe takes several weeks. Likewise, train and inland water navigation provide the means to transport large amounts of cargo at once (Aberle, 2003, pp. 19–20). But their drawback is that they are restricted to certain routes, namely the railway system and inland waterways, respectively. That is, they cannot directly connect to every company site (Heidmeier & Siegmann, 2008, p. 744). By contrast, trucks are more flexible as they are capable of providing door-to-door services. This is the reason why the transport modal split shifts to the detriment of train transport (Vahrenkamp, 2007, pp. 304–305). From a macro logistics perspective, it is desirable to employ trains instead of trucks because they are less expensive and use resources better. From a micro logistics perspective, an advantage of trains is that they are not exposed to overcrowded roads. Intermodal transport addresses this issue by employing multiple means of transport, thereby combining strengths of the transport modes involved (Vahrenkamp, 2007, p. 309).

2.1.2 Handling

Transferring cargo from one means of transport to another requires handling operations. Thus, handling is the next primary logistics function to be examined. Apart from changing between different means of transport, handling also refers to all activities involved in loading and unloading cargo (ten Hompel et al., 2007, p. 289). Handling is sometimes regarded as subtopic of transport. But ten Hompel et al. (2007, p. 289) mention that, in the context of transport, handling, and storage processes, it is generally considered as an equal function. The objective is to implement handling as efficient as possible. This has been achieved by introducing standard transport containers for intermodal transport (Fleischmann, 2008, p. 7).

The containers are standardised in size and handling mechanism. This abstraction enables handling them regardless of their actual content. Note, however, that there are certain restrictions which may not be exceeded, e. g., the weight of the container. Containers for special purposes exist (Frindik, 2008, p. 738). As an example, refrigerated containers are employed in order to transport perishable fruits. Special tank containers can carry liquids or gas. In this case, the round tank is surrounded by a steel framework having the standard shape of containers. Due to their standardised shape, containers cannot only be handled easier. Additionally, it is possible to stack them on container vessels or at container terminals in order to save space. Furthermore, the same standard size holds for all means of transport (Vahrenkamp, 2007, p. 322). This allows switching cargo between different means of transport without the necessity for re-packing. Shipping containers arriving by container vessel from East Asia can be loaded directly onto trucks, trains, or barges. Prior to containerisation the idle period of cargo vessels was significantly longer. Decreasing the idle period also decreases the respective costs (Levinson, 2006, p. 48). Today, shipping containers handle most of the intercontinental transport of general cargo (Vahrenkamp, 2007, pp. 322–324). Vahrenkamp explicates that containerisation is thus the foundation for global production processes. Containers can be handled by portal container cranes with load spreaders which bridge over container vessels, trains, and trucks; further handling means employed include straddle carriers and heavy trucks (Frindik, 2008, p. 739).

Handling is not restricted to switching cargo between different means of transport. Additionally, it is the interface between external and intra-company material flow. Warehouses have specific loading zones at which cargo is unloaded from means of transport and vice versa. Often, the loading zone is further divided into a receiving area and a shipping area. To prevent tractor units from wasting time while the cargo is being unloaded, shipping containers and swap bodies can be employed (Heidmeier & Siegmann, 2008, p. 747). However, neither shipping containers nor swap bodies are applied to store cargo in warehouses. It is thus necessary to change the load carrier,

i. e., commonly by re-packing onto pallets. Generally, this requires adjusting quantities, e. g., by concentrating or distributing the cargo.

2.1.3 Storage

Storage denotes those steps of the supply chain where materials handled rest (ten Hompel et al., 2007, p. 49). In general, it is not desired that material does not move because every standstill is accompanied with diverse costs (Schmidt & Schneider, 2008, p. 374). Firstly, the capital that corresponds to the worth of the material is tied up for the duration of storage. That is, during this time span no money is earned with the material. Secondly, storage itself generates costs. Thirdly, storage demands efforts regarding organisation and dispatch. Further risks are due to obsolescence as well as theft and loss of goods. (Martin, 2006, p. 310). Therefore, it is often preferred, to keep inventory levels low. Instead, one aims at delivering the material exactly at that point in time it is required (Martin, 2006, p. 310). This principle is denoted as just in time delivery (Vahrenkamp, 2007, pp. 5–6). But it is generally impossible to completely abolish storage; particularly whenever incoming and outgoing material flows are not synchronised (Weimar, 1973, p. 13). Then, storage has the purpose of bridging temporal gaps. Objectives for storage are manifold (Schmidt & Schneider, 2008, p. 374). One objective is to ensure both reliable delivery dates and high service levels. An aim in production logistics is to guarantee high utilisation of workshops, thereby also coping with interruptions (Bretzke, 2008, p. 7). Furthermore, storage allows buying material and products when prices are low.

Warehouses can be categorised according to different taxonomies. The taxonomies can be based on the function in a logistics process, the processing stage and type of the material, the degree of distribution, design and height, commodity type, as well as organisational and technological requirements (Schmidt & Schneider, 2008, p. 376). This section provides a brief introduction on the diversity of warehouse systems. A comprehensive overview is, for instance, provided by Jünemann (1989, pp. 143–187), Martin (2006, pp. 334–366), and ten Hompel et al. (2007, pp. 49–118). One distinction to be made is between warehouses for bulk and general cargo. Warehouses for bulk cargo (Martin, 2006, pp. 334–335) are out of the scope of this research because its particular interest is on general cargo. Instead, the focus is on warehouses for general cargo (Martin, 2006, pp. 336–358) which can be constructed as open-air storage or buildings. Open-air storage is only applicable for weatherproof logistics objects like shipping containers. Other goods must be kept in buildings. Common methods applied are floor and rack storage. Floor storage means that the general cargo is kept without any auxiliary means, either stacked or non-stacked. The advantage of this approach lies in the low capital investments required. But, in turn, the stacking height and thus the utilisation of space is limited (Martin, 2006, p. 338).

The degree of utilisation can be increased by block storage. Given that Last In, First Out (LIFO) is applicable, it is possible to keep large quantities of the same article directly next to each other without the necessity for lanes in between (Martin, 2006, p. 338). In contrast to single-storey warehouses, storing goods in racks enables a further improved utilisation of space because goods can be stacked higher (Haussmann, 1972, pp. 37–40). Goods can be kept either directly or in special containers. Line storage, compact storage, as well as combinations thereof can be distinguished regarding access (Martin, 2006, p. 339). With line storage it is possible to directly access all shelf positions at any time. By contrast, compact storage keeps goods at different depths. Mobile racks and storage carrousels are hybrid examples (ten Hompel et al., 2007, pp. 85–89). Racks can either be static or dynamic. In static racks, the goods stored are not moved. By contrast, flow rack stores pertain to the dynamic case with goods in motion (Weimar, 1973, p. 37).

Different strategies (Jünemann, 1989, pp. 175–177) exist for inventory management. On the one hand, they refer to allocation of storage positions. On the other hand, they also cover storage and retrieval. In this context, different aspects have to be considered. Regarding effective warehousing it is desirable to ensure fast access to the material stored. Following this insight, the ways to frequently requested goods have to be as short as possible. Likewise, an even and high utilisation of the storage space is desirable. Especially when dealing with perishable goods it is also important to avoid obsolescence. One possibility to implement inventory management is to define fixed positions for all articles. Jünemann (1989, p. 176) explains that this strategy allows accessing material even if the database administering the shelf positions is lost. This is possible because the storage position number correspond to the article number in this approach (Martin, 2006, p. 315). The disadvantage of this storage allocation strategy is its low degree of utilisation. The size and number of storage positions must be dimensioned always for the largest quantity (Martin, 2006, p. 316). Under the influence of transient quantities, it is virtually impossible to guarantee a high utilisation of the warehouse. Furthermore, changing ranges of products induce a high administration effort.

The problem of low utilisation is addressed by chaotic storage. In this approach the allocation storage place is completely free. This allows increasing the utilisation of warehouse capacity to almost 100% (Martin, 2006, p. 316) because storage space does not remain unused due to predetermined reservations. A combination of both approaches applies chaotic storage within fixed areas. This allows placing goods depending on their turnover rate near to or far from the loading zone (Vahrenkamp, 2007, p. 174). Distributing the same article over multiple lanes allows accessing the goods even if one storage and retrieval system fails.

Also storage and retrieval can be conducted with different strategies (Jünemann, 1989, pp. 175–177). First In, First Out (FIFO) prevents the material from becoming overage. Other storage systems, such as block storage, require LIFO in order to avoid unnecessary rearrangements. Another strategy is quantity adjustment. It aims at retrieving begun pallets first in order to prevent warehouse fragmentation. Finally, also storage and retrieval should prefer short ways to save time. Particularly double cycles allow preventing unnecessary movements of storage and retrieval systems.

2.1.4 Picking

In procurement and production logistics, articles are generally kept in item order during transport and storage. That is, shipping containers and pallets only contain articles of one type. Keeping great amounts of the same article together increases handling efficiency. But this does generally not meet customer demands. Instead, customers request single articles or multiple articles of different kind. Articles in item order must therefore be compiled in accordance with customer orders. This task is referred to as picking. A prototypical use case for picking is the distribution process of mail order businesses (ten Hompel et al., 2007, p. 251). In this context, the challenge is to compile small orders from an extensive range of products within short delivery time. Further applications include the supply of shops and outlets but also the provision of material for production. The articles to be picked are generally taken from a warehouse (Bode & Preuß, 2004, p. 271). Nevertheless, there is also the possibility to pick recently delivered articles directly without intermediate stockholding. This is referred to as cross docking (ten Hompel et al., 2007, p. 251).

Picking comprises the following work steps (Gudehus, 2007b, p. 686). Firstly, the articles to be picked must be allocated to supply stations. Then, the picker can collect the articles requested in the right amount. Afterwards, the articles picked have to be put on a conveyor system. All orders are consolidated at a order collection centre. In parallel, the supply stations have to be refilled. Information technology helps in assigning articles to supply stations and in optimising article picking order. Likewise, supply stations can be automatically refilled from the warehouse by respective conveyor systems. However, the picking operation itself is often too elaborate for robots (Bode & Preuß, 2004, pp. 297–298). Therefore, this task is generally carried out by humans. Following the picking process the orders have to be shipped. In the mail order business, distributing small shipments to many receivers is generally carried out by courier, express, and parcel (CEP) service providers (Vahrenkamp, 2007, pp. 137–161). For this purpose, packing is an important auxiliary function in order to protect the articles shipped (Fleischmann, 2008, p. 7). An introduction to packing systems is provided by ten Hompel et al. (2007, pp. 5–48).

2.2 Supply Networks

Managing logistics processes is a challenging task. Satisfying sophisticated demands from industry makes it necessary to combine the primary logistics functions to complex supply networks. The high number of participants and the arising uncertainty of demands constitute a major challenge in this context. Forrester (1961, pp. 21–22) points out that even small disturbances on the retail level may cause high oscillations in factory production rates. The order rates along the supply chain are increasingly fluctuating due to safety stocks created by all participants. Their behaviour is motivated by the fact that the visibility of each participant is generally limited to the orders of its direct customer. The effect of fluctuating order rates caused by this uncertainty of demands is commonly referred to as the bullwhip or Forrester effect (Papier & Thonemann, 2008, pp. 29–30).

However, the main focus of this research is not on order control. Instead, it puts emphasis on dealing with logistics requirements deriving from such orders. For this purpose, the primary logistics functions (Section 2.1) are the basic building blocks for logistics processes. They are combined in order to implement logistics networks that connect suppliers and consumers. Conventionally, supply network management is conducted from a centralistic perspective. Nevertheless, execution of processes is often partly or completely assigned to external service providers. Section 2.2.1 aims at identifying and categorising relevant actors in supply network management. The requirements for logistics processes are influenced by several current and future trends. These developments are discussed in Section 2.2.2.

2.2.1 Service Providers and Services Provided

A common approach to distinguish actors in logistics processes categorises parties of logistics providers. Figure 2.4 illustrates the bandwidth from companies who take the responsibility for their logistics demands themselves to fourth-party logistics providers who completely manage supply networks on behalf of their customers.

In the simplest case, companies satisfy their logistics demands themselves. In order to do so it is necessary that the company has the respective means to execute the primary logistics functions. For instance, this may include warehouses to store goods until they are sold. In order to supply customers including shops and outlets, means of transport, such as trucks, are required. Self-provision of logistics services can be referred to as first-party logistics, or 1PL in short. However, note that this term is generally not used in practice. Self-provision of logistics services coincides with a high degree of capital tie-up. Besides, transient demands of logistics services pose a major problem in this concept. As an example, consider subsidiaries which are supposed

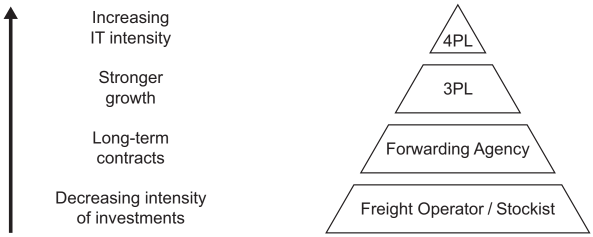

Fig. 2.4 Logistics service providers: The spectrum of companies offering their services ranges from freight operators and stockists to fourth-party logistics providers (4PL) who take responsibility for the whole supply network of their customers. The four properties develop for the provider types as indicated by the arrow (adapted from Vahrenkamp, 2007, p. 51).

to be supplied. Often, the period for delivery may be restricted to only a few hours per day for shops, e. g., in pedestrian precincts. Likewise, it might be necessary to supply a large number of shops simultaneously, but only on some weekdays. Apart from that, logistics demands may also fluctuate seasonally (Vahrenkamp, 2007, p. 49). This is particularly the case if shops are supplied with continuously changing ranges of products. These ranges may differ significantly regarding price, weight, and volume, thereby inducing different logistics demands (Schuldt, 2006, p. 4).

Another problem occurs if one aims at supplying customers of mail order businesses. To recapitulate, the challenge in this context is to distribute small shipments to a great many receivers. If there are insufficient shipments for one region it does not make sense to deliver them with an own truck. One possible reaction is to collect shipments for that area and deliver them in a consolidated way. This, however, means that customers receive their shipments with delay. These limitations indicate that it is not always adequate to satisfy logistics demands oneself. Instead, it is often more promising to buy logistics services in part or as a whole.

Employing external service providers which satisfy specific logistics demands can be referred to as second-party logistics, or 2PL in short. The term second-party logistics is also rarely used in practice. Gudehus (2007b, p. 1015) lists the following examples for 2PL providers. Firstly, freight operators transport cargo for their customers. Secondly, cargo handling providers carry out the respective primary logistics function. Thirdly, stockists store goods until they are supposed to be sold. Furthermore, specialised service providers take responsibility for bottling, packing, assembling, and repairing. A distinguishing characteristic of 2PL providers is that they perform primary functions themselves without relying on other companies (Scholz-Reiter, Toonen & Windt, 2008, p. 584).

Forwarding agencies are, in contrast, generally regarded as third-party logistics providers, in short 3PL. This is due to the fact that they act as a broker for the services offered by 2PL providers. It is, of course, possible that companies in the forwarding business additionally carry out transport and storage services themselves (Vahrenkamp, 2007, p. 47). Then, it is possible to assign them to both categories 2PL and 3PL. A particular advantage of third-party logistics providers is that they can choose from a broad spectrum of means of transport and carriers. Whenever one mode of transport does not suffice it is even possible to combine multiple modes. This enables 3PL to efficiently provide their customers with fitting solutions at competitive prices. Vahrenkamp (2007, p. 49) identifies the following additional advantages for customers of 3PL. External logistics service providers can compensate seasonally fluctuating demands by serving multiple customers with different seasonal peaks. The consolidated demands of their customers lead to an increased market power in buying logistics services. This enables them to offer their services at a lower price (Scholz-Reiter et al., 2008, p. 586). In contract logistics, the service provider also provides value-added services that exceed the primary logistics functions. This can, for instance, include customs clearance for imported goods or packing and picking services (Vahrenkamp, 2007, p. 48). In automobile logistics it is quite common that contract logistics providers prepare vehicle parts for overseas shipment. Likewise, the final inspection and assembly of cars can be conducted in the destination port. This allows mass production of cars. A customisation in accordance with national regulations and customer demands is then carried out as late as possible (Scholz-Reiter et al., 2008, p. 586). This strategy is referred to as postponement (Papier & Thonemann, 2008, p. 25).

A further advancement of 3PL is the so-called fourth-party logistics provider, in short 4PL. A widely accepted characteristic is that almost no logistics functions are executed by the 4PL itself (Gudehus, 2007b, p. 1015). Instead, the services of multiple 3PL providers are integrated in order to implement holistic supply network management. This high degree of integration requires standardised interfaces (Jünemann, 1989, p. 91). The own resources are generally restricted to elaborate information technology systems (Vahrenkamp, 2007, p. 50). The literature sometimes criticises the 4PL concept because it cannot be ensured to serve customers without own logistics resources (cf. Scholz-Reiter et al., 2008, p. 51). Both 3PL and 4PL providers can integrate a large number of subcontractors. They mediate between suppliers and buyers of logistics services. However, the large number of participants significantly increases the complexity of the logistics network because subcontractors may work for multiple 3PL and 4PL providers.

2.2.2 Developments and Influence Factors

The field of logistics is characterised by highly dynamic developments (Klaus & Kille, 2008, p. 951). As an example, consider the number of suppliers which has increased in recent years due to outsourcing and global sourcing

(Jünemann, 1989, p. 91). Hence, traditional supply chains evolve into complex supply networks with numerous suppliers and consumers. Complexity increases even more because both suppliers and consumers are themselves connected with several other participants within logistics networks. Vahrenkamp (2007, p. 3) suggests that recent developments in logistics can even be denoted as a logistics revolution. This statement also includes other influence factors that have to be regarded in order to understand the complexity and dynamics of logistics processes. Such factors influencing the freight business or logistics in general are referred to as effects (Aberle, 2003, pp. 91–98), impulses (Vahrenkamp, 2007, pp. 3–7), or megatrends (Klaus & Kille, 2008, pp. 951–957). This section reviews relevant influence factors.

The so-called goods structure effect (Aberle, 2003, pp. 93–94) refers to a development that occurs as a consequence of the transition from industrial to post-industrial societies. It can, for instance, be observed in North America, Western Europe, and parts of East Asia (Klaus & Kille, 2008, p. 952). In the past, a great demand for raw materials as well as mass production shaped logistics requirements. Train and inland water navigation are well suited to satisfy these demands. However, post-industrial societies are characterised by an increased amount of services instead of industrial products. This change leads to an increasing demand for exchanging documents, e. g., between planning offices in engineering, architecture, or culture and their clients (Vahrenkamp, 2007, p. 3). Apart from that mass customisation also impacts transport demands (Vahrenkamp, 2007, p. 4). In the past, mass production lead to a massive decrease in production costs. Today, individual customer preferences are increasingly considered, thereby retaining the effective principles of mass production. Nevertheless, this development increases demands on transport processes due to the great number of individual shipments to be handled. Likewise, also internet-based mail order businesses increase the amount of individual shipments (Vahrenkamp, 2007, p. 3). Particularly the rate of high-value consumer and investment products grows (Aberle, 2003, p. 93). Simultaneously, the relative weights and volumes of these goods decrease due to improved material and miniaturisation (Klaus & Kille, 2008, p. 952). Bulk freight transport systems, such as train and inland waterway navigation, are not suited to address these new demands. Instead, courier, express, and parcel service providers are becoming more important (Vahrenkamp, 2007, p. 3). To summarise, the size of individual shipments decreases while the total numbers of shipments and receivers significantly increase. Therefore, the goods structure effect leads to increased logistics requirements and thus increased complexity.

Aberle (2003, pp. 96–98) denotes another influence factor on developments in logistics as the integration effect. This effect refers to the international economic integration. Aberle (2003, p. 96) names the single market of the European Union as an example. Globalisation leads to the following developments (Klaus & Kille, 2008, p. 952). Firstly, it is possible to dislocate value-adding tasks to sites with best conditions. Secondly, companies can establish global production and value-adding networks. Thirdly, access to foreign markets and customers is significantly simplified. In this context, also the political and economical changes in Central and Eastern Europe play an important role (Vahrenkamp, 2007, p. 6). Apart from the European Union, worldwide attempts to ease international division of labour are made by the World Trade Organisation. Deriving from these developments, there is an increased demand for cargo transport over long distances. The increase in the amount of shipments, however, does not explain all growth in complexity of logistics processes. Additional complexity arises from the necessity to control and synchronise widely distributed logistics processes.

The so-called logistics effect arises from elaborate logistics concepts by industry and trading companies with increased requirements (Aberle, 2003, pp. 94–96). On the one hand, they concern the quality of the physical transport itself. On the other hand, they also cover due-date reliability. Customers often expect supply on demand, i. e., without long lead times. This is challenging due to the long distances to be bridged that derive from the integration effect. The challenge even aggravates due to the goods structure effect that leads to a great number of customers that must be delivered with a huge amount of small shipments on demand. Besides, Vahrenkamp (2007, pp. 3–4) points out that the high reliability and competitive pricing of logistics services enables dislocation of production facilities. Connection of the distributed facilities is then implemented by integrating them in supply networks. This, in turn, leads to the conclusion that the high quality of logistics today generates itself new demands for logistics services.

The new logistics requirements lead to an increased employment of courier, express, and parcel service providers and therefore increased road traffic. Independently from this development also transports formerly conducted by train or inland navigation are now relocated to trucks. This development is denoted as the substitution effect (Aberle, 2003, pp. 91–92) which means an individualisation of transport. According to Aberle, the particular properties of road freight traffic supported transferring cargo to the disadvantage of other transport modes. This shift of the so-called transport modal split is opposed to an increasing sensitivity regarding environmental concerns (Klaus & Kille, 2008, p. 953). Klaus and Kille describe an aversion against pollutive means of transports, particularly road freight traffic. A general objective in this context is to conduct as few transports as possible. One particularly aims at avoiding empty vehicle running, which in turn leads to cheaper transports. According to Aberle (2003, p. 13) pressures for cost containment and the fear of disadvantages in the competition also contribute to the willingness of companies to achieve this. Vahrenkamp (2007, p. 6) instances the food retail market. Due to decreasing profit margins optimised logistics processes are a crucial competitive factor.

To summarise, different factors lead to increased complexity and dynamics in logistics. Firstly, the size of individual shipments decreases while the total number of shipments and receivers increases. Secondly, the spatial distances to be bridged by logistics processes increase due to the globalisation. In order to address the elaborate logistics demands of industry and trading companies, more road freight traffic is required. Nevertheless, taking into account environmental issues and cost effectiveness it is desirable to reduce the total number of transports and to avoid empty vehicle running.

2.3 Challenges for Logistics Control

The primary logistics functions are applied in order to satisfy the logistics objectives (Section 2.1). Usually, each of these functions is not sufficient to satisfy the objectives on its own (Section 2.2). Instead, multiple functions contributed by multiple logistics service providers must be combined to a supply network. This is a challenging task because one aims at solutions that satisfy the logistics objectives both efficiently and effectively. In order to make the right decisions, it is thus important to prepare them with appropriate planning (Ellinger, Beuermann & Leisten, 2003, p. 2). This optimisation task is addressed by operational research, in short OR.

Dempe and Schreier (2006, p. 5) explain that there is no universally agreed definition of the term operational research. Nevertheless, it can be narrowed by its contributing disciplines, areas of application, problem types, and methods. Gal (1989, p. 15) categorises operational research as an interdisciplinary branch of science that integrates mathematics, systems theory, computer science, and decision theory. Ellinger et al. (2003, p. 7–8) give a survey on the areas in which operational research is applied. According to them, it has the highest diffusion rates in oil industry, chemical industry, iron and steel industry, electrical industry, aviation industry, automotive industry, mining, as well as paper industry. The application areas cover distribution, production, procurement, stock-keeping, human resource management, investment and financing, taxation, as well as integrated models (Ellinger et al., 2003, p. 7). In particular, Ellinger et al. (2003, pp. 8–11) distinguish the following problem types: combinatorial problems, stock-keeping problems, replacement problems, queueing problems, and concurrency problems.

Operational research has several branches. Linear programming optimises planning problems with respect to one or multiple objective functions (Domschke & Drexl, 2005, pp. 13–64). A standard method for this problem is the so-called simplex algorithm (Domschke & Drexl, 2005, pp. 21–30). Linear programming usually deals with continuous variables. Particularly in logistics, however, many problems demand discrete variables, e. g., pieces of general cargo. This class of problems is addressed by discrete linear and combinatorial optimisation (Zimmermann, 2005, pp. 307–333). Due to their combinatorial complexity, discrete optimisation problems are computationally significantly more demanding than continuous ones (Scholl, 2008, p. 48). Discrete linear programming is NP-complete (Hopcroft & Ullman, 1994, pp. 372–373). A prominent example for combinatorial problems in discrete linear programming are graph algorithms (Dempe & Schreier, 2006, pp. 231–267). Problems with nonlinear objective functions or constraints are dealt with by nonlinear programming (Ellinger et al., 2003, pp. 185–247). Dynamic programming addresses sequences of dependent decisions (Domschke & Drexl, 2005, pp. 157–173).

Apart from graph theory, also other fields contribute auxiliary means to operational research. Queueing theory allows modelling and analysing processes with objects waiting in a queue before being processed (Zimmermann, 2005, pp. 397–428). Game theory contributes modelling for actions of interdependent actors (Dempe & Schreier, 2006, pp. 320–344). Simulation allows the stochastic evaluation of problems which cannot be solved analytically (Domschke & Drexl, 2005, pp. 223–239).

A guiding principle in supply chain management is to take a holistic view on logistics systems in order to arrive at globally optimal solutions (Scheer, Angeli & Herrmann, 2001, p. 45). Nevertheless, Bretzke (2008, p. 6) points out that the actual research focuses only on encapsulated subproblems such as route planning. Operational research provides both optimal and heuristical solutions for these optimisation problems. For holistic control from a centralistic perspective, however, these approaches are limited by the following properties of logistics processes (Section 2.2):

1. Complexity
2. Dynamics
3. Distribution

Finding optimal solutions for logistics tasks requires computational effort. The mathematical models in operational research represent an abstract part of the real world (Ellinger et al., 2003, p. 4). Nevertheless, the effort increases with the number of logistics objects and parameters considered. Thus, it is often time-consuming to control supply networks from a centralistic perspective that incorporates all aspects. This challenge is even aggravated by the dynamic environment that influences logistics processes. Changes in the environment often cause solutions to become obsolete, therefore requiring re-computation. Besides, the high degree of distribution of logistics processes often prevents information from being centrally available for decision-making. The following sections present these properties of logistics processes in more detail and derive limitations for centralised control.

2.3.1 Complexity

In computer science, the computational complexity is an important property of algorithms (Saake & Sattler, 2006, p. 194). It helps in comparing algorithms and in estimating the computational effort required for solving particular problems. In order to compare algorithms, it is necessary to determine their asymptotic computational complexity, i. e., the relation between input and computational effort in the worst case. Thereby, one is generally not interested in constant factors which might depend on properties of the concrete computer executing the algorithm. Instead, one assigns them to so-called complexity classes (Saake & Sattler, 2006, pp. 199–200).

Common examples are depicted in Figure 2.5. For algorithms with linear computational complexity, referred to as $O(n)$, the computational effort increases with the same speed as the input. For instance, consider an unsorted list of elements. In order to find an entry in this list, it is necessary to examine all entries sequentially (Saake & Sattler, 2006, pp. 116–118). Therefore, doubling the size of the list means that the algorithm needs twice the time. In contrast, the computational effort is significantly lower if one aims at finding an entry in a sorted array. Binary search (Saake & Sattler, 2006, pp. 118–121) divides the search space into halves. Depending on the value of the middle entry, the searched entry is either in the left or in the right half. This half is again divided until the respective entry is found. Proceeding this way, only requires logarithmic effort, $O(\log n)$. If the number of entries in the array is doubled, the number of lookups is increased by only one. Therefore, also large amounts of data can be handled efficiently. For algorithms with quadratic complexity, $O(n^2)$, the computational effort increases quadratically with the input. A popular example algorithm pertaining to this class is bubble sort (Saake & Sattler, 2006, pp. 127–129). Even more complex problems can be found in $O(2^n)$ and $O(n!)$. Their effort increases exponentially and factorially, respectively, in relation to the input.

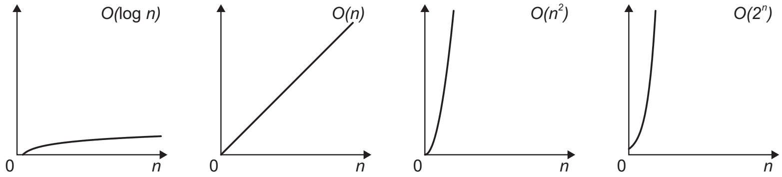

Fig. 2.5 Typical complexity classes in ascending order: logarithmic, linear, quadratic, and exponential. All curves are in the same scale. The axes are not labelled with units of measurement because constant factors are omitted in order to be independent from concrete computers.

Many logistics problems like the Transport Problem and the Travelling Salesman Problem (TSP) might appear rather simple at first glance. But they turn out to be still computationally demanding.

The classical Transport Problem is generally characterised by a bipartite graph (Domschke, 1995, p. 112). On the one hand, suppliers S_i with $i = 1, \dots, m$ offer s_i units of a certain good. On the other hand, consumers D_j with $j = 1, \dots, n$ demand d_j units of the same good. The costs for transporting one unit of the goods from S_i to D_j is c_{ij}, thereby assuming that transport relations exist from each supplier to each consumer. The question is then how many units of the good must be transported from supplier S_i to consumer D_j in order to supply all demands with minimal total transport costs. Transport Problems can be solved by linear programming, in particular by the simplex algorithm (Dantzig, 1951, pp. 360–365). It is worth mentioning that the Transport Problem is rather artificial compared to actual logistics processes. For instance, supply and demand are equal and only one type of goods is considered. Furthermore, there are no restrictions regarding delivery time and concurrent jobs. Despite of these confinements, the simplex algorithm still requires $2^n - 1$ iterations in the worst case (Klee & Minty, 1972, p. 174). Its asymptotical computational complexity is thus exponential, $O(2^n)$. However, Klee and Minty point out that, in practice, the computational effort is often significantly lower. A useful property of the Transport Problem is that the method is not limited regarding what is considered as supplier and consumer. Also applications apart from logistics exist, e. g., as a general distance metric (Rubner, Tomasi & Guibas, 2000).

The Transport Problem deals with assigning transport of goods to particular transport relations between suppliers and consumers. But it does not incorporate finding optimal routes through a transport network. A popular problem that deals with finding such a route is the Travelling Salesman Problem (Domschke, 1997, pp. 100–104). It pertains to the area of combinatorial problems which are addressed by discrete linear programming. It is defined as follows. A travelling salesman aims at visiting his customers at different locations. Subsequently, he wants to return to his starting point. The question is which way he should choose in order to minimise the total distance covered. Again, this is an artificial problem compared to real logistics requirements. Nevertheless, the complexity of the Travelling Salesman Problem even exceeds the one of the Transport Problem because the number of permutations of locations is $(n-1)!$ and thus factorial (Applegate, Bixby, Chvátal & Cook, 2007, p. 45).

A naive approach evaluates all these routes in order to find the optimal solution, i. e., the shortest way. Consequently, the computational complexity for such an algorithm is $O(n!)$. This means, adding just one additional location to be visited multiplies the previous computational effort by the total number of locations. This is due to the fact that $n!$ permutations exist in which the n locations can be arranged. The number of locations that can be considered by this approach is therefore highly limited. More elaborate algorithms consider additional knowledge about the structure of the problem in order to reduce its complexity. For instance, Applegate et al. (2007, pp. 513–515) apply a branch-and-cut strategy in order to determine an optimal route through $18,512$ locations in Germany. This procedure significantly reduces the computational effort for this particular problem. Still, however, finding the optimal solution takes approximately 57.5 CPU years on 2.66 GHz Intel Xeon processors (Applegate et al., 2007, p. 514).

An extension of the Travelling Salesman Problem is the so-called Vehicle Routing Problem (Domschke, 1997, pp. 204–213). It additionally considers multiple vehicles. Other more realistic applications address, for instance, fleet management for empty shipping containers for intermodal transport (Crainic & Kim, 2007, pp. 494–501) with even more parameters. Apart from exact approaches, it is also possible to apply heuristics which are less computationally expensive. It is generally not guaranteed that heuristics lead to optimal solutions. Common methods are simulated annealing, tabu search, and genetic algorithms (Domschke & Drexl, 2005, pp. 130–131). As an example, Schönberger (2005) applies memetic algorithms, i. e., genetic algorithms combined with local search, to vehicle routing. Timmermann (2008) applies tabu search to address the Multi Depot Vehicle Routing Problem with time windows.

These examples illustrate that even logistics problems which might seem trivial at first glance may exhibit a high computational complexity. To recapitulate, exponential complexity increases the computational effort with a fixed percentage rate for every additional input item. Correspondingly, computation takes more time. If the problem can be computed in parallel, it is alternatively possible to increase the computational power respectively by additional computers. Factorial complexity even multiplies the computational effort with the total number of inputs. Heuristics help reduce the complexity to some extent with a certain probability of deviating from global optima. Nevertheless, there are limitations with respect to the number of participating entities (Section 2.2.1). This challenge is even aggravated because in reality each entity may have a high number of parameters to be considered. These limitations must be considered when one aims at controlling whole supply networks from a centralistic perspective. It is particularly important to decide which aspects should be integrated into a model and how complexity can be reduced by abstraction or decomposition (Bretzke, 2008, pp. 30–31).

2.3.2 Dynamics

Solving logistics problems is often computationally complex and thus expensive. Nevertheless, it is worth finding optimal and not only efficient solutions for some problems. This makes particularly sense whenever decisions have long-term consequences. As an example, consider location planning (Domschke & Drexl, 1990, p. 3) for hubs in a logistics network. Establishing such hubs generally coincides with constructing respective buildings which are then used at least for several years. Due to the temporal impact of location planning, it might thus even be adequate to apply computers for several days or weeks in order to find an optimal location. If the locations were not optimally chosen, additional costs would arise later.

Location planning is a comparatively static problem. By contrast, this does generally not hold for logistics processes. For instance, controlling material flows cannot be approached by static planning. Consider an example from procurement logistics. In this example, the same types and amounts of goods are transported from the same suppliers to the same consumers every day. Then, one can compute a universally applicable optimal plan for resources and routes that holds for every day. But in real life, companies generally have different suppliers to deal with. Furthermore, the range of products might change, e. g., on a seasonal basis. This coincides with changing worth, weight, and volume of the goods to be transported. Thus, logistics demands change continuously, therefore preventing long-term static planning. Instead, planning must be conducted based on daily requirements.

Further motivation against static planning can be found in distribution logistics, e. g., in the mail order business. Here, orders by end-consumers are also transient. Although it might be possible to forecast the amount of goods sold to some extent, the exact figures and destinations of the shipments are unknown in advance. Thus, courier, express, and parcel service providers must cope with dynamically changing demands. Furthermore, also new requirements on logistics (Section 2.2.2) increase the dynamics underlying logistics processes. In the past, the push strategy was predominant in supplying outlets of trading companies. In this concept, the amount of goods assigned to each outlet is based on statistical data from the past. Today, companies switch over to the so-called pull strategy. That is, outlets are frequently resupplied based on the actual sales in the shops (Hellingrath, Hegmanns, Maaß & Toth, 2008, p. 468). As a consequence, the application of universally applicable static planning becomes impossible.

In the above examples from procurement and distribution logistics, one could argue that static planning is possible on a daily basis. Instead of universally applicable planning, the coverage of logistics planning is then at least one day. This is possible whenever the computational complexity is low enough in order to find an optimal solution in an acceptable period of time. For instance, computers could do the planning for the next day during several hours over night. However, apart from the external requirements on logistics, there are also inherent dynamics in the processes. These inherent dynamics are due to the complex interrelations in supply networks. In procurement logistics, there is a strong dependency on suppliers. Whenever they do not meet approved delivery times, further delays are caused (Bretzke, 2008, p. 8). Besides, delays in logistics processes can also be caused by other unpredictable situations, e. g., weather condition and traffic density.

It is thus necessary to implement continuous, instant re-planning in order to cope with the dynamics underlying logistics processes. From a centralistic perspective that incorporates all aspects of a supply network including all participants and parameters, this is often impossible due to the computational complexity (Section 2.3.1). Bretzke (2008, pp. 29–30) explicates that not only the frequency but also the extent of plan revisions is challenging.

Optimal plans might already be outdated in the moment their generation is finished. Conventional centralised control for whole supply networks is thus not applicable for instant re-planning.

2.3.3 Distribution

Apart from complexity and dynamics, distribution is the third challenge for centralised supply network management. Usually, logistics processes are distributed spatially and cross company boundaries. To understand the influence of distribution, it is necessary to distinguish between the different steps of supply chain management (Figure 2.2). In local production logistics, the influence of distribution is often rather limited. If workshops and workplaces in a company are to be supplied with material, the spatial distribution is generally limited to the company site. Likewise, the number of legal persons affected is limited if material and devices belong to the company itself. Thus, distribution does not prevent centralised control of this process because all relevant information for decision-making can be provided to a central entity. By contrast, this does not hold for procurement and distribution logistics. These parts of the supply chain are highly spatially distributed (Section 2.2.2). In procurement logistics, goods are procured from all over the world. In distribution logistics, goods are distributed throughout a country or even a continent. In order to be able to react from a global perspective on local changes, relevant information about all logistics entities must be available centrally. This is virtually impossible due to the costs of mobile communication. Hence, the spatial distribution prevents up-to-date information from being centrally available. Apart from spatial distribution, also processes crossing company boundaries are an argument against holistic centralised control. As elaborated in Section 2.2.1, logistics services are often not executed by the company itself but by external partners. Within supply networks, these partners generally serve multiple companies. Then, it is important that business data of subcontractors is treated confidentially (Cardeneo, 2008, pp. 732–733). Subcontractors are generally not interested in revealing internal information which might be required for centralised decision-making.

Not only information acquisition is challenging in distributed settings. Consider that there is a mathematical model that incorporates the whole supply network. Furthermore, consider that all information is available centrally. Disregarding complexity and dynamics, it is then possible to make decisions from the central perspective. However, it is still a problem to actually enforce these decisions. Bretzke (2008, p. 20) explains that, in order to determine a global optimum over company boundaries, it is important that the supply network has sharp boundaries. Sharp boundaries means in this context that companies pertain only to the respective network and to no others. Usually, however, suppliers work for multiple customers, in reality often even from completely different branches (Bretzke, 2008, p. 22), i. e., there are no sharp boundaries. This, in turn, means that there are factors which cannot be considered in the model but which nevertheless effect a global optimum. Hence, this kind of fragmentation would not be acceptable. Bretzke (2008, p. 22–28) explicates that such a ban leads to several problems. Firstly, outsourcing which means a further fragmentation of supply networks would not be allowed. Secondly, partners which do not adequately fulfill their tasks can hardly be exchanged. Finally, there is no incentive for improvement due to missing competition in the supply network.

Another question regarding centralised control of supply networks is who should take over leadership for the network (Bretzke, 2008, p. 25). An obvious choice is the focal company of the network. For a global optimum of the whole network, however, it might be necessary that the leader deviates from its own optimum. This leads to the question why he should do this if he has the power to impose his objectives on the other participants. A related question is why the other participants should trust the leader that it accounts for their interests. Apart from that, Bretzke (2008, p. 21) explains that sharp boundaries of supply networks mean that there must actually be as many networks as companies. As a consequence, there would be none (Bretzke, 2008, p. 21) or a planned economy (Bretzke, 2008, p. 30). Thus, also processes crossing company boundaries aggravate or even prevent centralised control of supply networks.

2.4 Conclusion

Supply network management mediates supplies and demands between suppliers and consumers. This objective can be achieved by providing the right quantity of the right objects in the right place at the right time in the right quality for the right price (Jünemann, 1989, p. 18). The primary logistics functions that can be applied are transport, handling, storage, and picking. To establish and control logistics processes, companies and their respective service providers combine the primary logistics functions. The evolving supply network consists of manifold interconnections between suppliers and consumers. This complexity is aggravated by increasing spatial distances and increasing amounts of individual shipments with decreasing size.

Operational research offers methods for optimised control of logistics. However, computational complexity is a challenge for the scalability of such centralised control of logistics processes. For problems with high complexity, the computational effort increases exponentially or even factorially with the number of logistics entities and parameters considered. Thus, the time needed to find optimal solutions frequently exceeds the time for executing the respective logistics functions. This is even aggravated by the fact that static planning cannot be applied. Transient logistics demands and dynamic environmental influences require continuous re-planning. Furthermore, spatial distribution often prevents decision-making from a centralised perspective because not all information is centrally available. Besides, distribution over multiple companies prevents centralised control because boundaries between supply networks are not clear cut.

These insights bring up the question how the limitations of conventional centralised control for holistic supply network management can be dealt with appropriately. As an alternative, it is possible to identify maximal subsystems that can be centrally controlled. To minimise the number of legal persons affected, such subsystems will most likely be found within the boundaries of one company. The computational complexity and the dynamics of the system restrict the number of entities and aspects that can be considered within such subsystems. The higher the dynamics, the more is the manageable problem size restricted by the computational complexity.

Identifying such subsystems follows the way conventional control is applied already today. For holistic control of supply networks, however, the resulting subsystems must be integrated. To this end, Bretzke (2008, pp. 30–33) proposes loosely, but intelligently coupled control systems. A potential deviation from a global optimum over the whole supply network is no restriction. As elaborated in this chapter, it is almost impossible to define and achieve a global optimum for whole supply networks. The question, however, is how to implement the integration of the independent subsystems.

References

Aberle, G. (2003). *Transportwirtschaft* (4th ed.). Munich, Germany: Oldenbourg.

Aßmann, R. (2008). Stückgutförderer in Logistiksystemen. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 613–645). Heidelberg, Germany: Springer-Verlag.

Applegate, D. L., Bixby, R. E., Chvátal, V. & Cook, W. J. (2007). *The Traveling Salesman Problem: A Computational Study.* Princeton University Press.

Arnold, D. (2008). Außerbetriebliche Logistikketten. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., p. 727). Heidelberg, Germany: Springer-Verlag.

Arnold, D., Isermann, H., Kuhn, A., Tempelmeier, H. & Furmans, K. (Eds.). (2008). *Handbuch Logistik* (3rd ed.). Heidelberg, Germany: Springer-Verlag.

Bode, W. & Preuß, R. W. (2004). *Intralogistik in der Praxis: Komplettanbieter der Intralogistik.* Suhl, Germany: Wirtschaftsverlag WV.

Bretzke, W.-R. (2008). *Logistische Netzwerke.* Heidelberg, Germany: Springer-Verlag.

Cardeneo, A. (2008). Straßengüterverkehr, Speditionen, Logistik-Dienstleistungen. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 727–736). Heidelberg, Germany: Springer-Verlag.

Crainic, T. G. & Kim, K. H. (2007). Intermodal Transportation. In C. Barnhart & G. Laporte (Eds.), *Transportation* (pp. 467–537). Amsterdam, The Netherlands: North-Holland.

Dantzig, G. B. (1951). Application of the Simplex Method to a Transportation Problem. In T. C. Koopmans (Ed.), *Activity Analysis of Production and Allocation* (pp. 359–373). New York, NY, USA: John Wiley & Sons.

Dempe, S. & Schreier, H. (2006). *Operations Research: Deterministische Modelle und Methoden.* Wiesbaden, Germany: B. G. Teubner Verlag.

Domschke, W. (1995). *Logistik: Transport* (4th ed.). Munich, Germany: Oldenbourg.

Domschke, W. (1997). *Logistik: Rundreisen und Touren* (4th ed.). Munich, Germany: Oldenbourg.

Domschke, W. & Drexl, A. (1990). *Logistik: Standorte* (3rd ed.). Munich, Germany: Oldenbourg.

Domschke, W. & Drexl, A. (2005). *Einführung in Operations Research* (6th ed.). Heidelberg, Germany: Springer-Verlag.

Ellinger, T., Beuermann, G. & Leisten, R. (2003). *Operations Research: Eine Einführung* (6th ed.). Heidelberg, Germany: Springer-Verlag.

Fleischmann, B. (2008). Begriffliche Grundlagen. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 3–12). Heidelberg, Germany: Springer-Verlag.

Forrester, J. W. (1961). *Industrial Dynamics.* Cambridge, MA, USA: MIT Press.

Frindik, R. (2008). Kombinierter Verkehr. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 736–743). Heidelberg, Germany: Springer-Verlag.

Gal, T. (Ed.). (1989). *Grundlagen des Operations Research: Einführung, Lineare Optimierung, Nichtlineare Optimierung, Optimierung bei mehrfacher Zielsetzung* (2nd ed.). Heidelberg, Germany: Springer-Verlag.

Gudehus, T. (2007a). *Logistik 1 — Grundlagen, Verfahren und Strategien.* Heidelberg, Germany: Springer-Verlag.

Gudehus, T. (2007b). *Logistik 2 — Netzwerke, Systeme und Lieferketten.* Heidelberg, Germany: Springer-Verlag.

Haussmann, G. (1972). *Automatisierte Läger* (O. K. Krausskopf, Ed.). Mainz, Germany: Krausskopf-Verlag.

Heidmeier, S. & Siegmann, J. (2008). Eisenbahngüterverkehr. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Hand-*

buch Logistik (3rd ed., pp. 743–757). Heidelberg, Germany: Springer-Verlag.

Hellingrath, B., Hegmanns, T., Maaß, J.-C. & Toth, M. (2008). Prozesse in Logistiknetzwerken – Supply Chain Management. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 459–486). Heidelberg, Germany: Springer-Verlag.

Hopcroft, J. E. & Ullman, J. D. (1994). *Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie* (3rd ed.). Bonn, Germany: Addison-Wesley Verlag.

Jünemann, R. (1989). *Materialfluß und Logistik: Systemstechnische Grundlagen mit Praxisbeispielen.* Heidelberg, Germany: Springer-Verlag.

Klaus, P. & Kille, C. (2008). Märkte für logistische Leistungen. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 947–969). Heidelberg, Germany: Springer-Verlag.

Klee, V. & Minty, G. (1972). How Good is the Simplex Algorithm? In O. Shisha (Ed.), *Inequalities* (Vol. III, pp. 159–175). New York, NY, USA: Academic Press.

Levinson, M. (2006). *The Box: How the Shipping Container Made the World Smaller and the World Economy Bigger.* Princeton, NJ, USA: Princeton University Press.

Martin, H. (2006). *Transport- und Lagerlogistik – Planung, Struktur, Steuerung und Kosten von Systemen der Intralogistik.* Wiesbaden, Germany: Friedrich Vieweg & Sohn Verlag.

Papier, F. & Thonemann, U. (2008). Supply Chain Management. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 21–34). Heidelberg, Germany: Springer-Verlag.

Rubner, Y., Tomasi, C. & Guibas, L. J. (2000). The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, *40*(2), 99–121.

Saake, G. & Sattler, K.-U. (2006). *Algorithmen und Datenstrukturen* (3rd ed.). Heidelberg, Germany: dpunkt.verlag.

Scheer, A.-W., Angeli, R. & Herrmann, K. (2001). Informations- und Kommunikationstechnologie als Treiber der Logistik. In H.-C. Pfohl (Ed.), *Jahrhundert der Logistik: Wertsteigerung des Unternehmens: customer related – global – e-based* (pp. 29–58). Berlin, Germany: Erich Schmidt Verlag.

Schmidt, A. & Schneider, M. (2008). Lager- und Materialflussprozesse. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 371–404). Heidelberg, Germany: Springer-Verlag.

Schönberger, J. (2005). *Operational Freight Carrier Planning: Basic Concepts, Optimization Models and Advanced Memetic Algorithms.* Heidelberg, Germany: Springer-Verlag.

Scholl, A. (2008). Optimierungsansätze zur Planung logistischer Systeme und Prozesse. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 43–57). Heidelberg, Germany: Springer-Verlag.

Scholz-Reiter, B., Toonen, C. & Windt, K. (2008). Logistikdienstleistungen. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 581–607). Heidelberg, Germany: Springer-Verlag.

Schuldt, A. (2006). *Management of International Logistic Processes: A Case Study at Tchibo* (Tech. Rep. No. 39). Universität Bremen: Technologie-Zentrum Informatik.

ten Hompel, M., Schmidt, T., Nagel, L. & Jünemann, R. (2007). *Materialflusssysteme — Förder- und Lagertechnik.* Heidelberg, Germany: Springer-Verlag.

Timmermann, T. (2008). *Ein zentraler Ansatz zur Berechnung des Multi Depot Vehicle Routing Problem with Time Windows (MDVRPTW).* Diplomarbeit, Universität Frankfurt.

Vahrenkamp, R. (2007). *Logistik — Management und Strategien* (6th ed.). Munich, Germany: Oldenbourg.

Weimar, H. (1973). *Hochregal-Lager: Grundlagen, Mittel und Kosten der Lagertechnik.* Mainz, Germany: Krausskopf-Verlag.

Zimmermann, H.-J. (2005). *Operations Research: Methoden und Modelle. Für Wirtschaftsingenieure, Betriebswirte und Informatiker.* Wiesbaden, Germany: Friedrich Vieweg & Sohn Verlag.

Chapter 3
Autonomous Control in Logistics

Supply network management aims at balancing supplies and demands between suppliers and consumers (Section 2.1). This is a challenging task due to the complexity, the dynamics, and the distribution that are inherent in logistics processes (Section 2.3). The autonomous logistics paradigm addresses these challenges by applying local control rather than centralised decision-making. To this end, each of the participating logistics entities is itself responsible for satisfying its predefined logistics objectives. Delegating both the autonomy and the ability to make decisions to the logistics objects coincides with the natural distribution observed in logistics. The advantages over previous methods are as follows. Firstly, it is possible to react locally on exceptions. It is thus not necessary to re-schedule the whole system which might even be impossible due to the complexity and the dynamics. Secondly, it is not necessary to reveal internal information and decision processes to a central entity. This is important if competing companies cooperate only in particular processes. Thirdly, handling standard processes and reacting on exceptions is still possible in cases of physical distribution with limited or even without communication bandwidth.

The objective of this chapter is to present the paradigm of autonomous control in logistics. Section 3.1 describes how this approach allows decreasing the complexity and coping with the dynamics and the distribution that are inherent in logistics processes. Based on this foundation, Section 3.2 introduces an architecture for logistics entities that are capable of decentralised control. In doing so, the technologies enabling autonomous control in logistics are examined. The objective of this investigation is to underline the particular importance of Distributed Artificial Intelligence in supply network management.

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics*,
DOI 10.1007/978-3-642-20092-2_3,

3.1 Paradigm Shift to Autonomous Control

Conventionally, supply network management is conducted from a centralised perspective. A central unit makes all decisions for planning and controlling logistics processes. This is a challenging task due to the high number of participants and parameters to be considered (Section 2.2.1). Furthermore, the complexity and the dynamics even increase due to new requirements on logistics (Section 2.2.2). These developments limit and sometimes even prevent centralised control. Instead, the paradigm of autonomous control in logistics, in short autonomous logistics, is a promising approach. Autonomous logistics aims at overcoming the limitations of conventional control in logistics by delegating decision-making to local entities (Herzog, Freitag & Scholz-Reiter, 2005, p. 222). The logistics entities are themselves responsible for satisfying the logistics objectives defined by their owners.

The general idea of autonomous control in logistics is presented in Section 3.1.1. It is particularly contrasted with conventional centralised control in order to examine how the new paradigm overcomes the limitations of centralised control. Based on this foundation, Section 3.1.2 investigates the potential for autonomous control in supply network management. Finally, limitations of autonomous logistics are discussed in Section 3.1.3.

3.1.1 Decentralised Decision-Making in Logistics

Following the discussion in Section 2.3, challenges for supply network management are the complexity, the dynamics, and the distribution that are inherent in logistics processes. The paradigm of autonomous control in logistics addresses the natural distribution of logistics processes by delegating decision-making to local entities (Figure 3.1). The term logistics entity denotes in this context both the material transformed as well as the facilities applied for transforming the material (Scholz-Reiter, Windt & Freitag, 2004, p. 362). That is, both providers and consumers of logistics services are considered active participants in the process (Hellenschmidt & Wichert, 2007, p. 99). Decentralised decision-making means that the local logistics entities are themselves responsible for achieving their logistics objectives (Section 2.1). For instance, consider a shipping container that is currently located in East Asia and that has to be transported to Europe. Being an active participant in the process, the container must then plan and schedule its route through the logistics network by itself. In order to be capable of this proactive behaviour, it is necessary to grant the respective autonomy to the shipping container. This particularly includes the permission to interact and to cooperate with other logistics entities in order to achieve its logistics objectives (Hellenschmidt & Wichert, 2007, p. 99).

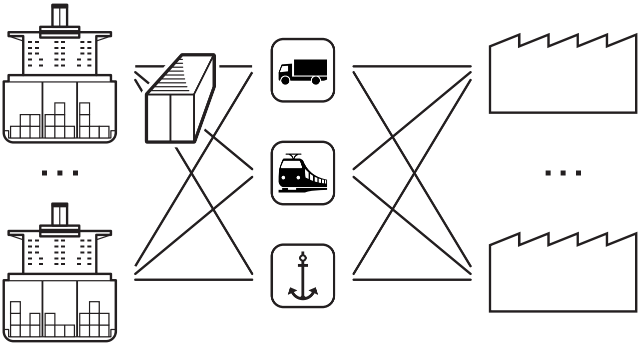

Fig. 3.1 An autonomous shipping container having sole responsibility for planning and scheduling its way through the logistics network. The container must cooperate with other entities such as means of transport and warehouses to achieve the logistics objectives imposed by the cargo owner.

Services provided by other entities like container vessels or trucks include transporting the container from its source to a sink. Other containers need the same resources. Hence, they compete for the logistics services offered. Because no central entity exists, containers and means of transport must themselves negotiate on the transport. In contrast to conventional centralised control, no hierarchy or structure is predefined on the logistics objects (Freitag, Herzog & Scholz-Reiter, 2004, p. 24). Instead, they flexibly interact based on their actual demands in a heterarchical way. Hence, the understanding of autonomous logistics here agrees with Windt and Hülsmann (2007, p. 8):

> "Autonomous Control describes processes of decentralized decision-making in heterarchical structures. It presumes interacting elements in non-deterministic systems, which possess the capability and possibility to render decisions.
>
> The objective of Autonomous Control is the achievement of increased robustness and positive emergence of the total system due to distributed and flexible coping with dynamics and complexity."

The idea of autonomous control in logistics is thus closely linked with the idea of self-organisation. Köhler-Bußmeier (2009) gives a general overview on the foundations of self-organisation. Hülsmann, Wycisk, Agarwal and Grapp (2007) particularly focus on aspects that underly autonomous logistics. An overview on the development of self-organisation in both information and communication technology is given by Becker, Kuladinithi, Timm-Giel and Görg (2007).

Autonomous control in logistics has several advantages over centralised approaches. Distributing decision-making to local entities coincides with the natural distribution of logistics processes discussed in Section 2.3.3. Proceeding this way particularly circumvents that all relevant information must be provided to a central decision-making entity (Freitag et al., 2004, p. 23). This step is no longer necessary because decisions are made locally. Control is therefore even possible without a permanent communication connection to a central entity. Apart from spatial distribution, also processes that cross company boundaries do not pose a problem. It is no longer necessary to disclose confidential information or reasons for decision-making. Instead, communication reduces to transmitting decisions taken directly by the local logistics entities.

The heterarchical organisation without a predefined structure of logistics entities allows reacting locally on dynamics occurring. Instead of updating the plan for the whole system, as it would have been necessary if centralised control was used, it is sufficient to modify only the plans of the entities that are directly affected. Decomposing problems into subproblems is a common approach in computer science, generally referred to as divide and conquer. Correspondingly, distributing control in logistics also decreases the problem complexity because each entity only has to consider its particular parameters (Windt, 2008, p. 352). This means a significant reduction of problem complexity compared to the centralised approach that incorporates all parameters of the whole system. With a limited number of parameters even problems with high computational complexity become manageable. As a further advantage, individual entities are only exposed to local dynamics and not to the dynamics of the whole logistics network (Windt, 2008, p. 352). A disadvantage, however, is that global optima are not necessarily achieved with decentralised control. For a more comprehensive discussion on global optima in supply networks, the reader is referred to Section 2.3.3.

Scholz-Reiter et al. (2004) divide logistics systems into three layers, namely the decision system, the information system, and the execution system (Figure 3.2). The execution system layer is already automatised to a high degree. Also autonomous systems have been successfully applied to this level. Think, for instance, of automated guided vehicles (AGV) which navigate through the real world (Schuldt & Gottfried, 2008a, 2008b). Examples can be found in intralogistics, e. g., in order to supply workshops and workplaces with material or to handle containers in automated container terminals. However, automated execution of primary logistics functions pertains rather to the field of robotics. In accordance with the general understanding of logistics (Chapter 2), also the notion of autonomous logistics used in this research focuses on planning and controlling processes instead of executing physical operations.

As illustrated in Figure 3.2, the decision and information system layers are currently less automatised. Hence, there is still potential for autonomously controlled technical systems. Applications for autonomous control in logistics can be found in all parts of the supply network. In procurement logistics, the intelligent container (Jedermann, Gehrke et al., 2007) is a novel application for autonomous logistics. This shipping container continuously monitors its content. In case of unexpected changes of the interior temperature or other parameters, the container can itself change its route or destination. This ensures that food loaded can still be sold and consumed before its shelf life is exceeded.

In production logistics, autonomous control can be applied to decrease the complexity of warehouse control (Trautmann, 2007). In distribution logistics, routing algorithms from computer and communication networks are applied to route packages through networks of courier, express, and parcel service providers (Wenning, Rekersbrink, Timm-Giel, Görg & Scholz-Reiter, 2007). Similar methods combine novel and conventional, i. e., centralised approaches (Berning & Vastag, 2007; Vastag, 2008).

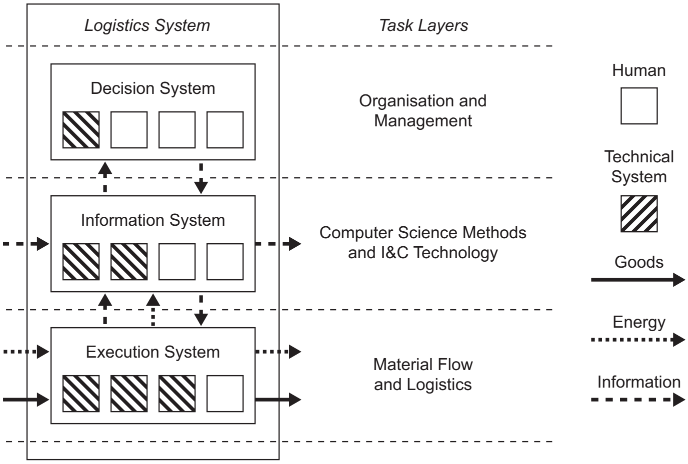

Fig. 3.2 The task layers in logistics divide into execution system, information system, and decision system. The execution level is already automatised to a high degree. By contrast, on the upper levels there is still potential for technical systems (adapted from Scholz-Reiter et al., 2004, p. 360).

3.1.2 Potential for Autonomous Control

The autonomous logistics approach aims at delegating decision-making to local logistics entities. In order to be able to make decisions on their own, these entities must have certain capabilities. Apart from the Artificial Intelligence to take the decisions themselves, they need identification and communication capabilities for interaction with other entities. In order to monitor their current state, they must be capable of localising themselves and sensing their environment. It is quite obvious that current logistics entities like cardboard boxes, pallets, containers, or trucks do not exhibit these capabilities. In order to implement autonomous control in logistics it is thus necessary to enhance them with the respective technologies required.

Augmenting physical objects with computational ability is possible because the miniaturisation observed in Section 2.2.2 does not only apply to the goods handled. It also has reduced the size of processors and related computer units (Hellenschmidt & Wichert, 2007, p. 94). According to Moore's law, the number of electronic components that can be placed on an integrated circuit doubles every 18 to 24 months (Moore, 1965, p. 115). That is, after this period the same computational power can be achieved with half the size. It turns out that this forecast is still accurate today. Interestingly, Moore's law does not only hold for processors but is also applicable for other technologies such as memory or communication bandwidth (Mattern, 2005, p. 43). However, Mattern (2005, p. 43) points out that Moore's law is eventually restricted by both physical and economical limitations.

Enhancing objects from the physical world as well as enhancing the physical environment is generally denoted by one of the following terms:

- Ubiquitous Computing
- Pervasive Computing
- Ambient Intelligence

Mattern (2005, p. 41) explains that the distinction between these terms is to a large extent an academical one. According to him, ubiquitous computing is a term that is mainly used as a vision for future developments. By contrast, the notion of pervasive computing is rather used by industry referring to solutions that can already be implemented today. While these terms are mainly used in North America, European scientists developed the notion of ambient intelligence which has a particular focus on human-machine interaction. These approaches have in common that they aim at assisting humans by Artificial Intelligence in the environment and interaction between physical objects, as in the case of autonomous logistics. Methods from Artificial Intelligence can be implemented on embedded systems that are attached to the physical object. Wöstmann (2006, p. 49) describes that RFID tags can even be integrated into cast components. Section 3.2 provides a more comprehensive overview on the key technologies enabling autonomous logistics.

3.1.3 Limitations of Autonomous Control

Autonomous control helps reduce the complexity of supply network management. Delegating decision-making to local logistics entities allows coping with both the dynamics and the distribution that are inherent in logistics processes. Therefore, it seems tempting to implement autonomous control at a very fine granular level. That is, to enhance every single article with the capability to make its own decisions. These articles are then expected to jointly control the logistics processes they are participating in. At first glance, this might seem the best strategy for reducing the complexity and for coping with the dynamics and the distribution. However, not only conventional centralised control is limited in its applicability. There are also limitations for autonomous logistics. Therefore, it is necessary to choose an adequate level at which autonomous control is applied (Windt, 2008, p. 350). Limitations that have to be considered for this choice are of technological, economical, and legal nature.

A limitation from the technological point of view is the computational power of autonomous entities. A particular question is which computational power can be expected with decreasing size of logistics entities. Embedded systems on comparatively small logistics entities, such as articles or packages, are rather limited in the power available. Power in this context not only refers to processor and memory, but also to the energy that can be consumed. Energy supply for mobile entities is generally implemented by battery. This is challenging because the miniaturisation of batteries does not keep up with the miniaturisation of processors (Mattern, 2005, pp. 42–43). Thus, energy must be used sparingly, i. e., computations should be reduced to a minimum. However, energy is not the only source of restriction to computational power. Furthermore, also the size of processor and memory is limited if the embedded system is affixed on small logistics entities. Limited memory demands efficient handling of knowledge acquired. Additionally, limited processing power requires knowledge bases to be kept manageable (Werner, Schuldt & Daschkovska, 2007, p. 15), particularly for complex reasoning tasks. This illustrates that with decreasing size of logistics entities also the number of computations that can be conducted decreases. Bigger entities, like warehouses, may have more processing power and are less limited regarding energy consumption. Nevertheless, it is worth remembering that also these entities are constrained by the asymptotical computational complexity (Section 2.3.1).

In addition to computational and energy considerations, interaction complexity is another potential limitation in autonomous logistics applications (Schuldt & Werner, 2007, p. 130). The more decision-making is distributed from one or few central to many local entities, the more communication is required for coordination. Therefore, it is important to categorise coordination mechanisms in accordance with the number of messages to be expected in relation with the number of participating entities. For this purpose, the complexity classes presented in Section 2.3.1 are applied accordingly. This allows comparing different approaches regarding their practical applicability. Interaction efforts depend on both the interaction complexity and the number of participants. Therefore, it is important to keep also the number of participating entities manageable in order to prevent a communication overhead. Otherwise, the decreased computational complexity of decentralised approaches is outweighed by the increased interaction complexity.

Apart from technological considerations also the economical perspective must be taken into account. The costs for components required to implement embedded systems for autonomous logistics are decreasing (Mattern, 2005, pp. 39, 42). Nevertheless, it is necessary to determine an acceptable granularity at which autonomous control is implemented. Enhancing every single article with the capability of decision-making requires providing these articles with the respective computational power. But many logistics applications do not require control on this fine granular level. Instead, it is often sufficient to stay on a coarser level where, for instance, a cardboard box or pallet is responsible for controlling all goods loaded. It might even be appropriate to implement autonomous control on the granularity of larger load carriers or means of transport such as containers or trucks.

From the discussion so far follows that the degree of granularity at which autonomous control in logistics is applied is important from both the technological and the economical perspective. The degree of granularity determines which logistics entities are capable of autonomous control. Apart from that, the degree of autonomy granted to logistics entities also is an important factor. On the one hand, this includes the temporal scope of their decisions, which may be operational, tactical, or strategic (Timm, 2006, pp. 7–11). On the other hand, the autonomy granted is closely linked with the degree of freedom for the decisions to be made. For instance, consider a shipping container that is transported by container vessel from East Asia to Europe. Due to a delay the container will not meet the estimated time of arrival. Depending on the autonomy granted the shipping container may itself initiate transshipping its cargo to an airplane. But transporting goods by airplane is significantly more expensive than delivering them by container vessel. Therefore, one might demand the autonomous container to consult a human dispatcher for this decision. Several legal aspects are closely linked with the autonomy granted to logistics entities (Nitschke, 2006, pp. 597–610). These include questions whether declarations of autonomous entities are legally effective and who is responsible in the case of misconduct. In this context, Matthias (2008) investigates under which circumstances intelligent autonomous entities might even be themselves accountable for their actions.

To summarise, the application of autonomous logistics is restricted by the degree of granularity at which autonomous control is applied and the degree of autonomy granted to logistics entities. The degree of granularity can be approached from both the technological and the economical point of view. It is thus necessary to find an appropriate level for the concrete logistics problem addressed. Thereby, it is important that increased cost and interaction complexity do not outweigh the decrease in computational complexity. Furthermore, the freedom in decision-making has to be defined in advance in order to implement a reasonable restriction in the autonomy granted to logistics entities.

3.2 Technologies Enabling Autonomous Control

Autonomous control of logistics processes means delegating decision-making to local entities. In order to implement the intelligent logistics entities required, new information and communication technologies have to be integrated (Section 3.1.2). Figure 3.3 proposes an architecture of the technologies demanded by autonomous logistics entities.

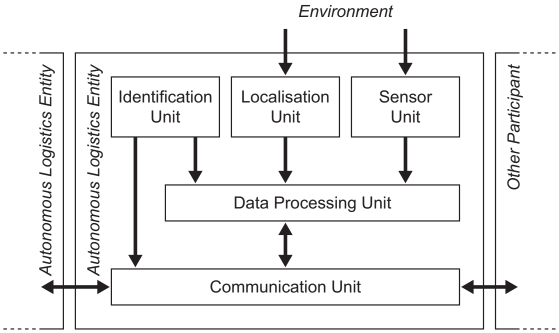

Fig. 3.3 Architecture for autonomous logistics entities. Enabling technologies include identification, localisation, and sensor technology, communication networks, and data processing.

In order to control a logistics process by a computer system, it is necessary to establish a link between the real world and the system. This holds for both centralised and decentralised approaches. Within the architecture presented, the identification unit provides the means to uniquely identify logistics entities (Section 3.2.1). To enable tracking and tracing of the goods transported, the localisation unit monitors the current location of the logistics entity (Section 3.2.2). For many applications, such as in the fresh food supply chain, it is additionally necessary to continuously monitor the condition of the cargo. The corresponding data about the logistics entity, its content, and its environment is contributed by the sensor unit (Section 3.2.3). Solving logistics tasks alone is only sufficient in special cases. Generally, autonomous entities have to cooperate with others, for instance, to negotiate on transport or storage capacities. Such interaction with other autonomous entities or other participants like humans is enabled by the communication unit (Section 3.2.4). Finally, the data processing unit (Section 3.2.5) integrates all incoming sensor data from the environment and coordinates interaction with the outside world.

The data processing unit is the central part of autonomous logistics entities because it has to actually implement the intelligence for the logistics object it represents. Like the data processing unit, the communication unit also is an essential part of the architecture. Without this unit, communication, and thus coordination, with other entities would be impossible. Likewise, the identification unit is indispensable because it provides the autonomous logistics entity with a unique identity. Its identifier can, for instance, be used by others to reason about the autonomous logistics entity and in order to address messages intended for it. Note that the identification unit has a direct link with the communication unit. Identification of objects is a frequent task in logistics, e. g., during shipping or receiving. Although access to identification information may be restricted, this restriction is generally not a task requiring intelligence. Thus, the design of the architecture enables identifying objects even without incorporating the data processing unit.

Localisation and sensor technology are also important in autonomous logistics. Nevertheless, these parts may be absent in autonomous logistics entities if the information can be acquired otherwise. As an example, consider a shipping container that monitors its interior and provides all packages loaded with the measurements obtained by its sensors. Another example incorporates a truck that takes responsibility for localisation for all logistics entities loaded. Then, the communication unit acts as a surrogate for the missing parts of the architecture. If also the data processing unit was replaced by the communication unit, one would arrive at centralised instead of autonomous control. The following sections provide a more detailed overview on the enabling technologies for autonomous logistics. Requirements for logistics and conventional solutions are discussed. Subsequently, more recent developments allowing the implementation of autonomous logistics entities in accordance with the architecture proposed are introduced.

3.2.1 Identification

Identification technology is the key aspect in order to synchronise information flows with their respective material flows. It is required in order to establish a link between real-world entities and their representation by computer-based control systems (Lampe, Flörkemeier & Haller, 2005, p. 69). Identification enables computer systems to become aware of movements and status changes of logistics entities. This allows reacting appropriately on changes in the real world. Approaches to identification are already applied today. The following paragraphs introduce two conventional systems and discuss their limitations. Based on this foundation, an innovative approach and its application to autonomous logistics is presented.

As a first example consider shipping containers. According to ISO standard 6346, each shipping container can be identified by a sequence of eleven characters and digits (Figure 3.4). The first three characters represent a code referring to the owner of the container. The fourth character is the product group code identifying the type of the equipment. The subsequent six digits form the serial number that is unique for all containers of one shipper. Finally, a check digit prevents data acquisition errors during handling.

The identification number is affixed as a marking on the surface of the container, either in horizontal or vertical orientation. Whenever a container is handled, for instance during loading or unloading, its identification number has to be recorded. Currently, this is mainly done manually which means a discontinuity of the information flow. Image processing methods constitute an alternative in that they support automated optical character recognition (OCR). Unfortunately, applying OCR in the shipping container domain is rather challenging. This can be explained by several reasons. Firstly, shipping containers have a large surface on which the marking may be affixed. This results in large-scale scans in order to find the marking. An effective scan requires a certain distance from the container. This requirement can often not be satisfied due to the narrow storage spaces of container terminals. Secondly, during maritime transport, shipping containers are directly exposed to the forces of nature. This results in dirt and abrasion which aggravate correct character recognition. Finally, illumination and weather conditions in the container terminal pose additional problems. Hadow (2005, p. 58) points out that current optical character recognition system achieve only a recognition rate of about 80% under these real-world conditions. This rate is far from being acceptable for professional applications.

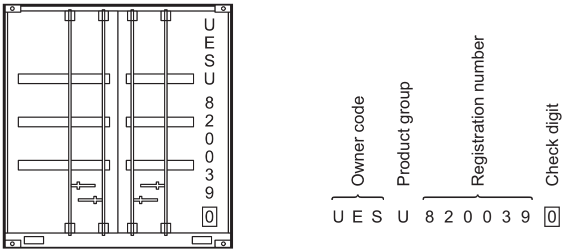

Fig. 3.4 According to ISO 6346, shipping containers are identified by markings on their surfaces. These markings are built up by human-readable sequences of characters and digits.

Another example for identification of real-world objects is the International Article Number EAN (Vahrenkamp, 2007, p. 60), formerly European Article Number. An EAN consists of 13 digits (Figure 3.5 left), for small-sized products a special version with only eight digits exists. EAN-13 is started by a Global Location Number (GLN) company prefix which allows identifying companies uniquely. The company prefix is five to seven digits in length. It is assigned by the Global Standards One (GS1) company. The remaining digits can be used by the respective company as a serial number for their products. The last digit serves as a check digit to ensure the validity of the code. EAN-13 codes are generally printed as a bar code on the respective packaging. Such article numbers identify articles as members of a certain article group. This allows, for instance, simply scanning the bar code at a point of sale in order to find out its price. Likewise, the entry of the respective article group in the inventory control system can be decremented by one when an article is sold. Coupled with automated replenishment, this allows keeping supplies coming.

A shortcoming, however, is that it is impossible to distinguish shipping units of the same article group during transport just by their EAN. All shipping units containing the same type of articles have the same EAN because the EAN always pertains to a whole group of articles. This problem is addressed by the Serial Shipping Container Code (Vahrenkamp, 2007, p. 60), in short SSCC, a numerical identifier with 18 digits. Following an extension digit, SSCC starts with the GLN company prefix like EAN-13 (Figure 3.5 right). The following digits minus the check digit can be assigned by the respective company itself. This allows, for instance, identifying complete shipping units like pallets. While identification on this level may be sufficient for some applications, it is not for all of them. Cardboard boxes on pallets may be re-packed on their way through a logistics network. Without unique identifiers for each cardboard box, it is impossible to implement a continuous monitoring, e. g., of the cold chain. Furthermore, a reliable permanent stocktaking is impossible without identification on the article level. Another shortcoming results from the bar code representation. The bar code has to be scanned each time a shipping unit is handled. Scanning is often carried out manually which again results in an information flow discontinuity. Furthermore, bar code scanning requires visual contact. As an example, this prevents cardboard boxes located on a pallet from being scanned when they are occluded by other cardboard boxes. Further challenges result, like in the shipping container example, from dirt and abrasion.

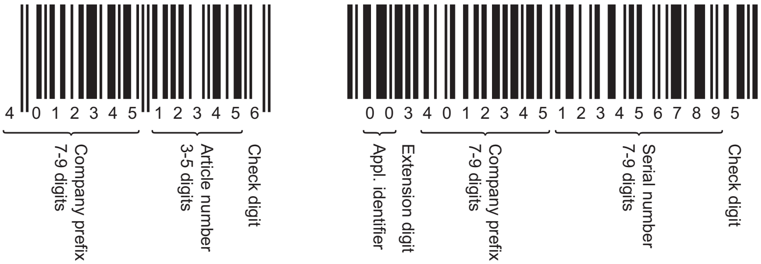

Fig. 3.5 EAN-13 (left) and SSCC (right) are intended to encode article numbers and numbers of shipping units, respectively. Both of them contain started by the GLN company prefix of their company. The application identifier 00 is part of the EAN128 bar code encoding standard for SSCC.

To summarise, the shortcomings of the current solutions presented above are as follows. Shipping units are mainly identified by non-digital codes. Dealing with analogue representations results in a discontinuity of the information flow because, in general, they have to be acquired manually. Automatic recognition is a hard task due to dirt, abrasion, poor illumination, and occlusion. Furthermore, identification is currently only applied on a high level of granularity of shipping units which is not sufficient for some applications. These issues are addressed by radio-frequency identification (RFID) which aims at automated identification without direct physical or visual contact, thereby avoiding information flow discontinuities (Lampe et al., 2005, p. 69). RFID tags consist of three components: a serial number identifying the object the tag is attached to, a transponder for wireless communication, and a microchip as data storage.

A standard to format serial numbers is the Electronic Product Code, in short EPC. Today, it is most commonly used with at least 96 bits in length. This length allows identifying single objects uniquely; reading serial numbers of this length does not pose a problem because it is done automatically. Nevertheless, the actual granularity at which objects are marked still depends on the demands imposed by the application at hand and the cost (Section 3.1.3). Originally, EPC was intended to be universally applicable for arbitrary applications in arbitrary domains (Flörkemeier, 2005, p. 90). This, however, is contradictory to existing industry standards. Thus, companies involved in the EPC standardisation process demanded to create customised sub-standards that are in compliance with existing formats like EAN (Flörkemeier, 2005, p. 90). Kuhlmann and Masuhr (2007, p. 257) explain that the intention is to protect previous investments and to support a smooth transition from bar codes to EPC. The EAN-based derivate of EPC is structured as follows. A header defines the format used. It is followed by a filter value that identifies the type of the load carrier, e. g., whether it is a pallet or cardboard box. A partition field defines where to divide the subsequent company prefix and item reference. Company prefix and item reference correspond to the EAN identifier. Finally, the serial number identifies single objects uniquely. In general, it is also possible to store additional data on RFID tags. Nevertheless, this is often avoided in order to save costs by keeping RFID tags simple. Instead, the EPC can be used as a reference in order to retrieve additional data (Flörkemeier, 2005, p. 89).

The EPC that is stored on an RFID tag can be readout with RFID readers. The energy supply for both transponder and microchip can either be passive, semi-active, or active (Lampe et al., 2005, p. 73). The first group of tags exclusively uses the energy field induced by the reader. Semi-active have an internal battery for the microchip, the transponder is served by the reader. In contrast, the energy supply of active tags is completely covered by their internal battery. Bulk scanning currently allows to read up to 400 transponders per second (Kuhlmann & Masuhr, 2007, p. 258). Different load carriers can be distinguished by the EPC filter value. This allows defining the level of granularity at which to scan, e. g., only pallets or cardboard boxes. Signal collisions may occur during bulk scan if multiple transponders respond in parallel. Such collisions can be prevented either by deterministic or probabilistic methods (cf. Lampe et al., 2005, p. 73).

While EPC and RFID enable the unique identification of objects, it is still necessary to link them to their representations in computer systems. For this purpose, the Internet of Things (Fleisch & Mattern, 2005; Bullinger & ten Hompel, 2007) aims at extending the common internet to objects in the real world (Figure 3.6). Therefore, all relevant objects are equipped with an RFID tag carrying the respective EPC identification. The connection to software applications is as follows (Flörkemeier, 2005, p. 89). Applications can access the EPC identification stored on the RFID tag either directly via the RFID reader or through a middleware. This middleware allows, for instance, filtering and bundling data streams with RFID tags. The EPC is only for identification purposes and does rarely provide any additional information about the object scanned. Additional information can be retrieved from an EPC Information Service (EPC IS). To avoid bottlenecks, there exists no central EPC Information Service. Instead, Information Services may be distributed, e. g., one for each company. An Object Naming Service (ONS) can be contacted in order to find out which EPC IS administers a real-world object. Hence, the ONS corresponds to the Domain Name Service (DNS) on the common internet.

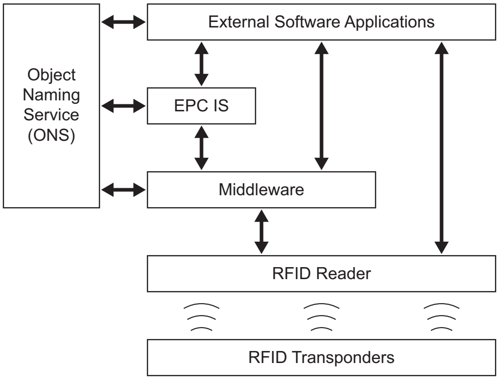

Fig. 3.6 The Internet of Things infrastructure encapsulates access to RFID tags by a middleware. The middleware queries the Object Naming Service (ONS) for EPC Information Services that provide additional information on the object scanned (adapted from Flörkemeier, 2005, p. 89).

EPC and RFID enable identifying objects more reliably and with less effort than previous systems. The information flow discontinuity of previous systems is abolished by wireless digital communication. Dirt and abrasion of bar codes no longer pose a problem, recognition rates are beyond 99% (Hadow, 2005, p. 58). In order to protect them from environmental influences it is possible to embed RFID tags into products or load carriers (Lampe et al., 2005, p. 70) and even cast components (Wöstmann, 2006, p. 49). This is possible because no visual contact is required. However, note that certain limitations exist because radio signals are attenuated, for instance, by metal and water. Lampe et al. (2005, p. 69) list the following advantages of RFID: decreased error rates, increased process efficiency, increased product quality, as well as cost saving by faster and better information processing. Example applications that can be implemented with identification technology include tracking lots through a factory or supply network as well as permanent stocktaking (Thiesse, 2005, pp. 114 –115). In autonomous logistics, electronic identification allows logistics entities becoming aware of both their own identity and the identity of other entities.

3.2.2 Localisation

Tracking and tracing goods is an important application in logistics (Scholz-Reiter, Toonen & Windt, 2008, pp. 596–597) to make logistics processes transparent to customers. Therefore, cargo owners are provided with a continuous visibility of their goods. Tracking in this context refers to the discrete or continuous localisation of goods. Tracing is the process of analysing and archiving the data records. The automatised identification of goods is one of the key technologies to implement tracking. However, it is usually not sufficient so that there is a demand for additional technology. This section outlines applications in logistics and different approaches to implement them.

In the courier, express, and parcel (CEP) business, it is quite common to provide customers with tracking and tracing services. These services are generally implemented by periodically scanning identification labels affixed to the packages transported (Cardeneo, 2008, pp. 787–788). The first time the CEP provider gets in contact with a package is when it is collected from the customer (Figure 3.7). The package is scanned for the first time then. Afterwards, the identification is recorded each time the package enters or leaves a hub. Finally, the package is also scanned when it is delivered to its recipient. In this context, identification is conventionally implemented with bar codes (Cardeneo, 2008, pp. 787–788). The temporal resolution of this localisation method is rather limited and allows only locating packages coarsely within the logistics network. Nevertheless, it is still sufficient to find out who is currently in charge of the shipment. Whenever a package should get lost during the transport process, this periodical scanning still allows determining the person responsible, e. g., for reimbursement. A more elaborate solution employs RFID and EPC, thereby enabling automated identification by radio-frequency scanning without visual contact (Section 3.2.1). Handling effort is massively reduced because bulk scanning can be applied.

Scholz-Reiter et al. (2008, p. 596) argue that tracking and tracing can also improve other transport processes. In particular, they mention intermodal transport which often involves several logistics providers in multiple countries. Hitherto, cargo owners normally lost contact with their shipping containers with pre carriage. The containers remained invisible until their arrival at the final destination. This is dissatisfying for several reasons. Firstly, like in the CEP domain, responsibility is an important issue. In particular, if potentially untrustworthy partners participate in the process, one must be able to trace

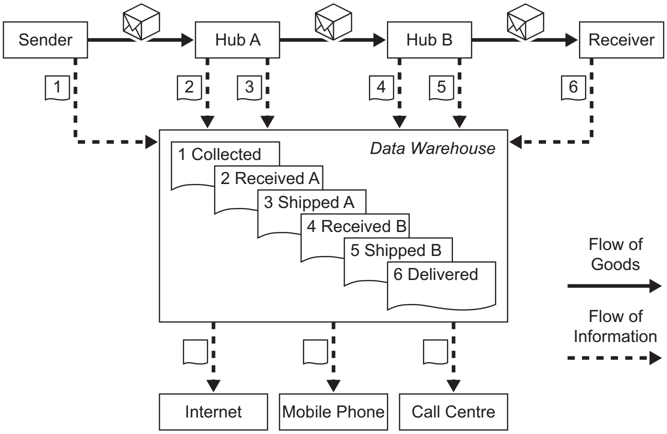

Fig. 3.7 Tracking and tracing in the courier, express, and parcel business. Packages are scanned when they are collected and delivered as well as when entering or leaving a hub in between (adapted from Cardeneo, 2008, p. 787).

responsibility if a shipping container is lost. However, the complete loss of containers is only the worst case. Secondly, without tracking and tracing it is impossible to reliably predict the time of arrival for shipping containers. This is due to the fact that delays may occur during transport. These delays may be caused, for instance, by bad weather conditions or overbooked container vessels. Therefore, tracking and tracing helps improve planning of production processes. If it is known in advance that cargo arrives late, production plans can be updated accordingly. An autonomous logistics entity might even reschedule its transport to a faster means of transport (Section 3.1.1). If the cargo is required on time, it might, for instance, be appropriate to change from container vessel to airfreight.

To enable such decisions in a timely manner it is necessary to implement localisation with high temporal resolution. Continuous updates of the cargo location are required. Pflaum and Hupp (2007, p. 109) point out that even RFID does not provide enough precision and therefore does retain one of the problems identified when bar codes are used. It is only possible to use RFID base stations as a beacon in order to find out whether or not a tag is located near a station. Localisation therefore requires a close-meshed network of base stations. In intralogistics, it is often no problem to install the readers required to enable at least periodic monitoring (Lampe et al., 2005, p. 70). But tracking RFID tags in transport logistics is significantly more challenging due to the long-range spatial distribution. A potential solution might include scanning EPC codes of cargo at each highway overpass (Hadow, 2005, p. 58). Similar solutions are already applied in order to track the movement of rail cars (Gallagher, 2002, p. 48). However, this has some disadvantages. Firstly, it is still necessary to install lots of additional hardware at the routes along which trucks or shipping containers move. Secondly, RFID readers at highway overpasses still do not enable real-time monitoring.

Apart from improved planning, security is another motivation for cargo tracking and tracing. Consider, for instance, trucks with valuable goods or hazardous material. A requirement might be to prevent such trucks from deviating from their scheduled route. This route can be defined by a corridor in which a truck can move freely. In general, such corridors should follow the course of highways, thereby also enabling trucks to leave the highway for refuelling. But if the predefined corridor is left, the autonomous logistics entity has to stop the car immediately and inform the dispatcher or the responsible authorities (Hannon, 2002, pp. 39–40). Likewise, this allows preventing trucks with hazardous materials from approaching vulnerable facilities such as nuclear power plants (Pekow, 2005, p. 14). RFID-based approaches may suffice to keep production plans up-to-date. But the temporal resolution of beacon-based approaches is too coarse for the security application intended.

As an alternative to beacon-based approaches, trilateration can be applied (Pflaum & Hupp, 2007, p. 112). Trilateration allows determining the position of objects based on their distance to reference points whose position is known. Figure 3.8 illustrates an example for the two-dimensional case. The position of point P is to be determined. The position of reference points R_1, R_2, and R_3 is known. Likewise, also the distances between P and the reference points are known. The distance to reference point R_1 constrains the position of P to a circle with radius $\overline{PR_1}$ around R_1. Adding reference point R_2 further constrains the position of P to the intersection of the circles around R_1 and R_2. Finally, adding R_3 reveals the actual position of P.

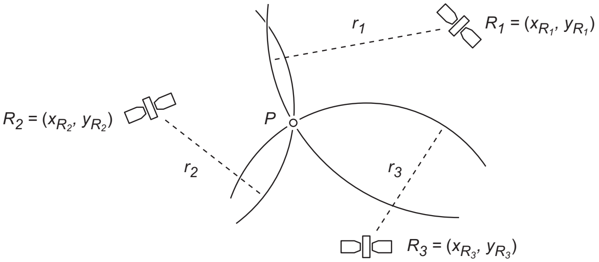

Fig. 3.8 A two-dimensional trilateration example. The position of P has to be determined by its distance to the reference points R_1, R_2, and R_3. P lies at the intersection of the distance circles around the reference points.

Instead of employing arbitrary landmarks, global navigation satellite systems (GNSS) are a generic alternative with worldwide coverage (Hofmann-Wellenhof, Lichtenegger & Wasle, 2008, pp. 4–6). Apart from the existing satellites, no additional equipment has to be installed in the environment. Only the autonomous logistics entity whose location is to be determined must be equipped with a receiver for position transmissions by the satellites

(Pekow, 2005, p. 14). Existing global navigation satellite systems include the US global positioning system GPS and the Russian GLONASS. In the future, also the European Galileo will be available. These systems have in common that they consist of a number of medium Earth orbit satellites (Tanenbaum, 2003, p. 114). The satellites are aware of their current position. Furthermore, they are equipped with high-precision clocks. Each satellite continuously broadcasts its current position and time (Pflaum & Hupp, 2007, p. 112). The distance to the satellite can be computed based on the delay with which the signal is received. Determining a position on the surface of Earth requires signals from at least four satellites (Hofmann-Wellenhof et al., 2008, pp. 8–9). Alternatively, one could also apply base stations of cellular networks. However, the precision of satellite-based systems is generally higher. Furthermore, while cellular networks do not cover oceans, global navigation satellite systems can also be applied to locate shipping containers worldwide.

In intralogistics, it is quite common to employ automated guided vehicles (AGV), for instance, to transport workpieces between different workplaces. The control and navigation of such vehicles also requires means for localisation. Unfortunately, the benefit of satellite-based localisation systems is rather limited in this context. This is due to the fact that the satellite signal is too weak to be reliably received inside buildings and through some materials (Pflaum & Hupp, 2007, p. 112). Conventional guidance systems are thus based on wire-guided tracks or optical following of surface markings (Martin, 2006, pp. 265–267). In the first case, the possibility to change the production layouts is rather limited; in the second case, abrasion can significantly decrease recognition rates (ten Hompel, Schmidt, Nagel & Jünemann, 2007, pp. 199–204). In contrast to machines, humans do easily succeed in such navigation tasks, even in dynamic environments. Hence, cognitively motivated spatial representations are a promising alternative (Schuldt & Gottfried, 2008a, 2008b).

In this context, RFID may support computer vision methods and help identify both stationary objects and other vehicles in the environment. Without RFID support, objects can be recognised by their visual appearance. In general, several visual properties such as size, position, orientation, colour, and texture are applicable for object recognition. According to Palmer (1999, p. 363), however, the visually most significant property of objects is their shape. For fast object recognition compact shape representations (Schuldt, Gottfried & Herzog, 2006a, 2006b, 2006c) can be applied. Choosing such compact representations can be motivated by their low computational complexity (Section 2.3.1) for shape comparison. Numeric representations characterise shapes even by only one numeric value (Duda & Hart, 1973; Garson & Biggs, 1992; Gottfried, Schuldt & Herzog, 2007). In order to ease object recognition, special visual markings can be attached in order to recognise categories of objects.

To summarise, localisation allows implementing services like tracking and tracing in logistics. Tracking and tracing is the foundation for both improving planning and increasing security in transport processes. The technologies that can be applied to localise objects range from beacon-based approaches to trilateration and recognition of the environment by computer vision. The concrete technology to be chosen depends on the application addressed. Likewise, also the granularity at which localisation is implemented depends on the concrete application (Section 3.1.3). For instance, it does not seem reasonable to equip every package transported on a truck with a GPS receiving unit. Instead, the receiver could be part of the respective truck. The truck would then use its communication unit to share the current global position with its packages.

3.2.3 Sensor Technology

Identification allows distinguishing autonomous entities in order to link them to computer-based control systems. With localisation technology, it is possible to determine the current location of logistics entities. Sensors additionally enable monitoring both objects and their environment. The necessity for monitoring can be motivated by actual requirements from logistics practice. This section present applications in logistics requiring sensor technology to ensure quality and security. Subsequently, it is examined how wireless sensor networks can be applied to address these demands.

One of the major objectives in logistics is to deliver goods in the right quality (Chapter 2). Hence, goods must be protected appropriately, e. g., by transporting them in refrigerated shipping containers. However, implementing protection alone does not suffice. Additionally, adequate quality monitoring has to be conducted in order to validate the protection. On the one hand, this allows reacting early if, for instance, the cooling system fails. On the other hand, regarding claims for reimbursement, monitoring allows finding out who is responsible for failures occurring.

As an example consider shipping containers that are transported by container vessel from East Asia to Europe. During their transport, these shipping containers undergo fluctuating extreme climate conditions, in particular regarding temperature and humidity (DHL Express Vertriebs GmbH & Co. OHG, 2005, pp. 393–394). Both temperature and humidity are influenced by ambient air and water. On deck of container vessels, containers are directly exposed to sunlight. On some shipping routes this leads to heating of up to 80° C on the surface of the container. Consequently, also the interior temperature increases, thereby heating the cargo to more than 50° C. Likewise, the water temperature heats containers in the body of the vessel on warm routes. Problems resulting are manifold (DHL Express Vertriebs GmbH & Co. OHG, 2005, p. 393). Extremely high temperatures often lead to changes in the physical state of matter, i. e., transitions between solid, liquid, and gas. As a consequence, packages may burst due to thermal expansion. Apart from that, growth of microorganisms may accelerate under increased temperature. This, in turn, may lead to earlier deterioration, self-heating, and even spontaneous combustion. Humidity is closely linked with temperature. Decreasing temperature results in water condensing on the goods loaded in the container because the absorption capacity of air decreases with lower temperatures. Condensation water, in turn, leads to a loss of quality of the goods carried (DHL Express Vertriebs GmbH & Co. OHG, 2005, p. 394).

Both temperature and humidity can be controlled by refrigerated containers. Legal obligations require to prove a gapless cold chain for some goods, e. g., in the food supply chain (Jedermann, Behrens, Westphal & Lang, 2006, p. 370). Therefore, it is necessary to monitor the interior climate of shipping containers (Thiesse, 2005, p. 113). Quality models exist that determine the perishability of particular forms of food in relation to the environmental temperature (Jedermann, Emond & Lang, 2007, pp. 233–234). Detecting a cooling system failure allows re-routing a container to a point of sale nearby. Another container with equal goods can then be re-routed to the original destination.

The positioning of the sensor significantly influences the measurements of the interior climate. In standard containers air temperature is higher near the hull of the container (DHL Express Vertriebs GmbH & Co. OHG, 2005, p. 393). Likewise, the temperature sensed in refrigerated containers depends on the vicinity to the cooling unit and air ventilation. Temperature differences of up to five kelvin require placing multiple sensors in the container (Jedermann, Stein, Becker & Lang, 2008). Besides temperature and humidity, also ethylene is an indicator for food quality (Jedermann, Schouten, Sklorz, Lang & Kooten, 2006, p. 3). The ethylene concentration influences the ripening process of fruits.

Apart from quality, other important issues to be regarded in logistics concern safety and security (Werner et al., 2007). The demand for security in logistics has been brought into focus after the terrorist attacks of September 11, 2001 (Daschkovska & Scholz-Reiter, 2007, p. 305). Shipping containers are particularly considered in these efforts due to their high throughput in the intercontinental transport of packaged goods. The two main objectives are protecting the cargo from thieves as well as preventing terrorists from smuggling dangerous goods. A conventional approach is to employ mechanical seals, e. g., numbered bolts. After loading, the seal is affixed on the container. Before unloading, it has to be validated that the seal is undamaged and that it has not been replaced (Tirschwell, 2005, p. 54).

Conventional seals are comparatively cheap because they only consist of a numbered bolt. A new seal has to be affixed each time a container has been opened legitimately because the seals are not reusable (Hadow, 2005, p. 58); the unique number (Field, 2005, p. 48) of the new seal has to be recorded. Despite of the low purchase price for mechanical seals, their handling cost is quite high because their manual inspection is time consuming (Tirschwell, 2005, p. 54). Mechanical seals increase the effort for tampering with a container. Nevertheless, their benefit is still limited. Hadow (2005, p. 58) elaborates that criminals can remove the doors of the container completely, thereby not damaging the seal. Alternatives are cutting a hole into another wall or creating a new seal after having finished.

Electronic seals are a more sophisticated approach as they notice tamper immediately and alert the cargo owner (Hickey, 2004, p. 34). Container security systems additionally include sensors in order to monitor tampering, theft, and placement of unintentional freight (Figure 3.9). Sensors applied range from light sensors and gamma ray detectors to chemical sensors (Schwartz, 2004, p. 16). In order to save energy and cost, the sensors applied have to chosen with respect to the concrete purpose. Thus, electronic seal and sensors must be able to establish ad hoc networks. Access to such networks has to be restricted in order to exclude untrustworthy sensors that have been placed by thieves or terrorists in order to inject manipulated data. To summarise, augmenting electronic security systems by additional sensors placed inside containers can significantly improve security.

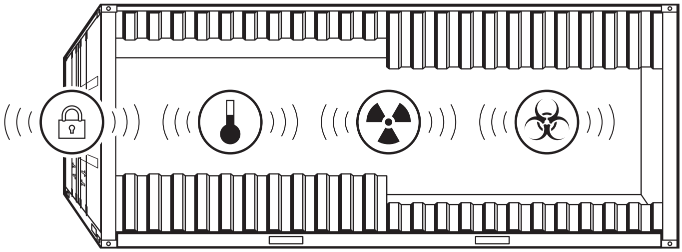

Fig. 3.9 Sensors can be applied in order to monitor the interior of shipping containers. The combination of sensors depends on the task at hand. Common sensors include door lock sensors, temperature sensors, gamma ray detectors, and chemical sensors.

The above examples demonstrate that sensors are an important technology in order to monitor logistics processes. Sensor technology provides the means to guarantee quality and security in logistics. In the applications discussed above, it is not sufficient to place single sensors within a shipping container. Instead, there is a demand for sensor networks. On the one hand, this is due to the fact that the data measured varies depending on the location of the sensor, e. g., temperature in refrigerated containers. On the other hand, different types of sensors may be required in order to measure different phenomena that correlate with, for instance, the ripening process of fruits. This can be achieved by wireless sensor networks (Akyildiz, Su, Sankarasubramaniam & Cayirci, 2002) which allow distributing sensors with a high degree of freedom. Jedermann, Behrens, Laur and Lang (2007, p. 384) describe an approach to automatically configure sensor systems depending on the goods transported. The configuration can be determined during loading of freight, e. g., by reading an RFID chip that is attached on the goods.

Apart from flexible configuration, energy consumption is another important issue when dealing with wireless sensor networks. Being distributed within a shipping container, sensors cannot be supplied from a central power source. Instead, they must be equipped with batteries. This, in turn, raises requirements regarding economical energy consumption because battery lifetime is limited (Section 3.1.3). A particularly high amount of energy is consumed by wireless communication between the sensor nodes and their base station. Therefore, sensor nodes should cooperate in order to decrease the amount of energy spent during the transmission of collected data. This can be accomplished by applying clustering algorithms that aim at optimising energy consumption for communication purposes. Clustering sensors by spatial proximity is a common approach. Routing data messages is then organised in a hierarchical way (Al-Karaki & Kamal, 2004). Each cluster collects the sensor data of its environment, aggregates it, and transmits it to the base station. A prominent approach is LEACH, which stands for low-energy adaptive clustering hierarchy (Heinzelman, Chandrakasan & Balakrishnan, 2000). In this method, some sensor nodes choose to be cluster-heads. The remaining nodes join the cluster that requires minimum communication energy. If sensor nodes generally choose the spatially closest cluster-head, a Voronoi (Klein, 2005, pp. 83–102) tessellation evolves (Figure 3.10). To distribute power consumption for long-range communication, the cluster-heads are regularly changed, which also leads to a new cluster partitioning.

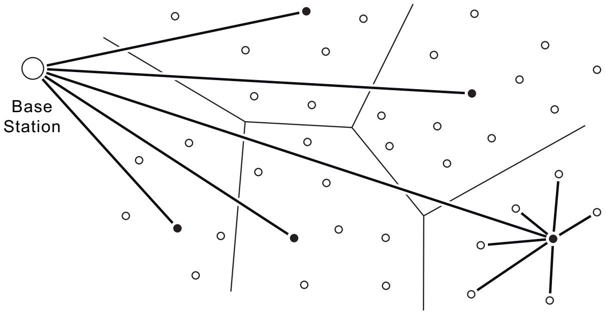

Fig. 3.10 Nodes (transparent circles) of a wireless sensor network. In order to save energy, sensors cluster themselves in order to aggregate data locally. Dynamically chosen cluster heads (opaque circles) then transmit the data sensed to their base station.

To summarise, sensor technology enables monitoring of goods handled in logistics processes. Applying sensors allows ensuring both quality and security. The actual type and number of sensors chosen depends on the concrete objective. Wireless sensor networks allow dynamically combining the independent sensor nodes required. The limited amount of power that is provided by batteries can be saved by appropriate clustering methods. The data acquired locally by wireless sensor networks can be applied in both conventional and autonomous logistics. However, pre-processing data and making decisions on the local level enables immediate reactivity without waiting for a central entity, a potential bottleneck. Additional technologies required for this objective are discussed in the following sections.

3.2.4 Communication

Identification, localisation, and sensor unit provide autonomous logistics entities with information about themselves, their condition, and their environment. Interaction with the outside world requires an additional unit, namely the communication unit. This part of the architecture provides the means to exchange with other autonomous entities or humans by sending and receiving messages.

Today, communication has become cheap for many applications. Broadband communication providers offer access to their guided networks with high bandwidth and low charges. But logistics entities are usually widely distributed, often over multiple continents. Furthermore, they are rarely stationary. Instead, they move through the logistics network. Therefore, autonomous logistics entities cannot employ classical guided transmission media, i. e., copper wire or fibre optics (Tanenbaum, 2003, p. 85). Instead, wireless or satellite communication has to be applied. From the point of view of an autonomous logistics entity it does not make much difference whether guided or wireless media are accessed. Architectures like the ISO OSI (Open Systems Interconnection) reference model and TCP/IP provide an abstraction from the underlying physical layer, thereby enabling end-to-end connections on the application layer (Tanenbaum, 2003, pp. 37–49).

This allows logistics entities communicating with each other independently of the underlying network infrastructure. Differences arise, of course, regarding availability and cost of communication. One possibility is to establish private wireless local area networks (WLANs) to connect autonomous logistics entities to guided networks. This approach, however, is only reasonable in intralogistics. For instance, consider trucks on a highway. Even assumed that connection to wired networks was possible, one would need many wireless access points. For containers on the ocean it is even more obvious that there is virtually no possibility to connect them to wired networks. Wireless communication over cellular networks or communication satellites could provide an alternative. The problem, however, is that bandwidth of these communication channels is comparatively limited. Therefore, higher utilisation fees are charged for these networks. Another cost factor for communication in autonomous logistics is the energy needed for mobile communication. This section presents the application of communication for reporting and interaction. Thereby, standards for message exchange and security issues are discussed.

Radio-frequency identification is one application for communication (Section 3.2.1). Furthermore, communication is also required to transmit localisation and sensor information in order to implement real-time visibility of supply networks. This information can then be centrally evaluated in order to react appropriately if the conditions of the cargo or its environment change. Consider detecting an increase in temperature level within a shipping container because its cooling unit fails. As described in Section 3.2.3, the remaining shelf life of food depends on the environmental temperature. In case of increasing temperature, it might thus be appropriate to re-route the container to a nearer sink or to re-pack the cargo onto another load carrier. In centralised control approaches one would simply provide a central unit with all sensor measurements.

Akyildiz et al. (2002, p. 403) point out that, that communication generally consumes more energy than sensing and data processing. Therefore, it is advisable to use communication as economically as possible. This can be achieved by pre-processing sensory data before transmitting it to other autonomous or centralised entities. For instance, it is not necessary to send all measurements by a temperature sensor to a dispatcher. By contrast, the dispatcher is only interested in being notified about substantial changes (Jedermann, Behrens et al., 2006, p. 370). Thus, it is sufficient to inform him if certain thresholds are exceeded. Autonomous logistics goes even one step further. In this paradigm, autonomous logistics entities may take actions required to react appropriately themselves or at least to propose possible reactions. Reasoning about adequate reactions is not conducted by the communication unit but by the data processing unit of the autonomous logistics entity (Section 3.2.5).

Apart from reacting to environmental changes, communicating autonomous logistics entities may also cooperate with others in order to achieve their logistics objectives. The communication unit is the interface to the outside world for coordination with other entities. Communication with such partners is generally conducted on an ad hoc basis (Section 3.1.1). Firstly, this requires a common vocabulary that is understood by all participating entities (Hellenschmidt & Wichert, 2007, p. 94, 102). This issue is, for instance, addressed by the EDIFACT standard, which is an acronym for Electronic Data Interchange for Administration, Commerce and Transport (Vahrenkamp, 2007, p. 56). Secondly, the communication unit is also a vulnerable point that must be secured appropriately (Werner et al., 2007).

As an example, consider a shipping container that is currently located at a container terminal. This container is loaded with hazardous material. Communication regarding the content of the container must then be restricted to trustworthy partners. From the perspective of safety it might be desirable for such containers to inform the environment about their hazardous content. But this is not the case from a security point of view. It is not advisable to broadcast the attractiveness of a container for terrorist attacks to everyone including the terrorists themselves (Hadow, 2005, p. 58). Only personnel at the container terminal, such as stevedores and truckers, is a legitimate recipient of some security-related data. Therefore, one must ensure that only legitimate recipients obtain data from the container security system. This can be accomplished by encryption. Additionally, it is important to clearly identify authorised cooperation partners. This issue is addressed by signatures.

Both demands for trustworthiness can be addressed by applying public key cryptography (Tanenbaum, 2003, pp. 752–755) which is based on pairs of asymmetric encryption keys. In this approach, the public key of each entity is known to everyone and is applied in order to encrypt the content of messages for the respective entity (Figure 3.11). Decrypting such contents can only be accomplished by applying the private key which is concealed and only known to the entity itself. A sender additionally signing the message with its own private key enables the receiver to validate its authenticity with the respective public key. To be capable of identifying trustworthy communication partners each entity must be provided with the public key of the company. The respective private key can, however, not simply be provided to all participants. Otherwise, the whole system runs into danger of being compromised, for instance, if a hand-held device with the key is lost. A finder or thief would then be able to decrypt all messages intended for the company. Instead, a public key infrastructure (Tanenbaum, 2003, pp. 768–771) has to be established. In this concept each entity gets its own private key that is signed by the private root certificate of the company. Validating the respective public keys with the public key of the company then reveals whether a communication partner is trustworthy. The problem of loss can be approached by expiring keys that must be renewed regularly.

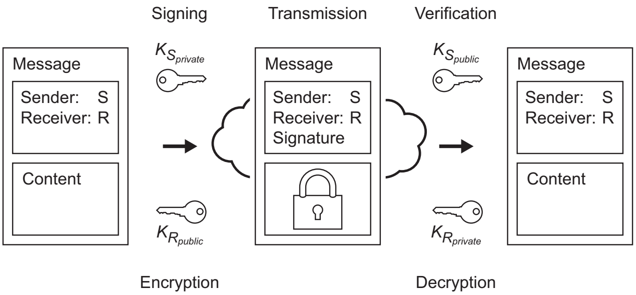

Fig. 3.11 Two autonomous logistics entities communicate with pairs of public and private keys. Sender S signs and encrypts the content of its message before transmission. The received message is then validated and decrypted by the receiver R.

To summarise, communication is needed for both cooperation between logistics entities as well as interaction with other participants in logistics processes. Cooperation may include coordination regarding joint utilisation of transport or storage capacities. Examples for other participants are owners who are informed about the current condition of their logistics entities. For security reasons communication must be restricted to trustworthy partners. This can be achieved by encryption and signatures. Like for localisation and sensor technology, it is not necessary to provide all autonomous logistics entities with all communication facilities. Instead, it might be sufficient that entities at a high granularity are capable of long-range communication. That is, trucks could have access to cellular networks. Container vessels may use a satellite uplink. Other entities loaded, such as packets or containers, can then share the communication infrastructure of the respective load carrier.

3.2.5 Data Processing

The technologies for autonomous entities examined so far lay the foundations for enabling autonomous control in logistics. The identification unit enables each entity to identify itself and to recognise other entities (Section 3.2.1). The localisation unit allows gathering tracking and tracing information for autonomous entities (Section 3.2.2). Combined with additional sensor technology, this helps ensure quality as well as safety and security in logistics (Section 3.2.3). The communication unit allows interacting with other logistics entities and to provide the cargo owner with information on the current state (Section 3.2.4).

However, these units are necessary but not sufficient to implement autonomous logistics. They could also be applied in order to improve conventional centralised supply network management. Then, all identification, localisation, and sensor information is provided to a central decision-making entity. By contrast, the autonomous logistics paradigm envisions that decisions are made locally by the logistics entities themselves. But none of the units investigated so far has the capability for decision-making. To this end, the data processing unit is the crucial part to implement autonomous control in logistics. In the architecture for autonomous logistics entities (Figure 3.3), the data processing unit integrates all other parts, thereby enabling decisions to be made locally.

The miniaturisation of integrated circuits allows attaching the data processing unit directly to the respective logistics entity (Section 3.1.2). Such embedded systems enable truly decentralised decision-making without centralised control (e. g., Jedermann, Antúnez Congil et al., 2007, p. 193–194). As an example, consider several shipping containers of the same company that are supposed to be loaded onto a container vessel in a port in East Asia. Furthermore, consider that it turns out that the vessel is overbooked and that not all containers can be loaded. It is then possible that the containers directly negotiate which of them are most important based on the priority of the cargo loaded. If the containers have the knowledge demanded for that decision, no intervention by a human dispatcher is required.

Distribution usually constrains the resources available (Section 3.1.2). On the one hand, resources are bound regarding the power available. But on the other hand, computational and memory capacities are also limited. As an alternative, intelligent representatives that are located on one or multiple central servers can act on behalf of the logistics entity represented by them. This is the variation of autonomous logistics that represents the smallest deviation from truly distributed decision-making. Nevertheless, the advantages of reduced complexity and the improved ability of coping with dynamics are retained. It is, of course, only applicable in environments with guaranteed communication connectivity to the autonomous logistics entity. The particular advantage is that a more powerful central server offers more computational power to representatives of logistics entities than embedded devices. Nevertheless, also central servers do not possess unbounded resources because they are likewise constrained by the limitations of asymptotical computational complexity regarding time and space (Section 2.3.1). Thus, sophisticated knowledge management approaches are required. Methods applied must consider learning of relevant, but also forgetting of irrelevant knowledge (Werner et al., 2007, p. 15).

3.3 Conclusion

Conventional supply network management from a centralised point is limited by the complexity, the dynamics, and the distribution that are inherent in logistics processes. In contrast to previous approaches, autonomous control in logistics delegates decision-making to the local logistics objects, e. g., sales units, cardboard boxes, pallets, containers, and trucks. These autonomous logistics entities are provided with logistics objectives defined by their owners. They must then cooperate in order to achieve these objectives. The miniaturisation of integrated circuits and other technologies allows enhancing logistics entities with the capabilities required. However, technological and economical limitations demand finding an adequate granularity at which autonomous control is applied. Additionally, also the degree of freedom in decision-making must be constrained. Technologies enabling autonomous logistics include identification, localisation, sensors, communication, and data processing. The data processing unit is the crucial part for autonomous logistics entities. The important question is how to actually implement the Artificial Intelligence required by autonomous logistics entities.

References

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless Sensor Networks: A Survey. *Computer Networks*, *38*(4), 393–422.

Al-Karaki, J. N. & Kamal, A. E. (2004). Routing Techniques in Wireless Sensor Networks: A Survey. *IEEE Wireless Communications*, *11*(6), 6–28.

Arnold, D., Isermann, H., Kuhn, A., Tempelmeier, H. & Furmans, K. (Eds.). (2008). *Handbuch Logistik* (3rd ed.). Heidelberg, Germany: Springer-Verlag.

Becker, M., Kuladinithi, K., Timm-Giel, A. & Görg, C. (2007). Historical Development of the Idea of Self-Organisation in Information and Communication Technology. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics: The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 45–56). Heidelberg, Germany: Springer-Verlag.

Berning, M. & Vastag, S. (2007). Simulation selbststeuernder Transportnetze. In H.-J. Bullinger & M. ten Hompel (Eds.), *Internet der Dinge* (pp. 119–126). Heidelberg, Germany: Springer-Verlag.

Bullinger, H.-J. & ten Hompel, M. (Eds.). (2007). *Internet der Dinge.* Heidelberg, Germany: Springer-Verlag.

Cardeneo, A. (2008). Kurier-, Express- und Paketdienste. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 782–788). Heidelberg, Germany: Springer-Verlag.

Daschkovska, K. & Scholz-Reiter, B. (2007). Electronic Seals for Efficient Container Logistics. In H.-D. Haasis, H.-J. Kreowski & B. Scholz-Reiter (Eds.), *1st International Conference on Dynamics in Logistics (LDIC 2007)* (pp. 305–312). Bremen, Germany: Springer-Verlag.

DHL Express Vertriebs GmbH & Co. OHG. (2005). *Logistik-Lotse 2005* (13th ed.). Bonn, Germany: Verkehrs-Verlag Fischer.

Duda, R. O. & Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* New York, NY, USA: John Wiley & Sons.

Field, A. M. (2005). Seals of Approval. *Journal of Commerce*, *6*(38), 46–48.

Fleisch, E. & Mattern, F. (Eds.). (2005). *Das Internet der Dinge.* Heidelberg, Germany: Springer-Verlag.

Flörkemeier, C. (2005). EPC-Technologie – vom Auto-ID Center zu EPCglobal. In E. Fleisch & F. Mattern (Eds.), *Das Internet der Dinge* (pp. 87–100). Heidelberg, Germany: Springer-Verlag.

Freitag, M., Herzog, O. & Scholz-Reiter, B. (2004). Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen. *Industrie Management*, *20*(1), 23–27.

Gallagher, J. (2002). Test Tracking. *Traffic World*, *266*(46), 48.

Garson, G. D. & Biggs, R. S. (1992). *Analytic Mapping and Geographic Databases.* Newbury Park, CA, USA: Sage Publications.

Gottfried, B., Schuldt, A. & Herzog, O. (2007). Extent, Extremum, and Curvature: Qualitative Numeric Features for Efficient Shape Retrieval. In J. Hertzberg, M. Beetz & R. Englert (Eds.), *30th Annual German Conference on Artificial Intelligence (KI 2007)* (pp. 308–322). Osnabrück, Germany: Springer-Verlag.

Haasis, H.-D., Kreowski, H.-J. & Scholz-Reiter, B. (Eds.). (2007). *1st International Conference on Dynamics in Logistics (LDIC 2007).* Bremen, Germany: Springer-Verlag.

Hadow, R. (2005). E-Seals and RFID. *Journal of Commerce*, *6*(43), 58.

Hannon, D. (2002). High-Tech Security Becomes Top Priority in Supply Chain. *Purchasing*, *131*(11), 39–41.

Heinzelman, W. R., Chandrakasan, A. & Balakrishnan, H. (2000). Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *33rd Hawaii International Conference on System Sciences (HICSS 2000)* (Vol. 8, pp. 8020–8029). Maui, HI, USA: IEEE Computer Society.

Hellenschmidt, M. & Wichert, R. (2007). Selbstorganisation: Dinge in eigenverantwortlicher Kooperation – eine Systemanalyse. In H.-J. Bullinger & M. ten Hompel (Eds.), *Internet der Dinge* (pp. 91–105). Heidelberg, Germany: Springer-Verlag.

Herzog, O., Freitag, M. & Scholz-Reiter, B. (2005). Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel. In Bundesvereinigung Logistik (BVL) e.V. (Ed.), *22. Deutscher Logistik-Kongress* (pp. 219–224). Berlin, Germany: Deutscher Verkehrs-Verlag.

Hickey, K. (2004). Insecurity Over E-Seals. *Traffic World*, *268*(3), 34.

Hülsmann, M. & Windt, K. (Eds.). (2007). *Understanding Autonomous Cooperation and Control in Logistics: The Impact of Autonomy on Management, Information, Communication and Material Flow.* Heidelberg, Germany: Springer-Verlag.

Hülsmann, M., Wycisk, C., Agarwal, R. & Grapp, J. (2007). Prologue to Autonomous Cooperation – the Idea of Self-Organisation as its Basic Concepts. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics: The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 23–44). Heidelberg, Germany: Springer-Verlag.

Hofmann-Wellenhof, B., Lichtenegger, H. & Wasle, E. (2008). *GNSS Global Navigation Satellite Systems: GPS, GLONASS, Galileo and more.* Wien, Austria: Springer-Verlag.

Jedermann, R., Antúnez Congil, L. J., Lorenz, M., Gehrke, J. D., Lang, W. & Herzog, O. (2007). Dynamic Decision Making on Embedded Platforms in Transport Logistics – A Case Study. In H.-D. Haasis, H.-J. Kreowski & B. Scholz-Reiter (Eds.), *1st International Conference on Dynamics in Logistics (LDIC 2007)* (pp. 191–198). Bremen, Germany: Springer-Verlag.

Jedermann, R., Behrens, C., Laur, R. & Lang, W. (2007). Intelligent Containers and Sensor Networks Approaches to apply Autonomous Cooperation on Systems with Limited Resources. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics: The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 365–392). Heidelberg, Germany: Springer-Verlag.

Jedermann, R., Behrens, C., Westphal, D. & Lang, W. (2006). Applying Autonomous Sensor Systems in Logistics — Combining Sensor Networks, RFIDs and Software Agents. *Sensors and Actuators A*, *132*(1), 370–375.

Jedermann, R., Emond, J.-P. & Lang, W. (2007). Shelf Life Prediction by Intelligent RFID – Technical Limits of Model Accuracy. In H.-D. Haasis, H.-J. Kreowski & B. Scholz-Reiter (Eds.), *1st International Conference on Dynamics in Logistics (LDIC 2007)* (pp. 231–238). Bremen, Germany: Springer-Verlag.

Jedermann, R., Gehrke, J. D., Becker, M., Behrens, C., Morales-Kluge, E., Herzog, O. et al. (2007). Transport Scenario for the Intelligent Container. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics: The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 393–404). Heidelberg, Germany: Springer-Verlag.

Jedermann, R., Schouten, R., Sklorz, A., Lang, W. & Kooten, O. van. (2006). Linking Keeping Quality Models and Sensor Systems to an Autonomous Transport Supervision System. In J. Kreyenschmidt & B. Petersen (Eds.), *2nd International Workshop on Cold Chain Management* (pp. 3–18). Bonn, Germany.

Jedermann, R., Stein, K., Becker, M. & Lang, W. (2008). UHF-RFID in the Food Chain – From Identification to Smart Labels. In J. Kreyenschmidt (Ed.), *3rd International Workshop on Coldchain Manangement.* Bonn, Germany.

Köhler-Bußmeier, M. (2009). Emergenz, Selbstorganisation und Komplexität. In R. v. Lüde, D. Moldt & R. Valk (Eds.), *Selbstorganisation und Governance in künstlichen und sozialen Systemen* (pp. 17–71). Münster, Germany: LIT Verlag.

Klein, R. (2005). *Algorithmische Geometrie: Grundlagen, Methoden, Anwendungen* (2nd ed.). Heidelberg, Germany: Springer-Verlag.

Kuhlmann, F. & Masuhr, D. (2007). Prozesse in offenen Systemen gestalten. In H.-J. Bullinger & M. ten Hompel (Eds.), *Internet der Dinge* (pp. 257–265). Heidelberg, Germany: Springer-Verlag.

Lampe, M., Flörkemeier, C. & Haller, S. (2005). Einführung in die RFID-Technologie. In E. Fleisch & F. Mattern (Eds.), *Das Internet der Dinge* (pp. 69–86). Heidelberg, Germany: Springer-Verlag.

Martin, H. (2006). *Transport- und Lagerlogistik – Planung, Struktur, Steuerung und Kosten von Systemen der Intralogistik.* Wiesbaden, Germany: Friedrich Vieweg & Sohn Verlag.

Mattern, F. (2005). Die technische Basis für das Internet der Dinge. In E. Fleisch & F. Mattern (Eds.), *Das Internet der Dinge* (pp. 39–66). Heidelberg, Germany: Springer-Verlag.

Matthias, A. (2008). *Automaten als Träger von Rechten. Plädoyer für eine Gesetzesänderung.* Berlin, Germany: Logos Verlag.

Moore, G. E. (1965). Cramming More Components onto Integrated Circuits. *Electronics*, *38*(8), 114-117.

Nitschke, T. (2006). Legal Consequences of Agent Deployment. In S. Kirn, O. Herzog, P. Lockemann & O. Spaniol (Eds.), *Multiagent Engineering: Theory and Applications in Enterprises* (pp. 597–618). Heidelberg, Germany: Springer-Verlag.

Palmer, S. E. (1999). *Vision Science: Photons to Phenomenology.* Cambridge, MA, USA: MIT Press.

Pekow, C. (2005). GPS Proven Best for Hazmat Tracking. *LP/GAS*, *65*(3), 14.

Pflaum, A. & Hupp, J. (2007). Sensornetzwerke und Lokalisierungsverfahren als Schlüsseltechnologien für die intelligente logistische Umwelt von morgen. In H.-J. Bullinger & M. ten Hompel (Eds.), *Internet der Dinge* (pp. 107–118). Heidelberg, Germany: Springer-Verlag.

Scholz-Reiter, B., Toonen, C. & Windt, K. (2008). Logistikdienstleistungen. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 581–607). Heidelberg, Germany: Springer-Verlag.

Scholz-Reiter, B., Windt, K. & Freitag, M. (2004). Autonomous Logistic Processes — New Demands and First Approaches. In *37th CIRP International Seminar on Manufacturing Systems* (pp. 357–362). Budapest, Hungary.

Schuldt, A. & Gottfried, B. (2008a). Selbststeuerung in der Intralogistik: Kognitive räumliche Repräsentationen für autonome Fahrzeuge. *Industrie Management*, *24*(4), 41–44.

Schuldt, A. & Gottfried, B. (2008b). Spatiotemporal Cooperation at the Geographic Scale. In B. Loos & V. Micelli (Eds.), *Interdisciplinary College (IK 2008)* (p. 32). Günne, Germany.

Schuldt, A., Gottfried, B. & Herzog, O. (2006a). A Compact Shape Representation for Linear Geographical Objects: The Scope Histogram. In S. Nittel & R. A. de By (Eds.), *14th ACM International Symposium on Advances in Geographic Information Systems (GIS 2006)* (pp. 51–58). Arlington, VA, USA: ACM Press.

Schuldt, A., Gottfried, B. & Herzog, O. (2006b). Retrieving Shapes Efficiently by a Qualitative Shape Descriptor: The Scope Histogram. In H. Sundaram, M. Naphade, J. Smith & Y. Rui (Eds.), *5th International Conference on Image and Video Retrieval (CIVR 2006)* (pp. 261–270). Tempe, AZ, USA: Springer-Verlag.

Schuldt, A., Gottfried, B. & Herzog, O. (2006c). Towards the Visualisation of Shape Features: The Scope Histogram. In C. Freksa, M. Kohlhase & K. Schill (Eds.), *29th Annual German Conference on Artificial Intelligence (KI 2006)* (pp. 289–301). Bremen, Germany: Springer-Verlag.

Schuldt, A. & Werner, S. (2007). Distributed Clustering of Autonomous Shipping Containers by Concept, Location, and Time. In P. Petta, J. P. Müller, M. Klusch & M. Georgeff (Eds.), *5th German Conference on Multiagent System Technologies (MATES 2007)* (pp. 121–132). Leipzig, Germany: Springer-Verlag.

Schwartz, E. (2004). The Cost of Safe Commerce. *InfoWorld*, *26*(45), 16.

Tanenbaum, A. S. (2003). *Computer Networks* (4th ed.). Upper Saddle River, NJ, USA: Pearson Education.

ten Hompel, M., Schmidt, T., Nagel, L. & Jünemann, R. (2007). *Materialflusssysteme — Förder- und Lagertechnik*. Heidelberg, Germany: Springer-Verlag.

Thiesse, F. (2005). Architektur und Integration von RFID-Systemen. In E. Fleisch & F. Mattern (Eds.), *Das Internet der Dinge* (pp. 101–117). Heidelberg, Germany: Springer-Verlag.

Timm, I. J. (2006). *Strategic Management of Autonomous Software Systems: Overview Article* (Tech. Rep. No. 35). Universität Bremen: Technologie-Zentrum Informatik.

Tirschwell, P. (2005). An Opportunity from Container Seals. *Journal of Commerce*, *6*(6), 54.

Trautmann, A. (2007). Multiagentensysteme im Internet der Dinge – Konzepte und Realisierung. In H.-J. Bullinger & M. ten Hompel (Eds.), *Internet der Dinge* (pp. 281–294). Heidelberg, Germany: Springer-Verlag.

Vahrenkamp, R. (2007). *Logistik — Management und Strategien* (6th ed.). Munich, Germany: Oldenbourg.

Vastag, S. (2008). Simulative Bewertung von Selbststeuerung in Transportnetzen. In M. Rabe (Ed.), *13th ASIM Dedicated Conference on Simulation in Production and Logistics (ASIM 2008)* (pp. 219–228). Berlin, Germany: Fraunhofer IRB Verlag.

Wenning, B.-L., Rekersbrink, H., Timm-Giel, A., Görg, C. & Scholz-Reiter, B. (2007). Autonomous Control by Means of Distributed Routing. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics: The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 325–335). Heidelberg, Germany: Springer-Verlag.

Werner, S., Schuldt, A. & Daschkovska, K. (2007). Agent-Based Container Security Systems: An Interdisciplinary Perspective. In R. Koschke, O. Herzog, K.-H. Rödiger & M. Ronthaler (Eds.), *37th Annual Conference of the German Society for Computer Science (GI 2007)* (pp. 11–16). Bremen, Germany: Gesellschaft für Informatik.

Windt, K. (2008). Ermittlung des angemessenen Selbststeuerungsgrades in der Logistik – Grenzen der Selbststeuerung. In P. Nyhuis (Ed.), *Beiträge zu einer Theorie der Logistik* (pp. 349–372). Heidelberg, Germany: Springer-Verlag.

Windt, K. & Hülsmann, M. (2007). Changing Paradigms in Logistics – Understanding the Shift from Conventional Control to Autonomous Cooperation and Control. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics: The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 1–16). Heidelberg, Germany: Springer-Verlag.

Wöstmann, F.-J. (2006). Intelligente Gussbauteile: Wenn Bauelemente mit ihrer Umgebung interagieren. *RFID im Blick*, *7*, 49.

Part II

Multiagent-Based Approach

Chapter 4
Agent Technology

The autonomous logistics paradigm aims at decreasing the overall problem complexity of supply network management (Section 3.1). To this end, process control is delegated to the participating logistics entities. The question is how this principle can be represented appropriately in a software implementation. Each local entity must itself be enabled to make its decisions. More precisely, the data processing unit of autonomous logistics entities is responsible for decision-making (Section 3.2). This specification is useful for distinguishing data processing from other parts of the logistics entity. The challenging task of implementing decentralised process control, however, is only shifted from the whole entity to one of its parts. In order to prevent ending up with an infinitely nested partitioning, it is important to specify how this unit can actually be implemented. Intelligent software agents are a promising candidate for implementing autonomous logistics (Jennings & Wooldridge, 1998, pp. 14–15). This finding can be motivated by the fact that the requirements for autonomous logistics entities are directly reflected by the description of software agents (Wooldridge, 2002, p. 15):

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

Amongst others, Parunak (1999, pp. 378–381) explicates that software agents are appropriate for many industrial applications. This position is supported by Jennings (2001, p. 35) who states that the agent-oriented software engineering paradigm is well-suited for implementing complex and distributed systems. Kirn, Herzog, Lockemann and Spaniol (2006, p. V) conclude that after several years of academic research, multiagent systems are mature for commercial application.

Jennings (2001, p. 37) particularly emphasises the decomposition as well as the abstraction and organisation of multiagent systems. First of all, agent-oriented software engineering allows decomposing computational problems by dividing them into appropriate subproblems (Jennings, 2001, pp. 37–38).

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics*,
DOI 10.1007/978-3-642-20092-2_4,

This directly corresponds to the approach of autonomous logistics. Each autonomous logistics entity can be represented by a software agent that acts on its behalf. The logistics entities and thus their agents are only loosely coupled (Parunak, 1999, p. 379). They interact flexibly on demand. Nevertheless, certain relationships exist between them. The agent paradigm offers concepts to represent such relationships (Jennings, 2001, p. 38), e. g., by interaction protocols and organisational structures. This allows agents modelling the relationships between their real-world counterparts.

Müller (1997, p. 218) proposes three minimal criteria for applications of agent technology. According to him, an intended application should exhibit a natural distribution, demand for flexible interaction, and be embedded in a dynamic environment. It is obvious that the logistics domain satisfies these criteria. Logistics processes and therefore their participants are highly distributed (Section 2.3.3), often over multiple continents. This is addressed by decentralisation in both autonomous logistics and multiagent systems. The demand for flexible interaction in logistics arises from highly dynamic environments (Section 2.3.2). For instance, participants are not known in advance. Hence, it is often virtually impossible to reliably predict which cargo has to be transported in the middle-term future.

Section 4.1 gives a general overview on the paradigm of intelligent software agents. Its objective is twofold. Firstly, it gives a more detailed introduction of what software agents are. Secondly, it gives an impression of the broad variety of possible agent implementations. Section 4.2 focuses on agent interaction in multiagent systems. Cooperation is necessary whenever agents cannot achieve their objectives on their own. Agents must thus be able to interact despite of their differences. Finally, Section 4.3 discusses organisation of agents for long-term collaboration as well as existing applications of agents in logistics.

4.1 Intelligent Software Agents

Intelligent software agents are supposed to represent autonomous logistics entities and to act on their behalf. It is thus necessary to create intelligent agents according to the architecture for autonomous logistics entities (Section 3.2). The agent definition by Wooldridge cited in the introductory section of this chapter covers two components, the agent and its environment. The agent interacts with the environment in order to achieve its objectives. This interaction is done by means of sensors and actuators (Figure 4.1). Sensors enable the agent to perceive its environment, actuators allow acting upon the environment. The integration into the architecture for autonomous logistics entities is thus as follows. The software agent implements the data processing unit of autonomous logistics entities. Its interfaces to the outside world are the identification, localisation, sensor, and communication unit (Figure 3.3). The identification unit allows identifying other agents and objects. The localisation unit enables the agent to find out where its logistics entity and thus itself are located. Applying the sensor unit allows getting measurements about the logistics object itself as well as its environment. Identification, localisation, and sensor unit of the autonomous logistics entity can thus be regarded as the sensors of the agent. The communication unit is both sensor and actuator of the agent because it can send and receive messages.

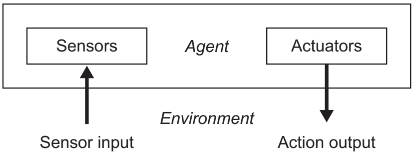

Fig. 4.1 An agent interacts with its environment. The agent can perceive the environment by means of its sensors. It can act upon the environment by means of its actuators (adapted from Wooldridge, 2002, p. 16).

Section 4.1.1 discusses important properties that characterise intelligent agents. Based on this foundation, Section 4.1.2 presents general models to implement agents in accordance with these characteristics.

4.1.1 Characteristics of Intelligent Agents

A huge amount of applications exist that already incorporate agent technology. Consequently, also the understanding of the term agent is not clear cut and differs between authors (Nwana & Ndumu, 1998, p. 29). However, today most authors agree on the minimum criteria for intelligent agents proposed by Wooldridge (1999, p. 32):

1. Autonomy
2. Reactivity
3. Pro-activeness
4. Social ability

The autonomy criterion refers to the fact that agents must be able to decide for themselves which actions they should take in order to achieve their goals. This is an important criterion for autonomous logistics entities that are situated in a dynamic environment. In supply network management, it is generally impossible to determine the exact sequence of necessary actions in advance. And even if such a universal sequence should exist for a process, it would be generally impossible to predefine the exact times at which the actions must be executed. Autonomous logistics objects and thus their agents must decide for themselves which actions are appropriate in a concrete situation. That is, agents decide for themselves whether and when they execute a particular action. Their autonomy also covers rejecting to execute actions requested by other agents. An example in autonomous logistics is a truck that may reject to transport a particular shipping container because it can earn a higher reward by transporting other cargo. The truck could even decide to transport no cargo at all. In doing so, it might achieve the best benefit if the transport rates do not cover fuel prices.

Granting an agent the autonomy to make decisions on its own is not sufficient for intelligent behaviour. Additionally, it has to be capable of flexible behaviour. This covers both reactivity and pro-activeness (Weiß & Jakob, 2005, p. 4). Reactivity means that an agent is able to react appropriately to changes in the environment, e. g., by adapting its planning. Such changes also include that actions of the agents may fail unexpectedly. An example requirement for reactive behaviour in autonomous logistics includes shipping containers with perishable fruits (Section 3.2.3). Consider that the temperature within the container increases, thereby shortening the lifetime of its goods. In order to adapt to this new situation, an intelligent agent might decide to re-route the container to another location nearby. Pro-activeness, in contrast, means that the agent takes adequate actions in order to achieve one or multiple goals. That is, the agent must anticipate future developments in order to plan appropriately. An example is an intelligent shipping container that plans and schedules its way through a logistics network itself (Section 3.1.1). In order to achieve its objectives, an agent must often collaborate with other agents. Agents are only loosely coupled with each other (Parunak, 1999, p. 379). Social ability is required in order to coordinate with others, e. g., by some form of communication. An intelligent shipping container must, for instance, negotiate with service providers like trucks about its transport.

The internal state of agents is completely encapsulated like the state of objects in the object-oriented programming paradigm (Genesereth & Ketchpel, 1994, p. 48). Information hiding enables agents from different companies to interact with each other without the necessity to reveal their internal structures for decision-making (Parunak, Savit & Riolo, 1998, p. 21). Despite of this similarity, agents clearly distinguish from objects. Objects are passive in the sense that they are manipulated by transition rules or programs (Herrler & Klügl, 2006, p. 579). Every other object can call methods on an object in order to request data processing, at least if the methods are publicly available. By contrast, software agents act autonomously, all interaction with the environment or other agents is conducted by perception and communication. In contrast to object-oriented programming, it is thus impossible to directly invoke certain methods on agents. As part of their autonomy, agents control their internal state themselves. Hence, agents also decide for themselves which actions they take. Other agents may try to convince them for some action by communication. Organisational structures may facilitate such requests (Köhler-Bußmeier, 2009a, p. 99). Despite of this difference, the software agent approach enables problem decomposition like object-oriented programming. Jennings (2001, p. 39) even points out that decomposition is often more effective by agents than by objects. He criticises that individual objects in object-oriented software engineering often have a too fine granularity. In particular, agents are not limited to representing individuals but can also cover abstract units like companies or other groups of individuals (Herrler & Klügl, 2006, p. 579). In autonomous logistics, one is particularly interested in agents that represent logistics entities. Besides, agents might also act as representatives for groups of entities.

The above criteria, autonomy, reactivity, pro-activeness, and social ability, are also referred to as the weak notion of agency (Wooldridge & Jennings, 1995, pp. 116–117). According to Wooldridge and Jennings, also stronger definitions exist which are particularly used by researchers in Artificial Intelligence. The strong notion of agency requires characterising agents by mentalistic notions (Wooldridge & Jennings, 1995, pp. 118–129).

4.1.2 General Agent Models

The preceding section discusses minimal criteria that characterise intelligent agents. These criteria only define a lower bound. There are still infinitely many ways to implement concrete agents. The actual implementation depends, for instance, on the logistics task to be solved. Nevertheless, it is possible to distinguish some abstract agent models which can be used to categorise concrete agents. On the most general level, Wooldridge (1999, pp. 36–41) distinguishes agents with and without state. Russell and Norvig (2010, p. 47) outline four general agent models, namely simple and model-based reflex agents as well as goal-based and utility-based agents. These models as well as example applications in logistics are discussed in the following paragraphs in order to illustrate the bandwidth of actual agent implementations.

The simplest agent model discussed by Russell and Norvig (2010, pp. 48–50) is the so-called simple reflex agent (Figure 4.2 left). This type of agent selects which actions it executes solely based on its current perception of the environment. That is, the agent does not maintain a history of its previous perceptions. The knowledge of such agents is represented by a set of rules. The name reflex agent is derived from the fact that the agent relies on these rules that define which actions to take given a particular state of the environment. An example application is tracking and tracing as described in Section 3.2.2. In that scenario, the reflex agent could act as a gatekeeper that scans incoming packages with RFID. Based on its internal rule table, it could then inform the owner of the respective package about successful reception. However, one could raise the question whether it is really necessary to apply an agent for this task. Or, even more provocative, whether one should call this software unit an agent. In order to cope with more complex tasks, it is necessary to extend the agent model. Agents can usually perceive only parts of their environment. A solution to this dilemma is to maintain previous perceptions in order to incrementally built up a model of the environment. This, however, is a particular problem for the simple reflex agent because it has no means for this task. The model-based reflex agent (Figure 4.2 right) overcomes this limitation of the simple reflex agent (Russell & Norvig, 2010, pp. 50–52). It maintains its own internal state as well as its knowledge about the world. Based on this knowledge and new perceptions, it updates its internal state. Subsequently, it chooses an action based on its internal model in the same way as the simple reflex agent. This allows implementing agents even for more sophisticated applications in logistics. Consider the quality monitoring scenario described in Section 3.2.3. A model-based reflex agent could, for instance, regularly monitor the temperature within a shipping container. Based on the changes in temperature, it could then discover that the shelf life of the goods loaded decreases. An adequate reaction could be to re-route the container to another destination nearby.

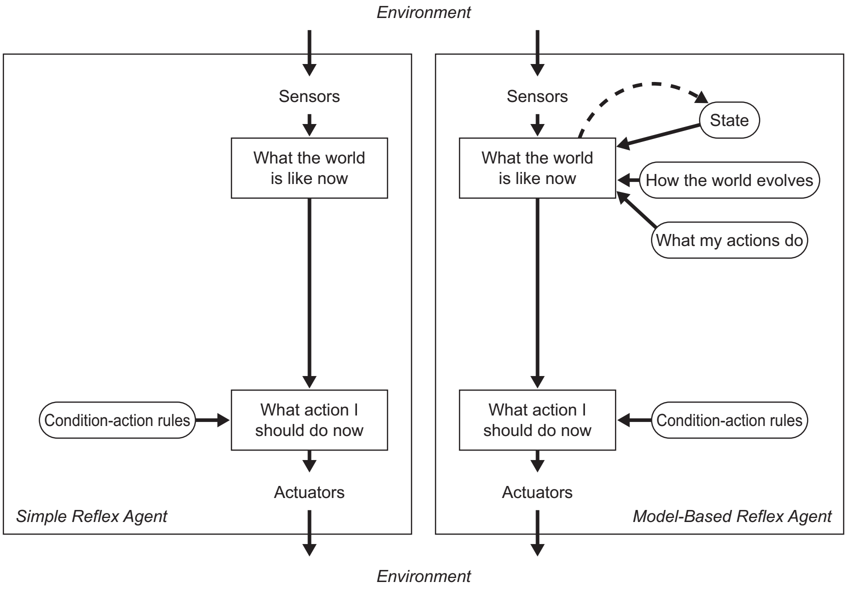

Fig. 4.2 Reflex agents follow a predefined set of rules in order to decide which actions to take. The simple reflex agent relies only on its current perception. By contrast, the model-based reflex agent maintains a model of its environment (adapted from Russell & Norvig, 2010, pp. 49, 51).

Knowing the current or former states of the environment is not always sufficient in order to decide which actions to take (Russell & Norvig, 2010, p. 52). Often an agent can choose between different actions in one situation. The so-called goal-based agent (Russell & Norvig, 2010, pp. 52–53) then chooses the most appropriate action to achieve its goals. This decision requires that the agent has a representation of its particular goals (Figure 4.3 left). Goals relieve the agent from having specific rules for each state of the environment. Instead, it must be able to evaluate the probable outcome of its actions with respect to its goals. A prominent example for goal-based agents is the BDI architecture by Bratman (1987) which distinguishes beliefs, desires, and intentions of agents. Rao and Georgeff (1997, p. 317) propose a modal logics formalism that allows representing alternative possible world states in the BDI architecture. Goal-based agents can, for instance, be applied in order to implement intelligent shipping containers. The agent would then have the task to plan and schedule its route through the logistics network.

The utility-based agent (Russell & Norvig, 2010, pp. 53–54) allows modelling the benefit of certain actions even finer. The goal-based agent simply decides whether an action is suitable to reach a certain goal. This binary distinction is not sufficient for coping with conflicting goals and for choosing the goal that is most likely successful. For this purpose, the utility-based agent applies a function that determines the utility for the agent (Figure 4.3 right). Utility corresponds in this context to a gradual measurement indicating the use of a particular action for the agent in that specific situation. Choosing the action with the maximum expected utility (Neumann & Morgenstern, 1944, p. 83) can be implemented by probabilistic methods. Conflict management for discrete goals has, for instance, been addressed with discourse agents (Timm, 2003, pp. 86–96). In autonomous logistics, a utility-based agent could be used in order to implement the data processing unit of a truck. Often, trucks can choose between different goods. The utility function enables the truck to choose the cargo with the highest freight rates.

To summarise, different models for agent implementation exist. Which type of agent is appropriate directly depends on the specific application addressed. Reflex agents are already sufficient to implement some basic tasks in autonomous logistics. However, for more sophisticated applications it is important to apply goal-based or utility-based agents. Within each of these general agent classes, there are infinitely many ways for concrete implementations. Likewise, there exist also infinitely many ways for internal knowledge representation and specific instantiations. Which agent technology is chosen depends on the intended purpose. Agents may be applied on different platforms. Agents may pertain to different companies. Agents may be implemented by different companies. Obviously, different types of agents may coexist in a logistics network. Despite of their differences it is important that they are able to collaborate.

4.2 Multiagent Systems

The broad variety of agents (Section 4.1.2) does not pose a major problem as long as each agent simply acts on its own. However, agents must often

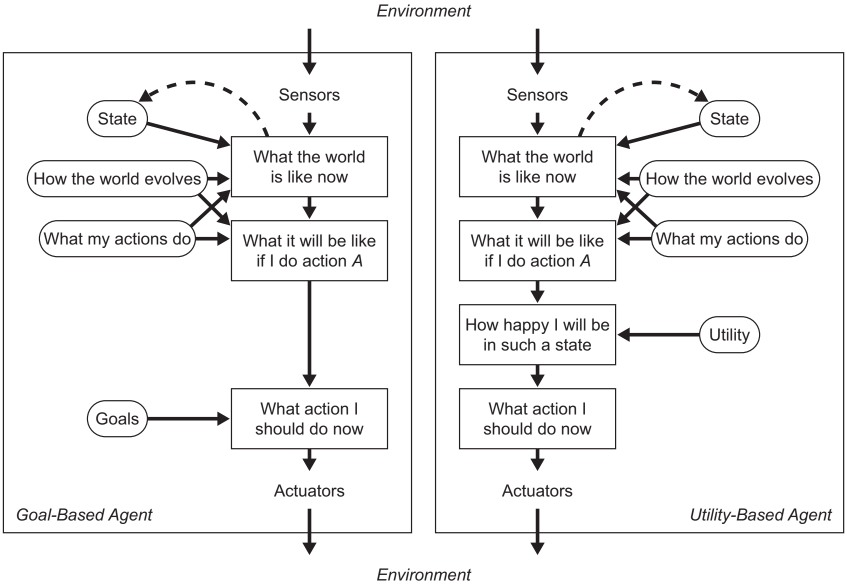

Fig. 4.3 Goal-based agents have a function that evaluates whether actions help in achieving a particular goal. Likewise, utility-based agents apply their utility function to evaluate the gradual utility of actions (adapted from Russell & Norvig, 2010, pp. 52, 54).

cooperate in order to achieve their objectives. Durfee (1999, p. 121) explicates that cooperation is necessary if agents cannot solve a task on their own or if they can accomplish their tasks better if they collaborate. Autonomous logistics has many examples for this finding. For instance, consider the shipping container discussed in Section 3.1.1. This container must cooperate with trucks in order to be transported from its source to a sink. However, cooperation is not the sole occurrence of interaction. Ferber (2001, p. 90) distinguishes eight degrees of interactions ranging from independence to collective resource conflicts, i. e., competition. Interaction requires interoperability standards (Singh, 2003, p. 37). These standards must provide an appropriate abstraction from different implementations, different platforms, and different companies (Hellenschmidt & Wichert, 2007, p. 97).

The IEEE Foundation for Intelligent Physical Agents (FIPA) has specified several standards for agent interaction (O'Brien & Nicol, 1998, p. 52). These standards specify

1. an architecture for multiagent platforms,
2. a message format for agent messages,
3. and agent interaction protocols.

Section 4.2.1 introduces the FIPA standard for multiagent platforms which covers the underlying message exchange infrastructure. Based on these foundations, Sections 4.2.2 and 4.2.3 give an overview on specifications how to format and interpret messages and their content. The particular focus of Section 4.2.4 is how such single messages can be combined to more complex agent interaction protocols.

4.2.1 Multiagent Platform

The FIPA architecture for multiagent platforms provides the basic infrastructure for message exchange. FIPA multiagent platforms (Foundation for Intelligent Physical Agents, 2004) comprise the following components (Figure 4.4). The message transport system enables software agents to communicate with each other. Agents can exchange messages with other agents on the same or another platform. This particularly requires a unique identification of agents (O'Brien & Nicol, 1998, p. 51). Note the parallelism to the identification unit of autonomous logistics entities discussed in Section 3.2.1. The agent management system administers a list of all agents on the respective platform, i. e., it acts as a white pages service. A yellow pages service is provided by the so-called directory facilitator. Agents can register their services with this component and query it for services provided by other agents. Both agent management system and directory facilitator can be agents themselves. This allows interacting with them via the message transport system like with any other agent (O'Brien & Nicol, 1998, p. 55).

4.2.2 Agent Message Structure

The FIPA standard for multiagent platforms specifies the communication layer for agent interaction. This layer is the means to exchange messages with each other. However, it does not define any semantics of the messages exchanged. Without semantics, agents cannot interpret messages they receive, i. e., messages do not have any meaning to them. One approach addressing this issue is KQML, the knowledge query and manipulation language (Huhns & Stephens, 1999, p. 88). Within the FIPA framework, the agent communication language (ACL) specifies a message format for agent communication (Poslad & Charlton, 2001, p. 110). According to the ACL standard (Foundation for Intelligent Physical Agents, 2002a), messages can have the following parameters. First of all, specifying the communicative act allows explicitly stating the intention of the message, e. g., whether it is an inform or a request. This is necessary to avoid ambiguities of the utterance (Huhns & Stephens, 1999, p. 87). A number of standard performatives has been specified (Foundation for Intelligent Physical Agents, 2002b). Introducing additional performatives is also permitted if required. Further fields in the message format cover the

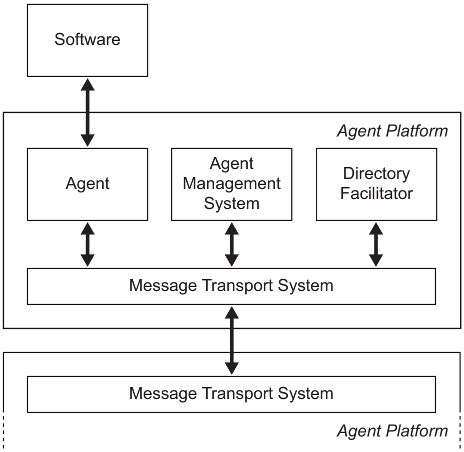

Fig. 4.4 Architecture of FIPA multiagent platforms. Agents can exchange messages with agents on the same or another agent platform via the message transfer system. The agent management system administers a white pages service with all agents on the platform. The directory facilitator administers the yellow pages with services offered by agents (adapted from Foundation for Intelligent Physical Agents, 2004, p. 2).

unique identifiers of sender and receivers. It is also possible to specify another agent than the sender to which subsequent messages should be directed to.

The actual content of the message depends on the application intended and can therefore be arbitrarily expressed. Defining a common architecture for multiagent platforms as well as a common message format is thus not sufficient for successful interaction between agents. Agents may have completely different internal representations of their environment (Section 4.1.2). Hence, it is additionally important to agree on a common content format. The ACL message format reflects this requirement by providing means to define also the semantics of the message content. Respective fields allow specifying the language and the encoding of the content as well as a reference to some ontology. The important question is which knowledge has to be considered and thus to be represented in agent messages. Reasoning about objects in logistics and other domains requires knowledge about their conceptual properties. However, the relevance of this knowledge is generally linked to spatial and temporal properties (Hübner, Spittel, Visser & Vögele, 2004, p. 80). Therefore, concept, location, and time are particularly relevant for intelligent agents in autonomous logistics (Schuldt & Werner, 2007b, p. 314). Conceptual properties refer to the condition of autonomous logistics entities. Location refers to their spatial position. Temporal properties play an important role because actions should be executed at specific points in time. The following section gives an exemplary overview of representations for concept, location, and time in autonomous logistics applications.

4.2.3 Message Content Formatting

As a motivation for appropriate knowledge representation, consider a shipping container that is expected to plan and schedule its way through the logistics network itself (Section 3.1.1). The intelligent agent representing the shipping container must know what is loaded inside the container. It depends on the cargo which means of transport it may choose. There might, for instance, be different rules for textiles and hazardous material (Section 3.2.2). The objective of such knowledge representations is to derive implicit findings from the explicitly represented knowledge (Nardi & Brachman, 2003, p. 2). Knowledge about object properties can be represented by description logics. Description logics are decidable fragments of first-order logic (Baader & Nutt, 2003, p. 44). Objects can be characterised as being instances of atomic concepts, i. e., classes of objects. As an example, the concepts C and D are designated by unary predicate symbols $C(x)$ and $D(x)$ respectively (Nardi & Brachman, 2003, p. 6). A terminology is built up by establishing is-a relationships between concepts. Figure 4.5 depicts an example taxonomy for consumer products. It comprises two branches. One branch characterises objects by their affiliation to particular article groups, e. g., textile, jewellery, and fruit. These concepts can be further refined, e. g., T-shirt as a subconcept of textile. The other branch describes properties of objects. Objects might, for instance, be perishable, valuable, or damaged.

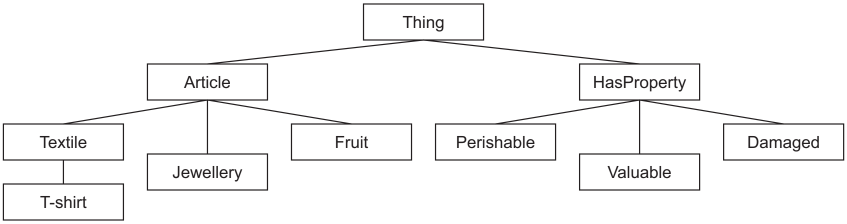

Fig. 4.5 An example terminology for cargo. It consists of two branches. One describes articles regarding their type, another characterises special properties. All concepts are derived from the most general concept Thing.

Apart from basic symbols, it is also possible to compose more complex concepts with constructors (Nardi & Brachman, 2003, p. 7), such as intersection $C \sqcap D$ which corresponds to $C(x) \wedge D(x)$. Figure 4.6 illustrates two example concepts, damaged T-shirts and damaged textiles. Subsumption can be computed in order to reveal the implicit knowledge that damaged textiles are a super concept of damaged T-shirts. This enables the shipping container to recognise that a selector for damaged textiles is capable of receiving damaged T-shirts. Furthermore, atomic roles designated by binary predicate symbols can be applied in order to express relationships between concepts. Concepts are defined within the TBox which maintains the terminology of the description logics knowledge base (Nardi & Brachman, 2003, pp. 13–15). The ABox contains membership assertions about individuals that are members of concepts defined within the TBox (Nardi & Brachman, 2003, pp. 15–16).

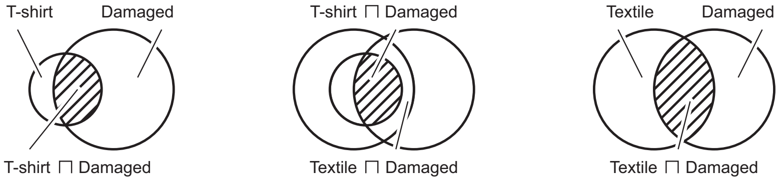

Fig. 4.6 The left hand concept characterises damaged T-shirts, while the right hand concept represents damaged textiles. Subsumption allows revealing the implicit knowledge that damaged textiles are a super concept of damaged T-shirts.

Description logics enable the example shipping container to choose an appropriate means of transport based on the goods loaded. However, spatial knowledge is required in order to choose a concrete transport relation. It is straightforward to provide every autonomous logistics with its geographic coordinates, i. e., latitude and longitude. This information can be directly derived from global navigation satellite systems (Section 3.2.2). However, such quantitative measures are not always sufficient and adequate in order to represent the location of autonomous logistics objects (Schuldt & Werner, 2007a, p. 194). As an example, imagine that two containers in a port are intended to be transported together on a truck. Their spatial distance is 500 metres. Although this is rather short, it does reveal almost nothing about the question whether a joint transport is eligible for these containers. They might be located in neighbouring container terminals of different operators which prevents them from being loaded on the same truck. A meaningful qualitative abstraction helps in deciding whether cooperation is possible for these shipping containers. One possibility is tessellating the port into distinct areas in order to apply topological relations (Figure 4.7) for spatial reasoning (Randell, Cui & Cohn, 1992, p. 172). Such regions can then be connected by a graph which can, for instance, be weighted by transport costs between different locations. Other qualitative representations also allow incorporating positional information for spatial reasoning (e. g., Schuldt & Gottfried, 2008a, 2008b).

Apart from conceptual and spatial knowledge, time plays also an important role in logistics. The example shipping container must be able to reason about temporal relationships in order to plan its way through the logistics network. Actions like transporting, handling, and storing the container must be executed in a particular order. The time spans during which some events occur can be represented by temporal intervals designated by particular start and end points. However, this quantitative information is often not sufficient. Instead, it is necessary to express also general relationships between temporal

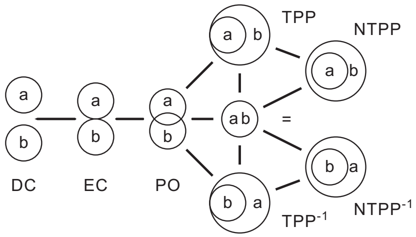

Fig. 4.7 The Region Connection Calculus (RCC) introduces eight qualitative relations in order to reason about pairs of regions *a* and *b* and their connections. The lines between the pictorial representations denote transitions between relations (adapted from Randell et al., 1992, p. 172).

intervals of autonomous logistics entities. For instance, it might be important that an action is executed before another one independent from the concrete times at which they occur. This can be expressed by the 13 temporal relations (Figure 4.8) between two intervals introduced by Allen (1983, p. 835). These relations allow distinguishing all qualitatively distinct configurations of two temporal intervals. The relations covered are before, meets, overlaps, starts, during, finishes, equals, and their inverses.

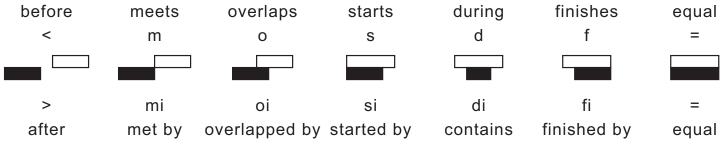

Fig. 4.8 Qualitative relations between two temporal intervals. The upper row lists the seven distinguishable relations, the lower row presents their inverses. Example configurations are depicted in the middle row (adapted from Allen, 1983, p. 835).

4.2.4 Agent Interaction Protocols

Standardising the structure of messages enables agents to communicate over the standardised multiagent platform. Specifying performative, participants, and content is sufficient in order to exchange and understand single messages. However, agent communication is often not limited to exchanging single messages. If multiple subsequent messages are exchanged, agents must be able to assign them to the respective thread of communication. Therefore, ACL allows senders specifying a reply-with key which other agents use in order to link their responses as in-reply-to a particular message. By using the reply-by attribute, a sender can inform responders about the time by which he would like to receive the responses. Additionally, ACL allows defining whether a message is part of some interaction protocol. Because agents can participate in multiple interaction protocols at the same time, it is necessary to assign a unique identifier to the whole conversation.

FIPA has not only issued standards regarding the architecture of multiagent systems and the message structure for agent communication. Additionally, standards for agent interaction have been specified (Poslad & Charlton, 2001, p. 101). Agent interaction protocols can be represented in the Agent Unified Modeling Language (Odell, Parunak & Bauer, 2000, pp. 124–125), in short AUML, which is an agent-specific extension of UML (Booch, Rumbaugh & Jacobson, 2005). AUML extends standard UML interaction diagrams with means to cope also with particular requirements of agents, e. g., regarding autonomy (Huget & Odell, 2004, p. 16). Agents participating in interaction protocols are represented by their so-called lifeline (Huget & Odell, 2004, pp. 23–24). It consists of a rectangle stating the name or class of the agent and a vertical dashed line below. Messages are represented by arrows between the lifelines of two agents. Messages are generally exchanged asynchronously because it is impossible to directly invoke methods on agents. This is reflected by the open head of message arrows (Huget & Odell, 2004, pp. 25–26). Cardinalities can be applied in order to indicate that messages have multiple senders or receivers. Messages are generally labelled by their respective performatives.

The FIPA request interaction protocol (Foundation for Intelligent Physical Agents, 2002d) allows requesting another agent to perform some action for the initiator (Figure 4.9). For this purpose, the initiator sends a request message to one or more participants. The participants decide whether they agree to perform the action requested. Remember that, unlike objects, agents are autonomous in their behaviour. Hence, an agent can also refuse to act for the initiator. However, if a participant agrees it will also inform the initiator about the outcome of the action or potential failures. The agreement message is optional. It is used to inform the initiator early if the actual task takes some time. Otherwise, the participant can also inform the initiator about successful execution without an explicit agreement message.

Simply requesting other agents to perform actions does not always suffice. Instead, it is often necessary to negotiate on the price that participants demand for their service. Smith (1977, p. 472) proposes the so-called contract net for delegating tasks to subcontractors. The FIPA contract net interaction protocol (Foundation for Intelligent Physical Agents, 2002c) defines the respective flow of messages (Figure 4.10). In this protocol, an initiator sends a call for proposals to all participating agents. The call describes the task to be solved for the initiator. Based on this description, initiators can decide whether they are able to solve the task. Like in the request protocol, participants must also decide whether they are willing to solve the task announced by the initiator. Depending on its decision, each participant can either refuse or make a proposal. Such a proposal generally includes the price at which the

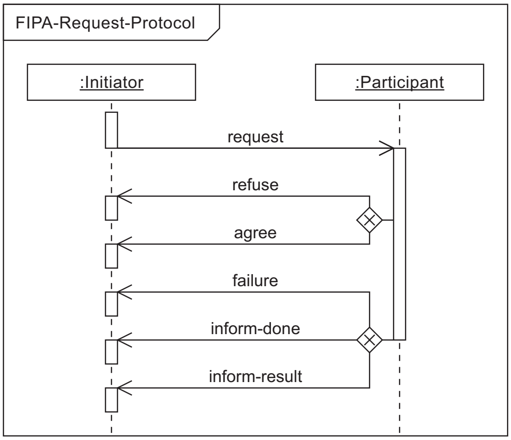

Fig. 4.9 Agents can request others to perform actions for them by applying the FIPA request interaction protocol (adapted from Foundation for Intelligent Physical Agents, 2002d, p. 1).

participant is willing to solve the respective task. Based on all proposals, the initiator decides which offers to accept and which ones to reject. The chosen participants inform the initiator either about the successful execution of the task or about failures that occurred during execution.

The standard protocols issued by FIPA already cover some tasks in autonomous logistics. For instance, an intelligent container might apply the FIPA contract net interaction protocol in order to negotiate its transport with some trucks. Other protocols enable auctions and subscription to services. Nevertheless, these standard protocols are not sufficient to cover all conceivable applications. Timm (2003, pp. 154–155) proposes to apply the concept of open adaptive communication if the respective situations are not known in advance. By contrast, if the circumstances of encounter are known it is possible to define own protocols depending the specific problem addressed (Huget & Koning, 2003, p. 180). Rosenschein and Zlotkin (1994, pp. 20–22) explicate that it is advisable to follow some general attributes when designing new interaction protocols. The attributes that Rosenschein and Zlotkin propose to interaction protocol designers are:

1. Efficiency
2. Stability
3. Simplicity
4. Distribution
5. Symmetry

Rosenschein and Zlotkin denote by efficiency that no unnecessary resources are squandered when an acceptable solution is found. As an example, they mention Pareto Optimality (Sandholm, 1999, p. 202) as a potential condition

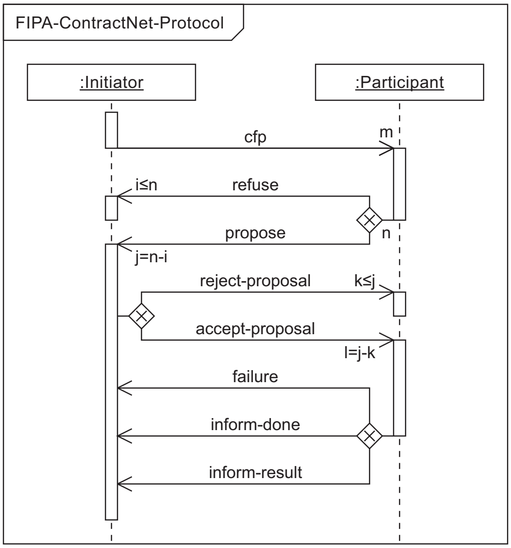

Fig. 4.10 The initiator in the contract net interaction protocol asks the participants for proposals for some tasks. The proposal of one or more participants may then be accepted. Finally, the chosen participants inform either about the result or about a failure (adapted from Foundation for Intelligent Physical Agents, 2002c, p. 2).

for termination of a protocol. A Pareto optimal solution is reached if no agent can improve its own benefit without demanding some other agent to derive less from the solution. Stability means that, as soon as an agreement is made, no agent has an incentive to deviate from it. Simplicity is closely related to both efficiency and stability. This criterion demands to spend as little (computational) power as possible for negotiation. This includes keeping the interaction complexity as low as possible. Distribution refers to the fact that interaction protocols should be independent from central decision-making entities because they are potential bottlenecks of the system. Finally, the symmetry criterion demands that distinctions between participating agents are only made if the criteria applied are appropriate. Rosenschein and Zlotkin (1994, p. 21) point out that this decision directly depends on the domain of the concrete application addressed.

Other criteria for evaluation of interaction protocols have been proposed by Sandholm (1999, pp. 202–204):

1. Social welfare
2. Pareto efficiency
3. Individual rationality
4. Stability
5. Computational efficiency
6. Distribution and communication efficiency

These criteria distinguish benefits for the whole system and for individual agents. Social welfare and Pareto efficiency measure the outcome of an interaction protocol from the perspective of the whole system. Sandholm (1999, p. 202) explains that solutions that maximise social welfare are a subset of Pareto efficient solutions. The criterion of individual rationality integrates the perspective of individual agents. Self-interested agents must have a personal benefit in order to decide for participating in a negotiation. The stability and efficiency criteria correspond to those of Rosenschein and Zlotkin. All criteria discussed in this section can be grouped into the categories efficiency, reliability, and complexity (Krempels, Spaniol, Scholz, Timm & Herzog, 2006, p. 387).

To summarise, agent interaction protocols allow specifying threads of communication. Several standard interaction protocols have been issued by FIPA. However, these standard protocols are not sufficient to cover all imaginable applications. Therefore, it is often necessary to design individual agent interaction protocols for concrete applications. Several attributes have been proposed that can be considered when specifying such new protocols.

4.3 Multiagent Organisation

Interaction protocols (Section 4.2.4) allow structuring the course of agent conversations. However, they cover only rather short-term interaction. By contrast, many tasks in autonomous logistics require long-term cooperation of intelligent agents and thus additional formalisms. This can be manifested by organisational structures, also called societies of agents (Huhns & Stephens, 1999, p. 112). Ferber (2001, p. 114) explains that organisations are characterised by two properties. On the one hand, by the roles that are assigned to classes of agents. On the other hand, by the abstract relationships between these roles. Common roles in multiagent systems are broker, directory, mediator, and moderator. Broker or middle agents provide agents with services of other agents, e. g., with knowledge acquired by other agents (Langer et al., 2006, p. 282). Directory facilitators (Section 4.2.1) administer lists of agents or services provided. Mediator agents translate between agents that have different languages. Moderators organise negotiations within groups of agents. Apart from these common examples, it is often necessary to define other roles based on the requirements of the concrete application at hand.

Timm, Scholz, Herzog, Krempels and Spaniol (2006, 46–48) propose to characterise structures in multiagent systems by three dimensions, namely capabilities, duration, and decision-making. The capability attribute refers to the question whether the jointly acting agents are homogeneous or heterogeneous in their capabilities. The possible duration of organisation spans from short-term to long-term. Timm, Scholz, Herzog, Krempels and Spaniol distinguish different degrees in freedom for decision-making of agents. Depending on the concrete structure, the autonomy of agents may be restricted, e. g., by democratic decisions or hierarchical orders. In economic systems, an additional fourth dimension helps distinguishing whether interaction is carried out horizontally, vertically, or diagonally. This classification allows, for instance, distinguishing cartel, collaboration, cooperation, alliance, department, institution, and location (Timm, Scholz, Herzog, Krempels & Spaniol, 2006, p. 47).

Section 4.3.1 discusses the concept of holonic agents as an approach to structure multiagent systems. Approaches to establishing such structures by team formation are then presented in Section 4.3.2. Finally, Section 4.3.3 gives example applications of multiagent systems to logistics.

4.3.1 Structuring Multiagent Systems

Multiagent systems often comprise a high number of agents. K. Fischer (1999, p. 34) states that interaction in such environments requires structure and organisation for efficient system behaviour. To this end, Schillo, Fischer and Siekmann (2003, p. 81) propose to derive long-term teams from successful cooperations. They suggest holonic agents, in short holons, as a compromise between hierarchical organisational structures and decentralised control (K. Fischer, 1999, p. 35). A holonic agent, also referred to as superholon, is formed by multiple subholons (Schillo et al., 2003, p. 82). This concept is recursive in that each of the subholons may itself be a holonic agent. Holons reduce the complexity of multiagent systems. The complete communication with the outside world is generally conducted by a distinguished head. The other agents that belong to the holon form its body. Their communication is restricted to other body agents and the head. That is, the concept of holonic agency follows the principle of encapsulation (Section 4.1.1). In this case, however, not only the internal state of one agent is encapsulated by the superholon but the whole group of subholons, i. e., head and body agents.

Different ways exist for organising holonic agents internally (K. Fischer, Schillo & Siekmann, 2003, pp. 75–78). These approaches to organisation differ regarding the autonomy that is left to the participating agents. The bandwidth ranges from agents that completely retain autonomy to agents that completely give up autonomy. In the first case, the holonic agent is modelled as a set of autonomous agents (Figure 4.11 left). The new superholon is then simply a new conceptual entity. Its properties are derived from the properties of its subholons (K. Fischer et al., 2003, pp. 75–76). The subholon agents are completely autonomous. They negotiate in order to come to an agreement how to act jointly. By contrast, another organisational structure for holonic agents merges the participating subholons into a new superholon (Figure 4.11 right). That is, a new agent is created that subsumes the properties of its participants (K. Fischer et al., 2003, pp. 76–77). Hence, the subholons completely give up their autonomy. However, K. Fischer et al. explain that subholons may be reactivated later when the superholon is dissolved. As a compromise between these two extremes, it is also possible to consider holons as a moderated association (Figure 4.11 centre). In this approach, one of the subholons takes the role of the head of the superholon (K. Fischer et al., 2003, pp. 77–78). This head represents the holon within the outside world. Its task is not necessarily limited to pure representation. It might also be authorised to make decisions on behalf of the holon.

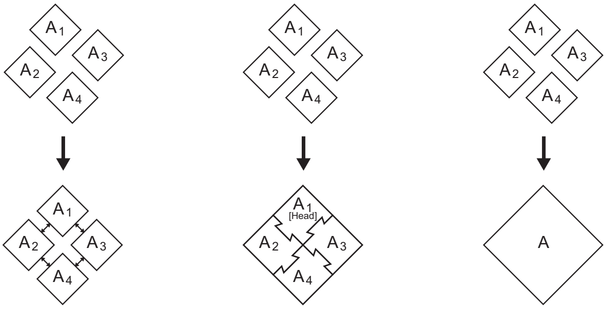

Fig. 4.11 The degree of autonomy for subholons within a holonic agent may vary. One extreme is that a holon consists of a set of loosely coupled autonomous agents (left). The other extreme is that several agents are completely merged into one (right). The centre depicts an alternative according to which a holon may be organised as a moderated association (adapted from K. Fischer et al., 2003, pp. 76, 77).

Multiagent organisations can be either predefined or emergent (Ferber, 2001, p. 143). Hillebrandt (2005, p. 43) distinguishes two types of delegation in multiagent systems. On the one hand, task delegation which is important in order to coordinate joint problem solving. On the other hand, social delegation which means that an agent becomes a representative for one or multiple agents and acts on their behalf. Hillebrandt (2005, p. 44) discusses four mechanisms for delegation:

1. Economic exchange
2. Authority
3. Gift exchange
4. Voting

Economic exchange is a standard market mechanism. It means that an agent is paid for executing some task or for taking over representation for another agent. Gift exchange is an alternative that aims at establishing long-term relationships between agents. Exchanging gifts allows establishing relationships of reciprocal trust (Hillebrandt, 2005, p. 44). Likewise, it helps identifying a lack of trust when another agent does not return a gift. Gift exchange demands agents to keep track of the current relationships with other agents. Authority is a concept that requires a non-cyclic set of power relationships between the participating entities (Hillebrandt, 2005, p. 44). This enables delegation along these relationships. In contrast to authority, voting allows groups of equal agents to jointly decide about delegation (Hillebrandt, 2005, p. 44).

One way for applying the concept of holonic agents in logistics is quite straightforward. Holons can be applied in order to structure agents in accordance to the company they belong to. For instance, consider a forwarding company that is represented by a holonic agent. The holon head then applies for transport contracts in the multiagent system. The subholons representing the trucks of the company then plan and schedule the actual execution of the transport process. This, however, contradicts to some extent to the principles of autonomous logistics in which all entities should be able to communicate in a heterarchical way. Bürckert, Fischer and Vierke (2000, pp. 711–713) propose to let the means of transport organise them by holons instead. In their concept, trucks and trailers form holons in order to be able to jointly address customer demands. Lind, Fischer, Böcker and Zirkler (1999, pp. 326–327) describe a respective approach for railroad transport.

4.3.2 Agent Team Formation

Köhler-Bußmeier (2009b) models team formation and action by Petri Nets. Another approach to formalise holonic applications has been proposed by Leitão, Boissier, Casais and Restivo (2003, pp. 61–62). They apply UML and Petri Nets in order to model dynamic aspects and to formally validate specifications of holonic agent systems. However, this does not cover the internal modelling of the participating agents. This gap is filled by the model for cooperation by Wooldridge and Jennings (1999). It consists of four consecutive steps (Figure 4.12), namely recognition, team formation, plan formation, and team action. In their model, the cooperative problem solving process is initiated by an agent that recognises a potential for cooperation. One motivation is that the agent is unable to achieve its goals on its own. However, an agent may also prefer to cooperate although it is able to achieve its goal on its own. This might be the case if multiple agents might solve a task better or more efficiently. Wooldridge and Jennings (1999, p. 575) explicate that believing that a group of agents can actually achieve the goal is important for recognition of a potential for cooperation. Having recognised such a potential, the respective agent initiates team formation. For this purpose, the agent must identify a group of agents of which it believes that they can solve the addressed task. It must then persuade these agents to assist in achieving its goals. The outcome of this step of the model for cooperation is a commitment of the group (Levesque, Cohen & Nunes, 1990, p. 94) to jointly address some goals. Based on this commitment, the agents then negotiate on their joint plan (Wooldridge & Jennings, 1999, p. 581). Finally, the agents collectively execute the plan. It is worth mentioning that each of these consecutive steps may fail. It might thus be necessary to return to a previous step (Wooldridge & Jennings, 1999, p. 574).

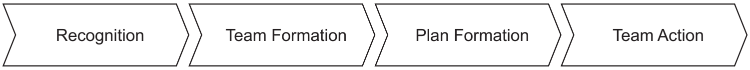

Fig. 4.12 According to the model for cooperation, the cooperative problem solving process is initiated by an agent that recognises a potential for cooperation. It is then necessary to form a team of collaborating agents. Subsequently, the participants negotiate a joint plan which is then executed in the final step.

Although Wooldridge (2000, p. 157) proposes a specific speech act (Section 4.2.2) for team formation, less effort has been spent on concrete interaction protocols (Section 4.2.4) for this purpose. Many approaches presume that agents have been grouped beforehand, either manually or by some unspecified method (cf. van de Vijsel & Anderson, 2004, p. 54). Nair, Tambe and Marsella (2002, p. 150) propose an approach for rapid team formation as it is, for instance, required in the RoboCupRescue domain. In this approach, agents form and reform teams ad hoc. Besides, the contract net interaction protocol (Section 4.2.4) is often used in order to distribute tasks among agents. However, Dignum, Dunin-Keplicz and Verbrugge (2000, pp. 150–151) argue that it is not flexible enough. Instead, they propose a theory for team formation by dialogue. Unfortunately, discussing about team formation requires more complex reasoning than simply applying a pre-defined protocol (Dignum et al., 2000, pp. 163–164). For real-world applications, it is thus often preferable to define specific interaction protocols. Ogston and Vassiliadis (2001, pp. 608–609) mention some criteria that can be considered when designing interaction protocols for team formation:

1. Set up cost
2. Distribution of information
3. Common language
4. Privacy

These criteria supplement the attributes by Rosenschein and Zlotkin and Sandholm (Section 4.2.4). Ogston and Vassiliadis denote the costs for enabling an agent to participate in an interaction protocol as set up cost. These costs include requirements for memory, processing, and communication resources. Distribution of information addresses the question to which extent bottlenecks are avoided. It is thus equivalent to the distribution attributes of the other authors. Distribution is closely related with the requirement for a common language throughout the system. The more an approach for team formation is decentralised, the more reduced is the necessity for a common language of all, potentially unrelated, agents. Finally, the privacy criterion judges whether team formation requires to broadcast request for team formation to all agents, including uninvolved ones.

To summarise the discussion so far, methodologies exist in order to characterise organisational structures in multiagent systems. These formalisations cover both the structure of multiagent systems as well as internal states of intelligent agents participating in team formation. However, less effort has been spent on concrete interaction protocols in order to establish teams of agents. Nevertheless, besides the general criteria for interaction protocol design, supplemental criteria for team formation protocols have been defined.

4.3.3 Applications of Agents in Logistics

Various projects exist that apply multiagent technology to the logistics domain. The tasks which agents undertake and the real-world entities which they represent depend on the concrete problems addressed. Hence, like for the broad bandwidth of actual agent implementations (Section 4.1.2), there exists also a broad variety of possible applications. Often, agents represent whole companies in order to automate interaction between them. For instance, Fink (2006) proposes to apply agents in order to solve coordination problems between companies. In this approach, agents negotiate on supply network coordination in an automated way. Likewise, Keller, Duguay and Precup (2004) describe an agent-based approach to supply network management in which agents act on behalf of computer manufacturing companies. The agents are responsible for acquiring customer orders or for supply with components (Precup, Keller & Duguay, 2006, p. 138). Pippow (2004) aims at reducing the Forrester effect (Section 2.2) for such agents that act on behalf of their company. In contrast to coordinating supply and demand between companies, the focus of the agent-based approach by Timm, Scholz and Herzog (2006, pp. 254–256) is on the manufacturing domain. In order to implement flexible and reliable manufacturing, process planning and production control is delegated to software agents that represent orders, resources, and services.

Apart from the general coordination of supply network or production processes, many approaches address the primary logistics functions (Section 2.1). A particular focus is on transport. For instance, Silva, Runkler, Sousa and Palm (2002) apply ant algorithms to optimise logistics processes with multiagent systems. They use the ant metaphor in order to represent components as food sources and orders as nests. The approach of van der Putten, Robu, La Poutré, Jorritsma and Gal (2006) represents logistics service providers by autonomous software agents. Their objective is to automate negotiation about allocation of transport orders between the companies involved in logistics networks. Lind et al. (1999) describe a system that plans and monitors plan execution in railroad transport. Their approach particularly considers routing intermediately coupled transport modules instead of conventional trains. Bürckert et al. (2000) use a multiagent-based approach for route planning, fleet management, and driver scheduling for trucks. Their agents represent means of transport, such as trucks and trailers, that can form holons to address transport order requirements. Hölscher, Knirsch and Kreowski (2005) combine generalised graph transformation units and multiagent systems to model transport networks. The rigorous semantics of the so-called autonomous units supports verification of agent interaction, e. g., negotiations (Hölscher, Knirsch & Luderer, 2007). Gehrke and Wojtusiak (2008) use machine learning for driving time prediction. This enables software agents to incorporate environmental conditions such as weather and traffic in adaptive route planning for trucks. Dorer and Calisti (2005) take the perspective of a logistics provider. Agents that represent trucks exchange the orders they fulfill in order to optimise transport requests.

In contrast to transport, the other primary logistics functions are addressed less frequently. Henesey, Davidsson and Persson (2006, 2008) apply multiagent systems in order to evaluate and improve operational policies within container terminals. They investigate transshipment of containers, i. e., handling. Their simulation model covers physical entities of the container terminal and agents representing management entities (Henesey, 2006, pp. 171–176). These agents simulate port captains, ships, stevedores, terminal managers, cranes, and straddle carriers. T. Fischer and Gehring (2006) describe a multiagent approach to transshipment in automobile logistics. They aim at optimising resource allocation and personnel deployment (T. Fischer & Gehring, 2006, p. 378). Storage as another primary logistics function is addressed by Triebig et al. (2005, p. 229). They model processes within high-bay warehouses with multiagent systems. Triebig et al. motivate that this allows evaluating the future behaviour of such facilities already during the planning period. Picking is the final primary logistics function to be discussed. It is a process with a considerably low degree of automation. Still today, automated picking is a challenge due to the broad bandwidth that objects exhibit in size and deformability. Nevertheless, engineers aim at building robots that are capable of these tasks. For instance, Hertzberg et al. (2003) apply agent technology in order to control picking robots.

To summarise, there is a broad variety of previous applications of multiagent systems in logistics. The bandwidth ranges between automating intercompany coordination, production logistics, and individual primary logistics functions. The agents applied usually represent either whole companies, individual decision-makers, or utilities that actually control processes today. Therewith, multiagent technology is applied in order to automate existing logistics processes. The paradigm of autonomous control in logistics goes even one step further (Section 3.1). It aims at delegating process control to previously inanimate objects such as general cargo units and their load carriers. Represented by software agents, autonomous logistics entities are expected to plan and schedule their way through logistics networks. Instead of simply automating existing processes, this demands for a completely new process design with a focus on the logistics entities themselves. To achieve their logistics objectives, these entities must interact with manifold logistics service providers and coordinate multiple primary logistics functions. Unlike other approaches discussed in this section, autonomous logistics aims at actual process control and not only simulation of logistics processes. Nevertheless, simulation can be applied in order to evaluate autonomous logistics strategies in advance.

4.4 Conclusion

Enabling logistics entities to take over control of autonomous logistics processes themselves requires implementing their data processing unit. Intelligent software agents are an appropriate means for this task due to their autonomy, reactivity, pro-activeness, and social ability. Different general agent models exist. Every model covers an infinite number of possible concrete implementations. Agents may be applied on different platforms, may be implemented by different companies, and may pertain to different owners. Hence, particular standards are required to enable interaction between different agents. Interaction is necessary because individual agents can often not achieve their logistics objectives on their own. Standards for agent interaction cover message transport, message formatting, as well as agent interaction protocols. Long-term relationships between intelligent agents manifest themselves in agent organisations or societies. Approaches for organisation, like holonic agents, allow structuring multiagent systems. Creating such structures is formalised by theories for team formation.

Today, several approaches already apply multiagent systems for conventional control of logistics processes. By contrast, less effort has been spent on specifying agent interaction for autonomous logistics. For effective cooperation, it is important to specify the interaction of intelligent agents that plan and schedule the execution of the primary logistics functions (Section 2.1). Thereby, it is particularly important to investigate the requirement for cooperation in autonomous logistics and to derive respective interaction protocols for team formation. These steps are the precondition for cooperation of arbitrary software agents that represent autonomous logistics entities.

References

Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, *26*(11), 832-843.

Baader, F. & Nutt, W. (2003). Basic Description Logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi & P. F. Patel-Schneider (Eds.), *The*

Description Logic Handbook: Theory, Implementation, and Applications (pp. 43–95). Cambridge, UK: Cambridge University Press.

Booch, G., Rumbaugh, J. & Jacobson, I. (2005). *The Unified Modeling Language User Guide* (2nd ed.). Amsterdam, The Netherlands: Addison-Wesley Longman.

Bratman, M. E. (1987). *Intention, Plans, and Practical Reason.* Cambridge, MA, USA: Harvard University Press.

Bürckert, H.-J., Fischer, K. & Vierke, G. (2000). Holonic Transport Scheduling with Teletruck. *Applied Artificial Intelligence*, *14*(7), 697–725.

Dignum, F., Dunin-Keplicz, B. & Verbrugge, R. (2000). Agent Theory for Team Formation by Dialogue. In C. Castelfranchi & Y. Lespérance (Eds.), *7th International Workshop on Agent Theories, Architectures, and Languages (ATAL 2000)* (pp. 150–166). Boston, MA, USA: Springer-Verlag.

Dorer, K. & Calisti, M. (2005). An Adaptive Solution to Dynamic Transport Optimization. In M. Pechoucek, D. Steiner & S. G. Thompson (Eds.), *Special Track for Industrial Applications at the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)* (pp. 45–51). Utrecht, The Netherlands: ACM Press.

Durfee, E. H. (1999). Distributed Problem Solving and Planning. In G. Weiss (Ed.), *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence* (pp. 121–164). Cambridge, MA, USA: MIT Press.

Ferber, J. (2001). *Multiagentensysteme: Eine Einführung in die Verteilte Künstliche Intelligenz.* Munich, Germany: Addison-Wesley Verlag.

Fink, A. (2006). Supply Chain Coordination by Means of Automated Negotiations Between Autonomous Agents. In B. Chaib-draa & J. P. Müller (Eds.), *Multiagent-Based Supply Chain Management* (pp. 351–372). Heidelberg, Germany: Springer-Verlag.

Fischer, K. (1999). Holonic Multiagent Systems – Theory and Applications. In P. Barahona & J. J. Alferes (Eds.), *9th Portuguese Conference on Artificial Intelligence (EPIA 1999)* (pp. 34–48). Évora, Portugal.

Fischer, K., Schillo, M. & Siekmann, J. H. (2003). Holonic Multiagent Systems: A Foundation for the Organisation of Multiagent Systems. In V. Mařík, D. C. McFarlane & P. Valckenaers (Eds.), *1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS 2003)* (pp. 71–80). Prague, Czech Republic: Springer-Verlag.

Fischer, T. & Gehring, H. (2006). Business Process Support in a Seaport Automobile Terminal — a Multi-Agent Based Approach. In B. Chaib-draa & J. P. Müller (Eds.), *Multiagent-Based Supply Chain Management* (pp. 373–394). Heidelberg, Germany: Springer-Verlag.

Foundation for Intelligent Physical Agents. (2002a). *FIPA ACL Message Structure Specification* (Standard No. SC00061G). Geneva, Switzerland.

Foundation for Intelligent Physical Agents. (2002b). *FIPA Communicative Act Library Specification* (Standard No. SC00037J). Geneva, Switzerland.

Foundation for Intelligent Physical Agents. (2002c). *FIPA Contract Net Interaction Protocol Specification* (Standard No. SC00029H). Geneva, Switzerland.

Foundation for Intelligent Physical Agents. (2002d). *FIPA Request Interaction Protocol Specification* (Standard No. SC00026H). Geneva, Switzerland.

Foundation for Intelligent Physical Agents. (2004). *FIPA Agent Management Specification* (Standard No. SC00023K). Geneva, Switzerland.

Gehrke, J. D. & Wojtusiak, J. (2008). Traffic Prediction for Agent Route Planning. In M. Bubak, G. D. van Albada, J. Dongarra & P. M. A. Sloot (Eds.), *8th International Conference on Computational Science (ICCS 2008)* (pp. 692–701). Kraków, Poland: Springer-Verlag.

Genesereth, M. R. & Ketchpel, S. P. (1994). Software Agents. *Communications of the ACM*, *37*(7), 48–53, 147.

Hübner, S., Spittel, R., Visser, U. & Vögele, T. J. (2004). Ontology-Based Search for Interactive Digital Maps. *IEEE Intelligent Systems*, *19*(3), 80–86.

Hellenschmidt, M. & Wichert, R. (2007). Selbstorganisation: Dinge in eigenverantwortlicher Kooperation – eine Systemanalyse. In H.-J. Bullinger & M. ten Hompel (Eds.), *Internet der Dinge* (pp. 91–105). Heidelberg, Germany: Springer-Verlag.

Henesey, L. (2006). *Multi-Agent Systems for Container Terminal Management.* Doctoral dissertation, Blekinge Institute of Technology.

Henesey, L., Davidsson, P. & Persson, J. A. (2006). Agent Based Simulation Architecture for Evaluating Operational Policies in Transshipping Containers. In K. Fischer, I. J. Timm, E. André & N. Zhong (Eds.), *4th German Conference on Multiagent System Technologies (MATES 2006)* (pp. 73–85). Erfurt, Germany: Springer-Verlag.

Henesey, L., Davidsson, P. & Persson, J. A. (2008). Agent Based Simulation Architecture for Evaluating Operational Policies in Transshipping Containers. *Autonomous Agents and Multi-Agent Systems*, *18*(2), 220–238.

Herrler, R. & Klügl, F. (2006). Simulation. In S. Kirn, O. Herzog, P. Lockemann & O. Spaniol (Eds.), *Multiagent Engineering: Theory and Applications in Enterprises* (pp. 575–596). Heidelberg, Germany: Springer-Verlag.

Hertzberg, J., Ritter, A., Bredenfeld, A., Mayer, J., Schöll, P., Schönherr, F. et al. (2003). Agententechnologie zur Integration heterogener Systeme am Beispiel eines Prototypen für ein Kommissioniersystem. In R. D. Schraft, G. Brandenburg & K. Bender (Eds.), *Elektrische Automatisierung, Systeme und Komponenten (SPS/IPC/DRIVES 2003)* (pp. 423–432). Nuremburg, Germany: VDE Verlag.

Hillebrandt, F. (2005). Sociological Foundation of the Holonic Approach Using Habitus-Field-Theory to Improve Multiagent Systems. In K. Fischer, M. Florian & T. Malsch (Eds.), *Socionics – Scalability of Complex Social Systems* (pp. 36–50). Heidelberg, Germany: Springer-Verlag.

Hölscher, K., Knirsch, P. & Kreowski, H.-J. (2005). Modelling Transport Networks by Means of Autonomous Units. In H.-D. Haasis, H. Kopfer & J. Schönberger (Eds.), *Annual International Conference of the German Operations Research Society (OR 2005)* (pp. 399–404). Bremen, Germany: Heidelberg, Germany.

Hölscher, K., Knirsch, P. & Luderer, M. (2007). Autonomous Units for Communication-Based Dynamic Scheduling. In *1st International Conference on Dynamics in Logistics (LDIC 2007)* (pp. 331–339).

Huget, M.-P. & Koning, J.-L. (2003). Interaction Protocol Engineering. In M.-P. Huget (Ed.), *Communication in Multiagent Systems* (pp. 179–193). Heidelberg, Germany: Springer-Verlag.

Huget, M.-P. & Odell, J. (2004). Representing Agent Interaction Protocols with Agent UML. In J. Odell, P. Giorgini & J. P. Müller (Eds.), *5th International Workshop on Agent-Oriented Software Engineering (AOSE 2004)* (pp. 16–30). New York, NY, USA: Springer-Verlag.

Huhns, M. N. & Stephens, L. M. (1999). Multiagent Systems and Societies of Agents. In G. Weiss (Ed.), *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence* (pp. 79–120). Cambridge, MA, USA: MIT Press.

Jennings, N. R. (2001). An Agent-Based Approach for Building Complex Software Systems. *Communications of the ACM*, *44*(4), 35–41.

Jennings, N. R. & Wooldridge, M. (1998). Applications of Intelligent Agents. In N. R. Jennings & M. Wooldridge (Eds.), *Agent Technology: Foundations, Applications, and Markets* (pp. 3–27). Heidelberg, Germany: Springer-Verlag.

Keller, P. W., Duguay, F.-O. & Precup, D. (2004). RedAgent-2003: An Autonomous, Market-Based Supply-Chain Management Agent. In N. Jennings, C. Sierra, L. Sonenberg & M. Tambe (Eds.), *3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)* (pp. 1182–1189). New York, NY, USA: IEEE Computer Society.

Köhler-Bußmeier, M. (2009a). Koordinierung in Agentensystemen. In R. v. Lüde, D. Moldt & R. Valk (Eds.), *Selbstorganisation und Governance in künstlichen und sozialen Systemen* (pp. 73–118). Münster, Germany: LIT Verlag.

Köhler-Bußmeier, M. (2009b). Koordinierung in Agententeams. In R. v. Lüde, D. Moldt & R. Valk (Eds.), *Selbstorganisation und Governance in künstlichen und sozialen Systemen* (pp. 337–376). Münster, Germany: LIT Verlag.

Kirn, S., Herzog, O., Lockemann, P. & Spaniol, O. (Eds.). (2006). *Multiagent Engineering: Theory and Applications in Enterprises.* Heidelberg, Germany: Springer-Verlag.

Krempels, K.-H., Spaniol, O., Scholz, T., Timm, I. J. & Herzog, O. (2006). Interaction Design. In S. Kirn, O. Herzog, P. Lockemann & O. Spaniol (Eds.), *Multiagent Engineering: Theory and Applications in Enterprises* (pp. 383–403). Heidelberg, Germany: Springer-Verlag.

Langer, H., Gehrke, J. D., Hammer, J., Lorenz, M., Timm, I. J. & Herzog, O. (2006). A Framework for Distributed Knowledge Management in Autonomous Logistics Processes. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, *10*(4), 277–290.

Leitão, P., Boissier, R., Casais, F. & Restivo, F. (2003). Integration of Automation Resources in Holonic Manufacturing Applications. In V. Mařík, D. C. McFarlane & P. Valckenaers (Eds.), *1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS 2003)* (pp. 35–46). Prague, Czech Republic: Springer-Verlag.

Levesque, H. J., Cohen, P. R. & Nunes, J. H. T. (1990). On Acting Together. In T. Dietterich & W. Swartout (Eds.), *8th National Conference on Artificial Intelligence (AAAI 1990)* (pp. 94–99). Boston, MA, USA: AAAI Press / MIT Press.

Lind, J., Fischer, K., Böcker, J. & Zirkler, B. (1999). Transportation Scheduling and Simulation in a Railroad Scenario: A Multi-Agent Approach. In *4th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 1999)* (pp. 325–344). London, UK: The Practical Application Company.

Müller, H. J. (1997). Towards Agent Systems Engineering. *Data & Knowledge Engineering*, *23*(3), 217–245.

Nair, R., Tambe, M. & Marsella, S. (2002). Team Formation for Reformation in Multiagent Domains like RoboCupRescue. In G. A. Kaminka, P. U. Lima & R. Rojas (Eds.), *Robot Soccer World Cup VI (RoboCup 2002)* (pp. 150–161). Fukuoka, Japan: Springer-Verlag.

Nardi, D. & Brachman, R. J. (2003). An Introduction to Description Logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi & P. F. Patel-Schneider (Eds.), *The Description Logic Handbook: Theory, Implementation, and Applications* (pp. 1–40). Cambridge, UK: Cambridge University Press.

Neumann, J. von & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton, NJ, USA: Princeton University Press.

Nwana, H. S. & Ndumu, D. T. (1998). A Brief Introduction to Software Agent Technology. In N. R. Jennings & M. Wooldridge (Eds.), *Agent Technology: Foundations, Applications, and Markets* (pp. 29–47). Heidelberg, Germany: Springer-Verlag.

O'Brien, P. D. & Nicol, R. C. (1998). FIPA — Towards a Standard for Software Agents. *BT Technology Journal*, *16*(3), 51–59.

Odell, J., Parunak, H. V. D. & Bauer, B. (2000). Representing Agent Interaction Protocols in UML. In P. Ciancarini & M. Wooldridge (Eds.), *1st International Workshop on Agent-Oriented Software Engineering (AOSE 2000)* (pp. 121–140). Limerick, Ireland: Springer-Verlag.

Ogston, E. & Vassiliadis, S. (2001). Matchmaking Among Minimal Agents Without a Facilitator. In J. P. Müller, S. Sen, E. Andre & C. Frasson (Eds.), *5th International Conference on Autonomous Agents (Agents 2001)* (pp. 608–615). Montreal, Canada: ACM Press.

Parunak, H. V. D. (1999). Industrial and Practical Applications of DAI. In G. Weiss (Ed.), *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence* (pp. 377–421). Cambridge, MA, USA: MIT Press.

Parunak, H. V. D., Savit, R. & Riolo, R. L. (1998). Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In J. S. Sichman, R. Conte & N. Gilbert (Eds.), *1st International Workshop on Multi-Agent Systems and Agent-Based Simulation (MABS 1998)* (pp. 10–25). Springer-Verlag.

Pippow, I. (2004). *Software-Agenten in Distributionsnetzen: Potenziale vertikaler Informationsteilung zur Senkung von Transaktionskosten.* Wiesbaden, Germany: Gabler Verlag.

Poslad, S. & Charlton, P. (2001). Standardizing Agent Interoperability: The FIPA Approach. In M. Luck, V. Mařík, O. Štěpánková & R. Trappl (Eds.), *9th ECCAI Advanced Course (ACAI 2001) and Agent Link's 3rd European Agent Systems Summer School (EASSS 2001)* (pp. 98–117). Prague, Czech Republic: Springer-Verlag.

Precup, D., Keller, P. W. & Duguay, F.-O. (2006). RedAgent: An Autonomous, Market-Based Supply-Chain Management Agent for the Trading Agent Competition. In B. Chaib-draa & J. P. Müller (Eds.), *Multiagent-Based Supply Chain Management* (pp. 135–154). Heidelberg, Germany: Springer-Verlag.

Randell, D. A., Cui, Z. & Cohn, A. G. (1992). A Spatial Logic Based on Regions and Connection. In B. Nebel, C. Rich & W. R. Swartout (Eds.), *3rd International Conference on Principles of Knowledge Representation and Reasoning (KR 1992)* (pp. 165–176). Cambridge, MA, USA: Morgan Kaufmann Publishers.

Rao, A. S. & Georgeff, M. P. (1997). Modeling Rational Agents with a BDI-Architecture. In M. N. Huhns & M. P. Singh (Eds.), *Readings in Agents* (pp. 317–328). San Francisco, CA, USA: Morgan Kaufmann Publishers.

Rosenschein, J. S. & Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers.* Cambridge, MA, USA: MIT Press.

Russell, S. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (3rd ed.). Upper Saddle River, NJ, USA: Prentice Hall.

Sandholm, T. W. (1999). Distributed Rational Decision Making. In G. Weiss (Ed.), *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence* (pp. 201–258). Cambridge, MA, USA: MIT Press.

Schillo, M., Fischer, K. & Siekmann, J. H. (2003). The Link between Autonomy and Organisation in Multiagent Systems. In V. Mařík, D. C. McFarlane & P. Valckenaers (Eds.), *1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS 2003)* (pp. 81–90). Prague, Czech Republic: Springer-Verlag.

Schuldt, A. & Gottfried, B. (2008a). Selbststeuerung in der Intralogistik: Kognitive räumliche Repräsentationen für autonome Fahrzeuge. *Industrie Management*, *24*(4), 41–44.

Schuldt, A. & Gottfried, B. (2008b). Spatiotemporal Cooperation at the Geographic Scale. In B. Loos & V. Micelli (Eds.), *Interdisciplinary College (IK 2008)* (p. 32). Günne, Germany.

Schuldt, A. & Werner, S. (2007a). A Clustering Protocol for Team Formation Based on Concept, Location, and Time. In M. Dastani & R. H. Bordini (Eds.), *5th European Workshop on Multi-Agent Systems (EUMAS 2007)* (pp. 193–196). Hammamet, Tunisia.

Schuldt, A. & Werner, S. (2007b). Towards Autonomous Logistics: Conceptual, Spatial, and Temporal Criteria for Container Cooperation. In H.-D. Haasis, H.-J. Kreowski & B. Scholz-Reiter (Eds.), *1st International Conference on Dynamics in Logistics (LDIC 2007)* (pp. 311–319). Bremen, Germany: Springer-Verlag.

Silva, C. A., Runkler, T. A., Sousa, J. M. & Palm, R. (2002). Ant Colonies as Logistic Processes Optimizers. In M. Dorigo, G. Di Caro & M. Sampels (Eds.), *3rd International Workshop on Ant Algorithms (ANTS 2002)* (pp. 76–87). Brussels, Belgium: Springer-Verlag.

Singh, M. P. (2003). Agent Communication Languages: Rethinking the Principles. In M.-P. Huget (Ed.), *Communications in Multiagent Systems* (pp. 37–50). Heidelberg, Germany: Springer-Verlag.

Smith, R. G. (1977). The Contract Net: A Formalism for the Control of Distributed Problem Solving. In R. Reddy (Ed.), *5th International Joint Conference on Artificial Intelligence (IJCAI 1977)* (p. 472). Cambridge, MA, USA: William Kaufmann.

Timm, I. J. (2003). *Dynamisches Konfliktmanagement als Verhaltenssteuerung Intelligenter Agenten.* Dissertation, Universität Bremen.

Timm, I. J., Scholz, T. & Herzog, O. (2006). Capability-Based Emerging Organization of Autonomous Agents for Flexible Production Control. *Advanced Engineering Informatics*, *20*(3), 247–259.

Timm, I. J., Scholz, T., Herzog, O., Krempels, K.-H. & Spaniol, O. (2006). From Agents to Multiagent Systems. In S. Kirn, O. Herzog, P. Lockemann & O. Spaniol (Eds.), *Multiagent Engineering: Theory and Applications in Enterprises* (pp. 35–51). Heidelberg, Germany: Springer-Verlag.

Triebig, C., Credner, T., Fischer, P., Leskien, T., Deppisch, A. & Landvogt, S. (2005). Agent-Based Simulation for Testing Control Software of High Bay Warehouses. In T. Eymann, F. Klügl, W. Lamersdorf, M. Klusch & M. N. Huhns (Eds.), *3rd German Conference on Multiagent Sys-*

tem Technologies (MATES 2005) (pp. 229–234). Koblenz, Germany: Springer-Verlag.

van de Vijsel, M. & Anderson, J. (2004). Coalition Formation in Multi-Agent Systems under Real-World Conditions. In L.-K. Soh & J. E. Anderson (Eds.), *2004 AAAI Workshop on Forming and Maintaining Coalitions and Teams in Adaptive Multiagent Systems* (pp. 54–60). Menlo Park, CA, USA: AAAI Press.

van der Putten, S., Robu, V., La Poutré, H., Jorritsma, A. & Gal, M. (2006). Automating Supply Chain Negotiations Using Autonomous Agents: A Case Study in Transportation Logistics. In H. Nakashima, M. P. Wellman, G. Weiss & P. Stone (Eds.), *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)* (pp. 1506–1513). Hakodate, Japan.

Weiß, G. & Jakob, R. (2005). *Agentenorientierte Softwareentwicklung.* Heidelberg, Germany: Springer-Verlag.

Wooldridge, M. (1999). Intelligent Agents. In G. Weiss (Ed.), *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence* (pp. 27–77). Cambridge, MA, USA: MIT Press.

Wooldridge, M. (2000). *Reasoning about Rational Agents.* Cambridge, MA, USA: MIT Press.

Wooldridge, M. (2002). *An Introduction to Multiagent Systems.* Chichester, UK: John Wiley & Sons.

Wooldridge, M. & Jennings, N. R. (1995). Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, *19*(2), 115–152.

Wooldridge, M. & Jennings, N. R. (1999). The Cooperative Problem Solving Process. *Journal of Logic & Computation*, *9*(4), 563–592.

Chapter 5

Potential for Cooperation in Autonomous Logistics

To transform logistics objects in accordance with customer demands, primary logistics functions must be applied (Section 2.1). The applicability of centralised control of supply networks is limited by the complexity, the dynamics, and the distribution of logistics processes (Section 2.3). This finding can be explained by the high number of logistics objects, their manifold parameters, and the dynamic environment. Conventional approaches take a centralised perspective which also requires that all information is centrally available. The paradigm of autonomous logistics aims at overcoming the limitations of conventional control by shifting the perspective to the logistics entities themselves (Section 3.1). These previously inanimate logistics units are provided with logistics objectives by their owners. The entities are then responsible for satisfying their predefined objectives autonomously by requesting execution of the primary logistics functions. Hence, the perspective shifts from individual logistics functions to coordinating all of them.

Agent technology is considered an appropriate means for implementing autonomous logistics (Section 4.1). Each logistics entity can be represented by an intelligent software agent that acts on behalf of the entity. Hereinafter, for the sake of readability, autonomous logistics entities and their agents are thus only distinguished if confusion might result. Not only service consumers but also service providers are implemented by software agents. As elaborated in Section 4.1.2, there is a broad variety of implementations for intelligent agents. Implementing all aspects of service providers is addressed by other authors (see Section 4.3.3) and thus out of the scope of this thesis. A fortiori, it is important to find an adequate abstraction by defining interfaces of the respective entities. Together with other static aspects, the specification of participating entities can be summarised as the structure of autonomous logistics networks.

Logistics entities must interact and cooperate with each other to achieve their logistics objectives (Section 3.1.1). Hence, the focus of this research project lies on the interaction between agents rather than on the internal implementation of their behaviour. This focus incorporates specifying interac-

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics*,
DOI 10.1007/978-3-642-20092-2_5

tion protocols that structure encounters in autonomous logistics. Autonomous logistics entities interact without predefined organisational structures (Section 3.1.1). A particular challenge is the interaction complexity of autonomous logistics processes (Section 3.1.3). As the interaction effort increases with the number of participating entities, an important research question is how autonomous logistics entities can by themselves establish organisational structures that reduce the interaction effort.

The implementation of the autonomous logistics paradigm with multiagent technology is guided by the steps of the model for cooperation by Wooldridge and Jennings (Section 4.3.2). This chapter investigates the potential for cooperation of autonomous logistics entities. To this end, structural aspects of autonomous logistics networks and organisational structures of their participants are examined. Subsequently, Chapters 6 and 7 address the interaction of autonomous logistics entities in terms of team formation and team action including plan formation. The remainder of this chapter is structured as follows. As a foundation, Section 5.1 describes the structure of autonomous logistics networks and identifies their major participants. This includes identifying the requirements for intelligent software agents that participate in autonomous logistics processes. Subsequently, Section 5.2 examines organisational structures that reduce the interaction effort and optimise autonomous logistics processes.

5.1 Participants in Autonomous Logistics

This research project examines interaction in autonomous logistics. Its focus is therefore on the dynamic aspects of logistics processes. An important foundation for this analysis, however, is a thorough investigation of the underlying logistics network itself. Autonomous logistics networks can be defined by two aspects. Firstly, the actors that populate the network. These actors are the participants in the interaction protocols to be examined later on. Secondly, the spatial representation of the logistics network itself. This representation is most fundamental because it is the environment in which the actors operate. A quantitative model of the environment would simply provide each logistics entity with its geographic coordinates, i. e., latitude and longitude. This data can be obtained easily with localisation technology (Section 3.2.2). Despite of its precision, however, such data reveals nothing about the qualitative relationship between two positions. To recapitulate, Section 4.2.3 gives an example of two shipping containers which are located nearby each other in a port. One might assume that they could be loaded onto a train together. Unfortunately, they are located in two neighbouring container terminals which are connected to different tracks. Joint transport by a shared train is thus impossible.

This example illustrates that it is useful to represent knowledge about the environment explicitly. This can be achieved by tessellating the environment into qualitatively distinct regions that cover whole company sites or logistics facilities. It must be ensured that geographic positions of logistics entities can be mapped to such regions unambiguously. To this end, it is important that the areas that evolve from the partitioning are disjoint. This requirement can be expressed by the topological relations (Figure 4.7) of the Region Connection Calculus (Randell, Cui & Cohn, 1992, p. 172). It must be ensured that all pairs of two different regions r, r' out of the set M of all regions are either disconnected (DC) or externally connected (EC):

$$\forall_{r \in M} \forall_{r' \in M}\ r \neq r' \rightarrow \big(\mathrm{DC}(r, r') \vee \mathrm{EC}(r, r')\big) \tag{5.1}$$

This means that there is no intersection between two regions. The whole topology of autonomous logistics networks can be modelled by a graph. In this graph, each relevant region is represented by a vertex. The restriction to relevant regions is intentional. It allows confining the graph to regions which actually correspond to company sites. All other regions can be left out to reduce reasoning complexity.

Transport is one of the primary functions in logistics (Section 2.1.1). In order to take this into account, transport relations between the locations of the logistics network are modelled by directed edges of the graph. The whole environment is thus defined as follows:

Definition 5.1 (Environment) *Let the set of locations within an autonomous logistics network be denoted by* $\mathcal{L}$. *Let the set of directed transport relations that connect locations in* $\mathcal{L}$ *be denoted by* $\mathcal{R} \subseteq \mathcal{L} \times \mathcal{L}$. *Together,* $\mathcal{L}$ *and* $\mathcal{R}$ *form a graph* $\mathcal{E}$ *that models the environment of the autonomous logistics network:*

$$\mathcal{E} := (\mathcal{L}, \mathcal{R})$$

Note that the transport relations do not directly correspond to road, train, or waterway networks. Instead, the semantics of an edge is that transport between the connected vertices is possible. Based on this representation, autonomous logistics entities can choose appropriate means of transport. Routing, however, is not of concern for these entities. This task is left to the transport service providers that actually execute the transport. For this purpose, they may have an additional representation of the underlying road, train, or waterway network. A foundation for efficient routing within such graphs are the algorithm of Dijkstra (1959, pp. 270–271) as well as A* by Hart, Nilsson and Raphael (1968, p. 102).

Logistics networks are populated by different types of actors. Previous approaches categorise the participants of logistics networks as being active or passive (Section 3.1.1). However, this categorisation does not apply to the actors in autonomous logistics networks. All autonomous logistics entities are active participants in the logistics processes. This even includes previously inanimate logistics objects. Nevertheless, the participants of autonomous logistics networks can be distinguished into consumers and providers of logistics services. As elaborated in Section 2.1, the main focus of this research is on general cargo. General cargo units must be transformed by appropriate processes from an initial to a final state (Figure 2.1). This may incorporate transformation of time, place, quantity, composition, and quality. The general cargo units are thus considered to be consumers of logistics services:

Definition 5.2 (Set of Logistics Service Consumers) *Let the set of all service consumers within a given autonomous logistics network be denoted by* $\mathcal{SC}$.

Moreover, they act as autonomous logistics entities to which process control is delegated. That is, the general cargo units are themselves responsible for requesting appropriate transformations. To achieve their logistics objectives, the cargo units thus negotiate with logistics service providers which are autonomous logistics entities as well.

Service providers can be distinguished by the primary logistics functions they offer, namely storage, picking, transport, and handling. The set of all service providers is thus defined as follows:

Definition 5.3 (Set of Logistics Service Providers) *Let the set of all service providers within a given autonomous logistics network be denoted by* $\mathcal{SP}$. *It is defined as the union of the sets of transport service providers* $\mathcal{TSP}$, *handling service providers* $\mathcal{HSP}$, *storage service providers* $\mathcal{SSP}$, *and picking service providers* $\mathcal{PSP}$:

$$\mathcal{SP} := \mathcal{TSP} \cup \mathcal{HSP} \cup \mathcal{SSP} \cup \mathcal{PSP}$$

The sets of service consumers and providers are not necessarily disjoint. Instead, an agent may take several roles, thereby, for instance, requesting services from providers and reselling these services to other consumers. Definition 5.3 does not restrict who actually is a provider of primary logistics functions. Other approaches consider companies to be such service providers (Section 4.3.3). By contrast, the concept of autonomous logistics chooses a finer granularity. In this paradigm, the actual facility is considered to be the logistics service provider. That is, logistics objects must negotiate, for instance, with concrete storage positions of a warehouse.

The logistics service consumers and providers cover most actors within autonomous logistics networks. Nevertheless, there also exist agents or agent roles which do not pertain to these groups. As an example, consider broker or directory agents (Section 4.3) which support autonomous logistics entities. They cannot be categorised as logistics service providers because their service is general and not directly related to logistics functions. Hence, the term auxiliary agents is introduced for these actors:

Definition 5.4 (Set of Auxiliary Agents) *Let the set of auxiliary agents within a given autonomous logistics network be denoted by* $\mathcal{A}$.

Based on the preceding Definitions 5.1 to 5.4, autonomous logistics networks are defined as follows:

Definition 5.5 (Autonomous Logistics Network) *Let $\mathcal{ALN}$ be an autonomous logistics network. It is defined by an environment graph $\mathcal{E}$, autonomous logistics entities that act as service consumers $\mathcal{SC}$ or as service providers $\mathcal{SP}$, and auxiliary agents $\mathcal{A}$:*

$$\mathcal{ALN} := (\mathcal{E}, \mathcal{SC}, \mathcal{SP}, \mathcal{A})$$

Autonomous logistics networks are dynamic and thus change over time. On the one hand, this applies to the participants which may join and leave the system during runtime. For instance, a general cargo unit joins the system if it demands logistics transformations. Likewise, it leaves the system as soon as the transformations are completed. On the other hand, also the environment graph may be dynamic. Depending on logistics demands, it may be extended or reduced by locations and transport relations. Hence, the static description of autonomous logistics networks is just a snapshot. Each snapshot characterises the state of a network at a specific point in time. Taking into account the dynamics, it is thus necessary to characterise each autonomous logistics network by a series of such descriptions:

$$\mathcal{ALN}_0 \rightarrow \mathcal{ALN}_1 \rightarrow \ldots \rightarrow \mathcal{ALN}_n \rightarrow \ldots \tag{5.2}$$

The following sections investigate the participants of autonomous logistics networks in more detail. Section 5.1.1 addresses general cargo units as the service consumers within logistics networks. Subsequently, Sections 5.1.2 to 5.1.5 examine the respective logistics service providers that offer transport, handling, storage, and picking to the cargo.

5.1.1 General Cargo Units

Cargo units are considered the service consumers in autonomous logistics processes. The emphasis here is on unit. The autonomous logistics paradigm envisions that logistics objects are represented by autonomous logistics entities which act on their behalf. Hence, it is necessary to establish a mapping between autonomous logistics entities and the corresponding cargo units. At first glance, this conclusion prevents bulk cargo from being considered by autonomous logistics. Bulk cargo is generally lumpy, granular, or dusty, i. e., it has a fluid state of matter (Section 2.1). Therefore, bulk cargo does not have a defined shape or clearly distinguishable units. It is thus impossible to identify individual units of bulk cargo that could act as autonomous logistics entities. A solution to this dilemma is to employ containers which allow handling bulk cargo, fluids, and gases as general cargo (Section 2.1). Then, the mapping between general cargo units and autonomous logistics entities becomes trivial.

For other general cargo, an adequate granularity at which units act as autonomous logistics entities is not that obvious. For instance, components, articles, sales units, cardboard boxes, pallets, or shipping containers could be chosen as autonomous logistics entities. This choice depends on the specific application intended. Consider, for instance, mail order businesses. In this domain, sales units can be considered atomic units because they do not change their composition throughout the whole supply network. A sales unit has the same condition when it leaves the vendor and when it arrives at the end customer. Hence, it is not necessary to choose a finer granularity.

Independently from the granularity chosen, each general cargo unit

1. has a location,
2. has logistics objectives to be satisfied,
3. is characterised by certain properties, and
4. has a load carrier with defined properties.

This leads to the following definition:

Definition 5.6 (General Cargo Unit) *A general cargo unit* $\mathrm{gcu} \in \mathcal{SC}$ *is defined by its location* $\mathrm{l} \in \mathcal{L} \cup \mathcal{R}$*, its logistics objectives* $\mathrm{o} \in \mathcal{O}$*, and the descriptors* $\mathrm{d_c} \in \mathcal{D}$ *for its cargo and* $\mathrm{d_{lc}} \in \mathcal{D}$ *for its load carrier:*

$$gcu := (l, o, d_c, d_{lc})$$

Each general cargo unit is an autonomous logistics entity. Thus, it has the capabilities specified by the architecture developed in Section 3.2. For instance, the localisation unit (Section 3.2.2) allows localising the autonomous logistics entity within the logistics network. Its location

$$l \in \mathcal{L} \cup \mathcal{R} \tag{5.3}$$

can be out of the set of vertices $\mathcal{L}$ of the environment graph $\mathcal{E}$. If a general cargo unit is currently transported, it is located on one of the transport relations $\mathcal{R}$. The concrete implementation of the localisation unit is out of the scope of this thesis. Depending on customer demands, it can be based on Global Navigation Satellite Systems (GNSS). This allows tracking and tracing with a fine resolution. Alternatively, it may be sufficient to apply the identification unit (Section 3.2.1) for beacon-based localisation at relevant locations within the network (Figure 3.7).

In accordance with the general objectives of logistics (Chapter 2), autonomous logistics entities are expected to arrive at a scheduled place and time in both the right quantity and quality. Moreover, they are supposed to satisfy these requirements for the right price. Some of the goals may be less precisely specified than others. For instance, a company may have different storage facilities. A general cargo unit might then be allowed to choose the storage facility that fits its demands best. Likewise, different alternatives for transport may exist. Furthermore, also the scheduled time of arrival is not necessarily fixed but may span a time window that depends on environmental changes measured, for instance, by the sensor unit (Section 3.2.3). As an example, consider an increase in temperature which reduces the shelf life of the cargo. A representation of logistics objectives depends on the specific agent model and implementation chosen (Section 4.1.2). Hence, the following definition of the set of logistics objectives is generic:

Definition 5.7 (Set of Logistics Objectives) *Let the set of logistics objectives for general cargo units be denoted by* $\mathcal{O}$.

In order to achieve their objectives, autonomous logistics entities must request execution of the primary logistics functions transport, handling, storage, and picking (Section 2.1). The communication unit (Section 3.2.4) enables negotiation with logistics service providers. An example is depicted in Figure 5.1. A general cargo unit of high monetary value has to choose an appropriate storage facility. On the one hand, this choice depends on the means of transport available. On the other hand, it is important to place emphasis on the term appropriate here. That is, the storage facility should be secured in order to prevent theft. Not only the properties of the cargo itself play an important role when choosing service providers. Besides, also the properties of its load carrier must be considered. For instance, goods in warehouses are usually stored on pallets. General cargo units that are delivered by shipping container must thus be re-packed on reception. This leads to two descriptors that characterise general cargo units. One for the cargo itself, the other for the current load carrier:

Definition 5.8 (Set of Descriptors) *Let the set of descriptors for general cargo units and their load carriers be denoted by* $\mathcal{D}$.

The descriptor $d_c \in \mathcal{D}$ characterises the cargo properties of the general cargo unit, $d_{lc} \in \mathcal{D}$ its load carrier properties. Definition 5.8 is generic in order to be independent from concrete implementations. The descriptors can, for instance, be defined by description logics concepts (Section 4.2.3). Then, it holds that

$$d_c \sqsubseteq \top \quad \text{and} \quad d_{lc} \sqsubseteq \top \tag{5.4}$$

As illustrated in Figure 4.6, applying description logics concepts allows determining a match with the properties of logistics service providers by computing the subsumption of the descriptors (Werner, 2006, p. 67). In the architecture of autonomous logistics entities, the descriptors can be integrated into the data processing unit. As an alternative, they can also be delegated to the identification unit. The descriptors enable classification rather than identification. Nevertheless, it can be motivated that this is identification of the membership to a particular class of objects.

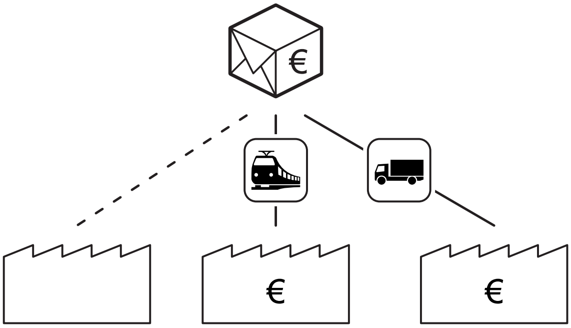

Fig. 5.1 A general cargo unit with high worth. When choosing a storage facility, it must consider that only secured storage facilities are appropriate. Besides, its choice also depends on the means of transport available.

5.1.2 Providers of Transport Services

Means of transport, such as trucks, trains, or barges, can be employed in order to bridge space (Section 2.1.1). They connect locations within the logistics network by enabling transport of general cargo units. To recapitulate, within the context of this research, the emphasis is not on routing. Means of transport must, of course, plan their way through a road, train, or waterway network. However, this does not concern the cargo that requests transport from one location to another. In autonomous logistics, means of transport are considered to be atomic transport service providers, i. e., individual trucks, trains, and barges. This is in contrast to some previous applications (Section 4.3.3) which consider whole forwarding companies as service providers. Each transport service provider

1. is capable of transporting specific goods,
2. on specific load carriers,
3. between locations,
4. with a defined capacity,
5. within a certain time span, and
6. for a certain price.

The definition of transport service providers is thus as follows:

Definition 5.9 (Transport Service Provider) *A transport service provider* $\mathrm{tsp} \subset \mathcal{TSP}$ *is defined by the transport relations* $\mathrm{R} \subseteq \mathcal{R}$ *it serves, the descriptors for cargo* $\mathrm{d_c} \in \mathcal{D}$ *and for load carriers* $\mathrm{d_{lc}} \in \mathcal{D}$ *it can transport, its* $\mathrm{lotSize} \in \mathbb{R}^+$*, as well as the mappings* $\mathrm{match_c}$*,* $\mathrm{match_{lc}}$*,* duration*,* capacity *and* cost*:*

$$tsp := (R, d_c, d_{lc}, lotSize, match_c, match_{lc}, duration, capacity, cost)$$

The transport relations R served by a transport service provider are a subset of all transport relations $\mathcal{R}$ of the logistics network:

$$R \subseteq \mathcal{R} \tag{5.5}$$

The locations and transport relations of the logistics network jointly form a directed graph (Definition 5.1). Hence, a directed connection between two locations is one way unless both directions explicitly belong to the set of transport relations. Different directions may differ in their price, e. g., if one direction is more utilised than the other.

In general, not every transport facility can transport all types of goods. Therefore, general cargo units must identify transport providers that offer appropriate services. Analogously to the example in Section 5.1.1, goods with high worth must be transported by secured means of transport. Another example are perishable goods which must be transported by a refrigerated means of transport. As elaborated in Section 5.1.1, also the load carrier of cargo can restrict the choice of logistics service providers. For instance, consider a tractor unit for swap bodies. This unit is not capable of transporting standardised pallets as long as they are not re-packed into a swap body. The properties of both cargo and load carriers are characterised by descriptors (Definition 5.8) that allow reasoning about their compatibility with transport service providers. To this end, also the service provider must be assigned respective descriptors. The following mappings determine whether cargo and load carrier of a general cargo unit *gcu* are compatible with the transport service provider *tsp*:

$$match_c : \mathcal{D} \times \mathcal{D} \to \mathbb{B} \quad \text{and} \quad match_{lc} : \mathcal{D} \times \mathcal{D} \to \mathbb{B} \tag{5.6}$$

$$match_c : (d_c^{gcu}, d_c^{tsp}) \mapsto b \quad \text{and} \quad match_{lc} : (d_{lc}^{gcu}, d_{lc}^{tsp}) \mapsto b \tag{5.7}$$

The mappings and descriptors with subscripts c and lc refer to cargo and load carrier properties, respectively. The superscripts *gcu* and *tsp* refer to general cargo units and transport service providers, respectively. It may happen that there is a match for the cargo itself but not for its load carrier. As explicated in Section 2.1.2, it is then necessary to change the load carrier with handling. To reiterate the example, consider goods that are stored on pallets in a warehouse. In order to transport them by a tractor unit, it is necessary to re-pack the goods into a suitable swap body.

Section 5.1.1 proposes to implement the descriptors with description logics concepts. Then, cargo and transport facility are compatible if the descriptor of the former is subsumed by the respective descriptor of the latter:

$$match_c(d_c^{gcu}, d_c^{tsp}) := \begin{cases} \text{true} & \text{if } d_c^{gcu} \sqsubseteq d_c^{tsp} \\ \text{false} & \text{otherwise} \end{cases} \tag{5.8}$$

$$match_{lc}(d_{lc}^{gcu}, d_{lc}^{tsp}) := \begin{cases} \text{true} & \text{if } d_{lc}^{gcu} \sqsubseteq d_{lc}^{tsp} \\ \text{false} & \text{otherwise} \end{cases} \tag{5.9}$$

Computing the subsumption between concepts does not incorporate instances of these concepts (Li & Horrocks, 2004, p. 335). Instances are not employed because TBox reasoning is more efficient than ABox reasoning (cf. Tessaris, 2001, pp. 26–27).

As an intermediary result, it is possible to determine whether a transport service provider is capable of transporting particular cargo. The next question to be answered is whether the provider actually has enough capacity. To determine how long cargo utilises the capacity of the provider, it must estimate the duration for transport on a relation at a certain time:

$$duration : \mathcal{R} \times \mathbb{N} \to \mathcal{I} \tag{5.10}$$

$$duration : (relation, date) \mapsto Interval \tag{5.11}$$

The date is a natural number that represents the points in time elapsed since a fixed start time. Depending on the time granularity demanded, it can, for instance, be measured in hours, minutes, seconds, or milliseconds. The outcome of the mapping is a discrete temporal interval:

Definition 5.10 (Discrete Temporal Interval) *Let the set of all discrete temporal intervals be denoted by $\mathcal{I}$. A discrete temporal interval* Interval $\in \mathcal{I}$ *is defined as a set*

$$Interval = \{t_b, \ldots, t_e\} \subseteq \mathbb{N}$$

with $\mathrm{t_b}$ *denoting the begin time,* $\mathrm{t_e}$ *denoting the end time, and* $\mathrm{t_b} \leq \mathrm{t_e}$.

Note that the actual duration, may additionally depend on external factors such as traffic and weather conditions.

In combination with the particular transport relation and the amount of cargo to be transported, the transport service provider can employ the duration interval in order to determine whether it has sufficient capacity:

$$capacity : \mathcal{R} \times \mathcal{I} \times \mathbb{R} \to \mathbb{B} \tag{5.12}$$

$$capacity : (relation, Interval, amount) \mapsto b \tag{5.13}$$

The amount of transport capacity demanded by cargo is not necessarily measured in terms of its current load carrier. The load carrier may change during handling or picking (Sections 2.1.2 and 2.1.4). Therefore, cargo must provide appropriate units when requesting transport capacity.

Usually, transport service providers have a certain minimum utilisation, the so-called *lotSize*. Consider, for instance, a truck that has the capacity to carry several pallets. A consumer that aims at employing this truck must book the whole truck even if it actually utilises only a fraction of the capacity available. An example is an individual sales unit that is transported by the truck. In fact, the truck could carry even hundreds of sales units. For optimal utilisation, it is thus often beneficial for general cargo units to cooperate in using service providers. The actual amount to be reserved can be determined with a mapping of the following signature:

$$\mathit{amount} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \tag{5.14}$$

$$\mathit{amount} : (\mathit{lotSize}, \mathit{amount}') \mapsto a \tag{5.15}$$

Given a particular *lotSize* and a desired *amount′*, the mapping determines the actual amount a. The mapping can be implemented as follows:

$$\mathit{amount}(\mathit{lotSize}, \mathit{amount}') := \left\lceil \frac{\mathit{amount}'}{\mathit{lotSize}} \right\rceil \cdot \mathit{lotSize} \tag{5.16}$$

The costs for transport depend on the transport relation served as well as on the temporal interval of the transport and the amount of cargo:

$$\mathit{cost} : \mathcal{R} \times \mathcal{I} \times \mathbb{R} \rightarrow \mathbb{R} \tag{5.17}$$

$$\mathit{cost} : (\mathit{relation}, \mathit{Interval}, \mathit{amount}) \mapsto \mathit{price} \tag{5.18}$$

5.1.3 Providers of Handling Services

Handling is applied for loading and unloading of cargo as well as switching means of transport. This often also includes adjusting quantities of goods (Section 2.1.2). To reiterate an example from Section 5.1.1, shipping containers are a common load carrier for transport. By contrast, one uses pallets in storage facilities, particularly in high-bay warehouses. For storage, it is thus necessary to re-pack sales units after their delivery from shipping containers onto pallets. This task is conducted by handling service providers such as automated palletising systems. A handling service provider

1. has a location,
2. is capable of changing the load carrier of cargo,
3. with a defined capacity, and
4. for a certain price.

Therefore, handling service providers are defined as follows:

Definition 5.11 (Handling Service Provider) *A handling service provider* $\mathrm{hsp} \subset \mathcal{HSP}$ *is defined by its locations* $\mathrm{L} \subseteq \mathcal{L}$*, the descriptors for initial load carrier* $\mathrm{d_{lc}}$ *and terminal load carrier* $\mathrm{d'_{lc}}$*, its* $\mathrm{lotSize} \in \mathbb{R}^+$*, as well as the* *mappings* $\mathrm{match}_{\mathrm{lc}}$, $\mathrm{match}'_{\mathrm{lc}}$, duration, capacity, *and* cost*:*

$$hsp := (L, d_{lc}, d'_{lc}, lotSize, match_{lc}, match'_{lc}, duration, capacity, cost)$$

The location of a handling service provider is defined by a subset of vertices in the graph of the autonomous logistics network:

$$L \subseteq \mathcal{L} \tag{5.19}$$

For atomic handling service providers, it holds that $|L| = 1$.

The initial and terminal load carrier of the cargo can be defined by descriptors $d_{lc} \in \mathcal{D}$ and $d'_{lc} \in \mathcal{D}$, respectively. Comparable to Equation 5.6, each handling service provider has two mappings $match_{lc}$ and $match'_{lc}$ that determine its compatibility with initial and desired terminal cargo states.

The mapping for determining the capacity of a handling service provider resembles the one of transport providers (Equation 5.12). The difference, however, is that the capacity depends on the respective location at which the service is provided rather than the transport relation served:

$$capacity : \mathcal{L} \times \mathcal{I} \times \mathbb{R} \rightarrow \mathbb{B} \tag{5.20}$$

$$capacity : (location, Interval, amount) \mapsto b \tag{5.21}$$

The same holds for the mapping determining the handling costs:

$$cost : \mathcal{L} \times \mathcal{I} \times \mathbb{R} \rightarrow \mathbb{R} \tag{5.22}$$

$$cost : (location, Interval, amount) \mapsto price \tag{5.23}$$

A prerequisite for finding out whether a handling service provider has enough capacity and which costs arise is the time span during which it is utilised. This temporal interval depends on the location at which the handling is conducted as well as the date. Like for transport providers (Equation 5.10), it can be estimated with the following mapping:

$$duration : \mathcal{L} \times \mathbb{N} \rightarrow \mathcal{I} \tag{5.24}$$

$$duration : (location, date) \mapsto Interval \tag{5.25}$$

5.1.4 Providers of Storage Services

The purpose of storage facilities (Section 2.1.3) is to bridge time, i. e., to establish a buffer for goods by storing them for a certain amount of time. This is necessary whenever incoming and outgoing material flows are not synchronised. Warehouses are a prominent example for storage facilities. Another example are container terminals which act as an intermediary store for shipping containers between their shipment and pre and onward carriage, respectively. Additionally, selectors can be regarded as storage facilities. These companies re-package or repair damaged goods. Abstracting from the process of quality improvement, they store the cargo for a certain amount of time. For the implementation of autonomous logistics processes, the granularity of modelling plays an important role not only for service consumers but also for service providers. Here, the particular question is what is considered a storage service provider and thus an autonomous logistics entity. In previous approaches, software agents represent, for instance, whole forwarding companies (Section 4.3.3). An alternative is to consider warehouses as service providers. Both choices, however, are rather coarse and thus centralised. For autonomous logistics, it is again necessary to identify atomic units as autonomous logistics entities. Following this principle, individual storage positions within storage facilities are considered logistics service providers. Each storage service provider

1. has a location,
2. is capable of storing specific goods,
3. on specific load carriers,
4. with a defined capacity, and
5. for a certain price.

The properties lead to the following definition:

Definition 5.12 (Storage Service Provider) *A storage service provider* $\mathrm{ssp} \in \mathcal{SSP}$ *is defined by its locations* $\mathrm{L} \subseteq \mathcal{L}$*, the descriptors for cargo* $\mathrm{d_c} \in \mathcal{D}$ *and for load carriers* $\mathrm{d_{lc}} \in \mathcal{D}$ *it can store, its* $\mathrm{lotSize} \in \mathbb{R}^+$*, as well as the mappings* $\mathrm{match_c}$*,* $\mathrm{match_{lc}}$*,* $\mathrm{capacity}$*, and* cost*:*

$$ssp := (L, d_c, d_{lc}, lotSize, match_c, match_{lc}, capacity, cost)$$

The location of storage facilities is defined by locations within the autonomous logistics network (Equation 5.19). For atomic storage service providers, it holds that $|L| = 1$. The descriptors and mappings for both cargo and load carriers that can be stored correspond to those of transport service providers (Section 5.1.2).

Storage service providers are limited in their space capacity, i. e., the maximum amount of goods that can be stored by their storage facility at the same time. Having figured out that cargo can be received by a storage service provider, the respective provider must thus check whether enough capacity is available. The capacity of storage service providers is defined as in Equation 5.20. The cost for storage can be determined like the cost for handling services (Equation 5.22). In contrast to transport and handling service providers, it is not necessary to have a mapping that estimates the utilisation duration (Equations 5.10 and 5.24). Instead, the utilisation interval is specified by the storage time requested by the consumer.

Each storage service provider may have its own concrete cost mapping. These mappings may incorporate different prices at different locations. Besides, storage providers may offer lower prices at times with low utilisation. As a notable example, consider container terminals. Usually, containers are granted a certain time during which they can remain at the container terminal for free. After this period of time, the owner must pay demurrage for late pickup. Depending on the contract of the container owner, demurrage may increase with time. Another cost factor to be considered is detention which must be paid for returning empty shipping containers late.

5.1.5 Providers of Picking Services

With regard to handling efficiency, pallets in a warehouse are usually composed in item order (Section 2.1.4). That is, all goods on a pallet are of the same type. This is in contrast to customer orders which are usually compiled from different articles. The gap is bridged by picking as the fourth primary logistics function. Its task is to compile customer orders from quantities in item order. A picking service provider

1. has a location,
2. is capable of changing the composition of cargo,
3. with a defined capacity, and
4. for a certain price.

The definition of picking service providers is as follows:

Definition 5.13 (Picking Service Provider) *A picking service provider* $\mathrm{psp} \in \mathcal{PSP}$ *is defined by its locations* $\mathrm{L} \subseteq \mathcal{L}$*, the descriptors for initial cargo state* $\mathrm{d_c}$*, terminal cargo state* $\mathrm{d'_c}$*, initial load carrier* $\mathrm{d_{lc}}$*, and terminal load carrier* $\mathrm{d'_{lc}}$*, its* $\mathrm{lotSize} \in \mathbb{R}^+$*, as well as the mappings* $\mathrm{match_c}$*,* $\mathrm{match'_c}$*,* $\mathrm{match_{lc}}$*,* $\mathrm{match'_{lc}}$*,* $\mathrm{duration}$*,* $\mathrm{capacity}$*, and* cost*:*

$$psp := (L, d_c, d'_c, d_{lc}, d'_{lc}, lotSize,\\ match_c, match'_c, match_{lc}, match'_{lc}, duration, capacity, cost)$$

In practical execution, handling and picking are clearly distinguished from each other. Handling is usually conducted with support by technical systems such as forklifts or palletisers (Section 2.1.2). By contrast, picking processes are generally carried out by humans (Section 2.1.4). Nevertheless, for a control perspective both handling and picking can be similarly abstracted. Like handling, picking also changes the quantity of the objects processed. The difference, however, is that atomic units of different kind are composed in accordance with customer orders. This means that the descriptor for the terminal state of the cargo differs from the descriptor of its initial state. The descriptor of a compiled order d_c^o can be determined based on the descriptors of the participating parts d_c^p, with D being the set of part descriptors. The subscript c stands for cargo, o and p refer to the order and its parts. In description logics, this can be expressed as follows:

$$d_{c_o} := \bigsqcup_{d_{c_p} \in D} d_{c_p} \tag{5.26}$$

5.2 Organisational Structures

The structure of autonomous logistics networks is defined by their environment and their participating entities (Section 5.1). Atomic units, e. g., sales units and storage positions, have been identified as an appropriate choice for autonomous logistics entities. Autonomous logistics assumes a heterarchy of the participating entities (Section 3.1.1). More precisely, for the sake of flexibility there are no or only weakly predefined organisational structures. This is a clear distinction from the hierarchical structure of previous approaches with centralised control.

At first glance, abandoning organisational structures might appear disadvantageous. Consider, for instance, a sales unit that has been manufactured in East Asia. This sales unit arrives in Europe and is expected to coordinate its storage until it is sold. Then, the sales unit must contact not only all warehouses but all individual storage positions in order to request offers for storage. Moreover, it is unlikely that a company has only one sales unit. By contrast, thousands or even millions must be handled. All of them must then communicate with the storage positions available. This leads to a high interaction effort which correlates with the number of sales units multiplied by the number of storage positions. Obviously, this results in a communication overhead. The high interaction effort is likely to outweigh the decrease in computational complexity that is gained by decentralisation (Section 3.1.3). Apart from reducing the interaction effort, organisational structures also facilitate optimisation. If an atomic storage position corresponds to the size of one pallet, it can usually contain more than only one sales unit. That is, multiple sales units of the same type should request re-packing to a pallet to jointly request the storage service.

Organisational structures help cope with these challenges (Section 4.3). Therefore, only predefined organisational structures are abandoned. The entities participating in autonomous logistics may establish organisational structures themselves whenever this is considered useful. Wooldridge and Jennings (1999, p. 578) define this usefulness as follows. There is a potential for cooperation (Section 4.3.2) for a specific goal if

1. a team can jointly achieve the goal

and

2. the agent cannot achieve the goal in isolation *or*
3. the agent has a goal conflict for all respective own actions.

Establishing organisational structures on demand allows reacting flexibly on changes of both the requirements and the environment. The lifespan of adaptively established organisations depends on the actual requirements and may range from short-term to long-term. Establishing organisational structures ad hoc presupposes cooperative behaviour. This requirement does not pose a problem if the cooperating entities belong to the same company. However, even in competitive environments, cooperation may prove advantageous. For instance, consider general cargo units of different companies which share a means of mass transport, e. g., train or barge. This allows them saving costs compared to employing trucks.

Organisations of agents can be modelled by holonic agents (Section 4.3.1). Following Fischer, Schillo and Siekmann (2003, p. 75), organisations are defined as follows:

Definition 5.14 (Team) *Let the set of all teams within a given autonomous logistics network be denoted by* $\mathcal{T} \subseteq \mathcal{A}$*. A* team $\in \mathcal{T}$ *is defined by the set of its* Managers*, the set of its* Members*, and the* commitment *of the members:*

$$team := (Managers, Members, commitment)$$

with commitment $\in \{C_{autonomous}, C_{association}, C_{merge}\}$*,* Members $\subseteq \mathcal{SC} \cup \mathcal{SP}$*,* Managers $\subseteq$ Members*.*

As depicted in Figure 4.11, the commitment distinguishes whether the members are autonomous, in an association, or merged. Also an individual agent $a \in \mathcal{SC} \cup \mathcal{SP}$ can be considered a team (Fischer et al., 2003, p. 75):

$$(\{a\}, \{a\}, C_{autonomous}) \tag{5.27}$$

Then, a is the only member and thus also the manager of its team.

Autonomous logistics entities can be distinguished into consumers and providers of logistics services (Section 5.1). Section 5.2.1 examines which organisational structures can be established by logistics service providers in order to decrease interaction effort. Correspondingly, Section 5.2.2 investigates which organisational structures help logistics service consumers achieve their objectives.

5.2.1 Teams of Logistics Service Providers

In the context of this research, atomic logistics units are considered to act as autonomous entities. This specification holds for both logistics service providers and consumers. Regarding storage, individual storage positions of larger facilities are considered atomic units. That is, service consumers must directly negotiate with storage positions if they request storage. As elaborated in the preceding introductory section, a potentially high interaction effort arises if every storage position negotiates individually and only on its own behalf. The question is thus which organisational structures of storage positions facilitate a decrease in interaction effort. A promising strategy is that storage positions with homogeneous properties form teams. One criterion for team formation can be spatial neighbourhood. As an example, consider storage positions that are located at the same storage facility. Note, however, that this criterion does often not suffice. Additional criteria are discussed below.

From the members of the team, a team manager is chosen that represents the team within the autonomous logistics network (Section 4.3.1). This representative is responsible for negotiation with service consumers (Figure 5.2). Hence, all service consumers that demand storage services can confine themselves to contacting distinguished team managers instead of all storage positions.

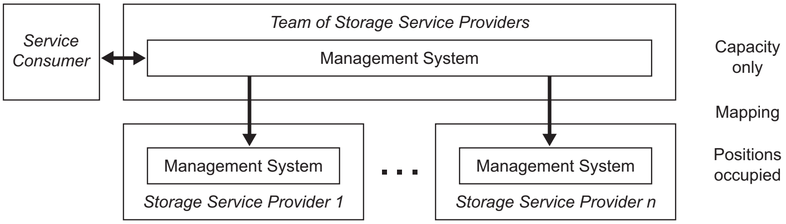

Fig. 5.2 A team of storage providers represents a homogeneous area within a physical storage facility. The team manager only administers the number of storage positions occupied. When the cargo is received, a mapping is performed to actual positions in the management system of the atomic service providers.

From the perspective of the service consumers, such teams can thus be regarded as black boxes (Figure 5.3). The storage capacity of all team members is aggregated and administered by the manager. The assignment to actual storage positions is then coordinated by the team members. Depending on the autonomy that is left to the members, they could also be merged, so that the manager also assigns actual storage positions itself.

Subsuming multiple storage positions within teams eases the process of matchmaking between service consumers and providers. The premise for team formation of storage positions is that they exhibit homogeneous properties.

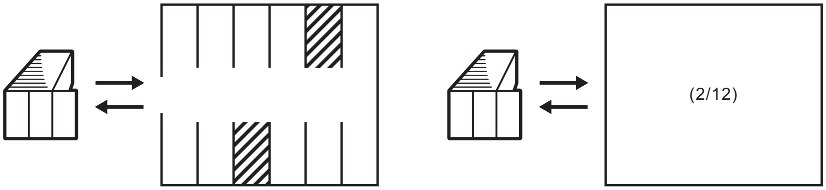

Fig. 5.3 General cargo units must contact every storage position of a storage facility in order to find out whether capacity is available (left). This leads to a high interaction effort. By contrast, storage positions can form a team (right). The team encapsulates its members so that cargo units have a single contact for the respective capacity.

In the discussion so far, spatial neighbourhood has been considered the criterion for homogeneity. However, spatial neighbourhood is only necessary but not sufficient for aggregating storage positions. Additionally, all participating storage positions must be able to store goods of the same properties for the same price. This requirement is not necessarily satisfied by storage positions that are located close to each other. As an example, consider special areas within a warehouse that are secured and thus capable of storing valuable goods while others are not. The top row of Figure 5.4 shows two possible configurations of such a warehouse with five of twelve storage positions occupied. If all twelve positions are aggregated within one team (Figure 5.4 bottom left), it is impossible to distinguish the two configurations. It is particularly impossible to decide whether an additional valuable shipping container can be received. By contrast, this can be decided if only storage positions with homogeneous properties are aggregated (Figure 5.4 bottom right). Then, two teams exist that administer the storage positions of the physical storage facility. In this case, it is sufficient that each team keeps track of its current capacity. A mapping to the concrete positions can be ensured by the team members. Note that the requirement for homogeneity is also the foundation for applying chaotic storage (Section 2.1.3).

The utilisation of a logistics service provider is represented as follows:

Definition 5.15 (Utilisation) *The utilisation of a logistics service provider is defined by a set*

$$Utilisations := \{utilisation_0, \ldots, utilisation_{n-1}\}$$

with

$$utilisation_i := (Interval, amount) \in \mathcal{I} \times \mathbb{R}; 0 \leq i < n, i \in \mathbb{N}$$

Each *utilisation* tuple comprises the *Interval* as well as the *amount* of storage positions of individual utilisations. The mapping *allocate* can be applied to add an additional $utilisation^+$:

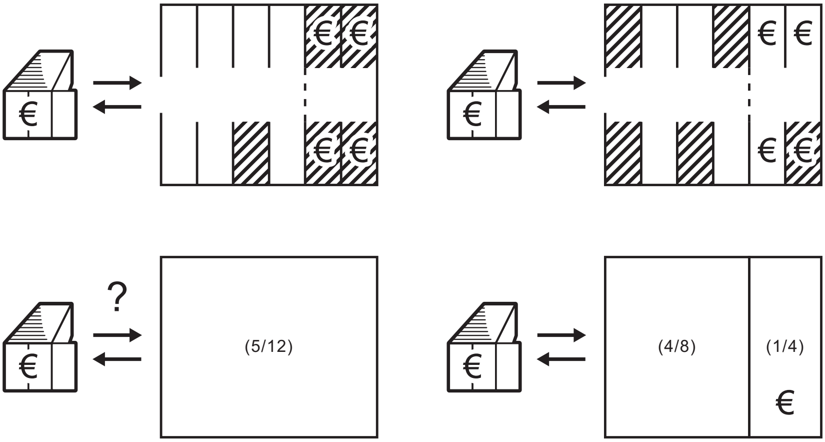

Fig. 5.4 Top row: A warehouse that is capable of storing both normal and valuable goods. In one example, all secured storage positions are already taken (left), in the other example, some secured positions are available (right). Bottom row: The examples cannot be distinguished in the abstraction (left) unless the warehouse is split logically (right).

$$\begin{aligned} \mathit{allocate} : \mathcal{P}(\mathcal{I} \times \mathbb{R}) \times (\mathcal{I} \times \mathbb{R}) &\to \mathcal{P}(\mathcal{I} \times \mathbb{R}) && (5.28)\\ \mathit{allocate} : (\mathit{Utilisations}, \mathit{utilisation}^{+}) &\mapsto \mathit{Utilisations}' && (5.29) \end{aligned}$$

More concretely, the mapping can be defined as follows:

$$\mathit{allocate}(\mathit{Utilisations}, \mathit{utilisation}^{+}) := \mathit{Utilisations} \cup \{\mathit{utilisation}^{+}\} \tag{5.30}$$

Capacity can be characterised by negative utilisation. When adding further utilisations of storage capacity, it must be guaranteed that the maximum capacity is not exceeded at any point in time. More formally, the check (Equations 5.12 and 5.20) whether there is enough capacity for an additional $\mathit{utilisation}^{+} = (\mathit{Interval}, \mathit{amount})$ can be defined as follows:

$$\forall_{\mathit{date} \in \mathit{Interval}} \; 0 \leq \sum_{\substack{u \in \\ \mathit{Utilisations}'}} \begin{cases} \mathit{amount}(u) & \text{if } \mathit{date} \in \mathit{interval}(u) \\ 0 & \text{otherwise} \end{cases} \tag{5.31}$$

In Equation 5.31, $\mathit{amount}(u)$ denotes the amount_u part of the tuple $u = (\mathit{Interval}_u, \mathit{amount}_u) \in \mathit{Utilisations}$ (Definition 5.15), i. e.:

$$\begin{aligned} \mathit{amount} : (\mathcal{I} \times \mathbb{R}) &\to \mathbb{R} && (5.32)\\ \mathit{amount} : (\mathit{Interval}_u, \mathit{amount}_u) &\mapsto \mathit{amount}_u && (5.33) \end{aligned}$$

Likewise, the mapping $\mathit{interval}(u)$ denotes the respective $\mathit{Interval}_u$:

$$\mathit{interval} : (\mathcal{I} \times \mathbb{R}) \rightarrow \mathcal{I} \tag{5.34}$$

$$\mathit{interval} : (\mathit{Interval}_u, \mathit{amount}_u) \mapsto \mathit{Interval}_u \tag{5.35}$$

Depending on the granularity of time modelling, a considerably high number of points in time must be checked, namely $|\mathit{Interval}|$. A similar problem in computer graphics is addressed by the scan-line algorithm (Foley & Van Dam, 1984, pp. 456–458). In order to draw a polygon on a computer display, it must be determined which pixels belong to the polygon. This is done line by line by the so-called scan line. Foley and Van Dam (1984, p. 457) explain that it is inefficient to check every single point on the line because most likely sequences of adjacent pixels belong to the polygon. Instead, it is sufficient to examine intersections of the outline and the scan line.

Algorithms 5.1 and 5.2 adapt this principle in order to determine whether a service provider has sufficient capacity for a new $\mathit{utilisation}^+$. For all existing utilisations, Algorithm 5.1 determines whether their *begin* and *end* times are within the *begin* and *end* times of the new $\mathit{utilisation}^+$. The respective dates are collected in the *Dates* set. These dates are relevant in the sense that the capacity used changes. In other words, between two successive dates it remains unchanged. Hence, it is sufficient to check these distinguished points in time instead of all of them in the respective interval. For each of these dates, the amounts of utilisations are summed up (Algorithm 5.1). Subsequently, Algorithm 5.2 determines whether this *sum* together with the *amount* of the new $\mathit{utilisation}^+$ exceeds the maximum capacity. The storage provider does not have enough capacity left if the maximum capacity is exceeded at any point in time.

In Algorithms 5.1 and 5.2, let *utilisation* and $\mathit{utilisation}^+$ be *utilisation* tuples (Definition 5.15). Then, $\mathit{begin}(\mathit{utilisation})$ and $\mathit{end}(\mathit{utilisation})$ denote the begin t_b and end t_e times

$$\mathit{begin} : ((\mathbb{N} \times \mathbb{N}) \times \mathbb{R}) \rightarrow \mathbb{N} \quad \text{and} \quad \mathit{end} : ((\mathbb{N} \times \mathbb{N}) \times \mathbb{R}) \rightarrow \mathbb{N} \tag{5.36}$$

$$\mathit{begin} : ((t_b, t_e), a) \mapsto t_b \quad \text{and} \quad \mathit{end} : ((t_b, t_e), a) \mapsto t_e \tag{5.37}$$

of the $\mathit{interval}(\mathit{utilisation})$ (Equation 5.34), respectively.

Teams of storage service providers reduce the number of communication partners and thus the effort for service consumers. However, storage is only one aspect in logistics. Likewise, team formation can be applied for service providers of the other primary logistics functions. Atomic transport service providers can be aggregated if they serve the same relations for similar cargo and load carriers and same conditions. Handling and picking service providers respectively can form teams if they have the same location and offer the same transformation for the same price.

To summarise, atomic logistics service providers can be aggregated in order to jointly offer their services. The potential for cooperation is derived as follows. Firstly, teams can actually jointly solve their task. Secondly, joint action is preferable over individual action because team managers act as

Algorithm 5.1 Capacity
$(\mathit{Utilisations} \subset \mathcal{P}(\mathcal{I} \times \mathbb{R}), \mathit{interval} \in \mathcal{I}) \rightarrow \mathcal{P}(\mathbb{N} \times \mathbb{R})$

```
Dates ⇐ {begin(interval)}
for all utilisation ∈ Utilisations do
   if begin(utilisation) ≥ begin(interval)
   and begin(utilisation) < end(interval) then
      Dates ⇐ Dates ∪ {begin(utilisation)}
   end if
   if end(utilisation) > begin(interval)
   and end(utilisation) ≤ end(interval) then
      Dates ⇐ Dates ∪ {end(utilisation)}
   end if
end for
Capacity ⇐ ∅
for all date ∈ Dates do
   sum ⇐ 0
   for all utilisation ∈ Utilisations do
      if begin(utilisation) ≤ date and date < end(utilisation) then
         sum ⇐ sum + amount(utilisation)
      end if
   end for
   Capacity ⇐ Capacity ∪ {(date, sum)}
end for
return Capacity
```

Algorithm 5.2 Check Capacity
$(\mathit{Utilisations} \subset \mathcal{P}(\mathcal{I} \times \mathbb{R}), \mathit{utilisation}^{+} \in \mathcal{I} \times \mathbb{R}) \rightarrow \mathbb{B}$

```
Capacity ⇐ capacity(Utilisations, interval(utilisation⁺))
for all entry ∈ Capacity do
   if amount(entry) + amount(utilisation⁺) < 0 then
      return false
   end if
end for
return true
```

proxies for their team members and thus reduce the interaction effort within the autonomous logistics network. A precondition is that team members offer homogeneous services, whereby the understanding of homogeneity differs for the primary logistics functions.

5.2.2 Teams of Logistics Service Consumers

As elaborated in the preceding section, aggregating logistics service providers in teams helps reduce the interaction effort in autonomous logistics. Likewise, organisational structures can be established by logistics service consumers. On the one hand, organisational structures of service consumers also reduce the interaction effort. On the other hand, cooperation helps optimise processes by using resources more efficiently.

To recapitulate, the probably most fundamental question regarding cooperation in autonomous logistics is which entities should actually cooperate (Section 5.2.1). One approach for organisation can be derived from physical groupings in existing logistics processes. In these processes, it is quite uncommon that a sales unit individually moves through a logistics network. Usually, individual sales units are sold to end customers. In all preceding steps of the supply network, they are grouped together on joint load carriers, e. g, cardboard boxes. Cardboard boxes allow handling multiple sales units together in an efficient way. For storage and transport, even larger units are created, namely pallets and shipping containers. These compositions of sales units usually last for a certain time. Sales units that jointly participate in such compositions generally have the same, or at least partially overlapping, logistics objectives, e. g., being transported from a common source to a common sink at the same time. This observation can be transferred to autonomous logistics. Multiple general cargo units can form teams based on joint logistics objectives. A match is determined by respective mappings:

$$match_o : \mathcal{O} \times \mathcal{O} \rightarrow \mathbb{B} \tag{5.38}$$

$$match_o : (o_1, o_2) \mapsto b \tag{5.39}$$

Such teams significantly reduce the interaction effort because the team manager can jointly request logistics services for the whole team. To save resources, agents can actually be merged into one (Definition 5.14). A reconfiguration is possible by employing handling and picking service providers. To reiterate a previous example (Section 5.1.1), during receiving at a storage facility general cargo units are extracted from a shipping container in order to be re-arranged on pallets. That is, new teams are created in the course of these logistics functions. In general, the autonomous logistics entities must themselves be able to establish teams because there is no prior knowledge about potential groupings

Besides, a more loosely coupling suffices in other applications. As an example, consider an application related to storage. Several general cargo units of the same type plan and schedule their way independently through the logistics network. Despite of their independence, it is desirable to receive them in the same storage facility. With regard to the subsequent distribution processes, this helps decreasing the number of truckloads by preventing empty vehicle running (Section 2.2.2). Thus, these load carriers must form a team in order to jointly allocate storage capacity. Load carriers can determine a match between their cargo by respective mappings:

$$match_d : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{B} \tag{5.40}$$

$$match_d : (d_{c_1}, d_{c_2}) \mapsto b \tag{5.41}$$

Due to their loose coupling, autonomous logistics entities are not restricted to the membership of only one team. Therefore, the participating agents remain their autonomy (Definition 5.14). Another team may be involved in transport operations. Load carriers that share both the same location and the same destination can jointly employ means of mass transport. For instance, joint transport by barge or train is cheaper than individual transport by truck. Finally, load carriers must also cooperate if they request time windows for receiving at a storage facility. This is necessary in order to coordinate handling requests by their priority. For this purpose, load carriers that wait at the same warehouse form a team.

To summarise, the potential for cooperation of service consumers is derived as follows. Firstly, teams of service consumers can jointly achieve their goal. Secondly, joint action is advantageous for autonomous logistics entities due to the decreased interaction effort and the increased resource utilisation efficiency. Hence, there is a demand for mechanisms facilitating flexible team formation.

5.3 Conclusion

The concept of autonomous logistics intends previously inanimate general cargo units to take over decentralised control of logistics processes. To this end, atomic units are identified and chosen to act as autonomous logistics entities. The term atomic refers, in this context, to the fact that the composition of the entity does not change within the part supply network modelled. Example units are components, articles, sales units, cardboard boxes, pallets, or shipping containers. Autonomous logistics does not assume any predefined organisational structures between the participating entities. Together with the fine granularity of modelling, this might lead to an increased interaction effort. This effort may even outweigh the decrease in computational complexity intended by decentralisation.

Organisational structures may help reduce the interaction effort and use logistics resources more efficiently. Furthermore, it has been shown that teams of service providers and consumers, respectively, can actually jointly achieve their goals. Hence, there is indeed a potential for cooperation in terms of the model for cooperation by Wooldridge and Jennings (Section 5.2). Autonomous logistics entities within such teams can be tightly or loosely coupled. In the case of loose couplings, agents may even belong to multiple teams. Usually, potential groupings are not known in advance. Therefore, the participating autonomous logistics entities must themselves be able to establish organisational structures. Hence, appropriate mechanisms for team formation are required that establish such structures flexibly during runtime.

References

Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, *1*(1), 269–271.

Fischer, K., Schillo, M. & Siekmann, J. H. (2003). Holonic Multiagent Systems: A Foundation for the Organisation of Multiagent Systems. In V. Mařík, D. C. McFarlane & P. Valckenaers (Eds.), *1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS 2003)* (pp. 71–80). Prague, Czech Republic: Springer-Verlag.

Foley, J. D. & Van Dam, A. (1984). *Fundamentals of Interactive Computer Graphics.* Reading, MA, USA: Addison-Wesley.

Hart, P. E., Nilsson, N. J. & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions of Systems Science and Cybernetics*, *4*(2), 100–107.

Li, L. & Horrocks, I. (2004). A Software Framework for Matchmaking Based on Semantic Web Technology. *International Journal of Electronic Commerce*, *8*(4), 39–60.

Randell, D. A., Cui, Z. & Cohn, A. G. (1992). A Spatial Logic Based on Regions and Connection. In B. Nebel, C. Rich & W. R. Swartout (Eds.), *3rd International Conference on Principles of Knowledge Representation and Reasoning (KR 1992)* (pp. 165–176). Cambridge, MA, USA: Morgan Kaufmann Publishers.

Tessaris, S. (2001). *Questions and Answers: Reasoning and Querying in Description Logic.* Doctoral dissertation, University of Manchester.

Werner, S. (2006). *Ontologie-basiertes Agentenmatching.* Diplomarbeit, Universität Bremen.

Wooldridge, M. & Jennings, N. R. (1999). The Cooperative Problem Solving Process. *Journal of Logic & Computation*, *9*(4), 563–592.

Chapter 6
Team Formation in Autonomous Logistics

Autonomous control delegates decision-making to autonomous logistics entities. The static specification of autonomous logistics networks distinguishes service consumers and providers as participants in autonomous logistics processes (Chapter 5). In particular, atomic units such as sales units and individual storage positions are regarded as service consumers and providers, respectively. This leads to a high number of participants which in turn increases the interaction effort for process control. Organisational structures (Section 5.2) established on demand help cope with the challenge of interaction complexity. Cooperation is one of the key principles underlying autonomous logistics. A potential for cooperation can be identified for the efficient control of all primary logistics functions (Section 5.2).

Therefore, this chapter focuses on team formation which is the second step of the model for cooperation (Section 4.3.2) by Wooldridge and Jennings. An important question is how the participating logistics entities can establish organisational structures themselves. Section 6.1 discusses requirements for team formation in autonomous logistics processes and reviews related work. Based on this foundation, Section 6.2 presents three interaction protocols for team formation of autonomous logistics entities. Finally, Section 6.3 analyses these protocols and compares their applicability in autonomous logistics.

6.1 Requirements and Related Work

Team formation is an important foundation for cooperation in autonomous logistics processes. Prior to investigating how autonomous logistics entities can efficiently and effectively coordinate team formation, it is necessary to identify the preconditions. As a foundation, Section 6.1.1 investigates general roles and tasks in team formation. Subsequently, Section 6.1.2 identifies particular requirements for team formation in autonomous logistics. Finally,

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics*,
DOI 10.1007/978-3-642-20092-2_6,

previous approaches to team formation are examined with respect to their applicability in Section 6.1.3.

6.1.1 Team Formation Roles and Tasks

The following roles can be distinguished for autonomous logistics entities participating in team formation (Definition 5.14):

1. Team member
2. Team manager

These general roles can be taken by logistics service consumers $\mathcal{SC}$ and providers $\mathcal{SP}$ (Definitions 5.2 and 5.3). Auxiliary agents $\mathcal{A}$ (Definition 5.4) such as brokers or directories may facilitate the team formation process.

Tasks to be handled by team participants and managers comprise:

1. Establishing a team
2. Joining a team
3. Leaving a team

In existing approaches such as the model for cooperation (Section 4.3.2), an initiator identifies a potential for cooperation. This initiator is then responsible for announcing its intention for team formation. In particular, it has to identify agents of which it believes that they can jointly solve the task addressed. Depending on the intended task, the team then develops a joint plan. This process is coordinated by the team manager, which may, for instance, be the initiator.

Sometimes, it is necessary that other agents can join an established team later. A possible reason is that the original members do not have the expected capability to achieve their goals. Besides, the team might be able to solve its task more efficiently with an additional member. In general, it is necessary to update the team plan if another member joins the team. Therefore, not only the new participant but also the manager is involved. The same holds if an existing member leaves the team. Members may decide to intentionally leave a team if their task is completed or if membership does not provide any benefit to them. However, it is also possible that they are accidentally lost. As an example in logistics, consider that a shipping container is lost on high sea. Obviously, this also affects the software agent that runs on an embedded system within the container.

Team changes are particularly challenging if the management agent leaves the team. If this agent leaves the team intentionally, it is necessary to transfer its knowledge to a new manager. If the manager is lost, it is necessary to regather its knowledge. Changing team management, however, is costly because all knowledge must be transferred. It is thus advantageous to avoid changes in management if possible. Various alternatives for the management of agent teams exist (Section 4.3.1). If all participating agents are merged into a new one, the choice of the manager is trivial because only one agent remains. Otherwise, either one of the existing agents can take over the management role or a new agent is created explicitly for this task. The challenge of changes in team management can be addressed by delegating team management to a distinguished agent. The sole responsibility of this agent is managing the respective team. Hence, it does not have a need to leave the team. This strategy successfully prevents changes in team management.

The focus of this chapter is on team establishment and joining members. Leaving members are dealt with in Chapter 7 on collaborative control of the primary logistics functions.

6.1.2 Requirements in Autonomous Logistics

Organisational structures in autonomous logistics decrease the interaction effort and optimise the utilisation of resources (Section 5.2). Team formation is applied in order to establish such structures on demand. A frequent motivation for team formation in other areas of application is that heterogeneous agents supplement each other in their capabilities. If an agent is not capable of solving (at least parts of) a plan, it tries to get support from others. A team is complete if all tasks are covered by its members.

The motivation in this research project is different. The applications in autonomous logistics discussed in Section 5.2 show that it is often useful if homogeneous logistics entities form teams, i. e., entities with similar properties or objectives. Usually, it is not necessary to restrict the number of participants. Consider, for instance, warehouse slots with similar properties (Section 5.2.1). In this example, a team represents all participating warehouse slots in the logistics network. Restricting the number of participants is neither useful nor desirable. By contrast, the advantage of grouping warehouse slots is that all slots with the same properties can be contacted through one proxy agent, i. e., their team manager. Another example is joint utilisation of logistics resources such as means of mass transport (Section 5.2.2). The capacity of an individual train or barge is limited. Nevertheless, it is not necessary to restrict the number of team members interested in transport because multiple trains or barges can be employed. The method for team formation must thus be capable of dealing with potentially many participants, i. e., efficiency is required.

The requirements for team formation in this context are as follows:

1. Unique teams
2. Flexible teams
3. Genericness

4. No prior knowledge
5. Decentralisation
6. Efficiency

The requirement for unique teams means that there should not be two teams $team_i, team_j \in \mathcal{T}$ with equal properties d_{team_i}, d_{team_j}:

$$\forall_{team_i, team_j \in \mathcal{T}} \; i \neq j \rightarrow d_{team_i} \neq d_{team_j} \tag{6.1}$$

As elaborated in the preceding paragraph, it should be clear which team, or team manager to be more precise, is responsible for which tasks. This requires dealing with concurrency because autonomous logistics entities act in parallel. That is, it must be detected and resolved if teams with equal properties are established in parallel.

Multiagent systems are dynamic in the sense that agents may join and leave during runtime (Ferber, 2001, pp. 338–339). This particularly holds for autonomous logistics networks (Definition 5.5) and their participants. Logistics entities join the system if they demand logistics transformations. They leave the system again if the transformations are completed. Because not all logistics entities exist right from the start, teams must be flexible and still accept new members after successful team establishment.

As an important property, a method for team formation in autonomous logistics should be generic. That is, it should not be restricted to particular descriptors. The discussion in Section 5.2 already reveals a broad bandwidth of team characterisations covering, for instance, logistics objectives (Definition 5.7) or descriptors of cargo and load carrier (Definition 5.8). The participating logistics entities are themselves aware of their properties and how to determine their similarity with other entities. However, they should not be forced to have prior knowledge about other participants of the autonomous logistics network or multiagent system in general. The system evolves dynamically. Providing every participant with information about all other entities is expensive. This is closely related to the set up cost addressed, for instance, by Ogston and Vassiliadis (Section 4.3.2).

The autonomous logistics paradigm decentralises process control and delegates decision-making to local logistics entities (Section 3.1). Consequently, team formation in autonomous logistics should also be conducted in a decentralised way. Otherwise, one runs into the danger of implementing bottlenecks into logistics networks. Even with decentralisation, however, team formation should be performed efficiently. This means that a modicum of effort is made for communication.

6.1.3 Previous Approaches

After the discussion of the requirements for team formation, previous approaches are examined regarding their applicability. At first glance, clustering algorithms such as k-means (MacQueen, 1967) might appear to be promising candidates. Their task is to partition a given set of objects into distinct groups. The objective is to achieve a maximal distance between different clusters and a minimal distance between the members of each cluster. As a drawback, however, such methods have a centralised perspective on the data to be clustered. Hence, they are not suited for the distributed setting of the problem at hand.

Distributed clustering approaches can be found in the area of wireless sensor networks (Section 3.2.3). Like the logistics domain, the application of sensor networks is highly distributed. To recapitulate, sensors are spread within a certain area in order to observe specific phenomena. Their primary tasks are acquiring data from the environment as well as transmitting the collected data to a base station (Figure 3.10). A common approach to save energy is to cluster sensors by spatial proximity. A prominent example, namely LEACH (Heinzelman, Chandrakasan & Balakrishnan, 2000), is described in Section 3.2.3. Apart from distribution also the requirement for flexibility is satisfied. Occasional re-clustering allows sensors to join clusters later. Unfortunately, the approach is not generically applicable. It requires at least implicit knowledge of the system. Sensor nodes send broadcast messages and receive responses from other nodes in their vicinity. This knowledge on spatial proximity does not apply to the problem addressed in autonomous logistics. In particular, one wants to incorporate also qualitative knowledge and not only quantitative (spatial) data. The difference is that the partitioning already exists implicitly in the agent properties. The task is thus rather finding potential team members without prior knowledge about the other agents.

Peer-to-peer approaches, such as the one by Ogston and Vassiliadis (2001), accomplish this task by providing each agent with an arbitrarily chosen set of other agents. Agents inform their peers about each other. Based on this foundation, they iteratively exchange their direct partners by others that are more similar. However, this setting is purely artificial for the addressed application in autonomous logistics. In particular, there is no meaningful choice for initial peers because the autonomous logistics entities are initially completely unaware of each other.

As discussed in Section 4.3.2, it is a common approach to apply the contract net interaction protocol for team formation. If an agent intends to form a team, it could use the contract net to announce the team description to all interested agents. A disadvantage, however, is that it is not flexible. Interested participants cannot join a team once it has been established. It is thus necessary to introduce new protocols because the existing ones are not appropriate for the application intended (Section 4.2.4).

6.2 Team Formation Interaction Protocols

Based on the requirement analysis (Section 6.1), Sections 6.2.1 to 6.2.3 present interaction protocols for team formation that are based on directory, broker, and multicast messages, respectively. Each of the protocols has advantages and drawbacks. Therefore, a thorough investigation is conducted in Section 6.3 in order to identify their adequacy for particular applications.

6.2.1 Team Formation by Directory

Different agent interaction protocols (Section 4.2.4) for team formation have been developed during the course of this research project. The first of them (Schuldt & Werner, 2007b, pp. 128–130) employs a directory service. Its flow is defined as follows (Figure 6.1). It is initiated by a participant:

Definition 6.1 (Participant) *Let* Participants $\subseteq \mathcal{SP} \cup \mathcal{SC}$ *be the set of all team formation participants. A* participant $\in$ Participants *is an agent role that is defined by a descriptor* $\mathrm{d} \in \mathcal{D} \cup \mathcal{O}$ *and a* match_t *mapping:*

$$participant := (d, match_t)$$

The $match_t$ mapping determines the similarity of two team descriptors:

$$match_t : \mathcal{D} \cup \mathcal{O} \times \mathcal{D} \cup \mathcal{O} \rightarrow \mathbb{B} \tag{6.2}$$

$$match_t : (d_1, d_2) \mapsto b \tag{6.3}$$

The participant deliberatively decides that it wants to join a team. This autonomous logistics entity is aware of its own properties that are related to team formation (e. g., Section 5.2). Furthermore, it has a mapping that allows deciding whether its properties match the properties of a team. The autonomous logistics entity does not have any additional knowledge about the multiagent system (Definition 5.5) it inhabits. Particularly, it does not know whether there are already existing teams or potential team members.

Hence, the participant, a cardboard box in the example in Figure 6.2, initially requests a list of existing team managers from a directory service:

Definition 6.2 (Directory) *Let* $\mathcal{DE}$ *be the set of all directory entries. A* directory $\in \mathcal{A}$ *is an auxiliary agent that administers a list* Entries $\subseteq \mathcal{DE}$ *of entries with team managers* manager $\in$ Participants*:*

$$Entries := \{entry_1, \ldots, entry_n\}$$

with

$$entry_i := (manager); 1 \leq i \leq n, i \in \mathbb{N}$$

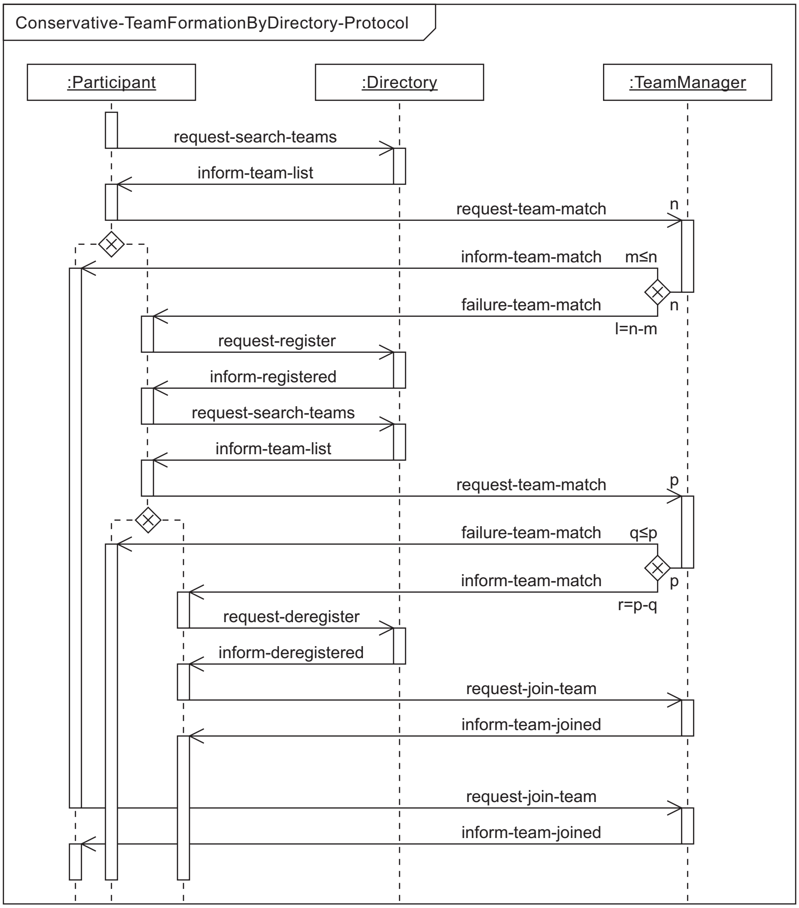

Fig. 6.1 A conservative approach to team formation by directory. A directory service exists that administers a list of existing groups. The protocol comprises two iterations because concurrency might lead to multiple groups with same properties otherwise (exceptional messages are omitted for the sake of readability).

The directory service is responsible for administering this list, i. e., keeping track of established and dissolved teams as well as providing the list upon request. The directory service provides the whole list of *Managers* without filtering matches between the participant and existing teams:

$$Participants \supseteq Managers = \bigcup_{\substack{e \in \\ Entries}} \{manager(e)\} \tag{6.4}$$

Let $e = (manager_e)$ be an entry singleton as defined in Definition 6.2. Then, $manager(e)$ denotes the $manager_e$:

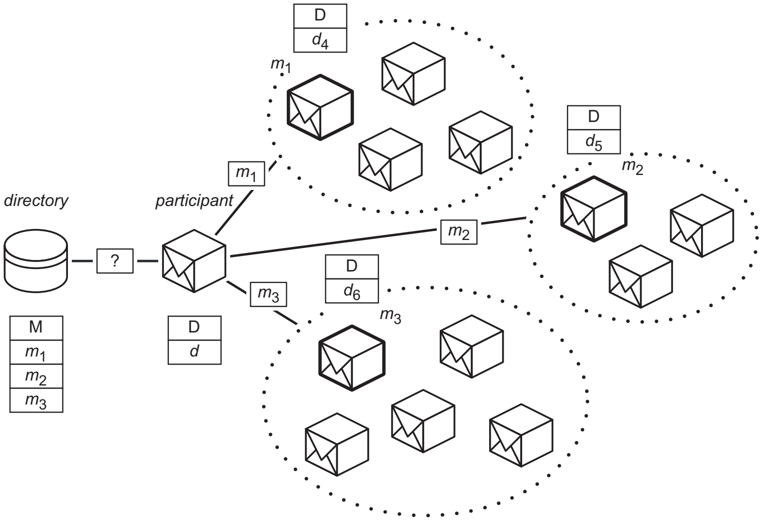

Fig. 6.2 In the concept of team formation by directory, the directory service administers a list of existing team managers m_i. This list can be requested by new participants, a cardboard box in this example, to provide the managers with their properties. The team managers determine whether there is a match with their properties.

$$manager : \mathcal{DE} \longrightarrow Participants \tag{6.5}$$

$$manager : (manager_e) \mapsto manager_e \tag{6.6}$$

Having received the manager list, the participant contacts all team managers and provides them with its descriptor. Each team can then decide for itself whether or not the participant matches the team descriptor. To this end, the respective $match_t$ mapping of descriptors is applied (Definition 6.2). All teams inform the participant about the outcome of this procedure.

If a team informs the participant about a positive match, the autonomous logistics entity may join the respective team. This is done by sending a join request. If none of the existing teams constitutes a positive match, the initiating agent may register itself with the directory service as a new team manager. A directory (Definition 6.2) receiving a register request, updates its *Entries* with the following mapping:

$$register : \mathcal{P}(\mathcal{DE}) \times Participants \rightarrow \mathcal{P}(\mathcal{DE}) \tag{6.7}$$

$$register : (Entries, manager) \mapsto Entries' \tag{6.8}$$

The mapping can be implemented as follows:

$$register(Entries, manager) := Entries \cup \{(manager)\} \tag{6.9}$$

At first glance, this procedure should suffice for establishing teams with distinct properties. However, agents act in parallel. Therefore, it may happen that two similar autonomous logistics entities register themselves concurrently as team managers with the directory service. This is possible because querying the directory and registering oneself is not an atomic operation (Schuldt & Werner, 2007a, p. 194). That is, after looking up the directory and before registering, another autonomous logistics entity may register itself as a team manager with the same properties. The only entity that has knowledge about all existing teams is the directory service. Unfortunately, it cannot resolve such conflicts because it only administers a list of existing teams but not their properties. Due to the decentralisation of multiagent systems, it is impossible to implement a semaphore (Dijkstra, 1965, pp. 31–34) for sequencing.

An iterated step of the protocol is required in order to address this issue. The participant again queries the directory for existing teams. Afterwards, the managers of all new teams

$$\mathit{Managers}' \setminus \mathit{Managers} \subseteq \mathit{Participants} \tag{6.10}$$

must again be contacted and provided with the properties of the initiator. If still no matching team is found, the protocol is terminated. This means that the participant has successfully established a new team that does not interfere with other existing teams. Note that, initially, the team manager is the only member of its team (Equation 5.27).

Additional actions must be taken if the second iteration reveals that one or more other teams match the properties of the initiator. In that case, all but one of the redundant team managers must deregister. The directory handles requests for deregistration with the following mapping:

$$\mathit{deregister} : \mathcal{P}(\mathcal{DE}) \times \mathit{Participants} \rightarrow \mathcal{P}(\mathcal{DE}) \tag{6.11}$$

$$\mathit{deregister} : (\mathit{Entries}, \mathit{manager}) \quad \mapsto \mathit{Entries}' \tag{6.12}$$

The deregistration of managers can be implemented as follows:

$$\mathit{deregister}(\mathit{Entries}, \mathit{manager}) := \mathit{Entries} \setminus \{(\mathit{manager})\} \tag{6.13}$$

To reiterate, there is no central entity for coping with concurrency. Hence, the affected agents must themselves be capable of coordinating deregistration. To this end, each positive response to match requests includes the timestamp at which the respective team has been registered with the directory service. Based on this information the younger teams deregister from the directory and join the oldest one subsequently. This solution covers almost all cases. The only exception is, however, that two timestamps are equal. In real-world operation this is rather unlikely although not impossible. However, it occurs more often in time-stepped or event-driven multiagent-based simulation because there is an artificial synchrony (Gehrke, Schuldt & Werner, 2008, p. 549) due to discrete time progression. In this case, an additional unambiguous criterion has to be applied. As long as there are no other distinguishing properties, the lexicographical order of the unique agent identifiers (Section 4.2.1) is an adequate choice:

$$\forall_{m_1,\ m_2 \in \mathit{Managers}}\ m_1 < m_2 \leftrightarrow \mathit{registered}(m_1) < \mathit{registered}(m_2) \vee \\ \mathit{registered}(m_1) = \mathit{registered}(m_2) \wedge \mathit{id}(m_1) < \mathit{id}(m_2) \quad (6.14)$$

The mappings *registered* and *id* denote the registration timestamp and unique agent identifier, respectively. Note that the same conflict can occur in the first iteration of the protocol. In order to prevent conflicts, the participant joins the oldest team also in this step.

Another potential exception caused by concurrency arises from the fact that requesting a match and joining a team is not an atomic action. That is, a participant has successfully found a matching team. But before the participant requests joining it, the team dissolves for some reason. Then, the team formation process initiated fails. Fortunately, the resolution is easy because the participant can simply restart the team formation interaction protocol. This resolution can also be applied for the alternative protocols discussed in the subsequent sections.

The protocol discussed so far can be categorised as being conservative. Each participant determines whether there is no conflict before it decides to establish a new team itself. Figure 6.3 depicts a corresponding protocol that takes an optimistic perspective. In this approach, participants act optimistically and assume that there is no conflict with existing teams. Based on this assumption, they simply register themselves with the directory service without any precautions. That is, the first iteration of the conservative protocol is omitted. Conflicts are detected and resolved in the formerly second iteration. As a drawback of the optimistic approach, more changes must be performed in the database of the directory service. As an advantage, however, it is possible to reduce the number of message cycles, i. e., pairs of request and response messages. To reiterate, interaction efficiency is an important criterion for interaction protocols (Section 4.2.4).

The protocol length is five message cycles if a matching team already exists. By contrast, only three cycles are necessary if a participant can successfully establish a new team. The number of cycles indicates how long it takes to apply a protocol. However, it does not measure how many messages actually have to be exchanged. It depends on the number of existing teams because every team has to be contacted at least once for the match request. The interaction complexity for one participant is thus $O(m)$ with m being the number of existing teams. For the whole system of n participating autonomous logistics entities, the asymptotic complexity is $O(mn) = O(n^2)$. Although the upper bound of interaction effort is quadratic, it is worth mentioning that $m \ll n$ for many applications, i. e., only a small part of the agents are also team managers.

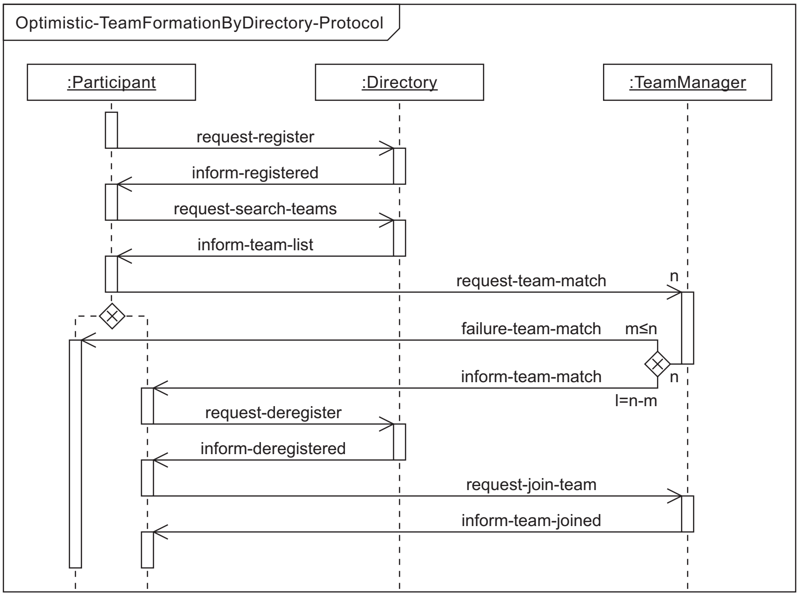

Fig. 6.3 An optimistic alternative of team formation by directory. In this protocol, each participant initially assumes that there is no conflict with existing teams. Hence, it registers itself. If a conflict is detected afterwards, it deregisters and joins the respective existing team.

To summarise, both interaction protocols for team formation by directory cover the steps of establishing and joining team. Whenever no existing team matches the properties of the participant, the participant establishes a new team. Otherwise, it joins the matching team. The directory service administers the list of existing teams. However, it does not have any knowledge about the properties of the respective teams. Thus, it is left to the teams to decide whether or not an agent matches their properties. The interaction complexity is $O(n^2)$ because each of the n participant contacts each of the m existing team at least once in order to announce its properties.

6.2.2 Team Formation by Broker

The interaction effort of the directory-based interaction protocol for team formation increases quadratically (Section 6.2.1). This is particularly challenging with respect to the high number of participating entities in autonomous logistics processes. It is thus desirable to reduce the complexity. The number of messages can be decreased effectively by delegating more responsibility to the directory service. In particular, it should be able to decide directly which

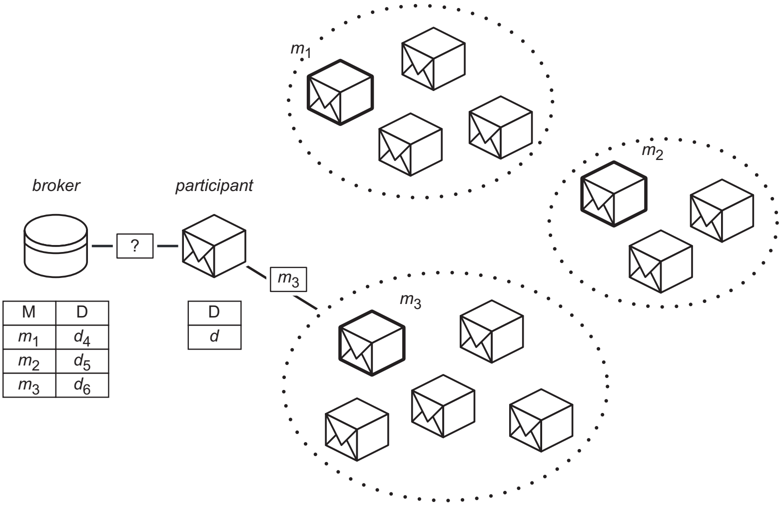

Fig. 6.4 In the broker-based approach, the broker administers not only the list of team managers but also their properties d_i. Provided with the properties of a new participant, it can itself determine a match. Hence, it is sufficient if the participant contacts only the matching team manager.

team properties match the properties of the agent. This turns the former directory service into a broker (Figure 6.4):

Definition 6.3 (Broker) *Let* $\mathcal{BE}$ *be the set of all broker entries. A* broker $\in \mathcal{A}$ *is an auxiliary agent that administers a list* Entries $\subseteq \mathcal{BE}$ *of team managers* manager $\in$ Participants*, their descriptors* d $\in \mathcal{D} \cup \mathcal{O}$*, and* match$_t$ *mappings:*

$$Entries := \{entry_1, \ldots, entry_n\}$$

with

$$entry_i := (manager, d, match_t); 1 \leq i \leq n, i \in \mathbb{N}$$

The flow of the adapted protocol is as follows (Figure 6.5). The protocol is still initiated by a participant (Definition 6.1) that aims at establishing a team. The agent acts optimistically and assumes that no other team exists with its properties. Therefore, it transmits its properties to the respective broker agent in order to register itself as a new team manager (Equation 6.4). The broker compares this descriptor with those of all *Entries*. If there is no match, the agent itself is registered as a new team manager (in analogy to Equation 6.7). If the properties resemble an existing team, the registration fails. The participant is informed about the conflicting team and may instead join it. The advantage of optimism is as follows. A conservative agent would first query the broker and register only if no matching team exists. Querying and registering however is not an atomic operation. Another participant with

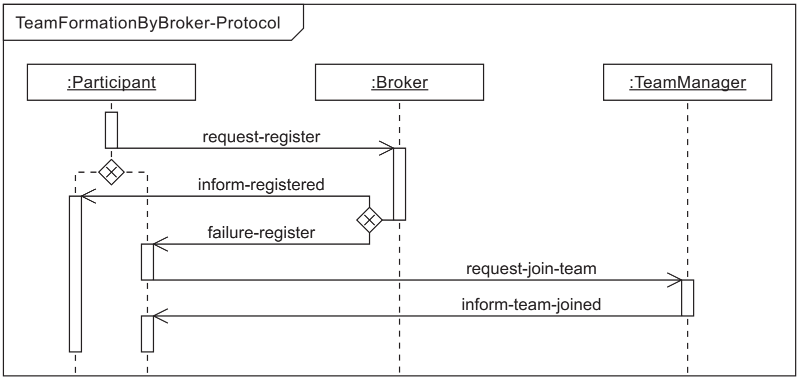

Fig. 6.5 Team formation by broker. A participant tries to register with the broker by providing its properties. This procedure is successful if no matching group is found. Otherwise, the agent may join the matching group that is provided by the broker.

the same properties may register in between. That is, registering may fail in both cases. Then, it is reasonable to act optimistically and decrease the number of message cycles by one.

The protocol length is at most two message cycles per agent. One message cycle suffices if the participant establishes a new team. The additional cycle is necessary if the participant joins an existing team. Every participant only exchanges messages with the broker and at most one team manager. Therefore, the asymptotical interaction complexity per agent is either $O(2)$ or $O(4)$ which is both constant, i. e., in $O(1)$. The interaction complexity for the whole system is thus linear, $O(4n - 2m) = O(n)$, with n being the number of participants and m being the number of team managers. Hence, the interaction effort for individual agents is significantly reduced compared to the directory-based protocol. The disadvantage, however, is a lack of decentralisation due to the centralistic entity that is responsible for all decisions about team matching.

6.2.3 Team Formation by Multicast

Team formation by broker (Section 6.2.2) decreases the interaction complexity of the original directory-based approach (Section 6.2.1). The broker allows reducing the number of messages to be exchanged in the multiagent system significantly. Another drawback of team formation by directory, however, is the directory itself. It is a potential bottleneck because all agents must contact this centralistic entity (Schuldt, 2009). Applying a broker even aggravates this lack of decentralisation because the central broker also makes decisions

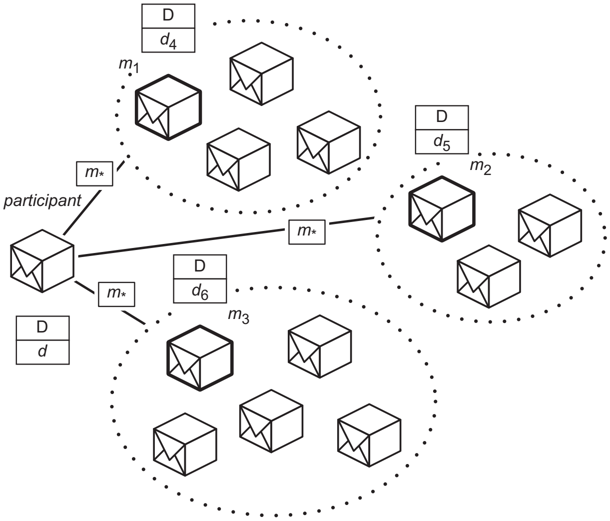

Fig. 6.6 Team formation by multicast completely abolishes centralistic entities. New participants do not have any knowledge about existing team managers. Therefore, they send a multicast message. To this end, the managers subscribe themselves to respective multicast addresses.

on behalf of the teams. In order to increase system robustness, it is desirable to abolish centralistic entities.

To recapitulate, participating agents do not have prior knowledge about existing teams or potential team members in the multiagent system. Furthermore, teams and partners cannot be discovered by broker or directory if these auxiliary agents are abolished. The alternative is to send a broadcast message to all agents within the multiagent system (Ferber, 2001, p. 338). Unfortunately, this increases the interaction effort even in the best case from $O(mn)$ to $O(n^2)$. Remember, that the number m of team managers is much smaller than the total number n of agents, $m \ll n$, for many applications (Section 6.2.1). Even if communication is affordable, one should aim at reducing the number of messages sent. Therefore, broadcast messages are usually not acceptable. As an alternative, it is possible to employ multicasting for team formation (Figure 6.6). Computer network reference models like OSI or TCP/IP implement multicasting on the network layer (Tanenbaum, 2003, p. 370). Participants can subscribe for receiving messages that are sent to particular multicast addresses. These requests are announced by the responsible network routers that deliver such messages. The application layer on which agents are located is disburdened from this task.

The protocol flow of the optimistic interaction protocol for team formation by directory (Section 6.2.1) can be adapted as follows (Figure 6.7). An optimistic participant (Definition 6.1) assumes that no teams exist with its properties. Therefore, it contacts the Message Transfer Service (MTS) in order to receive future multicast messages on team formation. Afterwards, it sends a match request to the respective multicast address. It provides its own properties in order to enable existing team managers (Equation 6.4) to compare its properties with their own properties. If no match is detected, the new participant has successfully established a new team. Otherwise, the participant deregisters from receiving multicast messages on team formation and joins the matching team instead. Like in the directory-based approach, measures are taken to resolve teams that are concurrently established. Note that team formation by multicast relieves participants from the message cycle to request the team list from the directory service.

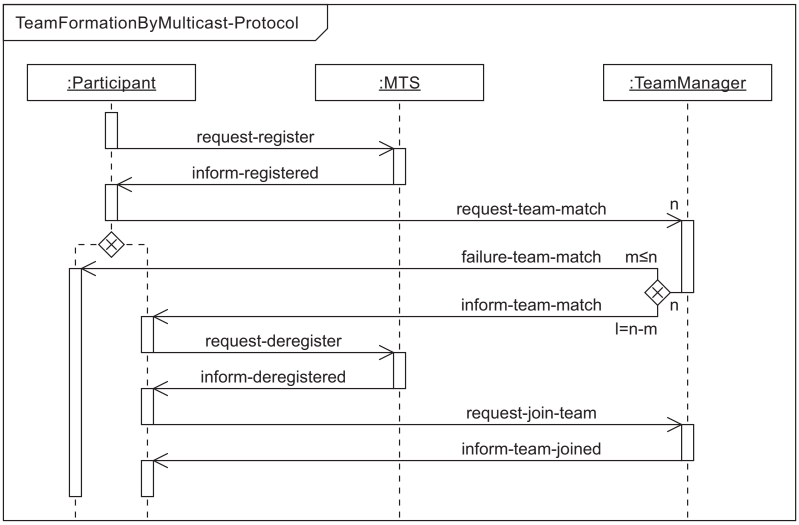

Fig. 6.7 Team formation by multicast. This protocol is completely decentralised because no central entity exists. All match requests are sent through the message transport system (MTS), which distributes them by multicasting. In order to act as a team manager, agents therefore just subscribe to respective multicast messages.

Team formation by multicast allows establishing teams without a centralistic entity. The protocol can, of course, only be applied if the respective network facilities for multicasting are available. This is, for instance, the case when the multicast service is implemented by routers on the network layer. The precondition generally does not hold for sensor networks where the communication network, and thus routing, is established by the sensor nodes themselves (Al-Karaki & Kamal, 2004).

6.3 Protocol Analysis and Comparison

Different interaction protocols for team formation in autonomous logistics processes have been introduced in Sections 6.2.1 to 6.2.3. They correspond with each other regarding some properties. However, they also differ in other aspects. Therefore, is is important to conduct a thorough examination that helps choose the adequate protocol for implementing autonomous logistics processes. Section 6.3.1 starts with investigating the compliance with the initial requirements elaborated in Section 6.1.2. Subsequently, Section 6.3.2 introduces additional criteria that distinguish the protocols. Finally, Section 6.3.3 categorises the protocols with respect to these criteria.

6.3.1 Compliance with Requirements

Initially, it is important to evaluate the protocols with respect to the requirements elaborated in Section 6.1.2. The first criterion demands interaction protocols to establish unique teams. That is, one must prevent that two teams with similar properties exist. Such redundancy might occur if new teams are established concurrently. In the broker-based approach, this does not pose a major challenge. The broker is a centralised entity that is aware of the properties of all teams. Therefore, it can be decided centrally whether two teams resemble each other. The protocols that are based on a directory or multicast messages cannot resolve redundant teams centrally. Instead, newly established teams are themselves responsible to detect conflicts. The respective procedure can also deal with identical groups that are detected late, e. g., if messages are delayed. An ordering criterion (Equation 6.14) ensures that other participants always join the right team. Redundant team managers can therefore simply join the actual team manager without the necessity for a group change of potential members. To summarise, the interaction protocols apply different mechanisms to ensure unique teams. However, they correspond in satisfying this requirement.

The second requirement is the flexibility of teams. That is, it should be possible to join teams after their initial establishment. All interaction protocols for team formation satisfy this requirement. It is implemented by integrating the process of establishing and joining a team. Participants join existing teams or create a new team if no matching team is found. The protocols are not limited in the properties by which teams are formed. Hence, also the requirement for genericness is satisfied. Applying the protocols does not require prior knowledge about other agents or the concrete structure of the multiagent system. Agents that join the system contact other agents through the directory service or the broker agent. In the protocol based on multicast messages, addressing other agents is even delegated to the underlying communication infrastructure.

The last two requirements from Section 6.1.2, decentralisation and efficiency, are satisfied to different extents by the interaction protocols. Therefore, it is important to examine them in more detail. Beforehand, other criteria that distinguish those protocols are collected.

6.3.2 Criteria for Estimation of Applicability

Different criteria for categorising agent interaction protocols have already been discussed in Section 4.2.4. Further criteria that particularly address team formation have been discussed in Section 4.3.2. One criterion by Rosenschein and Zlotkin is stability, i. e., agents should have no incentive to deviate from agreements. The interaction protocols developed in this research do not differ regarding their outcome. Therefore, stability is no distinguishing criterion. Nevertheless, it is worth mentioning that team members should generally be content with the outcome of the protocols which is a matching team. Likewise, also the simplicity criterion is satisfied by all protocols. It can, however, be argued that the optimistic protocol for directory-based team formation excels the conservative one in this aspect (Section 6.2.1). Also the symmetry criterion by Rosenschein and Zlotkin does not distinguish for the protocols introduced. They have in common that all participants are treated equally. The only exception is the first participant who automatically becomes the manager of the team. This, however, is only relevant during team establishment. The team members might decide later to elect another manager. To summarise, the protocols satisfy the criteria by Rosenschein and Zlotkin. However, most criteria are not distinguishing. In accordance with the discussion in Section 6.3.1, exceptions are efficiency and distribution which are to be discussed in more detail.

As elaborated in Section 4.3.2, Ogston and Vassiliadis supplement the criteria set up cost, common language, and privacy. Set up cost corresponds to the requirement for no prior knowledge on the system (Section 6.1.2). It is therefore already covered by the introductory discussion. However, the necessity of a common language and the privacy aspect are relevant in order to distinguish the new protocols and to estimate their adequacy for particular applications. In total, this leads to the following attributes that are considered here in order to compare the protocols with respect to their their particular advantages and drawbacks:

1. Decentralisation
2. Autonomy of the participant
3. Autonomy of the team
4. Interaction effort
5. Common language
6. Privacy

The degree of decentralisation refers to the question how decision-making is distributed within the system. It indicates whether multiple agents are applied instead of a centralistic entity which may turn out to be a bottleneck. A related aspect is the autonomy a protocol grants to the agents. Firstly, this refers to the autonomy of participating agents that may decide whether or not to join a team. Secondly, this refers to the autonomy of existing teams that may decide whether or not to affiliate an agent. The computational complexity for each single agent depends on the concrete reasoning algorithms applied for matchmaking. To judge the efficiency it is important to examine the interaction complexity of the different protocols, i. e., the number of message cycles necessary to solve the team formation problem addressed. The common language attribute addresses the question whether all agents must use the same language for expressing their properties related to team formation. The number of agents that are informed about the properties of participating agents depends on the structure of the interaction protocol applied. A particular question is also whether uninvolved (and potentially malevolent) agents can become aware of such information.

6.3.3 Protocol Categorisation

The directory-based agent interaction protocols for team formation leave all autonomy to the participants and team managers (Figure 6.8). Each participant can itself decide whether or not to initialise the team formation process. Likewise, all decisions about matchmaking can be made by the team managers that are contacted by interested participants. That is, the teams themselves decide which agents may join them. If an agent receives a positive response from a team, it can decide for itself whether or not to join the respective team. Delegating the autonomy for decision-making to the local participants already indicates a high degree of decentralisation. The only restriction is the directory which is a centralistic unit and thus a potential bottleneck. However, the directory only administers a list of existing team managers and does not make any decisions. Delegating decision-making about matchmaking to local agents, however, increases the demand for communication. As elaborated in Section 6.2.1, the interaction complexity for team formation by directory is $O(n^2)$. Compared to the conservative approach, the optimistic alternative of the protocol requires a slightly decreased interaction effort.

There is only a minor requirement for a common language in which properties of the participants are represented. Of course, team managers and their potential members must communicate in the same language to enable matchmaking. However, teams may be established for different applications (Section 5.2.2). It is possible to have different languages for different purposes, e. g., joint storage or transport. If a team manager receives a message that does not fit its purpose, it can simply discard the respective message. This also permits introducing descriptions for special purpose applications that were unknown at design time. Regarding privacy, it is important to distinguish two perspectives. On the one hand, there is a high privacy for team managers because they do not have to reveal their internal decision-making. Participants, however, must transmit their properties to all team managers. Nevertheless, there is no central entity which has all information about all participants or team managers.

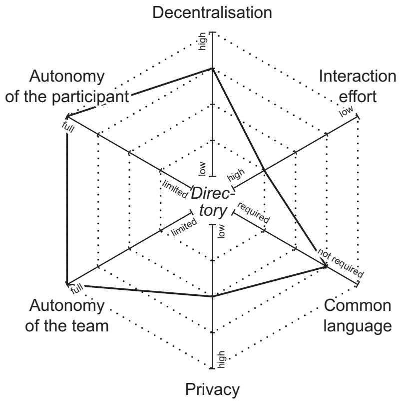

Fig. 6.8 Categorisation of the directory-based team formation interaction protocols. The criteria considered are decentralisation, interaction effort, common language, privacy, and autonomy.

The broker-based protocol can be assessed as follows (Figure 6.9). Participating agents have full autonomy regarding the decision whether to seek potential cooperation partners and whether to join a team if a match is found. By contrast, groups have only restricted autonomy in the matchmaking process. This is due to the fact that matchmaking is delegated to the central broker agent. From this arises the problem that teams never become aware of false negatives, i. e., agents the broker erroneously considers to be non-matching. The degree of decentralisation is lower because the broker is a potential bottleneck. In return for the limited decentralisation, interaction complexity decreases to $O(n)$ with n being the number of participants. For matchmaking, the broker must be supplied with all respective information. However, this is only necessary not sufficient. For the matchmaking process, the broker must understand the properties, i. e., a common language for team descriptions is required. Disclosing their decision processes is in contrast to the privacy of the team managers. From the point of view of the participants, however, it is advantageous that not all potentially uninvolved team managers, but only the broker receives the descriptors of all agents; at least, provided that the broker is trustworthy.

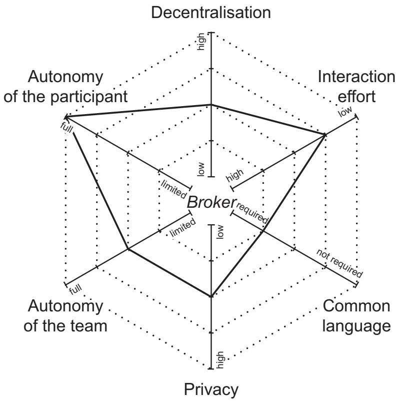

Fig. 6.9 In the broker-based approach to team formation, the degree of decentralisation is decreased because the broker as a central decision-maker constitutes a potential bottleneck. In turn, however, the interaction effort is decreased significantly.

In the team formation protocol based on multicasting (Figure 6.10), the degree of autonomy and the asymptotic complexity class for communication remain unchanged compared to the directory-based protocol. By contrast, the degree of decentralisation significantly increases because the directory as a central entity is avoided. All coordination is performed by the participating logistics entities themselves. Like for the directory-based approach, this reduces the requirement for a common language. The privacy for the team managers is high because they do not have to reveal their internal decision processes about matchmaking. By contrast, the degree of privacy for the participants is even further decreased because even completely uninvolved agents can subscribe to receive multicast messages on team formation. If there is a demand for security, it is necessary to restrict communication to trustworthy participants (Section 3.2.4).

6.4 Conclusion

To summarise, different agent interaction protocols for team formation have been introduced. The intended area of application is autonomous logistics to support autonomous logistics entities in collaborative process control. These protocols allow forming dynamic teams of agents sharing the same goals without any prior knowledge. They focus on interaction and are therefore generic in the choice of descriptions for logistics entities. The examination in this section reveals advantages and drawbacks of the respective protocols. The broker-based protocol decreases the interaction effort, thereby also limiting the degree of decentralisation. This protocol is particularly applicable if com-

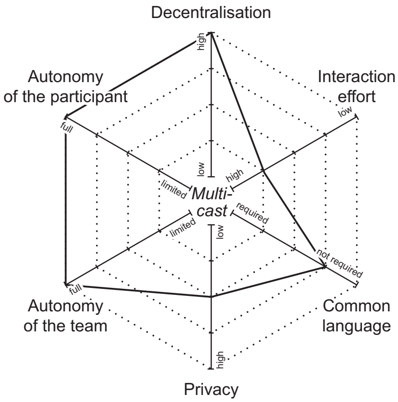

Fig. 6.10 Categorisation of team formation based on multicast messages. Compared to directory-based team formation, the degree of decentralisation is increased because the directory as a central entity is avoided.

munication is expensive. The protocol based on multicasting maximises the degree of decentralisation. In this protocol, all decision-making is left to the participating entities. In exchange, however, the interaction effort is higher, namely quadratic. This protocol can be applied if a high degree of decentralisation is required and enough bandwidth for communication is available. The directory-based protocol is less decentralised than the one based on multicast messages. However, it can be used as a fallback solution if multicasting is not available. It resembles its multicasting-based counterpart in most attributes.

References

Al-Karaki, J. N. & Kamal, A. E. (2004). Routing Techniques in Wireless Sensor Networks: A Survey. *IEEE Wireless Communications*, *11*(6), 6–28.

Dijkstra, E. W. (1965). *Cooperating Sequential Processes* (Tech. Rep. No. EWD-123). Technische Universiteit Eindhoven.

Ferber, J. (2001). *Multiagentensysteme: Eine Einführung in die Verteilte Künstliche Intelligenz*. Munich, Germany: Addison-Wesley Verlag.

Gehrke, J. D., Schuldt, A. & Werner, S. (2008). Quality Criteria for Multiagent-Based Simulations with Conservative Synchronisation. In M. Rabe (Ed.), *13th ASIM Dedicated Conference on Simulation in Production and Logistics (ASIM 2008)* (pp. 545–554). Berlin, Germany: Fraunhofer IRB Verlag.

Heinzelman, W. R., Chandrakasan, A. & Balakrishnan, H. (2000). Energy-Efficient Communication Protocol for Wireless Microsensor Networks.

In *33rd Hawaii International Conference on System Sciences (HICSS 2000)* (Vol. 8, pp. 8020–8029). Maui, HI, USA: IEEE Computer Society.

MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In L. M. Le Cam & J. Neyman (Eds.), *5th Berkeley Symposium on Mathematical Statistics and Probability* (pp. 281–296). Berkeley, CA, USA: University of California Press.

Ogston, E. & Vassiliadis, S. (2001). Matchmaking Among Minimal Agents Without a Facilitator. In J. P. Müller, S. Sen, E. Andre & C. Frasson (Eds.), *5th International Conference on Autonomous Agents (Agents 2001)* (pp. 608–615). Montreal, Canada: ACM Press.

Rosenschein, J. S. & Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers.* Cambridge, MA, USA: MIT Press.

Schuldt, A. (2009). Decentralisation and Interaction Efficiency in Cooperating Autonomous Logistics Processes. In H.-J. Kreowski, B. Scholz-Reiter & K.-D. Thoben (Eds.), *2nd International Conference on Dynamics in Logistics (LDIC 2009).* Bremen, Germany: Springer-Verlag.

Schuldt, A. & Werner, S. (2007a). A Clustering Protocol for Team Formation Based on Concept, Location, and Time. In M. Dastani & R. H. Bordini (Eds.), *5th European Workshop on Multi-Agent Systems (EUMAS 2007)* (pp. 193–196). Hammamet, Tunisia.

Schuldt, A. & Werner, S. (2007b). Distributed Clustering of Autonomous Shipping Containers by Concept, Location, and Time. In P. Petta, J. P. Müller, M. Klusch & M. Georgeff (Eds.), *5th German Conference on Multiagent System Technologies (MATES 2007)* (pp. 121–132). Leipzig, Germany: Springer-Verlag.

Tanenbaum, A. S. (2003). *Computer Networks* (4th ed.). Upper Saddle River, NJ, USA: Pearson Education.

Wooldridge, M. & Jennings, N. R. (1999). The Cooperative Problem Solving Process. *Journal of Logic & Computation*, *9*(4), 563–592.

Chapter 7
Team Action in Autonomous Logistics

The autonomous logistics paradigm envisions that autonomous logistics entities are themselves responsible for achieving their logistics objectives (Section 3.1). Choosing atomic units (Section 5.1) to be autonomous logistics entities leads to a high interaction effort for process control. Hence, there is a potential for cooperation to reduce the interaction effort (Section 5.2). Respective agent interaction protocols facilitate team formation in order to establish organisational structures on demand (Section 6.2). This chapter focuses on the team action step of the model for cooperation (Section 4.3.2). It examines how teams of autonomous logistics entities can actually coordinate their actions for collaborative process control. The third step of the model for cooperation, namely plan formation, is implicitly also addressed here. Predefined action schemes are employed which are instantiated for actual interaction. More complex planning is out of the scope of the interaction-centred focus of this project.

As elaborated in Section 5.2.1, organisational structures of logistics service providers are usually long-term. For instance, several atomic storage positions form a joint storage facility in order to offer their service together. In general, reasons to change the composition of these teams are rare. Therefore, the members of such teams can often be merged into one agent which acts on their behalf. Long-term organisations also exist among logistics service consumers (Section 5.2.2). An example are teams of sales units which move together through the logistics network. However, there are also short-term organisations which are established on demand, e. g., in order to jointly allocate specific resources from service providers. The particular focus of this chapter is on the interaction within such short-term teams. This is an important aspect because the members of these teams remain autonomous in their decision-making.

Team action in autonomous logistics is approached in three steps. Section 7.1 addresses the question how an individual general cargo unit can allocate a logistics service by negotiation with service providers. Section 7.2 investigates how this task can be coordinated by service consumers that act

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics*,
DOI 10.1007/978-3-642-20092-2_7,

collaboratively in teams. Finally, the extension to intra-agent coordination of different primary logistics functions is dealt with in Section 7.3.

7.1 Individual Allocation of Logistics Services

The objective of this chapter is to investigate team action in autonomous logistics, i. e., joint allocation of logistics services. For the sake of comprehensibility, it makes sense to restrict oneself to the individual case first and extend it to collaborative team action afterwards. To this end, Section 7.1.1 starts with specifying how autonomous logistics entities can express their demand for logistics services. Based on this foundation, the allocation of logistics services by individual service consumers is addressed in Section 7.1.2.

7.1.1 Specifying Demand for Logistics Services

As a foundation for negotiating with logistics service providers, general cargo units must specify their demand for logistics services:

Definition 7.1 (Set of Service Demands) *Let the set of all service demands be denoted by $\mathcal{SD}$. It is defined as the union of the sets of demands for transport services $\mathcal{TSD}$, handling services $\mathcal{HSD}$, storage services $\mathcal{SSD}$, and picking services $\mathcal{PSD}$:*

$$\mathcal{SD} := \mathcal{TSD} \cup \mathcal{HSD} \cup \mathcal{SSD} \cup \mathcal{PSD}$$

To start with storage, a specific storage demand is specified as follows:

Definition 7.2 (Storage Demand) *A storage service demand* $\mathrm{ssd} \in \mathcal{SSD}$ *is defined by a set of locations* $\mathrm{L} \subseteq \mathcal{L}$*, the variable discrete temporal interval* $\mathrm{V} \in \mathcal{V}$*, the descriptors* $\mathrm{d_c} \in \mathcal{D}$ *for the cargo and* $\mathrm{d_{lc}} \in \mathcal{D}$ *for the load carrier of the general cargo unit, as well as the* $\mathrm{amount} \in \mathbb{R}$ *demanded:*

$$ssd := (L, V, d_c, d_{lc}, amount)$$

Most elements of this demand specification have already been introduced in Chapter 5. When requesting storage capacity, it is possible to restrict the request to a set $L \subseteq \mathcal{L}$ of pre-selected locations (Definition 5.5). Alternatively, L can be assigned the empty set $\emptyset$ if one does not intend to make such a restriction. Furthermore, general cargo units provide the descriptors $d_c, d_{lc} \in \mathcal{D}$ of the cargo and the load carrier as defined in Definition 5.6. The unit for the *amount* of load carriers is specified by d_{lc}.

To allocate capacity of logistics resources, logistics service consumers must specify the time span during which they demand the service. At first glance, it seems reasonable to specify such a time span simply by a temporal interval (Definition 5.10). Given the begin t_b and end time t_e of such a temporal interval, logistics service providers like storage facilities can check (Algorithms 5.1 and 5.2) whether their capacity during this time span suffices for receiving the respective amount of goods. The enquiring consumer can then be informed whether its request can be processed.

This procedure perfectly addresses the demands of the enquirer if it results in a positive answer. By contrast, it is not satisfying otherwise because it does not give a hint why a request failed. As an example, consider a general cargo unit requesting storage capacity for a time span of two weeks. This request might fail because the provider does not have sufficient capacity for the first two days. For the requester, however, it might also be possible to arrive two days later, e. g., by remaining at the previous location a bit longer. Unfortunately, the consumer has to guess whether adapting its request might lead to a positive response because the reason why the request failed is not communicated in the procedure outlined above. One strategy might be to iteratively adapt the query, e. g., by shortening it always by one day. In general, however, this leads to a communication overhead caused by the stepwise approximation.

A solution to this problem can be derived from auctions. The efficiency (Section 4.2.4) of an English auction is limited because the final price is approximated by a potentially high number of small bids. An alternative is an auction with sealed bids (Vickrey, 1961, p. 20). In this approach, every bidder only places one bid. The winner pays the second highest bid. By this procedure, agents are motivated to reveal their full value so that the outcome of the auction equals the one of the English auction.

The efficiency aspect can be transferred to the allocation of logistics resources. The approximation procedure is only reasonable if there are variable bounds within the begin or the end point of an interval that can be chosen freely. In such cases, it is more promising to communicate these bounds instead of approximating a successful request. Communicating variable bounds enables the storage provider to adapt an incoming request without the necessity for further callbacks. To recapitulate, efficiency is an important requirement for agent interaction protocols (Section 4.2.4). Besides, the consumer receives a positive answer also if the original request is only approximately met. If a request fails, the consumer can be sure that modifying the request within the given parameters will not change the outcome.

The question is whether it might be harmful for a service consumer to reveal the maximally acceptable deviation from its original demands. In total, $3^2 = 9$ configurations for begin and end points of variable intervals can be distinguished qualitatively (Figure 7.1). Usually, a service consumer has a maximal demand and is willing to decrease it within given bounds. This is expressed by one of the four configurations within the dashed box in Figure 7.1. Then, also the service provider is motivated to fulfill the maximal demand to maximise its revenue. Only if there is actually not enough capacity, it offers less capacity. So, the service consumer has no reason to fear being cheated.

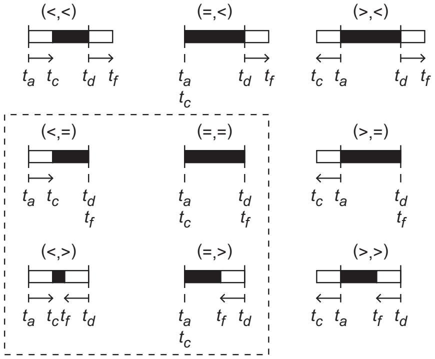

Fig. 7.1 Nine temporal configurations for the allocation of logistics resources can be distinguished qualitatively. The respective start and end points are either fixed by the service consumer or may be shifted by the service provider forward and backward within predefined bounds.

Variable bounds for the begin time

$$t_b \in \{\min(t_a, t_c), \ldots, \max(t_a, t_c)\} \tag{7.1}$$

are spanned by the two points in time t_a and t_c. Note that t_a is not necessarily smaller than t_c. By contrast, three possible qualitative relations (Vilain, Kautz & van Beek, 1990, p. 378) can be distinguished between t_a and t_c: $<$, $=$, and $>$. The special case of $t_a = t_c$ represents a fixed t_b. The other two cases have a different semantics. The value of t_a always marks the preferred time for t_b while t_c marks the least acceptable one. Correspondingly to the begin time t_b, the end time

$$t_e \in \{\min(t_d, t_f), \ldots, \max(t_d, t_f)\} \tag{7.2}$$

is specified by t_d and t_f. This leads to the following definition for variable discrete temporal intervals:

Definition 7.3 (Variable Discrete Temporal Interval) *Let* $\mathcal{V} \subset \mathcal{P}(\mathcal{I})$ *be the set of all variable discrete temporal intervals. A variable discrete temporal interval* $\mathrm{V} \in \mathcal{V}$ *is a set*

$$\begin{aligned} V = \{\{t_b, \ldots, t_e\} \mid t_b &\in \{\min(t_a,t_c), \ldots, \max(t_a,t_c)\} \wedge \\ t_e &\in \{\min(t_d,t_f), \ldots, \max(t_d,t_f)\}\} \end{aligned}$$

of discrete temporal intervals with the begin time $\mathrm{t_b}$ *being defined by* $\mathrm{t_a}$ *and* $\mathrm{t_c}$*, the end time* $\mathrm{t_e}$ *being defined by* $\mathrm{t_d}$ *and* $\mathrm{t_f}$*,* $\mathrm{t_b} \leq \mathrm{t_e}$*, and* $\mathrm{t_a}, \ldots, \mathrm{t_f} \in \mathbb{N}$.

Two inequalities should be taken into consideration when defining variable discrete temporal intervals:

$$\min(t_a, t_c) \leq \min(t_d, t_f) \quad \wedge \quad \max(t_a, t_c) \leq \max(t_d, t_f) \qquad (7.3)$$

Definition 7.3 demands that t_b is smaller than t_e. Hence, it does not make any sense that the lower bound for the end time is smaller than the one of the begin time. The same holds for the upper bound of the begin time which should not be greater than the one of the end time.

Not only for storage demand, but also for requesting transport capacity it is beneficial to provide variable bounds for the time of service consumption. Nevertheless, the temporal specification of transport demand differs from the one for storage demand:

Definition 7.4 (Transport Demand) *A transport service demand* $\mathrm{tsd} \in \mathcal{TSD}$ *is defined by a set of transport relations* $\mathrm{R} \subseteq \mathcal{R}$ *with* $\mathrm{R} \neq \emptyset$*, the discrete temporal interval* $\mathrm{I} \in \mathcal{I}$*, the descriptors* $\mathrm{d_c} \in \mathcal{D}$ *for the cargo and* $\mathrm{d_{lc}} \in \mathcal{D}$ *for the load carrier of the general cargo unit, as well as the* amount $\in \mathbb{R}$ *demanded:*

$$tsd := (R, I, d_c, d_{lc}, amount)$$

The transport demand specification particularly comprises a set of transport relations $R \subseteq \mathcal{R}$ on which the general cargo unit requests transport. Note the difference to the specification of storage demand (Definition 7.2). It is not intended to leave the transport relation unspecified because it is usually not reasonable to transport cargo without a defined destination. Furthermore, the transport demand includes the descriptors $d_c, d_{lc} \in \mathcal{D}$ and the amount of cargo to be transported.

At first glance, it may seem surprising that the time span for transport is defined by means of a fixed discrete temporal interval and not by a variable one. However, this can be explained by the fact, that it is not necessary to specify a desired duration for the service. In contrast to storage, transport has not the purpose to bridge time. Therefore, the duration of the service is determined by the provider by means of its *duration* mapping (Equation 5.10). The provider can then check whether there is enough capacity for the service. To this end, it chooses a time span with the respective *duration* within the interval provided by the consumer.

Demand for handling and picking services is defined correspondingly:

Definition 7.5 (Handling Demand) *A handling service demand* $\mathrm{hsd} \in \mathcal{HSD}$ *is defined by a location* $\mathrm{l} \in \mathcal{L}$*, the discrete temporal interval* $\mathrm{I} \in \mathcal{I}$*, the descriptors for initial load carrier* $\mathrm{d_{lc}}$ *and terminal load carrier* $\mathrm{d'_{lc}}$*, as well as the* amount $\in \mathbb{R}$ *demanded:*

$$hsd := (l, I, d_{lc}, d'_{lc}, amount)$$

Definition 7.6 (Picking Demand) *A picking service demand* psd $\in \mathcal{PSD}$ *is defined by a location* l $\in \mathcal{L}$*, the discrete temporal interval* I $\in \mathcal{I}$*, the descriptors for initial cargo state* $\mathrm{d_c}$*, terminal cargo state* $\mathrm{d'_c}$*, initial load carrier* $\mathrm{d_{lc}}$*, and terminal load carrier* $\mathrm{d'_{lc}}$*, as well as the* amount $\in \mathbb{R}$ *demanded:*

$$psd := (l, I, d_c, d'_c, d_{lc}, d'_{lc}, amount)$$

Note that both handling and service capacity can only be requested for a particular location $l \in \mathcal{L}$. This is due to the fact, that it does not make sense to perform these services at arbitrary locations but only at the storage facility chosen by the general cargo unit.

7.1.2 Negotiation about Logistics Services

Allocating capacity for logistics services requires that general cargo units contact appropriate logistics service providers. A service consumer must inform the service providers about its demands (Section 7.1.1). This can be accomplished with the contract net interaction protocol (Figure 4.10). Algorithm 7.1 outlines the respective procedure for logistics service consumers. At first, the consumer transmits its *demand* to the *Providers*, a set of storage service providers. These providers have been determined in advance by the service consumer. A common way to retrieve potential storage service provider is querying a directory service (Definition 6.2), i. e., the yellow pages service of multiagent systems. The demand for logistics services is transmitted to the respective service providers by applying the *sendCFP* function. This function sends the call for proposals (CFP) and receives the responses by the service providers.

Algorithm 7.1 Allocate Capacity
$(Providers \subset \mathcal{SP}, demand \in \mathcal{SD}) \rightarrow proposal \in \mathcal{SO}$

```
$Proposals \Leftarrow sendCFP(Providers, demand) \subset \mathcal{SO}$
if $|Proposals| > 0$ then
    $best \Leftarrow chooseBest(Proposals)$
    $sendAccept(sender(best))$
    for all $proposal \in Proposals$ do
        if $proposal \neq best$ then
            $sendReject(sender(proposal))$
        end if
    end for
    return $best$
end if
return null
```

Logistics service providers process a call for proposals in three steps. Firstly, they check whether they serve the right location or relation. Secondly, they apply their mappings for matching (Equation 5.6) to determine whether they are capable of offering the service requested. Finally, they apply Algorithms 5.1 and 5.2 to find out whether they have sufficient capacity to satisfy the demand. Based on the outcome of this process, they may offer their services:

Definition 7.7 (Set of Service Offers) *Let the set of all service offers be denoted by* $\mathcal{SO}$*. It is defined as the union of the sets of offers for transport services* $\mathcal{TSO}$*, handling services* $\mathcal{HSO}$*, storage services* $\mathcal{SSO}$*, and picking services* $\mathcal{PSO}$*:*

$$\mathcal{SO} := \mathcal{TSO} \cup \mathcal{HSO} \cup \mathcal{SSO} \cup \mathcal{PSO}$$

A specific storage offer is defined as follows:

Definition 7.8 (Storage Offer) *A storage service offer* $\mathrm{sso} \in \mathcal{SSO}$ *is defined by the location* $l \in \mathcal{L}$*, the discrete temporal interval* $\mathrm{I} \in \mathcal{I}$*, the storage service provider* $\mathrm{ssp} \in \mathcal{SSP}$*, the* $\mathrm{amount} \in \mathbb{R}$*, and the* $\mathrm{cost} \in \mathbb{R}$*:*

$$sso := (l, I, ssp, amount, cost)$$

In this quintuple, *ssp* and l denote the service provider and its location, respectively. $I \in V$ is an interval that is in the variable temporal interval specified by the storage demand (Definition 7.2). The *amount* may differ from the one specified by the storage demand if the service provider has a minimal utilisation.

The cost specified in the storage offer do not necessarily cover only on the actual costs for storage. Depending on the specific application, also other fixed and variable costs must be considered:

1. Storage
2. Receiving and shipping
3. Delivery to the storage facility
4. Distribution from the storage facility
5. Demurrage and detention

Receiving and shipping refer to the handling costs before and after storage. Another aspect to be considered when choosing a storage provider are the transport costs. For instance, consider a warehouse that is well connected by means of mass transport. Then, it may be preferable to choose this warehouse even if it is slightly more expensive than others because the price for transport is significantly lower. A problem when incorporating transport costs in the choice of storage providers is that the transport provider is usually not yet chosen. Therefore, it is necessary to estimate the transport costs. For this estimation, one could consider previous transports on the same transport relation. To this end, the average distribution to the available means of transport has to be multiplied with the respective costs. Likewise, also the costs for a subsequent distribution process can be considered. To this end, it must be estimated how the goods are usually partitioned to distribution centres and be multiplied with the respective transport costs. If the goods are received late, one must also consider demurrage and detention, for instance, if containers must remain at the container terminal.

Regarding the temporal interval I, a service provider must determine whether it is actually within the bounds of the requested variable temporal interval V requested (Definition 7.3), i. e., whether $I \in V$. For communication efficiency reasons, one will most likely not transmit the whole set of sets of points in time defined by V. Instead, it is sufficient to specify V by means of a quadruple (t_a, t_c, t_d, t_f). Likewise, a fixed interval (Definition 5.10) can be represented by a tuple (t_b, t_e). In this representation, however, it cannot simply be checked whether $I \in V$. Instead, the following mapping is applied:

$$\begin{aligned} \mathit{satisfies} &: \mathcal{I} \times \mathcal{V} \rightarrow \mathbb{B} && (7.4)\\ \mathit{satisfies} &: (I,\, V) \mapsto b && (7.5) \end{aligned}$$

It can actually be implemented as follows:

$$\mathit{satisfies}((t_b, t_e), (t_a, t_c, t_d, t_f)) := \begin{cases} \text{true} & \text{if } \mathit{between}(t_b,\, t_a,\, t_c) \wedge \\ & \quad \mathit{between}(t_e,\, t_d,\, t_f) \\ \text{false} & \text{otherwise} \end{cases} \tag{7.6}$$

In Equation 7.6, the mapping *between* checks whether a point in time lies between two other:

$$\begin{aligned} \mathit{between} &: \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{B} && (7.7)\\ \mathit{between} &: (t, t_1, t_2) \quad \mapsto b && (7.8) \end{aligned}$$

The respective implementation of the signature is as follows:

$$\mathit{between}(t, t_1, t_2) := \begin{cases} \text{true} & \text{if } \min(t_1, t_2) \leq t \leq \max(t_1, t_2) \\ \text{false} & \text{otherwise} \end{cases} \tag{7.9}$$

As soon as the service consumer has received all offers by the service providers, it must choose the best proposal. To this end, a mapping with the following signature is applied:

$$\begin{aligned} \mathit{chooseBest} &: \mathcal{P}(\mathcal{SO}) \quad \rightarrow \mathcal{SO} && (7.10)\\ \mathit{chooseBest} &: \mathit{Proposals} \mapsto \mathit{best} && (7.11) \end{aligned}$$

The implementation of the mapping for choosing the best proposal depends on the specific application at hand. In particular, it is not restricted to incorporating costs but may also consider aspects such as quality and trust. Based on the choice for one proposal, the providers are informed about the outcome by the *sendAccept* and *sendReject* functions. In this context, the mapping *sender* denotes the respective creator of a proposal.

The discussion so far covered storage services. However, the procedure is similar for the other primary logistics functions (Section 2.1). If a service consumer has a transport demand (Definition 7.4), transport service providers determine whether they can process the request. Usually, this means that the provider finds a route between the current location and the destination of the cargo. Based on this route, it makes a respective offer:

Definition 7.9 (Transport Offer) *A transport service offer* tso $\in \mathcal{TSO}$ *is defined by the transport relation* $r \in \mathcal{R}$*, the discrete temporal interval* $\mathrm{I} \in \mathcal{I}$*, the transport service provider* tsp $\in \mathcal{TSP}$*, the* amount $\in \mathbb{R}$*, and the* cost $\in \mathbb{R}$*:*

$$tso := (r, I, tsp, amount, cost)$$

The temporal interval I in the offer must satisfy the interval I' specified in the transport demand (Definition 7.4), i. e., $I' \subseteq I$. This can be checked with a mapping of the following signature:

$$satisfies : \mathcal{I} \times \mathcal{I} \rightarrow \mathbb{B} \tag{7.12}$$

$$satisfies : (I, I') \mapsto b \tag{7.13}$$

The implementation simply applies Equation 7.6:

$$satisfies((t_b, t_e), (t'_b, t'_e)) := satisfies((t_b, t_e), (t'_b, t'_e, t'_b, t'_e)) \tag{7.14}$$

Offers for handling and picking services are defined correspondingly:

Definition 7.10 (Handling Offer) *A handling service offer* hso $\in \mathcal{HSO}$ *is defined by the location* $l \in \mathcal{L}$*, the discrete temporal interval* $\mathrm{I} \in \mathcal{I}$*, the handling service provider* hsp $\in \mathcal{HSP}$*, the* amount $\in \mathbb{R}$*, and the* cost $\in \mathbb{R}$*:*

$$hso := (l, I, hsp, amount, cost)$$

Definition 7.11 (Picking Offer) *A picking service offer* pso $\in \mathcal{PSO}$ *is defined by the location* $l \in \mathcal{L}$*, the discrete temporal interval* $\mathrm{I} \in \mathcal{I}$*, the picking service provider* psp $\in \mathcal{PSP}$*, the* amount $\in \mathbb{R}$*, and the* cost $\in \mathbb{R}$*:*

$$pso := (l, I, psp, amount, cost)$$

The understanding of autonomous control in the context of this research project assumes that service consumers and providers directly negotiate with each other without a centralised entity. In large networks, however, it does not make sense to send the request for capacity to service providers that are too distant to process the demand. Therefore, it is not desirable to broadcast the demand specification to all providers. Instead, it is promising to apply a concept like the regions of relevance proposed by Gehrke (2009, pp. 103–104) which allows confining oneself to specific regions and time intervals.

7.2 Inter-Agent Collaboration

The discussion so far covers the individual allocation of service capacity (Section 7.1). As elaborated in Section 5.2.2, it is often advantageous for process optimisation to jointly allocate logistics services. Examples are general cargo units that employ means of mass transport together. Another application are similar general cargo units that are stored at the same storage facility. With regard to the subsequent distribution process, this allows reducing empty vehicle running. The coordination required can be accomplished by teams of autonomous logistics entities. The agent interaction protocols for team formation (Section 6.2) allow forming teams of agents with similar cargo types. The team manager (Definition 5.14) is then responsible for coordinating the allocation of service capacity.

Section 7.2.1 describes how agents can jointly allocate logistics services. Service consumers may have to deal with logistics providers that require a minimal utilisation of their services. Section 7.2.2 investigates an optimistic approach to these challenges while Section 7.2.3 examines a conservative.

7.2.1 Joint Allocation of Logistics Services

Teams of logistics service consumers can be formed based on the team formation interaction protocols (Section 6.2) and the criteria discussed in Section 5.2.2. Having joined a team, general cargo units must provide their service demand (Definition 7.1) to the respective team manager. This can be accomplished by requesting service capacity from the team manager with the FIPA request interaction protocol (Figure 4.9).

Algorithm 7.2 outlines this procedure from the perspective of the team manager. The team manager administers the $\mathit{ChosenProviders} \subseteq \mathcal{SP}$, a set of storage service providers which have already been chosen by this team before. Initially, this set is empty. The manager continuously waits for messages sent by team members which provide their service demands. Think, for instance, of general cargo units that aim at joint storage. If the team manager receives a demand from one of its members, it applies the contract net interaction protocol (Figure 4.10) as specified in Algorithm 7.1. The parameters are the demand submitted as well as the set of providers chosen before. The confinement to providers previously chosen by this team, ensures joint storage if possible. The outcome of the contract net is a matching provider. To recapitulate, a potential outcome of Algorithm 7.1 is that no appropri-

Algorithm 7.2 Collaboratively Allocate Capacity

```
$ChosenProviders \Leftarrow \emptyset$
loop
    $demand \Leftarrow receiveMessage()$
    if $demand = $ null then
        continue
    end if
    $offer \Leftarrow allocateCapacity(ChosenProviders, demand)$
    if $offer = $ null then
        $OtherProviders \Leftarrow retrieveAllProviders() \setminus ChosenProviders$
        $offer \Leftarrow allocateCapacity(OtherProviders, demand)$
        if $offer = $ null then
            exception "No matching provider found"
        end if
    end if
    $ChosenProviders \Leftarrow ChosenProviders \cup sender(offer)$
    $sendInform(sender(demand), offer)$
end loop
```

ate storage service provider is found. This is particularly the case if the set of *ChosenProviders* is empty at the beginning. Another reason is that the capacity of a previously chosen storage provider is exhausted. This may, for instance, be the case for very large teams. Then, it is necessary to employ an additional provider.

In order to find further potential service providers, the team manager queries a directory service (Definition 6.2) with the *retrieveAllProviders* function. The previously chosen providers are subtracted from this set because they were already queried in the first iteration. Again, Algorithm 7.1 is applied with the new storage service provider candidates. The best one is selected and added as a new member of the set of chosen providers. By means of the response of the previously initiated FIPA request interaction protocol, the initiator is informed about the successful allocation. As an exceptional case, it may happen that even in this step no matching provider is found. This case cannot be handled by the system because additional capacity is required by service providers that are currently not part of the logistics network. This issue must be handled by a human dispatcher.

To summarise, collaborative allocation of service capacity can be applied by combining both FIPA contract net and request interaction protocol (Figure 7.2) with a team formation interaction protocol. Initially, similar general cargo units form a team in order to coordinate joint storage. One of the autonomous logistics entities is chosen to be the team manager. All team members submit their storage demands to the team manager by applying the FIPA request protocol. The team manager then uses the FIPA contract net protocol in order to negotiate with appropriate storage service providers. The approach is capable of flexible reaction if the storage capacity of one service provider is exhausted. Then, additional providers may be added. Note

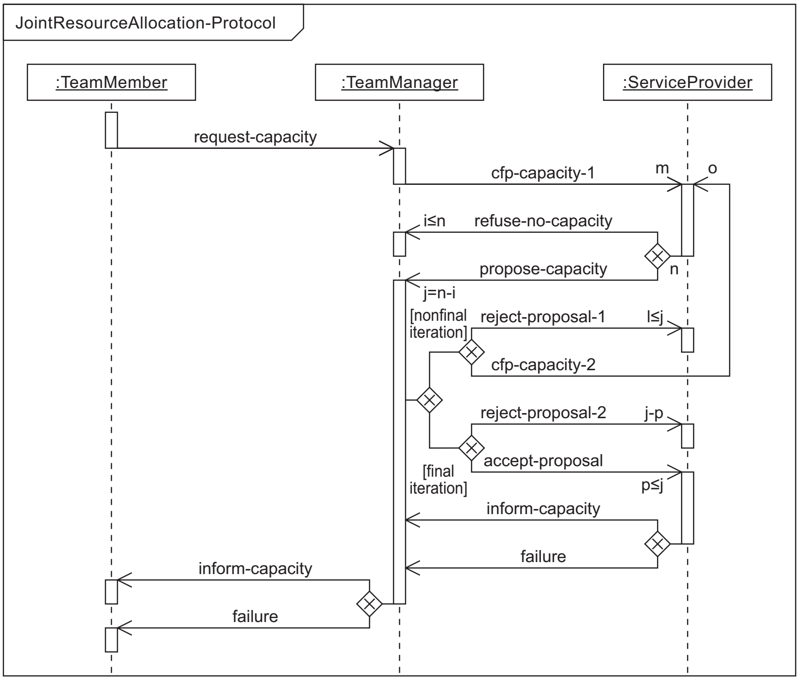

Fig. 7.2 In the combined interaction protocol for joint resource allocation, members inform their team manager about their demand by the FIPA request protocol. The team manager then negotiates with the logistics service provider based on the FIPA iterated contract net protocol.

that the combined protocol depicted in Figure 7.2 incorporates the FIPA iterated contract net protocol (Foundation for Intelligent Physical Agents, 2002) rather than its non-iterated counterpart because resource allocation may require up to two steps (Algorithm 7.2).

Teams for joint allocation of logistics resources are formed based on common objectives or properties (Section 5.2.2). These objectives and properties depend on the primary logistics function (Section 2.1) addressed. As an example, remember that storage teams are formed by similar cargo properties while transport teams are formed based on common location and destination. Teams are established flexibly on demand. If currently no team exists with specific properties, a new one is formed. Furthermore, teams can be dissolved if all members have achieved their logistics objects, i. e., if all members have left the logistics network.

7.2.2 Optimistic Allocation of Logistics Services

Some service providers require a minimal utilisation of their services. For instance, consider a shipping container that aims at being transported from its current location to a destination. Then, employing a means of mass transport like a train is cheaper than transport by truck. This finding, however, is only true if the container is not transported on its own. Instead, multiple containers must be transported together by train in order to save costs. If a shipping container is individually transported by train, the costs are most likely to be higher than transport by truck. Hence, teams must appropriately deal with minimal utilisation of service providers.

This challenge can be approached in different ways. One of them is to optimistically allocate logistics services. In this approach, the team manager accepts a higher amount of service utilisation than initially requested. The surplus of service capacity must then be administered and distributed by the team manager. To this end, the team manager itself becomes a service provider for these resources. Subsequently, the team manager adds itself to the set of chosen providers. This ensures that future service requests by team members are primarily served from the previously allocated capacity.

Proceeding this way is simple and effective. However, an optimistic team manager has the risk of allocating more resources than actually needed by its team. This can be addressed by accepting offers with minimal utilisation only if there is an indicator that the additional resources will be consumed by the team members. This can, for instance, be accomplished with statistics from past occasions. Another alternative is to cancel previous resource allocations if they turn out to be insufficiently exhausted. This cancellation can lead to decommitment penalties as, for instance, investigated by Mao, ter Mors, Roos and Witteveen (2007, pp. 139–142).

7.2.3 Conservative Allocation of Logistics Services

An alternative to optimistic allocation of logistics services is to act conservatively. A conservative team manager does not immediately negotiate about logistics resources. Instead, it collects demands from its team members until there is sufficient demand for joint utilisation. If not enough demand accumulates, the team manager must employ individual service providers instead. Administering a set of demands by service consumers comprises two tasks. Firstly, the team manager must prioritise demands in order to serve its members in accordance to their urgency. To this end, it has a mapping with the following signature:

$$priority : \mathcal{SC} \times \mathbb{N} \quad \to \mathbb{R} \tag{7.15}$$

$$priority : (gcu, date) \mapsto p \tag{7.16}$$

Secondly, the manager must temporally coordinate the demands of its team members (Schuldt & Werner, 2007, p. 127–128). Time supplements the conceptual and spatial criteria for team action discussed in Section 5.2.2. In particular, one is not interested in specific quantitative data when relating temporal intervals with each other. Instead, one aims at describing general qualitative relationships, e. g., by applying the set of 13 relations (Figure 4.8) introduced by Allen (1983, p. 835). The definition of the qualitative relation between two discrete temporal intervals $I_1, I_2 \in \mathcal{I}$ is as follows:

Definition 7.12 (Temporal Relation) *Let* $\mathrm{I}_1, \mathrm{I}_2 \in \mathcal{I}$ *be discrete temporal intervals. The relation* $\mathrm{I}_{1\,\mathrm{I}_2}$ *of* I_2 *with respect to* I_1 *is characterised as*

$$I_{1\,I_2} \in \{<, >, \mathrm{m}, \mathrm{mi}, \mathrm{o}, \mathrm{oi}, \mathrm{s}, \mathrm{si}, \mathrm{d}, \mathrm{di}, \mathrm{f}, \mathrm{fi}, =\}$$

The intervals can, for instance, represent the time during which an autonomous logistics entity resides at one location. For joint transport of two entities, it would then be necessary that the end dates equal. The relations are restricted to pairs of temporal intervals. However, teams in autonomous logistics (Definition 5.14) are usually not restricted in size, i. e., they can comprise more than two members. Consequently, also more temporal intervals have to be related to each other. This can be reflected by representing the relations between all pairs of intervals in a matrix (Figure 7.3):

Definition 7.13 (Temporal Relation Matrix) *Let* $\langle \mathrm{I}_1, \ldots, \mathrm{I}_\mathrm{n} \rangle \in \mathcal{I}^n$ *be an ordered sequence of discrete temporal intervals. The temporal relation matrix*

$$TM = \langle I_1, \ldots, I_n \rangle \times \langle I_1, \ldots, I_n \rangle$$

with

$$\forall_{I_i = \{t_{b_i}, \ldots, t_{e_i}\}} \; t_{b_i} < t_{b_{i+1}} \;\vee\; t_{b_i} = t_{b_{i+1}} \;\wedge\; t_{e_i} \leq t_{e_{i+1}}$$

characterises the relations $\mathrm{I}_{\mathrm{i}\,\mathrm{I}_\mathrm{j}}$ *between all discrete temporal intervals in* $\langle \mathrm{I}_1, \ldots, \mathrm{I}_\mathrm{n} \rangle$ *with* $\mathrm{I}_{\mathrm{i}\,\mathrm{I}_\mathrm{j}}$ *being the entry in row* I_i *and column* I_j*.*

To arrive at a unique matrix, the participating temporal intervals are ordered in accordance with their occurrence.

Based on the matrix of relations of temporal intervals, it is then possible to define predicates which must hold for particular joint actions:

Definition 7.14 (Temporal Relation Matrix Restriction) *Let* TM *be a temporal relation matrix. Restrictions on the matrix can be described by a set* Rel *of temporal relations and one of the following predicates:*

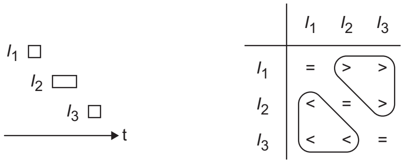

Fig. 7.3 Three temporal intervals ordered in accordance with their occurrence (left) and the matrix of their interrelationships (right). It can be derived from the matrix that the intervals do not intersect.

$$\begin{aligned} \mathit{wholeMatrix}(\mathit{TM}, \mathit{Rel}) &= \forall_i \forall_j \; (I_{i\,I_j} \quad \in \mathit{TM} \land i \neq j) \rightarrow I_{i\,I_j} \quad \in \mathit{Rel} \\ \mathit{matrixRow}(\mathit{TM}, \mathit{Rel}) &= \exists_i \forall_j \; (I_{i\,I_j} \quad \in \mathit{TM} \land i \neq j) \rightarrow I_{i\,I_j} \quad \in \mathit{Rel} \\ \mathit{matrixDiagonal}(\mathit{TM}, \mathit{Rel}) &= \forall_i \qquad I_{i\,I_{i+1}} \in \mathit{TM} \qquad\qquad \rightarrow I_{i\,I_{i+1}} \in \mathit{Rel} \end{aligned}$$

As an example, the left hand side of Figure 7.3 shows the times at which three shipping containers have to be transported. The respective temporal intervals are already ordered in accordance with their start and end times. Consider a truck that is able to transport one shipping container at a time. The question is then whether the three depicted shipping containers can employ the same truck one after another. To answer this question, the right hand side of the same figure depicts the matrix relating all intervals with each other. The precondition for utilising the same truck is that no pair of temporal intervals intersects. This can be determined by examining the whole matrix without its main diagonal. Note that the main diagonal always contains the identity relation =. Indeed, it turns out that only < and > relations appear, i. e., there are no intersections:

$$\mathit{wholeMatrix}(\mathit{TM}, \{<, >\}) \tag{7.17}$$

From this follows that the three shipping containers examined can actually employ the same truck one after another.

Another example is depicted in Figure 7.4. It comprises the temporal intervals during which three general cargo units populate a logistics network. They plan a joint team for transport. The question is whether one of them can take over team management or whether an additional agent is required. That is, they have to find out whether the lifetime of one entity subsumes the one of all others. A row in the matrix comprises the relations of interval to all others. Therefore, a row has to be found that only contains the relations s, d, f, and =:

$$\mathit{matrixRow}(\mathit{TM}, \{\mathrm{s}, \mathrm{d}, \mathrm{f}, =\}) \tag{7.18}$$

In the example, this holds for the first row. Hence, the autonomous logistics entity with the first interval can actually become the team manager.

Figure 7.5 illustrates a third configuration. The temporal intervals represent the duration at which three general cargo units are stored in a warehouse. Starting from this location, the cargo units have the same destination. There-

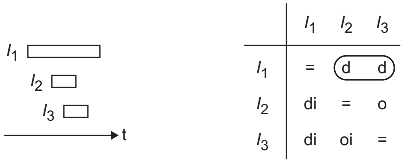

Fig. 7.4 Another configuration of three temporal intervals (left) and the matrix of their interrelationships (right). The matrix reveals that one interval covers the duration of all others.

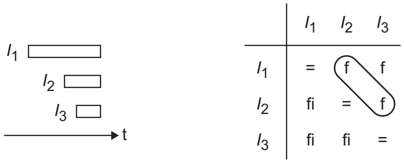

Fig. 7.5 A configuration of three temporal intervals (left). The matrix of their interrelationships (right) shows that all intervals share a common end.

fore, they aim at sharing a means of mass transport. To find out whether this is possible, the matrix of temporal relations is created. The precondition for joint transport is that the intervals share the same end point for storage at the previous location. It holds when the relations between all pairs of subsequent intervals are either *f* or =. Pairs of subsequent intervals can be found in the diagonal of the matrix:

$$\mathit{matrixDiagonal}(\mathit{TM}, \{\mathrm{f}, =\}) \tag{7.19}$$

7.3 Intra-Agent Coordination

As an intermediary result, autonomous logistics entities can jointly allocate logistics resources (Section 7.2). However, general cargo units usually do not only demand transformation by one logistics function (Section 2.1). By contrast, multiple functions must be combined. This is challenging because one is not interested in arbitrary combinations (Section 2.3). Instead, one aims at optimising the execution of the primary logistics functions according to the logistics objectives imposed by the cargo owner. In particular, fixed dates must be met by general cargo units. Such dates may, for instance, be due to the sales start or quality assurance of the respective products.

Autonomous logistics entities are themselves responsible for planning and scheduling their way through the logistics network. To this end, they must coordinate the execution of the primary logistics functions. In order to ease this task, it is important to investigate whether a general order exists for these functions. Actually, two orders can be distinguished. Section 7.3.1 focuses on the execution order of the primary logistics functions. Subsequently, Section 7.3.2 elaborates on the planning order of these functions. Based on these foundations, the coordination of the primary logistics functions is then approached in Section 7.3.3. Finally, Section 7.3.4 discusses supply network exception management.

7.3.1 Execution Order of Logistics Functions

Logistics objects that demand some transformations (Figure 2.1) enter the logistics network at one location. After all necessary transformations have been applied, the objects leave the network at another location. In this process, transport is an important aspect, be it within one company site or between different sites (Section 2.1.1). Consider, for instance, consumer products that are delivered from the manufacturer to the end customer. Often, there is no direct connection between two locations. Direct connections between all pairs of two connections would require a complete meshing which is rather ineffective. An alternative is to employ storage facilities like warehouses and distribution centres in between (Section 2.1.3). The products are delivered from the manufacturing site to a warehouse which stores them until their sales date. Then, they are transported to distribution centres which are responsible for coordinating delivery to the end consumer.

Following the discussion so far, the primary logistics functions of transport and storage alternate. Both operations, however, cannot be applied directly one after another. As elaborated in Section 2.1.2, handling is the interface between transport and storage because this operation unloads cargo from means of transport in order to receive it at a storage facility. In this process, quantities are often adjusted. To reiterate a previous example, goods are unloaded from a shipping container and re-packed to pallets. The other direction is shipping, i. e., transferring cargo out of the storage facility onto means of transport. This can also be done by handling. However, often particular customer orders are to be satisfied. Hence, it is not sufficient to simply adjust quantities but also the composition of the cargo. This can be accomplished by picking (Section 2.1.4). According to Definition 5.13, picking is considered a more general concept that subsumes handling. Therefore, picking can be applied during shipping even if only handling operations were necessary. To summarise, this leads to the following cyclic order (Figure 7.6):

1. Picking
2. Transport
3. Handling
4. Storage

So far, it is assumed that storage and transport operations alternate. For storage, this is quite obvious. If cargo is stored at one location and supposed to be stored at another, it must be transported. Hence, transport is necessary between storage operations at different locations. Regarding transport, however, one could argue that two transport operations can be executed directly one after another without storage in between. Handling allows transferring cargo directly from one means of transport to another (Section 2.1.2). An example are hub and spoke networks as they are, for instance, applied in the courier, express, and parcel (CEP) business (Figure 3.7). In this concept, a logistics object passes several locations on the way from its origin to its destination. One could, of course, argue that transshipment always includes storage, at least with a minimum amount of time. However, the actual line of argument is different because here the perspective is on general cargo units as autonomous logistics entities. Hubs and spokes are internal structures of the network of CEP providers. As elaborated in Section 5.1, autonomous general cargo units do not consider routing but only request transport from one location to another one. Therefore, they can abstract from the routing task of the service provider. From the service consumer perspective, the execution order of primary logistics functions is thus actually the one illustrated in Figure 7.6.

7.3.2 Planning Order of Logistics Functions

The execution order of the primary logistics functions (Section 7.3.1) derives from their interdependencies. Likewise, also the order in which the operators are chosen is not arbitrary. Nevertheless, the planning order distinguishes from the execution order as follows:

1. Allocate storage capacity
2. Allocate transport capacity
3. Allocate picking and handling capacity

Based on the interdependencies, this order can be motivated as follows. It is impossible to choose means of transport on a particular transport relation without defining the destination in advance. From its current storage location within the logistics network, the autonomous logistics entity must choose its next destination to achieve its logistics objectives. Depending on the logis-

Fig. 7.6 The primary logistics functions of storage and transport alternate. Two storage locations can be connected by one transport operation. Between each pair of storage and transport operators, it is necessary to apply handling and picking operators, respectively.

tics objectives defined by the cargo owner, this destination may already be fixed to one concrete storage facility. Otherwise, the entity may itself select an adequate storage facility. The commitment to particular storage capacity allows allocating other logistics services in the next step.

Choosing a storage provider constrains the choice for potential transport service providers to those who serve the transport relation between origin and destination. When choosing the actual date for transport, one aims at minimising costs for storage, e. g., due to demurrage and detention at one location. However, one is also constrained by the transport capacity available. From the choice of transport service provider also the shipping date at the former location and the receiving date at the next destination arise. It is then possible to organise both shipping and receiving, i. e., to allocate capacity for picking and handling.

7.3.3 Coordinating the Logistics Functions

The planning order of the primary logistics functions has been examined in Section 7.3.2. Based on that foundation, this section investigates how these functions can actually be coordinated. Consider an autonomous logistics entity that is currently stored in a storage facility. The cargo owner demands it to be at another location at a given time. The first step is to identify an appropriate storage facility at the destination. This identification task also includes finding out whether similar goods of the same cargo owner are already stored in a facility in that region (Section 7.2.1). Therefore, the entity initiates a team formation process for joint storage with similar cargo at its scheduled destination. The similarity of the cargo is determined by means of the respective cargo descriptors (Definition 5.8). The destination is specified by a location of the environment graph (Definition 5.1). The respective team manager can then choose an adequate storage facility.

Choosing an appropriate storage facility is necessary but not sufficient. Additionally, it is important to specify the temporal interval for the storage at the new facility. As elaborated in Section 7.1.1, the temporal demand for storage can be specified variably (Definition 7.3). In the example currently examined, the end point for storage is fixed. It corresponds to the point in time at which the cargo owner wants to consume the cargo at the destination, e. g., by delivering it to an outlet. The begin time for storage at the new destination could be chosen in accordance with the end date of storage at the origin storage facility. Proceeding this way, however, would constrain the possible time for transport to exactly one date. Such a restriction is not desirable, particularly against the background that no transport resources might be available at that time.

Instead, it is advantageous to choose overlapping temporal intervals for storage at the former and the next storage facility. With such overlaps, one is more flexible regarding the transport date. The transport can be conducted at some time during the overlap. The time at which the transport is actually scheduled is not arbitrary but depends on the enframing storage contracts. If storage at the initial storage facility is cheaper, it is desirable to conduct the transport as late as possible in order to benefit from the low storage rates. Otherwise, an early transport is desirable. This can be reflected by specifying the respective temporal interval in the transport demand (Definition 7.4) in descending or ascending order, respectively. The actual transport is coordinated within a team of other autonomous logistics entities sharing the same location and destination.

The desired overlap can be reflected by a variable begin time for storage at the destination (Definition 7.3). Such a variable begin time has two bounds. One bound is specified by the end time of storage at the previous facility. The other one lies some time before this date so that both storage contracts overlap initially. The actual overlap depends then on the capacity available at the destination storage facility (Section 7.1.2). When the transport date is appointed, the storage contracts can be adapted accordingly.

Following the allocation of storage and transport resources, capacity for picking and handling can be allocated (Section 7.3.2). To this end, negotiations with the respective service providers (Definitions 5.11 and 5.13) are required. The choice of appropriate service providers is constrained by the selected storage service providers. Only those picking and handling providers are eligible that operate at the locations of the previous and next storage provider, respectively. The time for conducting the picking and handling operation is constrained by the contract with the transport service provider. Preferably, shipping and receiving should be conducted immediately, or at least shortly, before and after transport. Consider a truck that delivers general cargo units. It is not acceptable if trucks have to wait for several hours or even days. Instead, it is particularly important to unload the truck soon because the tractor unit and its driver cannot earn money as long as they await loading and unloading. If the respective picking and handling capacity cannot be allocated at acceptable times, also the transport contract must be cancelled. Instead, the general cargo unit must request earlier or later transport. This procedure must be repeated until picking, transport, and handling capacity match. The allocation of logistics resources is thus not guaranteed to be efficient under these circumstances.

In contrast to trucks, shipping containers and swap bodies are less time critical. They can wait some time until they are unloaded because the tractor unit and its driver can already complete other jobs after having transported the unit to the storage facility (Section 2.1.2). Therefore, it is possible to handle them more flexibly than trucks. Shipping containers and swap bodies may have different priorities. A general cargo unit that is collected from and delivered to a storage facility is only aware of its own priority. For coordinated shipping and receiving in accordance with the priorities, it is thus necessary that the general cargo units cooperate. This can be accomplished by forming teams based on the location $l \in \mathcal{L}$ at which they demand handling. To this end, one of the agent interaction protocols for team formation introduced in Section 6.2 can be applied. When joining such a team, each team member transmits its priority (Equation 7.15) to the responsible team manager. The team manager can then establish a priority queue that orders all members in accordance to the time frame requested for picking and handling as well as the priority. Based on this prioritisation, the team manager can allocate capacity for its team members (Section 7.2.1).

7.3.4 Supply Network Exception Management

Autonomous logistics automates supply network management by delegating process control to the participating entities. Section 7.3.3 describes process control for standard cases. An important property of autonomous logistics entities and software agents in general is that their actions may fail. Such exceptional cases originate from two sources:

1. The inner state of the autonomous logistics entity
2. The environment of the autonomous logistics entity

Changes of the inner state of the autonomous logistics entity can be monitored by sensor technology (Section 3.2.3). An example is increasing temperature within a refrigerated shipping container. Such an increase in temperature may decrease the shelf life of food loaded. An exception occurs if the food will most likely be perished at the arrival of the container.

Examples for exceptions originating from the environment are as follows. Consider a general cargo unit that is transported by truck through the logistics network. Due to traffic congestion, it is in question whether the truck can actually hold the delivery time estimated before transport. Another example deals with shipping containers that are transported by container vessel from East Asia to Europe. Due to weather conditions, the vessel may arrive late at its destination.

Exceptions must be dealt with appropriately. Another important property of software agents is their reactivity (Section 4.1.1). The capability for local re-planning is a particular advantage of autonomous logistics over previous control approaches. Only the affected entities have to update their planning and scheduling. Two types of exceptions must be distinguished:

1. Exceptions that can be resolved by the participating entities
2. Exceptions that go beyond the boundaries of the system

It is important to investigate which exceptions pertain to which class. In particular, exceptions must be identified that pertain to the second class, therefore demanding attention by human dispatchers.

Hardware defects, be it in the identification, localisation, sensor, communication, or data process unit, cannot be resolved by the autonomous logistics entity itself. By contrast, state changes of the cargo itself can be reflected by the entity. As an example, a decreased shelf life can be addressed by updating the former schedule in order to re-route the shipping container (Section 7.3.3). However, in handling internal exceptions, further exceptions may occur that are due to the environment. A prototypic example for external exceptions are limited resources. An important reason for insufficient resources is due to contracts which offer pre-negotiated capacity for fixed costs. Often, autonomous logistics entities can cope with insufficient resources by adapting the time of service consumption. However, this is only possible if capacity is available at another acceptable time. The boundaries of the system are then shaped by the capacity of the logistics resources available. If capacity does not suffice, it is necessary to extend the logistics network with additional capacity or additional service providers (Section 7.2.1). To this end, action of a human dispatcher is required who adds further logistics resources.

7.4 Conclusion

To summarise, this chapter introduces an approach coordinating the participants in autonomous logistics processes. As a starting point, efficient interaction between service providers and individual service consumers is investigated. Such individual action may suffice for some applications. In general, however, there is a potential for cooperation in autonomous logistics networks. In particular, Chapter 5 identifies an increased interaction efficiency and an increased utilisation efficiency of logistics resources as reasons for cooperation. Following the model for cooperation by Wooldridge and Jennings (Section 4.3.2), the initial approach of individual action has thus been extended to joint action of service consumer teams. These teams are formed on a short-term basis by means of the interaction protocols developed in Chapter 6. The purpose of each of these teams is to coordinate the joint utilisation of a particular logistics resource. Therefore, inter-agent collaboration of multiple team members is only one step towards coordinated process control. Additionally, it is necessary to coordinate the utilisation of multiple resources. This can either be conducted by individual entities or by long-term teams, e. g., multiple sales units that move jointly on a common load carrier through the logistics network.

Challenges for logistics control have been examined in Chapter 2. As elaborated in Section 2.3, these challenges are due to the complexity, the dynamics, and the distribution of logistics processes. Although operational research finds optimal or near-optimal solutions, it can only be applied to control subsystems in reasonable time. At first glance, the approach of autonomous logistics with its decentralised decision-making may seem contradicting. Actually, however, it is supplemental because it constitutes an instantiation of the loosely, but intelligently coupled control systems discussed by Bretzke

(2008, pp. 30–33). This becomes apparent when examining service provider teams in autonomous logistics. These service providers resemble the intelligent control systems of Bretzke (Section 2.3). Within these self-contained entities, the powerful methods from operational research can be applied successfully. Think, for instance, of route planning by transport service providers. Considering service providers separately, reduces problem complexity and distribution. Furthermore, such subsystems are more robust in the sense that they can react on dynamics flexibly on the local level without the need for re-planning the whole system.

Autonomous general cargo units as service consumers embrace the individual logistics service providers. More precisely, the service consumers combine the required services flexibly on demand. Today, it is quite common that individual logistics functions are optimised by means of operational research methods. The holistic view over multiple functions from the perspective of general cargo units, however, is usually taken by human dispatchers. The principle of autonomous logistics implemented with Distributed Artificial Intelligence automates this task in supply network management for standard cases. The limitations of this automation are exceptional cases which cannot be handled without support by human dispatchers.

References

Allen, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, *26*(11), 832-843.

Bretzke, W.-R. (2008). *Logistische Netzwerke.* Heidelberg, Germany: Springer-Verlag.

Foundation for Intelligent Physical Agents. (2002). *FIPA Iterated Contract Net Interaction Protocol Specification* (Standard No. SC00030H). Geneva, Switzerland.

Gehrke, J. D. (2009). Evaluating Situation Awareness of Autonomous Systems. In R. Madhavan, E. Tunstel & E. Messina (Eds.), *Performance Evaluation and Benchmarking of Intelligent Systems* (pp. 93–111). Heidelberg, Germany: Springer-Verlag.

Mao, X., ter Mors, A., Roos, N. & Witteveen, C. (2007). Coordinating Competitive Agents in Dynamic Airport Resource Scheduling. In P. Petta, J. P. Müller, M. Klusch & M. Georgeff (Eds.), *5th German Conference on Multiagent System Technologies (MATES 2007)* (pp. 133–144). Leipzig, Germany: Springer-Verlag.

Schuldt, A. & Werner, S. (2007). Distributed Clustering of Autonomous Shipping Containers by Concept, Location, and Time. In P. Petta, J. P. Müller, M. Klusch & M. Georgeff (Eds.), *5th German Conference on Multiagent System Technologies (MATES 2007)* (pp. 121–132). Leipzig, Germany: Springer-Verlag.

Vickrey, W. (1961). Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, *16*(1), 8–37.

Vilain, M., Kautz, H. & van Beek, P. (1990). Constraint Propagation Algorithms for Temporal Reasoning: A Revised Report. In D. S. Weld (Ed.), *Readings in Qualitative Reasoning about Physical Systems* (pp. 373–381). San Francisco, CA, USA: Morgan Kaufmann Publishers.

Wooldridge, M. & Jennings, N. R. (1999). The Cooperative Problem Solving Process. *Journal of Logic & Computation*, *9*(4), 563–592.

Part III
Application and Evaluation

Chapter 8
Implementing Autonomous Logistics

The complexity, the dynamics, and the distribution of logistics processes are major challenges in supply network management (Chapter 2). The paradigm of autonomous logistics addresses these challenges by delegating process control to local logistics entities (Chapter 3). Distributed Artificial Intelligence and particularly agent technology have been identified as appropriate means for implementing autonomous control in logistics (Chapter 4). Chapters 5 to 7 specify a respective agent-based approach. The chapter at hand describes the actual implementation of this specification. Like for all software systems, it is important to test and evaluate the new approach before practical application. Multiagent-based simulation allows testing the behaviour of multiagent systems. In contrast to other kinds of simulation, it reflects the actual system behaviour by directly transferring agents from operation to simulation and vice versa.

Section 8.1 describes the implementation of autonomous logistics by means of multiagent systems. This description covers both the underlying multiagent platform and the actual software agents and their behaviour. Subsequently, Section 8.2 turns the attention to multiagent-based simulation of autonomous logistics processes.

8.1 Multiagent-Based Implementation

The implementation of autonomous control with multiagent systems is approached in three steps. Section 8.1.1 presents the underlying multiagent platform. Based on this foundation, Sections 8.1.2 and 8.1.3 describe the actual multiagent-based implementation. Thereby, Section 8.1.2 lays a particular focus on the implementation of team formation while Section 8.1.3 deals with team action of autonomous logistics entities.

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics,*
DOI 10.1007/978-3-642-20092-2_8,

8.1.1 Multiagent Platform

Prior to implementing autonomous logistics processes with software agents, it is important to choose an appropriate tool or, to be more precise, an adequate multiagent platform. Weiß and Jakob (2005, pp. 279–283) give an overview over existing platforms and categorise them regarding several attributes. Interoperability is most important for applications that go beyond particular academic questions. This especially holds in the logistics domain with a potentially high number of participants from different companies (Section 2.2). Respective standards for agent interoperability have been issued by FIPA (Section 4.2). Therefore, it is desirable that a multiagent platform to be applied in autonomous logistics processes is in accordance with these standards. This only holds for some of the systems investigated by Weiß and Jakob. Out of these systems, JADE claims to be the probably most widespread (Bellifemine, Caire & Greenwood, 2007, p. 1).

JADE is an acronym for the Java Agent Development Framework which has been developed by Telecom Italia Lab and the University of Parma (Weiß & Jakob, 2005, pp. 202). JADE is based on Java and capable of executing software agents distributed over multiple computers. On the one hand, this means that the platform is scalable because agents can be executed in parallel. On the other hand, it is possible to distribute agents in accordance with the real-world distribution of the objects represented. The Lightweight and Extensible Agent Platform (Adorni, Bergenti, Poggi & Rimassa, 2001; Bergenti & Poggi, 2001), in short LEAP, even allows migrating JADE agents to mobile devices with limited resources.

The JADE architecture is as follows (Bellifemine et al., 2007, pp. 32–34). As depicted by the UML (Booch, Rumbaugh & Jacobson, 2005) class diagram in Figure 8.1, the overall agent platform is composed of several agent containers. These agent containers are actually populated by the software agents and may be distributed over a computer network. A designated main container is responsible for coordinating the other containers. In accordance with FIPA standards, an agent management system and a directory facilitator provide white and yellow pages services (Figure 4.4). By means of the message transport system, agents can exchange messages. Each container delivers these messages within the container, between containers, and over the borders of the multiagent platform. Since JADE version 3.5, agents can subscribe to multicast topics. It is thus possible to benefit from multicast messages as demanded by one of the agent interaction protocols for team formation in autonomous logistics (Section 6.2.3).

Within agent containers, software agents act in parallel. To this end, each agent is executed as an operating system thread. The actual behaviour of an agent is implemented within so-called behaviour classes (Figure 8.1). Instances of these classes are then scheduled in accordance with specific scheduling policies defined by the agent programmer, e. g., sequential or in parallel. A specific subtype of behaviours are cyclic behaviours which do not have

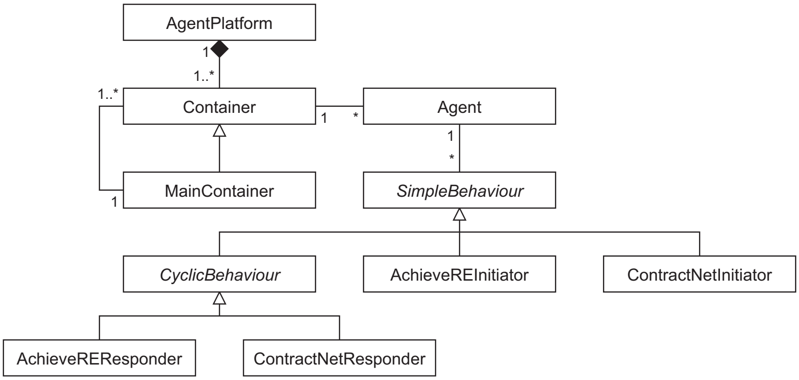

Fig. 8.1 UML class diagram illustrating the architecture of the JADE multiagent platform. Software agents populate agent containers which may be distributed over a computer network. Behaviour of agents is implemented in classes with corresponding names (adapted from Bellifemine et al., 2007, pp. 33, 92).

a specified end. Instead, they are continuously executed, e. g., in order to process requests by other agents.

In JADE, participation in interaction protocols is also implemented by behaviours. As depicted in Figure 8.1, initiators for the FIPA request and FIPA contract net interaction protocols can be derived from simple behaviours. Correspondingly, the responders for these protocols extend the abstract cyclic behaviour. Note that the internal implementation of the behaviours implementing protocols applied in this project slightly deviates from the standard JADE implementation. This is for the sake of compatibility with time synchronisation in multiagent-based simulation (Section 8.2.2). Figure 8.2 illustrates the general principle of behaviours implementing participation in interaction protocols. Callback methods are employed in order to prepare messages to be sent and to handle messages received. The FIPA interaction protocols can be applied in order to construct more complex protocols like those for team formation (Section 6.2). Each message cycle in the team formation protocols can be implemented by means of the FIPA request interaction protocol (Section 4.2.4).

8.1.2 Implementation of Team Formation

The team formation interaction protocols introduced in Chapter 6 have been implemented as depicted in Figure 8.3. The implementation can be divided into several parts. Firstly, the underlying JADE implementation providing the fundamental agent and behaviour classes. These basic behaviours also include the behaviour-based implementation of the FIPA request protocol

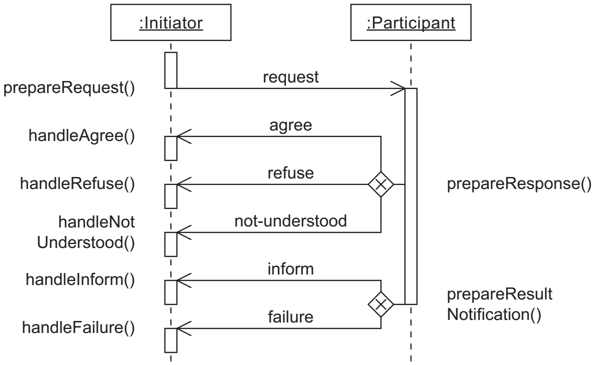

Fig. 8.2 Callback methods handling the states of the FIPA request interaction protocol (Gehrke & Schuldt, 2009, p. 1176).

discussed in the preceding Section 8.1.1. Secondly, auxiliary agents such as the broker and the directory service. Finally, the actual implementation of behaviours related to team formation.

The agent class in the upper left hand side of Figure 8.3 is the base class for all agents within the multiagent system. The UML class diagram depicts two specific agents derived from the general agent class, namely the broker (Definition 6.3) and the directory (Definition 6.2) service. Each of these agents has only one behaviour, the broker responder and the directory responder, respectively. Both behaviours are implemented as responders of the FIPA request interaction protocol (Section 4.2.4) which itself is a cyclic behaviour (Section 8.1.1). The responders administer the entries of the broker and the directory, respectively. To this end, they continuously wait for messages by other agents and handle search (Equation 6.4), register (Equation 6.7), and deregister (Equation 6.11) requests.

Team managers and participants are not specialised subclasses derived from the agent base class. Instead, managing a team and participating in team formation are roles that can be taken by every agent that decides to do so. In order to participate in team formation, an agent may use a team formation initiator class which extends the simple behaviour. The team formation initiator is an abstract base class for participants in the team formation interaction protocols (Section 6.2). One implementation is the participant in broker-based team formation (Section 6.2.2). It is implemented as a finite state machine that is composed of two sub-behaviours. On the one hand, a behaviour that implements registering with the broker. On the other hand, a behaviour that is responsible for joining teams in case that registering with the broker fails. Both sub-behaviours are derived from the general initiator of the FIPA request protocol (Section 8.1.1).

The implementations for team formation by directory and team formation by multicast are also derived from the abstract team formation initiator. Both

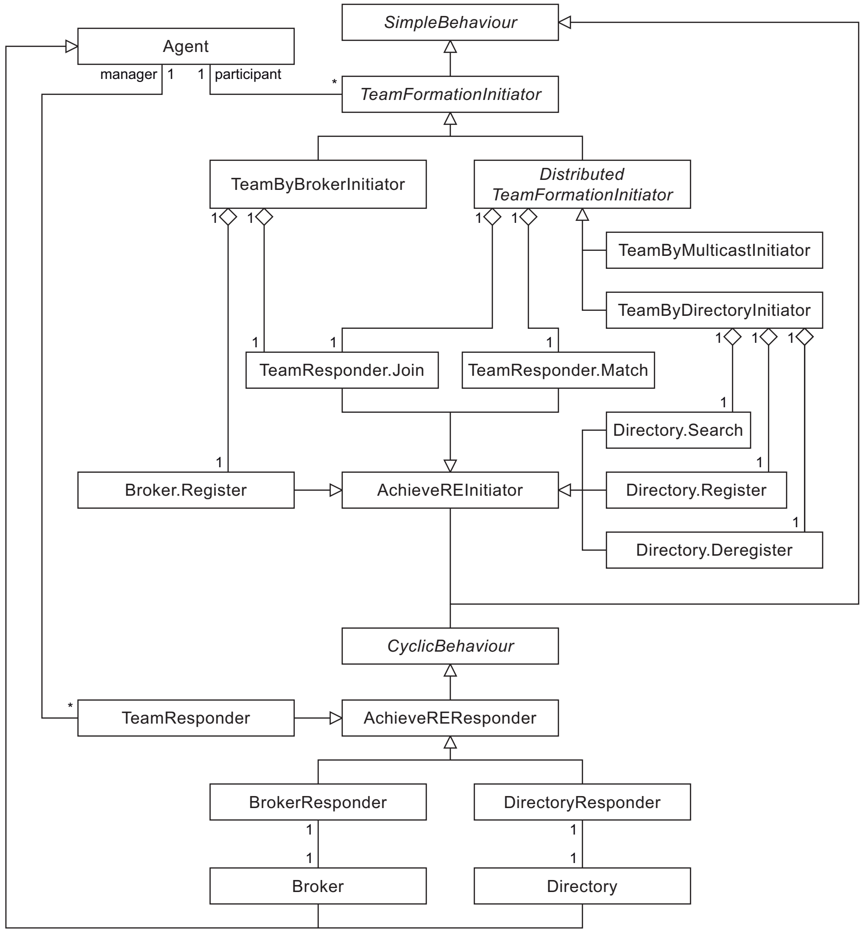

Fig. 8.3 UML class diagram of software agents and their behaviours related to team formation. In particular, this also includes the broker and the directory service. Besides, agents may have additional behaviours.

the directory-based and the multicast-based protocol have in common that decision-making is distributed to the local entities. Therefore, the initiators of both protocols share a joint abstract base class for distributed team formation. The distributed team formation behaviour has two sub-behaviours. One for determining matches with existing teams, another for joining existing teams. The derived initiator for multicast-based team formation (Section 6.2.3) has no additional sub-behaviours because its communication with existing teams is based on multicast messages. The directory-based initiator (Section 6.2.1) has sub-behaviours for registering and deregistering with the directory as well as for searching the directory. All sub-behaviours are derived from the general FIPA request initiator (Section 8.1.1). Both the directory-based and the multicast-based protocol implementations are capable of optimistic or conservative behaviour.

During the course of a team formation interaction protocol, the initiating agent may itself become a team manager. This can be reflected by applying the team responder behaviour. This behaviour is derived from the FIPA request responder behaviour. It waits for match and join requests by other agents and process them correspondingly.

8.1.3 Implementation of Team Action

The implementation of team formation (Section 8.1.2) lays the foundation for team action in autonomous logistics. This section approaches the implementation of team action in two steps. Firstly, the implementation of logistics service providers (Chapter 5) is described. Secondly, the interaction with service consumers (Chapter 7) is approached.

The UML class diagram of the logistics service provider implementation is depicted in Figure 8.4. For each primary logistics function (Section 2.1), an agent is derived from the agent base class that represents a respective service facility. These agents provide basic functionality and can be extended for specific purposes. Each service facility agent has a specific behaviour that is derived from the abstract service provider base class. This behaviour is actually responsible for providing services to service consuming agents. The service provider base class is implemented as a contract net responder which itself is a cyclic behaviour (Section 8.1.1). This means that all negotiations with other agents can be conducted based on the FIPA contract net interaction protocol (Section 4.2.4).

The service provider behaviour aggregates several interfaces which have to be implemented for concrete instantiations. These interfaces follow Definitions 5.9, 5.11, 5.12, and 5.13. Firstly, each service provider has at least one place, i. e., a location or transport relation, at which it offers its services (Definition 5.5). Furthermore, it has two to four descriptors which characterise the cargo types as well as the load carriers that can be handled (Definition 5.8). The interface of the descriptors already includes the mapping for descriptor matches. The interfaces duration, capacity, and costs define the corresponding mappings. A basic implementation of the capacity interface is in accordance with Algorithms 5.1 and 5.2. More sophisticated implementations can integrate existing software systems of the respective service providing company (Section 2.2.1). As elaborated in Section 5.1.4, storage service providers do not have a duration mapping. Hence, the multiplicity of this interface is either zero or one.

The classes implementing joint action of service consumers as well as their interaction with service providers (Chapter 7) are illustrated in Figure 8.5. As elaborated in Section 2.1, autonomous logistics particularly deals with

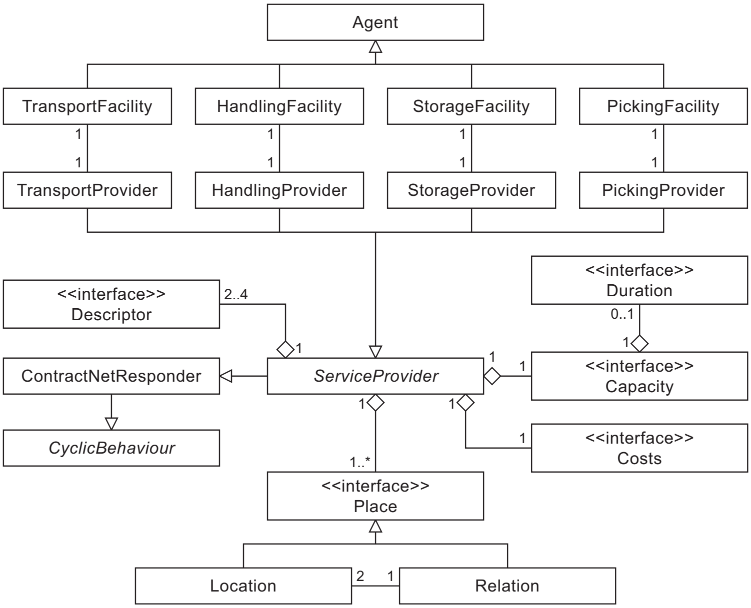

Fig. 8.4 UML class diagram of the software agents and behaviours representing service providers in autonomous logistics.

general cargo units (Definition 5.6) that are themselves responsible for satisfying their logistics objectives. Hence, the general cargo unit implementation is derived from the base agent class. General cargo units have an abstract service consumer behaviour. This class is a simple behaviour that is implemented based on a specific process at hand. It coordinates the execution of the primary logistics functions (Section 7.3.3).

The service consumer behaviour is composed of service team member subbehaviours. A concrete implementation of this abstract class exists for each of the primary logistics functions (Section 2.1). The service team member behaviour reflects that logistics resources are allocated in teams. Initially, a sub-behaviour initiates team formation for the respective primary logistics function (Section 8.1.2). Having identified a matching team, the logistics demands are transferred to the team manager by the service allocator behaviour. As described in Section 7.2.1, the service allocator is implemented as an initiator of the FIPA request interaction protocol. Its counterpart is the service team manager behaviour. This is an abstract cyclic behaviour that responds to requests for service allocation. Like for the service team manager, specific implementations for the primary logistics functions exist. The service team manager works in accordance with Algorithm 7.2 and applies a directory search to retrieve appropriate service providers. As specified in Algorithm 7.1, another sub-behaviour then negotiates with the respective service providers by means of the FIPA contract net interaction protocol.

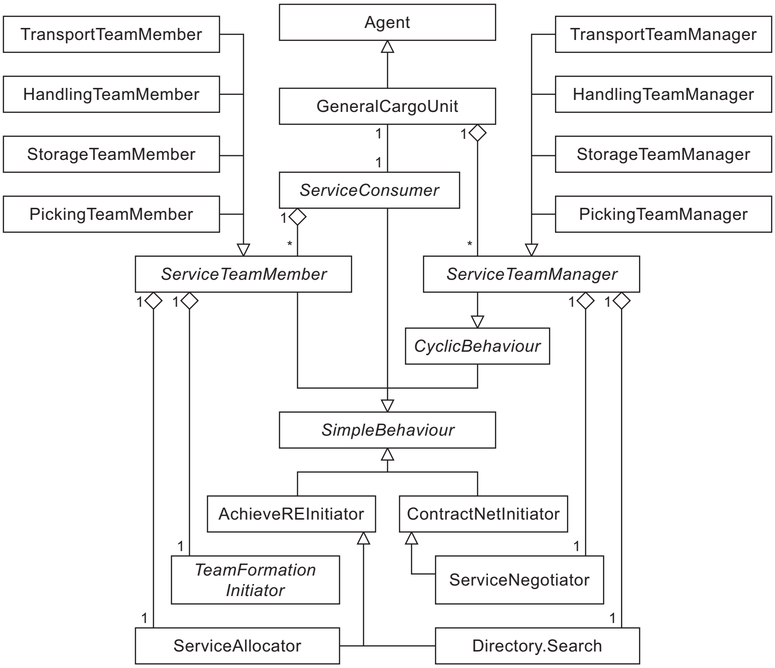

Fig. 8.5 UML class diagram of general cargo units as service consumers in autonomous logistics. The implementation covers service team members on the one and service team managers on the other hand.

Each general cargo unit may act as a service team manager for one or multiple primary logistics functions. A precondition is that no existing team for the specific purpose is discovered during the team formation phase. Note the relationship to Figure 8.3 in the preceding Section 8.1.2. The team responder behaviour is responsible for communication about team matching and joining. The service team manager behaviour takes the role of actually managing and satisfying the demands of the team members.

8.2 Multiagent-Based Simulation

Multiagent systems are a powerful means to implement autonomous logistics processes (Section 4.2). However, distributed processes in supply network management are often highly complex and highly dynamic (Section 2.3). Hence, it is generally impossible to predict runtime interactions between agents at design time (Jennings, 2001, p. 38). This means that also the outcome of such processes cannot be predicted analytically in advance. Therefore, like in the general software development process, testing and evaluation play an important role also in multiagent systems (Herrler & Klügl, 2006, p. 575). However, it is generally not desirable to test software systems in their actual deployment (Herrler & Klügl, 2006, p. 575). Firstly, it is quite expensive and time-consuming to test software in its real environment. This is particularly the case for distributed application scenarios like in logistics. Secondly, testing might compromise the integrity of actual processes which again leads to high costs. In order to avoid these problems, simulation is a common means to evaluate logistics strategies (Kuhn & Wenzel, 2008, pp. 73–94). Particularly multiagent-based simulation (MABS) is appealing. It applies the concept of multiagent systems to simulation (Herrler & Klügl, 2006, p. 579). Agents and their behaviour can be easily transferred (Parunak, Savit & Riolo, 1998, p. 21), which makes MABS a promising approach in order to examine them with minimal effort.

Approaches to simulation can be distinguished regarding their model granularity (Davidsson, 2000, p. 97). Macro simulation is generally based on equation-based modelling. This method models the world by a set of global variables. The values of these valuables are defined by differential equations that are evaluated during simulation (Parunak et al., 1998, p. 10). Multiagent-based simulation differs from such approaches. It pertains to the category of micro simulation. Micro simulation applies individual-based modelling. In this method, the behaviour of individuals is encapsulated by logical processes. Variables are assigned to individuals and thus also evaluated and validated on this level. The difference between macro and micro simulation can be explained by traffic simulation, e. g., the movement of pedestrians (Klügl & Rindsfüser, 2007). On the macro level, one would model pedestrians as streams defined by differential equations. The system behaviour is thereby derived from the average behaviour of individuals. In turn, this means that differing personal behaviours of individuals cannot be considered. This issue can be addressed on the micro level where each individual has its own behaviour. Simulating pedestrians pertains to the area of social simulation. In this field of application, software agents are employed to imitate the behaviour of humans. In this research, however, multiagent-based simulation is applied in order to test and evaluate the behaviour of software agents from real-world logistics applications in simulation.

Section 8.2.1 introduces the time model and the synchronisation mechanism applied to evaluate this research. Subsequently, Section 8.2.2 discusses additional requirements for synchronisation that are imposed by the application of agents as logical processes in simulation. Finally, Section 8.2.3 describes the simulation middleware applied in this project.

8.2.1 Time Model and Synchronisation Mechanism

Time plays an important role in simulation. Hence, it is necessary to distinguish the following notions of time (Fujimoto, 2000, pp. 27–28):

1. Physical time
2. Simulation time
3. Wallclock time

Their interrelationship is as follows. Physical time refers to the time of the real world, i. e., the time at which simulated events would happen in reality. Physical time is modelled in simulation by the so-called simulation time. Finally, wallclock time is the time that is consumed by the simulation system in order to execute the simulation. That is, wallclock time is measured in physical time.

In the real world, physical time progresses continuously. By contrast, time progression in multiagent-based simulation is discrete. Real-world events must thus be mapped to discrete timestamps in simulation time. One way to implement discrete time progression is time-stepped simulation (Herrler & Klügl, 2006, pp. 577–578). In this approach, physical time is mapped onto equidistant steps of simulation time, which are then computed one after another. Obviously, proceeding this way is only to a minor degree efficient. Time progression in equidistant steps means that also time steps without any events must be considered although the simulation state remains unchanged. By contrast, time progression is driven by events in discrete event simulation (Herrler & Klügl, 2006, pp. 577–578). In this approach, the simulation state is only updated when events occur. That is, the steps in simulation time bridge the gap between two successive events.

Time progression may differ for different agents. This can be motivated by the following reasons. In distributed simulations multiple logical processes, i. e., agents in MABS, run concurrently on different platforms or processors. These platforms may differ regarding their computational power. Furthermore, also the agents may differ regarding their computational demands, depending, for instance, on the events to be processed. Agents are generally executed as operating system threads. Therefore, even on one single CPU simulation platform simulation times of agents may diverge depending on their computational demands. Consequently, each logical process has its own local virtual time. Concurrency and thus diverging local virtual times do not pose a major problem as long as agents are independent from each other. However, causality problems (Fujimoto, 2000, p. 52) may arise because agents usually interact (Section 4.2). Interaction between agents with different local virtual times may lead to so-called straggler messages. Consider an agent A that passes a message m to agent B that is already advanced in its local virtual time. B has already made decisions that lie after the arrival of m. If B were aware of the message on time, it might thus have taken other decisions. To ensure the correctness of simulation results, it is therefore important that events are processed according to the order of their timestamps. For this purpose, synchronisation must be applied in order to handle diverging local virtual times. Synchronisation methods can be optimistic (Fujimoto, 2000, p. 97) or conservative (Fujimoto, 2000, p. 54).

Optimistic synchronisation does generally not restrict logical processes in their local virtual time progression. Agents may thus simply process their events without any constraints. The synchronisation mechanism comes into operation whenever a straggler message is received by an agent. Then, it is necessary to rollback the state of this agent to the local virtual time of the message received. An implementation of this time management approach is the time warp mechanism (Jefferson, 1985, pp. 409–420) with its cancelback protocol extension (Jefferson, 1990, pp. 80–86). Note that this also influences all other agents that have received messages from the respective agent in the meantime. Proceeding optimistically has the advantage of more efficient simulation execution. In particular, fast logical processes do not have to wait for slower ones. Time consumption is therefore a reason for optimistic synchronisation. This holds, at least, as long as the occurrence of straggler messages is limited.

However, optimistic synchronisation has potentially high requirements regarding memory (Fujimoto, 2000, p. 138). All past states of every agent must be stored, starting from the minimal local virtual time of all agents. In conventional parallel discrete event simulation, this generally means continuously storing several variables. In MABS, however, agents may possess extensive knowledge bases. These knowledge bases may change over time and must therefore be stored frequently. Space complexity may be reduced by introducing time windows for synchronisation (Lees, Logan & Theodoropoulos, 2004; Pawlaszczyk & Timm, 2006). Nevertheless, runtime performance may decrease significantly if state saving requires frequent and extensive input and output operations.

Throughout this research, however, conservative synchronisation with tree barriers is applied (Fujimoto, 2000, p. 67–68). In contrast to optimistic synchronisation, conservative synchronisation methods constrain time progression in order to prevent causality problems from occurring. In this approach, agents must commit to send no further messages before a specified point in simulation time. This means that all events before this commitment can be processed safely. Time progression is therefore potentially slower with conservative synchronisation. The application of conservative synchronisation can be motivated by the fact that, in turn, also the memory requirements are significantly lower because it is not necessary to store past agent states. Davidsson (2000, p. 99) names two additional reasons for applying conservative synchronisation. On the one hand, if the simulation incorporates human interaction, some processes may have to wait for user input. It is then not desirable that other processes advance arbitrarily far in time. On the other hand, also visualisation is an argument for conservative synchronisation. If the simulation has a visualisation component, one generally aims at visualising the states of all processes at the same time.

Despite of the potentially restricted speedup, conservative synchronisation is therefore an adequate means in domains requiring for human interaction or monitoring. MABS usually incorporates a great number of agents. To recapitulate, interaction cannot be predicted at design time (Jennings, 2001, p. 38). Implementing conservative synchronisation would thus require each agent to continuously synchronise with all others. This is obviously an overhead of synchronisation and thus not adequate. Instead, there is a demand for coordinated synchronisation control, e. g., provided by barrier synchronisation known from parallel computation.

8.2.2 Agent-Specific Message Handling Requirements

Agents are autonomous in their behaviour and thus also not restricted in message handling. That is, they can deliberately choose when to handle incoming messages from other agents. Messages may even be completely ignored. That is, messages in multiagent-based simulation are not just a simulation-specific representation of events. Instead, they are an integral part of the domain modelled, thereby representing the flow of information between agents (Gehrke, Schuldt & Werner, 2008, p. 548). Several general quality criteria exist that must be considered for simulation (Wenzel, Weiß, Collisi-Böhmer, Pitsch & Rose, 2008). Additionally, it is necessary to ensure also quality criteria especially for message handling in MABS (Schuldt, Gehrke & Werner, 2008, p. 110):

1. Time model adequacy
2. Causality
3. Reproducibility

It is worth mentioning that causality is sometimes also referred to as correctness, reproducibility is also referred to as repeatability. The following paragraphs give a short motivation for these requirements regarding message handling in MABS.

Regarding time model adequacy it is important to choose an appropriate granularity of time progression. Remember that time progression in MABS is discrete while physical time in real world is continuous (Section 8.2.1). Simulated events must therefore be mapped to certain timestamps in simulation time. On the one hand, a very fine-grained simulation time might lead to decreased runtime performance. On the other hand, a coarse granularity increases the artificial synchrony, i. e., events occur simultaneously in simulation although they would not in reality. A reasonable value for the minimal time progression must thus be chosen in advance (Gehrke et al., 2008, p. 549). It depends on both the domain modelled and the purpose of the simulation. Closely related to minimal time progression is another issue that is depicted in Figure 8.6. It shows two scenarios with agents A and B that act as logical processes in simulation. Both agents interact by exchanging a sequence of messages which might, for instance, be part of an interaction protocol like negotiating a contract (Section 4.2.4). The axes t represent the continuous physical time while the axes t_{sim} correspond to the respective discrete timestamps in simulation time. On the left hand side of Figure 8.6, multiple messages are exchanged at the same timestamp in simulation time although sending messages would consume physical time. Hence, the result of the communication process occurs earlier in simulation than it would in real world (Schuldt et al., 2008, p. 111). To prevent corruption of simulation results, transmitting each single message must at least coincide with a minimal time progression (Schuldt et al., 2008, p. 111). The right hand side of Figure 8.6 depicts the respective scenario with messages being mapped to different timestamps.

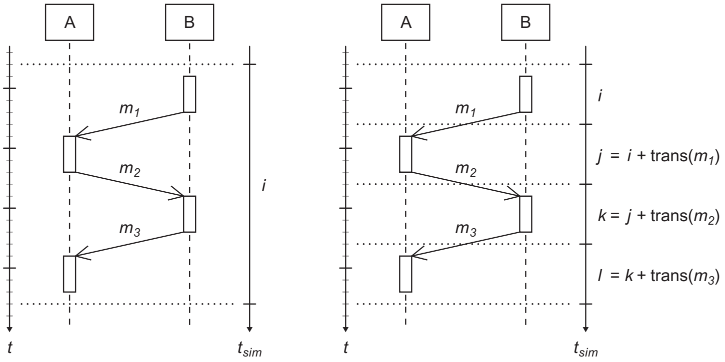

Fig. 8.6 If message passing does not consume time (left), results of communication processes like negotiations might occur earlier in simulation than it would be possible in real world. Therefore, different timestamps (right) must be assigned to the messages (Schuldt et al., 2008, p. 110).

However, even if it is assured that message passing consumes simulation time, it might occur that messages arrive early (Gehrke et al., 2008, p. 551). An example is given in Figure 8.7. Again, two scenarios with two agents A and B are depicted. At timestamp i, agent A passes a message m to agent B. The order in which agents are executed depends on operating system thread scheduling (Tanenbaum, 2001, pp. 132–153). Simulation systems can generally not influence this order. Hence, also the timestamp at which m arrives at B depends on scheduling. If A is scheduled after B (Figure 8.7 left), the message arrives in the message inbox of B at timestamp j. This is in accordance with the adequacy requirement. But if A is scheduled before B, the message might already arrive at the same timestamp i. In classical parallel discrete simulation, this would not pose a major problem. Processing early messages is simply deferred until the local virtual time arrives at the timestamp intended. In MABS, however, a causality problem might arise. Remember that agents are autonomous regarding their message handling. That is, an agent might also access such early messages in its inbox. The agent is then aware of information that it would receive in the future in the real world. In order to ensure causality, it is thus important to constrain the visibility of these messages (Schuldt et al., 2008, p. 112). An example is depicted by message m' on the right hand side of Figure 8.7.

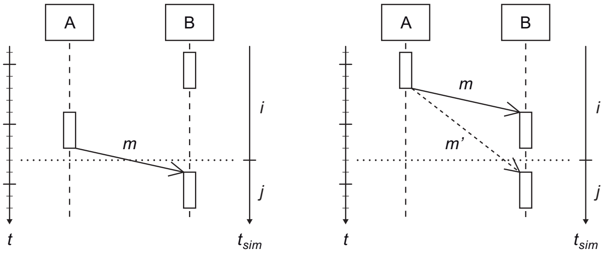

Fig. 8.7 Due to their autonomy in accessing the message inbox, agents could process messages early. In order to ensure the causality constraint, message visibility has to be controlled (Schuldt et al., 2008, p. 111).

Ensuring the causality criterion prevents messages from being visible early. Nevertheless, this does not guarantee reproducible results (Gehrke et al., 2008, p. 551). An example is illustrated by the two scenarios in Figure 8.8. Two agents A and C send a message to agent B, m_A and m_C respectively. The ordering of the messages in the inbox of C depends still on scheduling of the operating system. If A is scheduled before C the position of m_A is before m_C and vice versa. Note that this does not affect simulation result accuracy. But one is generally also interested in simulation results that are repeatable over multiple simulation runs. Reproducibility allows traceability and analysis of occurring effects and modelling errors (Schuldt et al., 2008, p. 112). For this purpose, Schuldt et al. (2008, p. 112) propose to introduce an additional ordering criterion besides arrival time of messages. This can, for instance, be the unique agent identifier of the sender. In order to prevent a bias in simulation results, it is important to choose attributes for ordering carefully (cf. Fujimoto, 2000, pp. 84–86).

To summarise, the autonomy of intelligent software agents distinguishes multiagent-based simulation from other types of simulation. This property turns messages from a simulation-specific representation of events into an integral part of the domain modelled. It is necessary to consider additional requirements regarding message handling because they represent the information flow between agents. These requirements are addressed by three quality criteria, namely time model adequacy, causality, and reproducibility. These criteria constrain how agents can access and process messages.

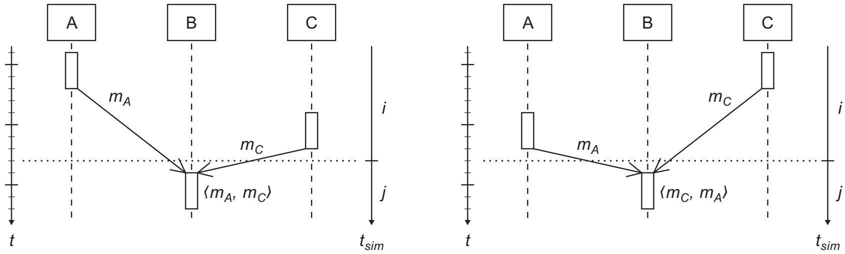

Fig. 8.8 Reproducibility problem of inbox queue order. The ordering of messages depends on the operating system scheduling of the respective senders (Schuldt et al., 2008, p. 112).

8.2.3 Middleware for Multiagent-Based Simulation

Several environments for multiagent-based simulation exist. Overviews of respective systems are provided by Klügl (2001, pp. 99–106) as well as Herrler and Klügl (2006, pp. 581–584). Many systems are intended for social simulation. By contrast, the focus here is on evaluating software agents and their interaction. That is, software agents must be transferred from real-world application to simulation and vice versa. This is in contrast to social simulation systems which investigate the behaviour of, for instance, human beings. For that purpose, agents are only observed. However, their behaviour is usually not transferred back to humans.

Within the scope of this research, software agents are implemented with the JADE multiagent platform (Section 8.1.1). Therefore, it is particularly desirable that software agents implemented with JADE can be easily evaluated in simulation. This purpose is addressed by PlaSMA (Gehrke & Ober-Blöbaum, 2007, p. 416), which stands for platform for simulations with multiple agents. PlaSMA is a simulation middleware that extends JADE with means for event-driven multiagent-based simulations. It is in accordance with the quality criteria for message handling discussed in Section 8.2.2. Furthermore, it provides a statistics library for pseudorandom number generation (PRNG).

Software agents can be transferred easily between JADE and PlaSMA. Only minor modifications of the source code are required regarding synchronisation. In particular, agents must explicitly request time progression. It is, however, intended that synchronisation is conducted implicitly in the future without burdening agents and their programmers. First steps towards this uniform agent design for simulation and operation (Figure 8.9) exploit knowledge about agent interaction for implicit synchronisation (Gehrke & Schuldt, 2009, pp. 1175–1176).

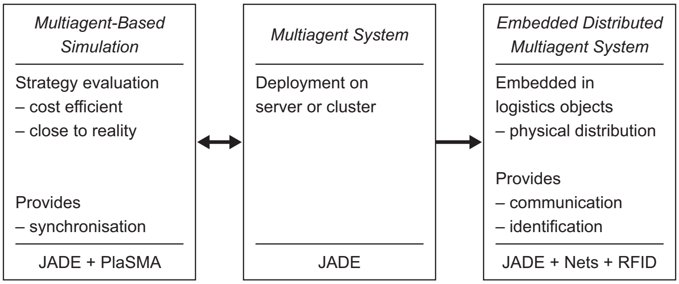

Fig. 8.9 The idea of uniform agent design envisions transparent transfer of agents between simulation and operation. The next step is to release software agents from a server to physically distributed real-world application.

8.3 Conclusion

Autonomous logistics envisions that process control in logistics is delegated to formerly inanimate logistics objects. Agent technology is an appropriate means to implement decentralised supply network management. For this research project, JADE has been chosen as the underlying multiagent platform. JADE is one of the most widespread agent platforms that is in accordance with FIPA standards which ensure interoperability even over company boundaries. The actual agent implementation follows the specification from the preceding chapters. It is divided into two parts, one of them focusing on team formation, the other on team action in autonomous logistics. The implementation constitutes an abstract framework that does not model a specific process. Instead, it provides the means to model a broad variety of autonomous logistics processes. To demonstrate its applicability, it is therefore important to apply it in a specific industrial application.

It is desirable to evaluate the likely outcome of autonomous logistics processes even before releasing the respective software agents to their actual deployment. This is addressed by multiagent-based simulation which allows testing the actual system behaviour by transferring software agents between simulation and operation. The quality criteria of adequacy, causality, and reproducibility should by satisfied by a MABS system. Within this project, PlaSMA is the chosen simulation middleware for JADE because it is in accordance with the aforementioned criteria.

References

Adorni, G., Bergenti, F., Poggi, A. & Rimassa, G. (2001). Enabling FIPA Agents on Small Devices. In M. Klusch & F. Zambonelli (Eds.), *5th International Workshop on Cooperative Information Agents (CIA 2001)* (pp. 248–257). Modena, Italy: Springer-Verlag.

Bellifemine, F., Caire, G. & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE.* Chichester, UK: John Wiley & Sons.

Bergenti, F. & Poggi, A. (2001). LEAP: A FIPA Platform for Handheld and Mobile Devices. In J.-J. C. Meyer & M. Tambe (Eds.), *8th International Workshop on Intelligent Agents (ATAL 2001)* (pp. 436–446). Seattle, WA, USA: Springer-Verlag.

Booch, G., Rumbaugh, J. & Jacobson, I. (2005). *The Unified Modeling Language User Guide* (2nd ed.). Amsterdam, The Netherlands: Addison-Wesley Longman.

Davidsson, P. (2000). Multi Agent Based Simulation: Beyond Social Simulation. In S. Moss & P. Davidsson (Eds.), *2nd International Workshop on Multi-Agent-Based Simulation (MABS 2000)* (pp. 97–107). Boston, MA, USA: Springer-Verlag.

Fujimoto, R. (2000). *Parallel and Distributed Simulation Systems.* New York, NY, USA: John Wiley & Sons.

Gehrke, J. D. & Ober-Blöbaum, C. (2007). Multiagent-Based Logistics Simulation with PlaSMA. In R. Koschke, O. Herzog, K.-H. Rödiger & M. Ronthaler (Eds.), *37th Annual Conference of the German Society for Computer Science (GI 2007)* (pp. 416–419). Bremen, Germany: Gesellschaft für Informatik.

Gehrke, J. D. & Schuldt, A. (2009). Incorporating Knowledge about Interaction for Uniform Agent Design for Simulation and Operation. In K. S. Decker, J. S. Sichman, C. Sierra & C. Castelfranchi (Eds.), *8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)* (pp. 1175–1176). Budapest, Hungary: International Foundation for Autonomous Agents and Multiagent Systems.

Gehrke, J. D., Schuldt, A. & Werner, S. (2008). Quality Criteria for Multiagent-Based Simulations with Conservative Synchronisation. In M. Rabe (Ed.), *13th ASIM Dedicated Conference on Simulation in Production and Logistics (ASIM 2008)* (pp. 545–554). Berlin, Germany: Fraunhofer IRB Verlag.

Herrler, R. & Klügl, F. (2006). Simulation. In S. Kirn, O. Herzog, P. Lockemann & O. Spaniol (Eds.), *Multiagent Engineering: Theory and Applications in Enterprises* (pp. 575–596). Heidelberg, Germany: Springer-Verlag.

Jefferson, D. R. (1985). Virtual Time. *ACM Transactions on Programming Languages and Systems*, *7*(3), 404–425.

Jefferson, D. R. (1990). Virtual Time II: Storage Management in Conservative and Optimistic Systems. In C. Dwork (Ed.), *9th Annual ACM Symposium on Principles of Distributed Computing (PODC 1990)* (pp. 75–89). Quebec City, Canada: ACM Press.

Jennings, N. R. (2001). An Agent-Based Approach for Building Complex Software Systems. *Communications of the ACM*, *44*(4), 35–41.

Klügl, F. (2001). *Multiagentensimulation — Konzepte, Werkzeuge, Anwendung.* Munich, Germany: Addison-Wesley.

Klügl, F. & Rindsfüser, G. (2007). Large-Scale Agent-Based Pedestrian Simulation. In P. Petta, J. P. Müller, M. Klusch & M. Georgeff (Eds.), *5th German Conference on Multiagent System Technologies (MATES 2007)* (pp. 145–156). Leipzig, Germany: Springer-Verlag.

Kuhn, A. & Wenzel, S. (2008). Simulation logistischer Systeme. In D. Arnold, H. Isermann, A. Kuhn, H. Tempelmeier & K. Furmans (Eds.), *Handbuch Logistik* (3rd ed., pp. 73–94). Heidelberg, Germany: Springer-Verlag.

Lees, M., Logan, B. & Theodoropoulos, G. (2004). Time Windows in Multi-Agent Distributed Simulation. In *5th EUROSIM Congress (EUROSIM 2004).* Paris, France.

Parunak, H. V. D., Savit, R. & Riolo, R. L. (1998). Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In J. S. Sichman, R. Conte & N. Gilbert (Eds.), *1st International Workshop on Multi-Agent Systems and Agent-Based Simulation (MABS 1998)* (pp. 10–25). Springer-Verlag.

Pawlaszczyk, D. & Timm, I. J. (2006). A Hybrid Time Management Approach to Agent-Based Simulation. In C. Freksa, M. Kohlhase & K. Schill (Eds.), *29th Annual German Conference on Artificial Intelligence (KI 2006)* (pp. 374–388). Bremen, Germany: Springer-Verlag.

Schuldt, A., Gehrke, J. D. & Werner, S. (2008). Designing a Simulation Middleware for FIPA Multiagent Systems. In L. Jain et al. (Eds.), *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2008)* (pp. 109–113). Sydney, Australia: IEEE Computer Society Press.

Tanenbaum, A. S. (2001). *Modern Operating Systems* (2nd ed.). Upper Saddle River, NJ, USA: Prentice Hall.

Weiß, G. & Jakob, R. (2005). *Agentenorientierte Softwareentwicklung.* Heidelberg, Germany: Springer-Verlag.

Wenzel, S., Weiß, M., Collisi-Böhmer, S., Pitsch, H. & Rose, O. (2008). *Qualitätskriterien für die Simulation in Produktion und Logistik.* Heidelberg, Germany: Springer-Verlag.

Chapter 9
A Case Study in Container Logistics

A concept for autonomous control of complex supply networks has been developed in Chapters 5 to 7. This abstract specification has been implemented in Chapter 8. But apart from theoretical considerations, it is also important to examine the application to real industrial logistics processes. Beforehand, it is necessary to investigate the status quo of supply network management in industry. Therefore, a case study has been conducted to examine the current procurement logistics processes of Tchibo (Schuldt, 2006). The following reasons motivate why the logistics of Tchibo is an adequate subject of this case study. Firstly, a high percentage of the suppliers is located in East Asia. As a consequence, the logistics department of Tchibo has to control a complex international supply network. Secondly, Tchibo supplies a great amount of outlets throughout Europe with a weekly changing range of products. This fact underlines the amount of goods that have to be procured and distributed, but also leads to high dynamics in logistics processes. The case study has been conducted in 2006 during a three-month internship at the forward logistics department of Tchibo. It is based on interviews with employees of this department.

Section 9.1 introduces the Tchibo company. It is supposed to motivate why this particular company is an appropriate choice for the practical examination. Therewith, this section lays the foundation for understanding the sophisticated logistics demands of Tchibo. Section 9.2 investigates the procurement logistics processes of Tchibo in more detail. The focus is on the containerised transport of goods which have been produced and purchased in East Asia. After their arrival in European ports, the shipping containers have to be transported to appropriate warehouses. Here, the picking is conducted which precedes the distribution into the outlets throughout Europe. Section 9.3 identifies and characterises the participating logistics entities.

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics,*
DOI 10.1007/978-3-642-20092-2_9,

9.1 Company Background

Tchibo is a major retailer of consumer products in Germany and its neighbouring countries. To understand the sophisticated logistics requirements of Tchibo it is necessary to learn more about the company. Section 9.1.1 starts with providing a brief historical overview of the company. It covers the time from 1949, the year in which Tchibo has been founded, until the beginning of 2006, the year in which the case study at hand has been conducted. Afterwards, the general concept of the range of products and their sales strategy is presented in Section 9.1.2. This section provides the foundations to understand the logistics requirements of Tchibo. Section 9.1.3 presents further general facts outlining the size of company, its structure, as well as its current market position.

9.1.1 Company History and Development

The history of Tchibo goes back to 1949 when the company has been founded by Max Herz and Carl Tchilling-Hiryan. They recognised the problem that roasted coffee demands for fast delivery to the end customer due to its highly limited shelf time. At that time, the available time-frame was too short for most of the established coffee traders. So, Herz and Tchilling-Hiryan decided to supply their customers with roasted coffee by mail order. The name of their company Tchibo is an abbreviation that derives from the combination of the first syllables of Tchilling and Bohne, whereby Bohne is the German word for (coffee) bean. About six years later, in 1955, the company extended its business and opened its first shop in Hamburg. Like the whole company at that time the shop was specialised in selling coffee. In particular, the shop offered its customers to taste the products of the company already before buying them for consumption at home. Until today, the total number of shops has grown to about 1,200. But this is not the total number of outlets to be supplied. In addition, Tchibo started in 1963 to place its products in bakeries and cake shops. This shop-in-shop concept was extended in 1987 to food trading outlets in general. Today, Tchibo supplies more than 56,000 outlets in total. They are spread all over Europe as Tchibo started the international expansion of its outlet business in the 1990s. More precisely, the outlets are located in German-speaking Europe, the Czech Republic, the Netherlands, Poland, as well as the United Kingdom.

In addition to its original coffee business, Tchibo extended its activities also to non-food products in 1973. In that year, the company started to offer its customers a limited range of consumer products. Actually, this idea has been derived from another concept that has been applied already in the first years of Tchibo. At that time, coffee was a high-priced product that was only sold in small amounts. Therefore, Tchibo decided to sell it in practical sideline products instead of standard paper bags. For example, kitchen towels pertained to this kind of packages. Later on, the idea was extended to selling other products related to coffee or kitchen. This concept bothered other German retailers in the 1970s. So they took legal measures which prevented Tchibo from further bundling coffee with non-food products. Instead of completely taking off from the non-food market, Tchibo decided to adapt its concept and started to sell coffee and non-food products separately. Today, the distribution of the non-food product line is no longer limited to the outlets. Instead, Tchibo revived its former mail order business in 1996. The German online shop, which has been established in 1997, and its successors in other European countries also pertain to this channel of distribution.

9.1.2 Range of Products and Sales Strategy

Apart from special offers, the range of coffee sold by Tchibo is permanently available in its shops. By contrast, for the non-food business another approach is applied. The range of consumer products changes on a weekly basis. According to Tchibo its concept of "a new experience every week" is globally unique. The products are always grouped together by particular topics, e. g., depending on the season. They might differ considerably in value, weight, and volume. The articles range for example from prepaid cards for mobile phones to jewellery, from pillow cases to tabletop football, as well as from cutlery to mattress. In general, only one occurrence of each article is offered. An exception from this principle are textiles as they are offered in different sizes. All occurrences of one article are internally grouped together into one project. Each of the weekly sales phases consists of about 30 projects. The total number of articles within one sales phase varies depending on the respective topic. All products sold by Tchibo are developed by product managers within the purchase department. This department is supported by trend scouts as well as market researchers in order to serve current trends. All articles are exclusively produced for Tchibo (mainly in East Asia) and sold under its own brand: TCM. This acronym stands for Tchibo Magazine[1] which is a weekly catalogue presenting the offers by Tchibo.

9.1.3 Company Structure and Key Figures

Tchibo claims to have an awareness level of about 100% in Germany. Taking the market share of 27% in the coffee business as a basis, Tchibo is the leading coffee roaster in Germany. With its worldwide market presence in

---

[1] Note that Tchibo changed the meaning to Tchibo Certified Merchandise in 2007.

the coffee roasting and exporting business Tchibo is number five worldwide. Besides, Tchibo retails its coffee as well as its non-food products in several countries throughout Europe. Apart from Germany, these countries include Austria, the Czech Republic, the Netherlands, Poland, Switzerland, as well as the United Kingdom (Figure 9.1). Except from the Czech Republic and Poland, Tchibo operates also e-business activities in these countries. Especially the German online shop is very successful as it is the second largest in Germany, directly behind the Amazon internet store and even ahead of Otto and KarstadtQuelle which are long-established in the mail order business. This follows from the number of visitors measured by Nielsen Netratings in February 2006 (Die Welt, 2006).

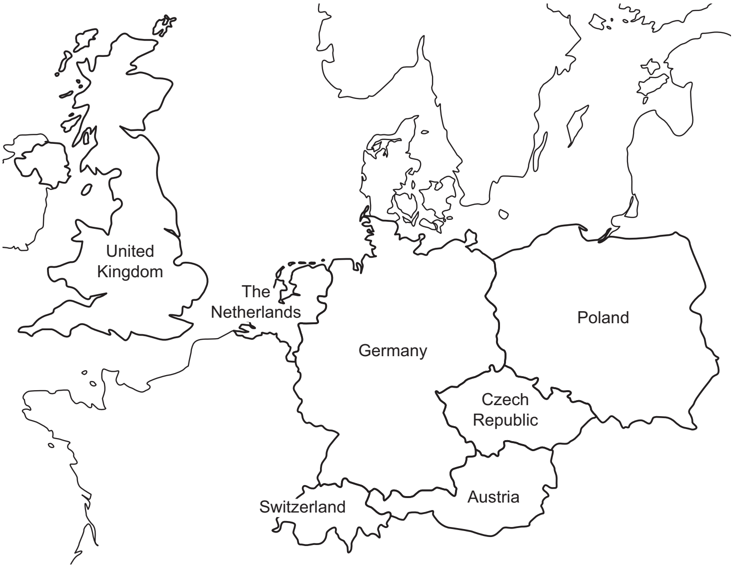

Fig. 9.1 European countries with outlets that are supplied by the logistics department. These countries include Austria, the Czech Republic, Germany, the Netherlands, Poland, Switzerland, as well as the United Kingdom.

Tchibo is structured as follows (Figure 9.2). Since 1988, the Tchibo Holding AG[2] acts as the parent company for many different companies. It is wholly owned by members of the family of Max Herz, one of the founders of the company. The original coffee and non-food business is subsumed within the Tchibo GmbH, which is a wholly-owned subsidiary of the holding. Apart from this company, the Tchibo Holding AG is involved in another major company, namely the Beiersdorf AG. Beiersdorf is a globally operating producer of beauty products. Tchibo started getting involved in Beiersdorf with a non-controlling interest in 1977. Since 2004, Tchibo owns 50.46% of the Beiersdorf

---

[2] Note that the company changed its name to maxingvest ag in 2007.

shares and therefore holds a controlling interest. In total, the Tchibo Holding AG had an annual turnover of about 8.8 billion euro in the year 2005 (Tchibo Holding AG, 2006). From this turnover about 4.0 billion euro have been earned within Tchibo GmbH. This is, for instance, comparable with the turnover figures of the department stores of Karstadt in the same year (KarstadtQuelle AG, 2006). This comparison illustrates the size of Tchibo. The number of employees of Tchibo GmbH lies at 12,796, the whole group employs 29,619 people (from which 16,783 belong to Beiersdorf and 40 to the holding).

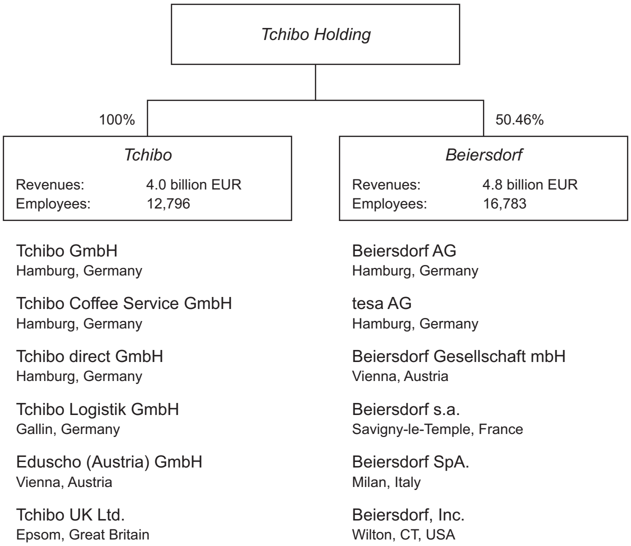

Fig. 9.2 The holding company subsumes two main branches. The first one is involved in the original coffee and non-food business. The second one is a producer of beauty products (Tchibo Holding AG, 2006).

Tchibo GmbH itself is the parent company to several subsidiaries. These enterprises include mainly companies in the coffee business as well as subsidiaries in foreign countries. From the point of view of logistics particularly Tchibo Logistik GmbH, a wholly-owned subsidiary, is of special interest. This company manages the logistics demands of Tchibo in general. It is paid per year for a fixed number of cargo handlings. Although the headquarters of Tchibo is located in Hamburg, the administration of the logistics department is located in Bremen. This fact is related to the takeover of the Eduscho group, one of the former major competitors in the German coffee business. Because Eduscho operated its logistics department at its headquarters in Bremen, this was chosen to become also the new site for the joint logistics department.

9.2 Procurement Logistics Processes

Subsequent to the general company information, this chapter deals with the procurement logistics processes that are applied in order to implement the concept of Tchibo. As an introduction, the general structure of the supply network is examined in Section 9.2.1. Subsequently, the process of transporting goods from East Asia to Europe is examined in Section 9.2.2. The onward carriage to warehouses is described in Section 9.2.3. In order to represent the logistics processes, the ARIS (Seidlmeier, 2006) representation of event-driven process chains (EPC) is applied.

9.2.1 Supply Network Reorganisation

Tchibo has to deal with a huge amount of suppliers from East Asia. Through the ports in East Asia, they are connected to Europe by several carriers. In the past, each vendor applied his own carrier for the transport to his preferred port of discharge in Europe (see left hand side of Figure 9.3). From here, transport continued to a number of warehouses that store the goods until their distribution into the outlets. This organisation lead to many transport relations between the participating entities.

During a fundamental reorganisation of its logistics processes Tchibo has tightened the structure of its supply network by reducing the number of participating entities and thereby the number of transport relations between them. Today, Tchibo confines itself to a limited number of preferred carriers. In order to ensure low prices and a good quality of transport a framework agreement was concluded. Thereby, each participating carrier is guaranteed to receive a fixed percentage of the shipping assignments by Tchibo. In return, the preferred carriers provide appropriate prices and the quality of transport that is required by Tchibo. Out of the general logistics objectives (Chapter 2), it is particularly important that scheduled dates are met. This is due to the number of containers that changes from week to week. However, transport is not limited to those preferred carriers because they serve several routes together with partnering companies (Wood, Barone, Murphy & Wardlow, 1995, pp. 106–126). Likewise, the number of ports for discharge of cargo has been limited to Bremerhaven and Hamburg in Northern Germany. Both of them are well connected to the most significant innovation within the reorganised supply network: the new logistics centre at the Neustädter Hafen (LCNH) in Bremen, which has been established in 2003. It serves as a central high-bay warehouse instead of many smaller ones before. It is the focal entity of the supply network where all transport relations are concentrated. However, this strict concept does only hold in theory as in practice not all articles can actually be stored in the high-bay warehouse.

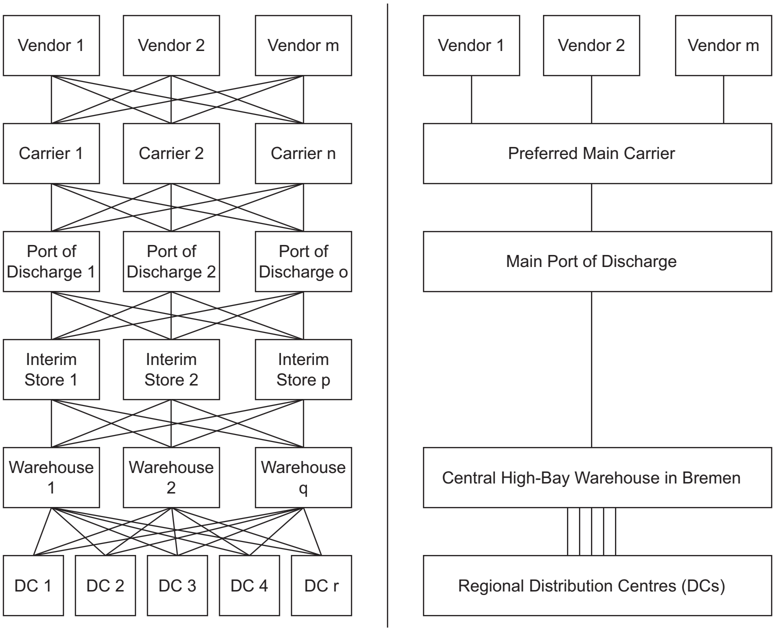

Fig. 9.3 The former (left) and current (right) organisation of the supply network. Today, the central warehouse is the focal entity of the supply network (adapted from Tchibo GmbH, 2004, p. 8).

Additionally, Tchibo has contracts with several other warehouses which can be chosen alternatively. The LCNH is owned and operated by the BLG Logistics Group AG & Co. KG (BLG). At the time this case study has been conducted, the capacity of the LCNH was extended to over 200,000 pallets by adding a third warehouse slot (BLG in.add.out Logistics GmbH & Co. KG, 2006). In 2004, the logistics department of Tchibo won the German Logistics Award. This award was set up in 1984 by the German Logistics Association (BVL), a non-profit organisation characterising itself as an expert network of the German logistics sector. This award honours the logistics department for the successful supply network reorganisation (Tchibo GmbH, 2004).

Figure 9.4 shows a generalised organisational chart of the logistics department in 2006. As the focus of this case study lies on procurement logistics processes, the forward logistics branch of the company is depicted in more detail than the other parts. Starting from the top, the Tchibo Logistik GmbH is divided into two parts. One of them deals with operations and processes, while the other is responsible for business and project management. The first one is again divided into four parts. Forward logistics for non-food products, food and reverse logistics, as well as operations in the north/west and operations in the south/east. The two latter ones are concerned with managing the distribution centres employed. Compared to the forward logistics processes, that deal with supplying the outlets, the amount of goods handled by the reverse logistics division is much smaller. In order to obtain divisions of comparable in size, this part of the department is also concerned with distributing coffee, the original product of Tchibo. As illustrated in Figure 9.4, the forward logistics division itself consists of four sections. They include the processes that are related to containerised transport by sea from East Asia to Europe which are described in Section 9.2.2. A second part, that is discussed in Section 9.2.3, deals with the onward carriage to warehouses. Finally, a third part of the division is responsible for distributing the articles into the outlets. As this case study especially concentrates on containerised transport, the distribution is not further investigated here. The same holds for the last part which deals with process standards.

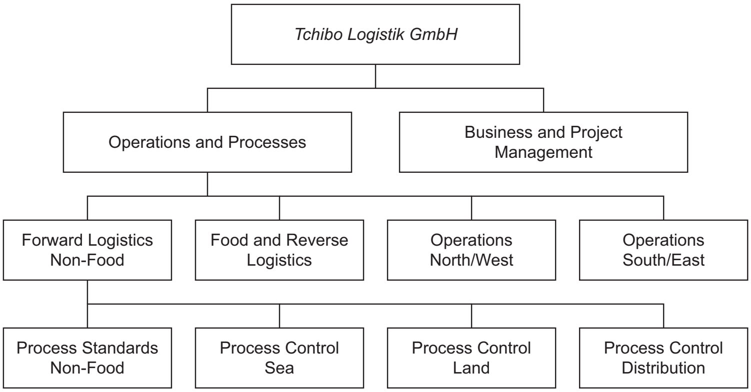

Fig. 9.4 A generalised organisational chart of the logistics department with a particular focus on forward logistics processes for non-food articles.

9.2.2 Transport from East Asia to Europe

About 70% of the non-food products by Tchibo are purchased on international markets, mainly in East Asia. The logistics department has to ensure that all products arrive in Europe on time. On time means in this context, that each product is supposed to be present in the shops in its scheduled main sales phase. To achieve this objective it is necessary to consider also time for the onward carriage from the port to the warehouse, for quality assurance, as well as for the subsequent distribution into the shops. Depending on the amount of goods of a project, its transport can be split into multiple shipments. This section investigates the processes that are handled by the sea forward logistics division in more detail. These processes include booking capacity on container vessels and, afterwards, managing the shipping itself.

Another task of the sea forward logistics division, that is not further examined, deals with handling the billing of shipment costs as well as customs clearance.

Booking Capacity on Container Vessels

In order to ship a container with a specific vessel, a booking has to be placed at least ten days before the estimated time of delivery, i. e., the date when goods are handed over in the port. Because it depends on the manufacturer when the goods are ready for shipping, he or rather his respective vendor in Germany has to initiate the booking (Figure 9.5). In order to support a central organisation by Tchibo, he is not allowed to place a booking directly at the carrier. Instead, he is supposed to send his booking to the booking agent that is employed by Tchibo. Each booking includes the respective project, the number of containers needed, as well as the port of loading. After having received this booking request, the booking agent starts with assigning a priority to each of the containers affected. The priority varies from one to three and depends on the time that is left until the start of the main sales phase. As soon as the vessel is chosen, the respective carrier may be provided with a space forecast. In this case he reserves space on the vessel chosen. Subsequently, a pre-advice is created by the booking agent and sent to the logistics department of Tchibo. Having received the pre-advice, the task of the logistics department is to validate it concerning several parameters. These parameters include the time, the number and type of containers, the ports of loading and discharge, as well as the priority and the carrier chosen (Figure 9.6).

Time is validated regarding the question whether the predefined time frame is met or not. If the delivery is late, the logistics department has to check whether or not the status is critical. Critical in this context means that it is in question whether or not the main sales phase can be met. If status is deemed critical, the purchase department has to decide whether a faster vessel or even a plane should be used. In this case it is also possible to choose a way of transport that does not belong to a preferred carrier. Whenever the vendor is responsible for the delay, the additional costs for faster transport may be passed down to him. Not only late delivery, but also early delivery can pose a problem because additional costs for warehousing will arise. If delivery is more than 30 days before the estimated time of delivery, the purchase department of Tchibo has to confirm the early loading. In this context it has to be decided whether Tchibo pays the additional costs of warehousing or passes on the charges to the vendor.

The second parameter to be validated is the amount of cargo that is shipped within one shipping. In particular, it has to be checked whether or not all versions of the article are available within one shipping. Furthermore, it is necessary to validate the container type chosen by the vendor. This task is supported by a software for load optimisation.

Fig. 9.5 The booking process is initiated by the vendor, prepared by the booking agent, validated by the logistics department, and finally placed by the booking agent again.

Afterwards, also the chosen ports of loading and discharge have to be validated. The port of discharge is of special interest for Tchibo. It is always either Bremerhaven or Hamburg in Germany. Which port is actually chosen depends on the type of cargo and, to a minor degree, on warehouse capacity. In general, all cargo that can be auto-palletised is stored in the high-bay warehouse in Bremen. The shortest and also cheapest way to Bremen is to discharge these containers in Bremerhaven and to transport them by barge directly to the high-bay warehouse. A second, but considerably more expensive way of transporting goods from Bremerhaven to Bremen is by truck. All cargo that is not auto-palletisable is routed to the port of Hamburg. From here, the cheapest way of transporting goods to Bremen is by train. Alternatively, trucking is also an option.

Additionally, it has also to be checked whether all containers are assigned a correct priority. Finally, the logistics department validates whether or not the chosen carrier belongs to the group of preferred carriers. Having received the confirmation by the logistics department, the booking agent places the booking at the carrier. As soon as the carrier has confirmed the booking, the booking agent notifies the vendor about the scheduled departure of the vessel booked.

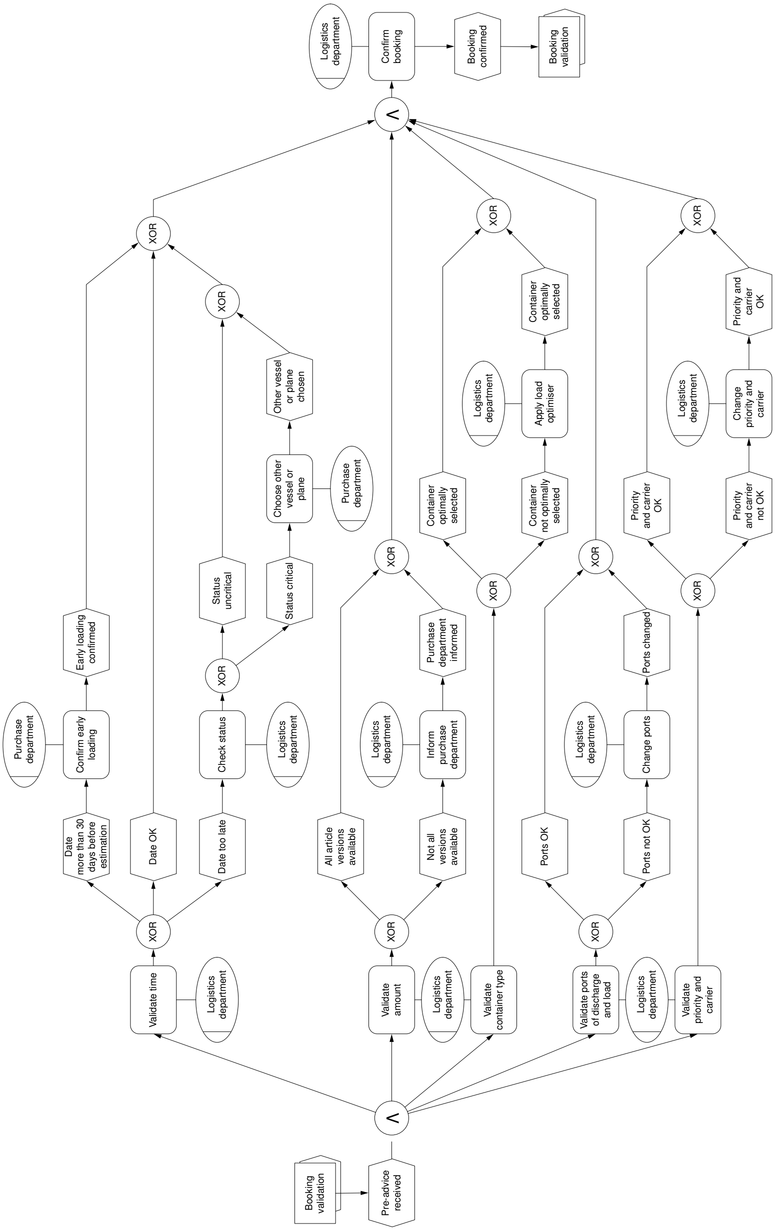

Fig. 9.6 Each pre-advice has to be validated by the logistics department. Validated parameters include the time, the number and type of containers, the ports of loading and discharge, as well as the chosen priority and carrier.

Shipping Containers to Europe

As soon as the cargo is ready for shipping, the shipping process itself starts. As depicted in Figure 9.7 this process is initiated by the manufacturer, who checks whether or not the scheduled vessel can be reached. If he realises, that he will miss the selected vessel, the booking agent has to be informed by sending him a new suggestion, so that the booking agent can start a new booking process. In this case, the current shipping progress is finished temporarily. Otherwise, i. e., if the vessel can be reached, the containers have to be prepared for shipping (Figure 9.8). This subprocess includes providing empty container equipment which is done by the booking agent. The pickup of the empty containers can either be accomplished by the manufacturer himself or be delegated to the booking agent. In the latter case the fee for this service has to be paid by the manufacturer or the vendor.

When the empty containers are available, the manufacturer can start filling them with his goods. Some products demand to be shipped in fumigated containers. This is the case, for instance, when transporting wooden products that have to be protected from bugs. Whenever a container is fumigated it has to be marked accordingly. Proceeding this way ensures that it is ventilated after its arrival in Europe.

After the preparation of the containers is finished they are delivered to the port either by the manufacturer or by the booking agent. Besides the purchased products themselves it is obligatory for the vendor to deliver spare cardboard boxes and replacement units for each project. Spare cardboard boxes are used in the case of transport damages. They are applied whenever only the package of a product is broken while the product itself remains undamaged. Replacement units are articles that are interchanged as a whole in the warranty case, parts which may be missing in the cardboard box of a product, as well as parts that are necessary to repair damaged products. After their arrival, Tchibo delivers the replacement units to a service centre which is responsible for the whole process of warranty handling.

After the containers have arrived in the port it is the task of the carrier to load them on the scheduled vessel (Figure 9.9). In order to do so, he has to check whether or not there is actually enough space on the vessel. Sometimes it may happen that not enough space is available. In these cases the priority of the affected containers has to be checked. If their priority is hot or even very hot, it is important that they arrive on schedule in order to meet the planned main sales phase. In this case, containers have nevertheless to be loaded, e. g., by unloading other ones. Otherwise, if the containers have a normal priority, they can be rolled to the next vessel. Whenever containers are rolled to another vessel, a black list has to be transmitted to the booking agent, who in turn informs the logistics department. In order to confirm the new vessel it is again validated by applying the same criteria as before. In the context of loading the freight onto the vessel it is worth taking a short look at the relevant selling terms. In the context of Tchibo, the Incoterms 2000,

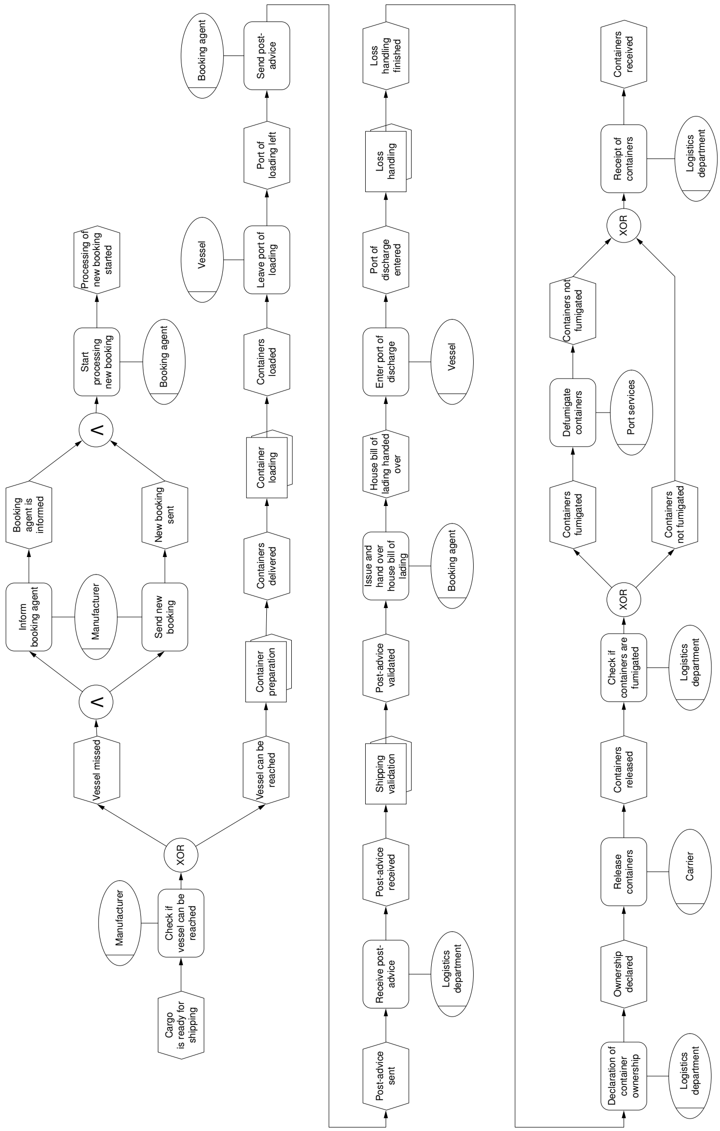

Fig. 9.7 The process of shipping containers from East Asia to Europe. The process chain starts with the preparation of the containers by the vendor. It ends with the arrival in Europe.

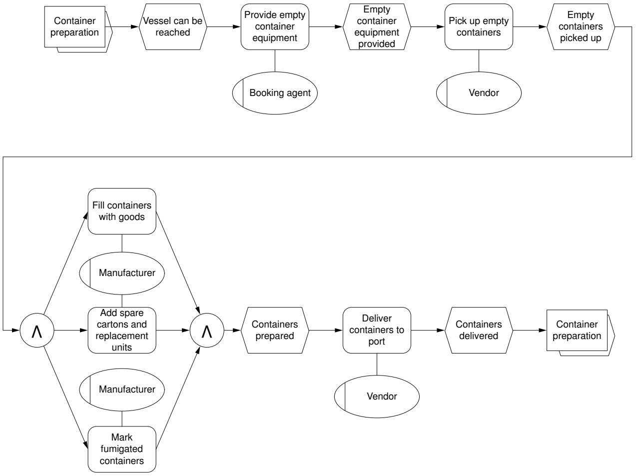

Fig. 9.8 The process of container preparation includes ordering empty containers, filling them, as well as delivering them to the port of loading.

which have been defined in the year 2000 by the International Chamber of Commerce, are of interest. Especially, the FOB term is applied:

> "**Free on Board (. . . named port of shipment).** Free on Board means that the seller delivers when the goods pass the ship's rail at the named port of shipment. This means that the buyer has to bear all costs and risks of loss of or damage to the goods from that point. The FOB term requires the seller to clear the goods for export. This term can be used only for sea or inland waterway transport. [...]" (DHL Express Vertriebs GmbH & Co. OHG, 2005, p. 697)

Besides, Tchibo applies the so-called Tchibo Term (TCT) which extends FOB with some Tchibo-specific agreements. These extensions address the responsibility of the vendor for any quality loss during transport, e. g., due to rust or mould. In return, Tchibo agrees to perform the product testing within ten days after delivery. Independently from the term chosen, the booking agent creates a post-advice from the already existing pre-advice shortly after the vessel has left the port of loading. This again has to be validated by the logistics department in order to synchronise it with its own system. In doing so, the notes of the booking agent, article availability, container types, and the number of containers are checked (Figure 9.10). Additionally, it is validated whether or not spare cardboard boxes and replacement units have also been shipped.

After the post-advice has been validated by the logistics department the booking agent issues the house bill of lading and hands it over to the vendor. The bill of lading represents a substitute for the actual goods in the exchange process (Wood et al., 1995, p. 246). The goods covered by a bill of lading are owned by the consignee that is named in the document. The vendor hands over the bill of lading to his bank in order to get paid for the goods.

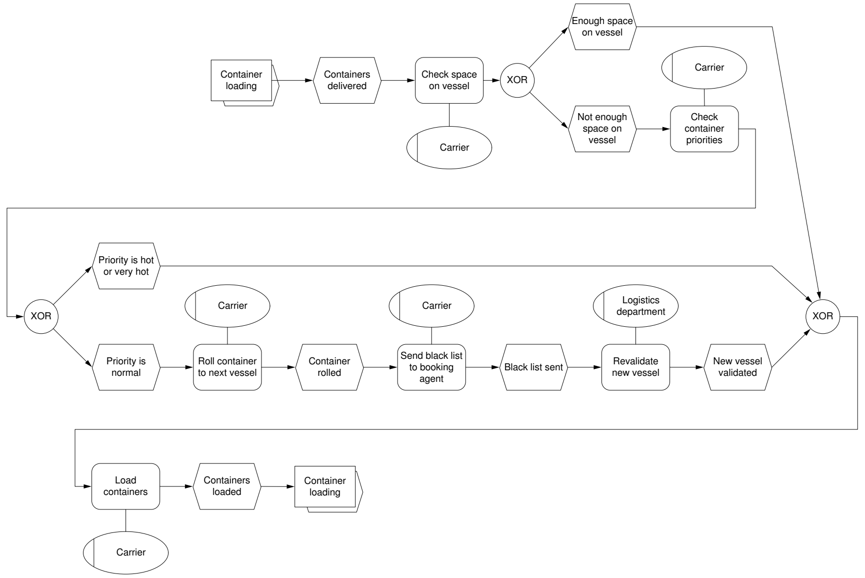

Fig. 9.9 During the process of container loading, the carrier has to determine whether there is enough space on the vessel.

Subsequently, the bank sells the bill of lading to Tchibo. After the vessel has arrived at the port of discharge, the logistics department needs the bill of lading in order to declare its ownership of the shipped containers. Therefore, the bill of lading is handed over to the carrier, who releases the containers in turn. Before the containers can finally be received by the logistics department, it has to be checked whether or not they have to be defumigated before further handling.

Unfortunately, it is not always ensured that all containers which have been loaded on a vessel in East Asia actually arrive in Europe. Due to manifold reasons, e. g., bad weather, it may happen that containers get lost during the transport. Additionally, it may also be necessary to unload containers on high sea by throwing them overboard in order to prevent vessel and crew from further damage (DHL Express Vertriebs GmbH & Co. OHG, 2005, p. 361). In each case the carrier has to inform the logistics department about the loss of containers which, in turn, informs its insurance company (Figure 9.11).

9.2.3 Onward Carriage to Warehouses

Subsequent to their arrival in European ports, the containers await their onward transport. This section describes how the onward carriage to warehouses is managed by the land forward logistics division. This task is influenced by

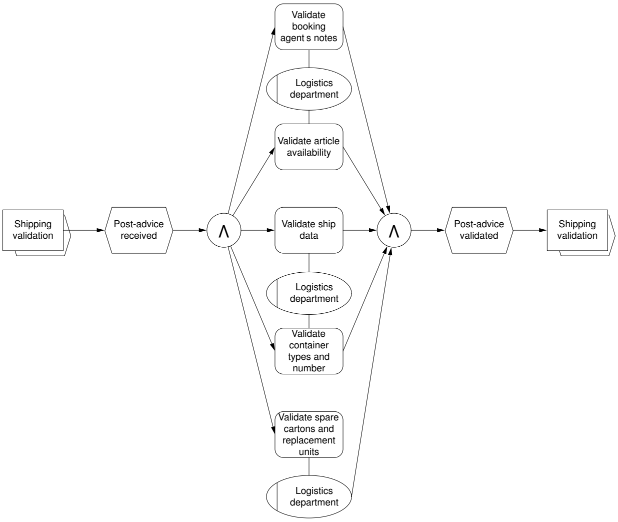

Fig. 9.10 As soon as a post-advice has been received, the logistics department validates the actual shipping parameters in order to synchronise them with its own system.

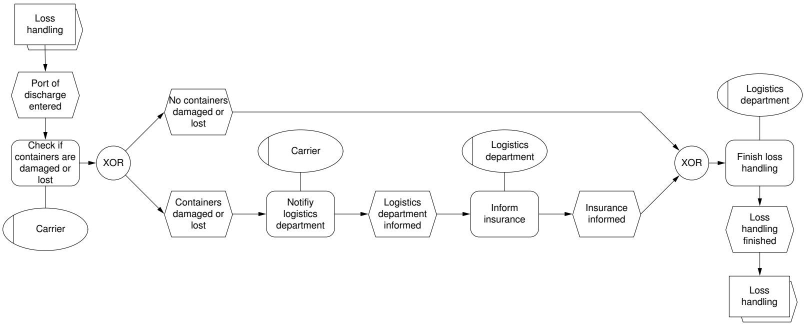

Fig. 9.11 The carrier has to inform the logistics department in case that containers are lost. In turn, the insurance company is informed by the logistics department.

a set of numerous parameters. For instance, the nature of the article itself affects the choice of the warehouse as not all warehouses are capable of receiving all types of articles. Additionally, warehouse capacity has to be taken into consideration. Finally, it has also to be decided how a container is supposed to be transported to its respective warehouse. Especially, the fluctuating number of containers per week is challenging. As for the transport by vessel this number directly influences the transport capacity required. In this context, it is worth mentioning that the logistics department is mainly responsible for assigning containers to warehouses and transport relations. The transport itself is executed by an external service provider. In accordance with the general understanding of logistics, the focus of the logistics department is on planning and controlling processes rather than executing the respective operations (Chapter 2).

Dispatch for Onward Carriage

The dispatch process (Figure 9.12) is conducted everyday for all containers that have not yet been dispatched. It starts with checking the release status of the respective container (Figure 9.13). This is due to the fact that a container can only be handled if it has already been released by the carrier. In general, each container that has been handled by the sea forward logistics division should already be released. However, this does not hold for all containers. For instance, sometimes the house bill of lading does not arrive on time. As a consequence, the carrier cannot release the respective containers. Otherwise he would risk responsibility for any unauthorised release by the lawful owner of the container. If a bill of lading is not available on time, the logistics department has to check whether it is already on its way to Tchibo. Whenever the respective articles are needed urgently and it is estimated that the bill of lading will arrive soon, the purchase department can decide to deposit a security for such a container. This security covers the worth of the cargo and it allows the carrier compensating the actual owner if the container has been taken unauthorised. In this case, the respective container can be released immediately. By contrast, if the current location of the bill of lading is unclear, deposing a security coincides with a high risk. Then, container dispatch must be delayed.

In general, the quality assurance department examines samples of all articles at the earliest when they arrive in their respective warehouse. Apart from that, an increasing amount of goods is already pre-examined in East Asia by a subsidiary of Tchibo in Hong Kong. Therefore, it may already be known that major parts of a project will be rejected due to insufficient quality (Figure 9.14). In this case, the cargo is not routed to one of warehouses employed by Tchibo. By contrast, it is directly delivered to a selector. This service provider recovers the required quality of the concerned projects by removing broken articles or repairing damaged ones (Section 2.2.1). Following this process the cargo is returned to the warehouse. Subsequently, the logistics department has to update the amount of the respective articles as their number is generally decreased in this case.

After the completion of the release and rejection handling, the priority of the container is checked. If a container has only a low priority, e. g., because there remains some time until its respective main sales phase starts, it is delayed for later dispatch. All containers are dispatched manually with computer support. Hence, delaying low-priority containers allows concentrating on containers with high priority.

The most important question for each container is whether or not its content is of high value (Figure 9.12). All containers having high worth are di-

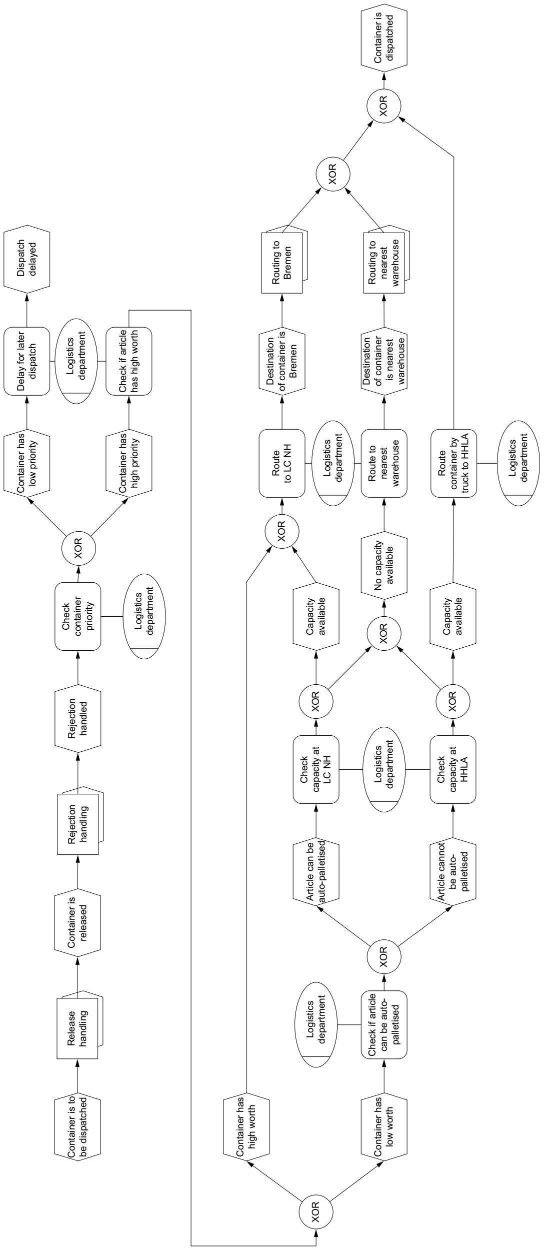

Fig. 9.12 Dispatch for onward carriage is conducted every day for all containers that are not already dispatched. It starts with handling the release of the container and generally ends with the container being dispatched.

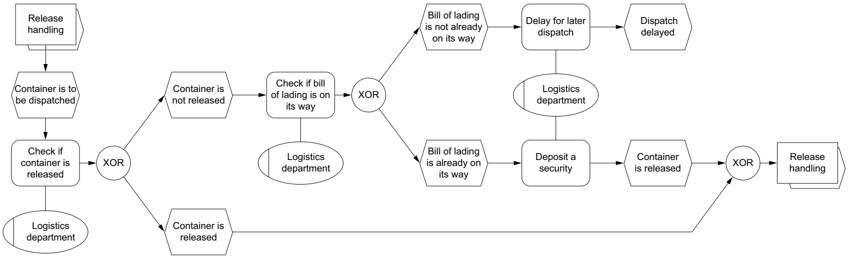

Fig. 9.13 For each container it has to be checked if it has been released yet. If a container is not yet released but the bill of lading is on its way, it is possible to deposit a security in order to handle it soon.

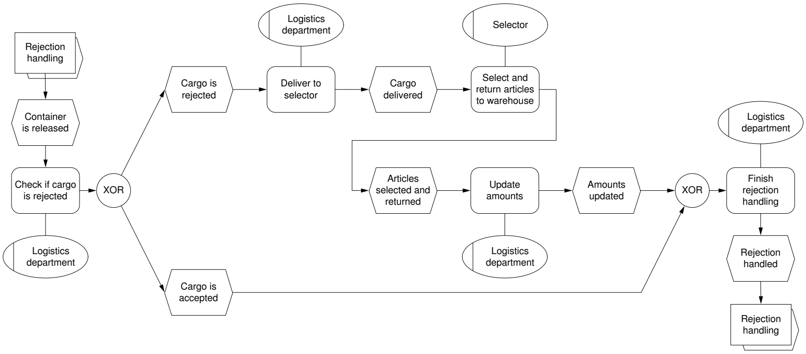

Fig. 9.14 If it is already known that major parts of a project will be rejected due to insufficient quality, the articles are generally directly delivered to a selector who recovers them.

rectly routed to the logistics centre at the Neustädter Hafen in Bremen. For all the remaining containers it is checked whether or not their content can be auto-palletised. This is the precondition for all articles that are stored in the central logistics centre. Because it is an automatised high-bay warehouse only goods that are placed on pallets can be handled. If the LCNH has free storage capacity the respective containers are routed there. Otherwise, they are routed to a warehouse that is near to the current location of the container. In general, the reason why an article cannot be auto-palletised is its size. In this case, the respective container is routed to a warehouse in Hamburg that is operated by the Hamburger Hafen und Logistik AG (HHLA). However, if this warehouse does not have enough capacity they are also routed to another warehouse nearby.

Whenever the destination of a container is set to Bremen (Figure 9.15), its current location has to be determined. This can either be Hamburg or Bremerhaven. If the container is located in Hamburg, it can be scheduled for train or truck transport. As one might expect, the train is the cheaper alternative. However, because trains are not always available it has to be checked,

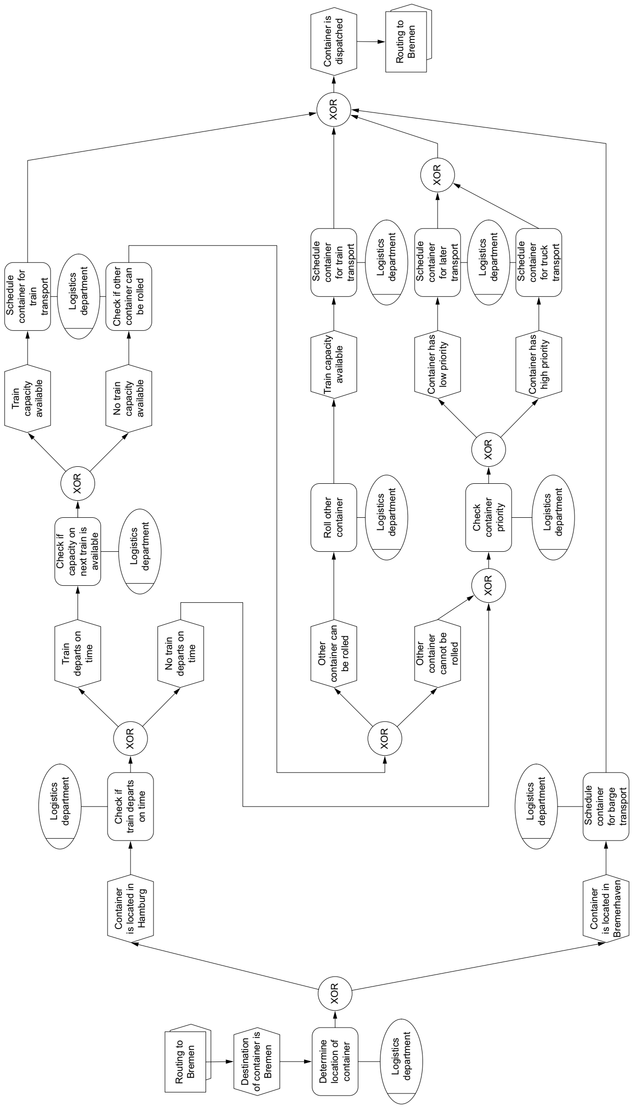

Fig. 9.15 There exist several possibilities for routing a container to Bremen. Which one is chosen depends, for instance, on its current location.

if a train departs on time. This depends, for instance, on the number of containers available. If there are not enough containers waiting for transport, ordering a train is too expensive. Furthermore, not all container terminals offer train departures on all days of the week. Even if an appropriate train has been found, it has to be checked whether or not it has capacity for further containers. Only if capacity is available, a container can be assigned to the respective train. As an alternative, it can be checked whether or not another container that is already assigned to the train has a lower priority. In this case it may be rolled so that capacity becomes available for the container to be assigned. Whenever there is currently no capacity the priority of the container has to be checked. If it has only a low priority it may scheduled for a later transport. Otherwise, it has to be transported by truck to Bremen.

Containers located in Bremerhaven have to be handled in a similar way. Bremerhaven is not connected by train to Bremen. By contrast, it is possible to transport containers by barge on the Weser river directly to the port of Bremen where the central high-bay warehouse of Tchibo is located. In general, capacity does not pose a problem in the context of barge transport. Like a train also a barge can only carry a limited number of containers at once. Nevertheless, a barge can depart multiple times a day because it is not necessary to reserve a railway in advance.

Whenever there is no capacity available at the LCNH or at the HHLA, the respective container is routed to a warehouse nearby (Figure 9.16). Therefore, it is first checked whether or not the container is the first part of its project. This is due to the fact, that each project is supposed to be kept together in order to simplify its management during distribution. If the container is not the first part of the project, the location of the other parts has to be determined. If they are already routed to the same warehouse that is scheduled for the current container no further steps have to be taken regarding onward carriage. By contrast, if the destination of the other parts is another warehouse it has to be checked if a rerouting is still possible. Subsequently, the other containers are rerouted to the same warehouse as the current one. Whenever it is not possible to proceed this way, the current container is not routed to the nearest warehouse but to that one in which the other parts of the same project are already kept.

Customs clearance for each container takes place as soon as it arrives at its respective warehouse. Due to their number the containers are generally not investigated by the authorities themselves. In general, the process of customs clearance includes only the transmission of the respective documents. Also subsequently to their arrival at the warehouse the content of the containers undergoes an examination by the quality assurance department.

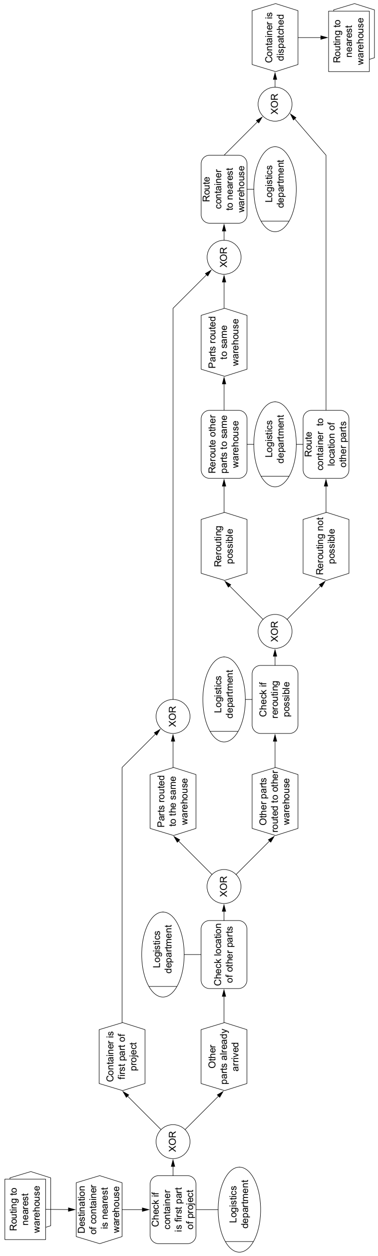

Fig. 9.16 Containers are routed to the nearest warehouse if no other capacity is available. In this case it has to be checked where other parts of the same project are located in order to keep them as close together as possible.

9.3 Participating Logistics Entities

After the examination of the procurement logistics processes of Tchibo this section identifies and characterises the main participating entities. Especially those participants are considered that are directly involved in onward carriage from the ports to the warehouses. This is particularly important in view of the later modelling of these processes. It is rather obvious that the shipping containers are the most important entities in this scenario (Section 9.3.1). Further participants are the ports (Section 9.3.2) and warehouses (Section 9.3.3) involved, which are the sources and sinks, respectively. Finally, the container transport relations connecting ports and warehouses are also examined (Section 9.3.4). Each of these entities is discussed together with a list of questions that have to be answered during dispatch.

9.3.1 Shipping Containers

The containers involved in the dispatch process are mainly characterised by their content. All articles carried by a container belong to a specific project. This project determines the date of the respective main sales phase. This date implies the deadlines for the arrival at all points between the port in East Asia and the outlet in Europe. Further questions concerning the content of a container are whether or not its cargo can be auto-palletised and whether or not it needs special treatment due to its worth.

Apart from their cargo, containers can be distinguished by their ports of loading and discharge. In this context it is particularly interesting when the container is scheduled to be returned to the carrier after discharge. In general, this topic depends on the involved carrier. The framework agreement with the preferred carriers grants a longer time-frame than other carriers. Furthermore, detention, the fee for returning shipping containers late is more moderate for the former group of carriers. This is an important point for the decision which containers have to be dispatched and received first in order to reduce costs. On average Tchibo receives a number of about 300 containers per week. Each of the following questions has to be answered in the dispatching process with respect to shipping containers:

- When is the estimated time of departure (ETD) in East Asia?
- Which is the port of loading (POL) in East Asia?
- When is the estimated time of arrival (ETA) in Europe?
- Which is the port of discharge (POD) in Europe? At which terminal?
- Does the container arrive later than estimated? Is it late?
- When is the main sales phase supposed to start?
- Is the container the first part of its project?
- Did other parts of the same project arrive already?

- What is the location of the recent parts?
- Is the article carried valuable?
- Can the article carried be auto-palletised?
- How many pallets emerge from one container?
- Is the container already released?
- What is the rejection status of the container?
- When has the empty container to be returned to the carrier?
- Which fee has to be paid for returning the container late?

9.3.2 Ports of Discharge

Tchibo has restricted its ports of discharge in Europe to Bremerhaven and Hamburg, both of them located in Northern Germany. Their geographical location directly supports a fast transport of incoming cargo from the port to the central warehouse in Bremen. However, not every carrier serves both ports. Some of them are only connected to Bremerhaven or Hamburg respectively in order to reduce detention in the ports. Despite of the restriction to two ports the actual number of involved container terminals is higher. This is due to the fact that container terminals in both ports are operated by different companies. These companies include the Eurogate GmbH & Co. KGaA, KG based in Bremen and the HHLA in Hamburg. Eurogate is a subsidiary of the Eurokai KGaA and the BLG. The HHLA is owned by the Free and Hanseatic City of Hamburg.

Containers that arrive in Bremerhaven are handled by the Eurogate Container Terminal Bremerhaven (CTB). For containers delivered to Hamburg there exist four alternatives: Tollerort Container Terminal (TCT), Container Terminal Altenwerder (CTA), and Container Terminal Burchardkai (CTB), which are operated by the HHLA, as well as the Eurogate Container Terminal Hamburg (CTH). While all other container terminals in Hamburg are located at the west side of the Elbe river, TCT is an exception as it is located on the eastern riverside. This fact is a disadvantage as it makes it difficult to combine containers from multiple terminals onto the same train. Transport by train is cheap and therefore generally desired. But containers located at TCT have to wait several days until a full train can depart from this terminal. The only alternative is the more expensive transport by truck. Therefore, Tchibo has reduced the number of containers arriving at TCT through the selection of its preferred carriers. The container terminals are not just paid for loading and unloading containers. Besides, demurrage has to be paid for a late pickup of containers at the terminal. Hence, Tchibo is interested in collecting containers as soon as possible in order to store them at the logistics centre until they are unloaded. The following questions are of interest when dispatching containers with respect to the port of discharge:

- How long does it take to unload a container from the vessel?
- Which fee has to be paid for leaving a container at the terminal?

9.3.3 Warehouses

Tchibo uses a number of different warehouses. The warehouses range from conventional single-storey warehouses to high-bay warehouses. At the time of this case study, Tchibo used in total 15 warehouses. They are mainly located in Northern Germany, most of them in the area of Bremen. The warehouses can be distinguished with respect to their type (Vahrenkamp, 2007, p. 176–181), i. e., whether they are single-storey or high-bay warehouses. This also influences the type of cargo that can be stored. For example, an automatised high-bay warehouse can generally only store palletised articles. Apart from these physical limitations also political restrictions can be established, e. g., storing all valuable goods in a secured warehouse. A further distinction can be made regarding the storage capacity as well as the capacity at receiving and shipping.

The costs of a warehouse depend on two factors. One of them is the fixed basic charge for the service of providing capacity. A variable part depends on the amount of cargo that is actually stored in the respective warehouse as well as the turnover at receiving and shipping. The central warehouse of Tchibo is the LCNH logistics centre at the Neustädter Hafen in Bremen. Apart from the LCNH, Tchibo applies two other high-bay warehouses. A second one in Bremen serves mainly as a storage for coffee, while the warehouse in Gallin receives and handles mainly articles returned by the outlets. All other ones are single-storey warehouses that serve as alternatives if the central logistics centre does not have sufficient capacity. However, the HHLA warehouse in Hamburg especially handles oversized articles. The following questions concern the process of container dispatch with respect to the warehouses:

- Which warehouses should be filled with priority?
- Which articles can be received (e. g., valuable, auto-palletisable)?
- Which fill level does a warehouse have (today and in the future)?
- Which other sources request warehouse capacity?
- Which capacity for receiving and shipping is available?
- How much basic charge must be paid for a warehouse?
- How much time does it take to receive a container?
- Which fee must be paid for receiving and shipping of goods?
- Which fee must be paid for storing a pallet per day?

9.3.4 Transport Relations

As already discussed in Section 9.2.3 there exist different methods of transport for onward carriage, namely barge, train, and truck. The barge connects Bremerhaven to Bremen. It is the cheapest way for transporting containers from the port to the central high-bay warehouse in Bremen. This is due to the fact that the barge can carry multiple containers per way. However, a barge is only available between Bremerhaven and Bremen because it depends on inland waterways, which is in this case the Weser river. By contrast, there exists no such connection from Hamburg to Bremen. Nevertheless, that connection is served by train that also offers good rates. Trains potentially depart from Hamburg on multiple days per week. A train is only used if there are enough containers available. Whenever no barge or train is available (Section 2.1.1), containers have to be transported by truck from both Bremerhaven and Hamburg. This leads to highest costs, because of the limited capacity of trucks compared to trains and barges. The contract between Tchibo and its transport service provider includes fixed rates for container transport on each relation. Dispatching demands requires answers to the following questions:

- Which transport relations connect a terminal and a warehouse?
- Which transport capacity is available at a given time?
- Which fee has to be paid for the transport from the terminal to the selected warehouse?
- Which is the cheapest transport relation available?
- How much time is needed for the transport?

9.4 Conclusion

This case study examines real world logistics processes which are hitherto centrally organised. It is intended to serve as a foundation for examining autonomous cooperation in the processes observed. In this context the choice of the procurement logistics processes of Tchibo as the subject of investigation can be motivated by their complexity as well as the underlying dynamics. The processes considered include shipping of containers from East Asia to ports in Europe, as well as their onward carriage into the warehouses in Northern Germany. Today, assigning containers to warehouses and respective transport relations is carried out centrally by the logistics department. This task demands monitoring the total system which is a complex task due to the number of 300 shipping containers that have to be dispatched per week. Furthermore, each container is linked with a high number of interrelated parameters that have to be taken into consideration for dispatch (Section 9.3). This is even more challenging as the container dispatch is conducted manually to a major extent. The support for the employee responsible for fulfilling this task is rather limited. Despite of a computer application supplying all relevant information, it is still the task of the human dispatcher to combine them for dispatch. The number of parameters for each container implies that it is rather impractical to take all aspects into consideration for the manual dispatch. Therefore, only some questions are currently considered while the rest of them is simply discarded (Section 9.2). Furthermore, dispatch is restricted to those containers with the highest priority, while the other ones are delayed for later dispatch. Another challenge is the low redundancy of the process of dispatch. It is a highly specialised task that is accomplished only by a limited number of employees. Problems may arise in the case of holidays, illness, as well as transfer of personnel.

In this context, it seems promising to support the employees by automating at least standard situations. Proceeding this way allows them concentrating on exceptional cases which require more attention. Due to the high degree of complexity and dynamics underlying the distributed processes central control is applicable only to a minor degree (Section 2.3). It has to be investigated to which extent an autonomous control of the participating entities decreases the complexity of dispatch. Thereby, each entity mainly resorts to its own parameters and tries to optimise them by cooperating with other involved participants.

Acknowledgement

Several employees of the forward logistics department of Tchibo acted as interviewees and contributed valuable information to this case study. Therefore, the author should like to thank Tchibo as well as its employees.

References

BLG in.add.out Logistics GmbH & Co. KG. (2006). *The High Bay Warehouse in Bremen.* Bremen, Germany.

DHL Express Vertriebs GmbH & Co. OHG. (2005). *Logistik-Lotse 2005* (13th ed.). Bonn, Germany: Verkehrs-Verlag Fischer.

Die Welt. (2006, May 9). *Tchibo feiert mit Bestellservice Erfolge.*

KarstadtQuelle AG. (2006). *Geschäftsbericht 2005.* Essen, Germany.

Schuldt, A. (2006). *Management of International Logistic Processes: A Case Study at Tchibo* (Tech. Rep. No. 39). Universität Bremen: Technologie-Zentrum Informatik.

Seidlmeier, H. (2006). *Prozessmodellierung mit ARIS: Eine beispielorientierte Einführung für Studium und Praxis* (2nd ed.). Wiesbaden, Germany: Friedrich Vieweg & Sohn Verlag.

Tchibo GmbH. (2004). *Jede Woche eine neue Logistik-Welt.* Hamburg, Germany.

Tchibo Holding AG. (2006). *Geschäftsbericht 2005.* Hamburg, Germany.

Vahrenkamp, R. (2007). *Logistik — Management und Strategien* (6th ed.). Munich, Germany: Oldenbourg.

Wood, D. F., Barone, A., Murphy, P. & Wardlow, D. L. (1995). *International Logistics.* New York, NY, USA: Chapman & Hall.

Chapter 10
Transition to Autonomous Logistics

In autonomous logistics, the participating logistics entities are themselves responsible for achieving the objectives imposed by their owners. Delegating decision-making to the local entities is a significant difference to conventional approaches with centralised control. An operationalisation for autonomous control of logistics processes has been developed in Chapters 5 to 7. The actual implementation with multiagent systems is described in Chapter 8. As a foundation for a transition from centralised to autonomous control, it is important to evaluate the new method. For some aspects, this evaluation can be conducted analytically. Hence, there is no need for simulation in these cases (Wenzel, Weiß, Collisi-Böhmer, Pitsch & Rose, 2008, p. 15). For more complex runtime interactions of autonomous logistics entities, however, simulation is an appropriate means of investigation. As discussed in Section 8.2, multiagent-based simulation is particularly suited for examining the actual agent behaviour as it would be in real-world operation.

The evaluation conducted in this chapter follows the structure of the proposed approach (Chapters 5 to 7). Firstly, the potential for cooperation in autonomous logistics is analytically investigated in Section 10.1. Secondly, Section 10.2 examines analytically the limitations for cooperation in autonomous logistics that can be derived from the arising interaction effort. Finally, the real-world logistics process examined in Chapter 9 is revisited. Section 10.3 describes and examines its transition to autonomous logistics by means of multiagent-based simulation.

10.1 Potential for Cooperation

The potential for cooperation in autonomous logistics has already been discussed in Chapter 5. To recapitulate, Section 5.2.2 gives the following reasons for establishing organisational structures:

1. Decreasing the external interaction effort

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics*,
DOI 10.1007/978-3-642-20092-2_10,

2. Increasing the resource utilisation efficiency

On the one hand, the number of interaction partners can be significantly reduced if autonomous logistics entities act jointly to achieve their objectives. Then, only a distinguished team manager is responsible for communication with the outside world. That is, the number of interaction partners for other entities decreases from n team members to one manager. On the other hand, cooperation supports process optimisation by using logistics resources such as means of mass transport more efficiently.

The discussion so far was primarily qualitative. Deciding whether and when cooperation in autonomous logistics is eligible, however, requires a quantitative analysis. Therefore, Section 10.1.1 examines how joint action can actually reduce the interaction effort of autonomous logistics entities. Section 10.1.2 investigates the benefit from jointly utilising logistics resources. Based on these foundations, Section 10.1.3 revisits the question of an appropriate degree of autonomous control.

10.1.1 Decreasing the External Interaction Effort

As elaborated in Section 5.2, autonomous logistics entities can act jointly if they have common properties or objectives. One advantage of cooperation is that the external interaction effort can be reduced significantly. To reiterate an example from Section 5.2.1, storage positions in a storage facility can jointly offer their service. Then, interested service consumers have one distinguished contact for their service request. The internal processing of such requests is left to the team members. It depends on the commitment (Definition 5.14) team members have to their team. If the team manager may act on behalf of its members, it can directly respond without forwarding the message to its members. However, contacting only the team manager is also beneficial if the manager has to forward the message to its members. In the storage facility example, only one message has to be sent through a wide area network. Message forwarding can then be implemented as a broadcast of the local area network of the storage facility.

To summarise the discussion, decreasing the external interaction effort is beneficial. It leads to a decrease in costs because less communication is required. The decrease in interaction effort is expressed by the ratio of the external interaction effort with and without teams:

$$eie_r(\mathit{Agents}, \mathit{Teams}) := \frac{\min(|\mathit{Agents}|, |\mathit{Teams}|)}{|\mathit{Agents}|} = \frac{|\mathit{Teams}|}{|\mathit{Agents}|} \tag{10.1}$$

Thereby, $\mathit{Agents} \subseteq \mathcal{SC} \cup \mathcal{SP}$ is the set of all agents considered. The set of teams represented by their team managers is referred to as $\mathit{Teams} \subseteq \mathit{Agents}$. The equation

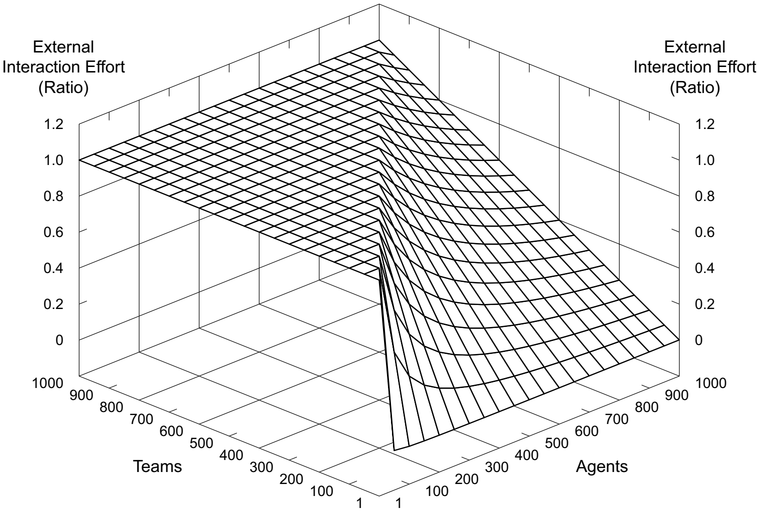

Fig. 10.1 Joint action in teams reduces the external interaction effort because other participants can simply contact the team manager instead of all members. The decrease is maximal if all agents form only one team which corresponds to centralised control.

$$\min(|Agents|, |Teams|) = |Teams| \tag{10.2}$$

holds because *Teams* is a subset of *Agents*, i. e., the size of *Teams* cannot exceed the size of *Agents*:

$$|Agents| \geq |Teams| \geq 1 \tag{10.3}$$

If there is only one agent, there is exactly one team, the singleton team in which the agents itself is its team manager.

Figure 10.1 depicts eie_r (Equation 10.1) for up to 1,000 agents which are organised in up to 1,000 teams. In the middle diagonal between the axes for teams and agents, the number of teams equals the number of agents. This means there are only singleton teams and consequently no decrease in interaction effort for external interaction partners. On the left hand side of this diagonal, the number of teams would even exceed the number of agents. As explained in Equation 10.2, this is impossible. For the sake of recognisability, however, Figure 10.1 depicts these values as 1.0 which corresponds to an unchanged interaction effort. On the right hand side of the middle diagonal in Figure 10.1, there are less teams than agents. That is, at least one team manager represents more than one agent. To recapitulate, external interaction partners can confine themselves to interacting with the team managers. Consider, for instance, that the number of teams is 10% of the total number of agents. Obviously, the external interaction effort then also decreases to

10%. A maximal decrease of the external interaction effort is reached if all agents are represented by only one team manager.

Following the analysis of the external interaction effort, the important question is how the findings can be interpreted. At first glance, one might be tempted to employ Equation 10.1 in order to minimise the external interaction effort. As explained in the preceding paragraph, this is achieved if only one team exists. This solution, however, represents the special case of centralised control. Consequently, such a solution contradicts the paradigm of autonomous logistics which aims at overcoming shortcomings of centralised control (Chapter 3). Hence, Equation 10.1 should be interpreted as a guideline for reducing the interaction efficiency. In particular, joint properties and objectives of autonomous logistics entities should guide the decision for cooperation (Section 5.2). Equation 10.1 then indicates the expected reduction in external interaction effort. This is particularly important if several alternatives for team formation exist. Apart from other criteria, the external interaction effort can be applied to compare such alternatives.

10.1.2 Increasing the Resource Utilisation Efficiency

In addition to reducing the interaction effort, cooperation can also help increasing the efficiency of resource utilisation. Chapter 5 gives some examples. Think, for instance, of sales units that are to be transported by truck (Section 5.1.2). Each truck has a certain capacity, e. g., several pallets. This capacity is the lot size, i. e., the minimum utilisation service consumers must accept when requesting the respective transport service. The lot size usually exceeds the size of one sales unit. Hence, it is beneficial for autonomous logistics entities to cooperate by utilising resources jointly (Section 5.2.2). Another example given in Section 5.2.2 refers to consolidating similar goods in a limited number of storage facilities. The purpose is to prevent empty vehicle running in the subsequent distribution processes. This means, consolidating similar goods early is a preparation for utilising logistics resources, transport in this case, more efficiently.

In order to answer the question of the potential for cooperation, it is fundamental to investigate the influence of the lot size. In the preceding Section 10.1.1, the focus is on autonomous logistics entities which form one or multiple teams. By contrast, the perspective in this section is another one. Here, the focus is on autonomous logistics entities which act either individually or in one team. Acting individually, each entity can choose a service provider itself. Acting jointly, all entities choose the same service provider. In order to focus on the actual question, this analysis abstracts from the costs of specific service providers. This means that it is assumed that all service providers have the same costs and offer the same service. Furthermore, it is assumed that all providers have unlimited capacity to abstract also from this unintended influence factor. Remember that the actual approach for joint action incorporates these aspects by considering multiple providers if the demand exceeds the capacity of a single one (Algorithm 7.2).

If autonomous logistics entities act individually, there exist numerous combinations how these entities can be distributed to logistics service providers. The distribution of consumers to providers can be regarded as integer partitions (Zoghbi & Stojmenovic, 1998, pp. 319–320). As an example, the possible integer partitions of four are:

$$4 = 3 + 1 = 2 + 2 = 2 + 1 + 1 = 1 + 1 + 1 + 1 \tag{10.4}$$

Thinking of four service consumers, they could be all utilising one service provider. Another possibility is that there is one service provider serving three consumers and one provider serving one consumer and so on. The possible partitions can be written as a set of multisets:

$$\{\{(4,1)\},\{(3,1),(1,1)\},\{(2,2)\},\{(2,1),(1,2)\},\{(1,4)\}\} \tag{10.5}$$

More generally, the integer partitions for arbitrary natural numbers can be defined as follows:

Definition 10.1 (Integer Partitions) *Let* $\mathrm{N} \subset \mathbb{N}$ *be a set of natural numbers, let* $\mathrm{M} = (\mathrm{N}, \mathrm{m})$ *be a multiset over* N *with* $\mathrm{m} : \mathrm{N} \to \mathbb{N}^{+}$ *defining the multiplicity of the elements of* N. $\mathrm{M} = (\mathrm{N}, \mathrm{m})$ *is a partition of* $\mathrm{i} \in \mathbb{N}$ *if* $\sum_{\mathrm{n} \in \mathrm{N}} n \cdot m(n) = i$. *The set of all partitions of* i *into* j *summands is*

$$\mathit{partitions}(i,j) := \left\{ (N,m) \;\middle|\; \sum_{n \in N} m(n) = j \wedge \sum_{n \in N} n \cdot m(n) = i \right\}$$

The total set of all partitions of i *is*

$$\mathit{partitions}(i) := \bigcup_{j=1}^{i} \mathit{partitions}(i,j)$$

Definition 10.1 is the foundation for examining the possible partitions of consumers to service providers. Algorithms that compute all possible integer partitions have, for instance, been proposed by Zoghbi and Stojmenovic (1998, pp. 325–327). Figure 10.2 depicts a graph of the number of possible partitions (Lint & Wilson, 1992, p. 132), $|\mathit{partitions}(i,j)|$, for the natural numbers between 1 and 100. The graph is in logarithmic scale because the number of partitions grows exponentially. The maximum is $11,087,828$ for 18 summands of 100. That is, there are more than 10^7 combinations how 100 service consumers can be distributed to 18 different service providers.

The actual utilisation for each of these distributions given a particular lot size can be determined as follows. Let c be the total number of consumers, let p be the total number of providers examined. The multiset

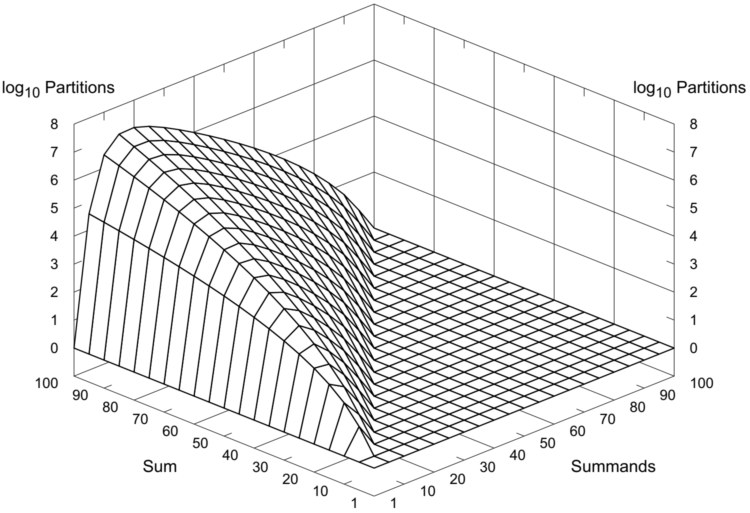

Fig. 10.2 The number of different integer partitions for all integers between 1 and 100. The values in the right half of the diagram are zero because there is no partition if the number of summands exceeds the sum. Note the logarithmic scale, the maximum is $11,087,828$ for 18 summands of 100.

$$M = (N, m) \in \mathit{partitions}(c, p) \tag{10.6}$$

is a partition of consumers to providers in accordance with Definition 10.1. That is, an $n \in N$ denotes the number of consumers served by a particular provider. By applying the *amount* mapping from Equation 5.16 with a *lotSize*, it is then possible to determine the actual amount utilised. The mapping $m(n)$ denotes how many providers serve n consumers. Altogether, the total utilisation for M given a particular *lotSize* can be determined as follows:

$$\mathit{utilisation}((N, m), \mathit{lotSize}) := \sum_{n \in N} m(n) \cdot \mathit{amount}(\mathit{lotSize}, n) \tag{10.7}$$

Figure 10.3 visualises this relationship of the number of providers chosen and the lot size for $1,000$ service consumers. Thereby, the number of providers and the lot size vary from 1 to $1,000$. To recapitulate, a considerably high number of possible partitions exists which exceeds 10^7 already for 100 consumers (Figure 10.2). Therefore, not all possible partitions can be depicted in the visualisation. Out of the variety of possible partitions, Figure 10.3 considers those partitions $M = (N, m) \in \mathit{partitions}(c)$ with the following numbers of consumers served:

$$N = \left\{ \left\lfloor \frac{c}{p} \right\rfloor, \left\lceil \frac{c}{p} \right\rceil \right\} \tag{10.8}$$

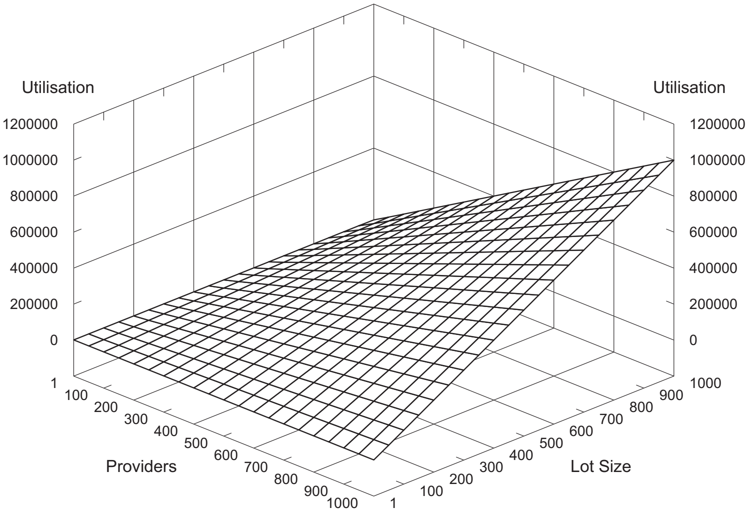

Fig. 10.3 The potential for cooperation of 1, 000 service consumers by joint utilisation of logistics resources. The higher the lot size of the logistics service providers, the higher is the benefit from jointly utilising only a few selected providers.

The multiplicity of the numbers of consumers served in M is:

$$m = \left\{ \left(\left\lfloor \frac{c}{p} \right\rfloor, p - (c \bmod p) \right), \left(\left\lceil \frac{c}{p} \right\rceil, c \bmod p \right) \right\} \tag{10.9}$$

For the sake of understandability, the partition can also be written as a sum. It corresponds to the following equation:

$$c = \left\lfloor \frac{c}{p} \right\rfloor \cdot (p - (c \bmod p)) + \left\lceil \frac{c}{p} \right\rceil \cdot (c \bmod p) \tag{10.10}$$

Note that the remainder $c \bmod p$ may be zero. Then, the right element in the sets of Equations 10.8 and 10.9 is omitted. This is in accordance with Definition 10.1 which demands multiplicities to be positive. In this case, all providers serve the same number of consumers. Otherwise, it holds that all providers have approximately the same number of consumers, i. e., the difference of the number of consumers served does not exceed one:

$$\forall_{n_1, n_2 \in N} \ |n_1 - n_2| \leq 1 \tag{10.11}$$

The graph depicted in Figure 10.3 can be interpreted as follows. Given the minimal lot size of only one for all service providers, it does not make a difference how many service providers are employed by the service consumers. With increasing lot sizes, however, also the total utilisation increases if the consumers employ too many service providers. The extreme case is that all $1,000$ service consumers employ $1,000$ different providers. With a lot size of one, the total utilisation is $1,000$. By contrast, if the lot size is $1,000$ per provider, the total utilisation increases to $1,000,000$ although $1,000$ would suffice. This means that the higher the lot size of the logistics service providers, the higher is the benefit from jointly utilising only a few selected providers. Hence, there is a massive potential for cooperation.

10.1.3 Appropriate Degree for Autonomous Control

Following the investigations of the preceding two sections, it is worth revisiting the question regarding an appropriate degree of granularity for autonomous control. Section 5.1.1 names components, articles, sales units, cardboard boxes, pallets, and shipping containers as possible levels at which general cargo units can be considered atomic. One answer to the question of an appropriate granularity can be given based on the technical limitations discussed in Section 3.1.3.

This answer, however, is not sufficiently extensive. The question is not only what is an appropriate choice for atomic units. Additionally, one must identify an adequate level at which the autonomous entities act jointly. On the one hand, Section 5.2 has already indicated that purely individual control is not desirable. This qualitative finding is supported by the quantitative analyses in Sections 10.1.1 and 10.1.2. On the other hand, the paradigm of autonomous logistics addresses the challenges for logistics control (Section 2.3) by delegating process control to the participating logistics entities themselves (Section 3.1). Consequently, it is not reasonable that all logistics entities form one team with one team manager which is then responsible for process control. This extreme case of autonomous control is actually centralised control.

Hence, the lower and upper bounds for an appropriate degree of granularity for cooperation in autonomous logistics are as follows. The lower bound is shaped by two influence factors, namely the external interaction effort (Section 10.1.1) and the resource utilisation efficiency (Section 10.1.2). The upper bound is due to the insight that autonomous logistics entities should not cooperate arbitrarily. Cooperation only makes sense if the entities share similar properties and objectives (Section 5.2.2). In the case of similar properties and objectives, there is a potential for synergies because negotiations can be conducted jointly by one responsible team manager. Contrariwise, cooperation is not reasonable in the absence of such similarities. In that case, the autonomous logistics entity acting as a team manager would have to incorporate aspects that are widely disjoint from its own problems (cf. Figure 5.4). Consequently, the decision-making regarding this problem should be decomposed and delegated to the concerned entities themselves.

10.2 Effort and Limitations of Cooperation

In the preceding Section 10.1, the potential for cooperation has been examined. The outcome is that cooperation in autonomous logistics processes is actually beneficial. Consequently, also the interaction effort arising for cooperation must be investigated. Based on the agent interaction protocols (Section 4.2.4) applied, the interaction effort can often be derived easily. As an example, consider the FIPA request or contract net interaction protocols employed for coordinating team action (Chapter 7). The number of respondents in these protocols is comparatively constant in autonomous logistics. Think, for instance, of storage service providers that offer their services in negotiations (Section 7.1.2). Unless new storage facilities are added to the logistics network, the number of storage providers in one region does not change. As a consequence, the interaction effort is constant for individual participants and increases linearly with the number of participants.

The team formation interaction protocols (Chapter 6) lay the foundation for cooperation in autonomous logistics. Their asymptotic interaction complexity has already been examined in Section 6.2. To estimate the real effort for team formation in actual applications, however, it is desirable to obtain more precise figures. This is particularly important because interaction complexity is not only a cost factor but also a potential limitation for autonomous control. Analysing the actual effort of these protocols is challenging. In contrast to the FIPA request and contract net protocols, the roles of the participants may change over several runs. During one run of a team formation protocol, a participant may become a new team manager, thereby increasing the number of respondents by one for subsequent runs.

Sections 10.2.1 and 10.2.2 examine the interaction effort for team formation by directory and broker, respectively. The effort for team formation by multicasting equals the directory-based case. Based on these foundations, Section 10.2.3 derives limitations for autonomous control in logistics.

10.2.1 Effort of Team Formation by Directory and Multicast

The asymptotic interaction complexity of directory-based team formation is $O(mn) = O(n^2)$ (Section 6.2.1). However, the number m of team managers is usually only a small fraction of all n participants, i. e., $m \ll n$. The same observation holds for the respective interaction protocol based on multicast messages (Section 6.2.3). The actual interaction effort for both protocols depends on two factors:

1. The total number of team managers
2. The times at which participants become team managers

Every agent that joins the multiagent system contacts all existing team managers in order to find a matching team. To determine the interaction effort for a particular agent, it is thus necessary to find out how many team managers already exist at its creation time. The question how many agents in the multiagent system are team managers at a certain point in time can be reformulated. It is equivalent to the question to how many teams the agents in the multiagent system pertain. The questions are interchangeable because the first agent of each team is the responsible team manager.

The question of the number of teams can be approached in two steps. Firstly, one must determine the probability that an agent of a particular team has already been created. Secondly, the resulting probability must be accumulated for all teams in order to get the total number of different teams existing. A foundation for this investigation is the hypergeometric distribution (Dörfler & Peschek, 1988, pp. 387–388). It is defined as follows:

Definition 10.2 (Hypergeometric Distribution) *Let* $\mathrm{A} \in \mathbb{N}$ *be the size of the population, let* $\mathrm{T} \leq \mathrm{A} \in \mathbb{N}$ *be the number of successes in the population, let* $\mathrm{a} \leq \mathrm{A} \in \mathbb{N}$ *be the number of draws, and let* $\mathrm{t} \leq \mathrm{A} \in \mathbb{N}$ *be the number of successful draws. The probability for* t *given* a*,* T*, and* A *is*

$$P(X = t) := h(t \,|\, a, T, A) = \frac{\binom{T}{t}\binom{A-T}{a-t}}{\binom{A}{a}}$$

Within the scope of this section, let A denote the total number of agents. All agents are numbered by a in the order of their creation. That is, $a - 1$ agents have been created before agent a with $1 \leq a \leq A$. Let T be the number of agents pertaining to a particular team. Based on Definition 10.2, the probability that at least one member of the team is in the total number of agents examined so far can be derived as follows:

$$P(X > 0) := \sum_{t=1}^{T} h(t \,|\, a, T, A) = 1 - h(0 \,|\, a, T, A) \tag{10.12}$$

Therewith, the question of the probability that at least one agent of a team has already been created is answered. The extension to all teams follows.

The respective partitioning of agents to teams, or to descriptors (Definition 5.8) to be more precise, can be considered an integer partition (Definition 10.1). The respective partitioning of all agents A is represented by the multiset

$$M = (N, m) \in \mathit{partitions}(A) \tag{10.13}$$

Summing up the probabilities for the existence of individual teams (Equation 10.12) for all teams (Equation 10.13) in the multiagent system reveals the number of already existing teams at the creation time of agent a:

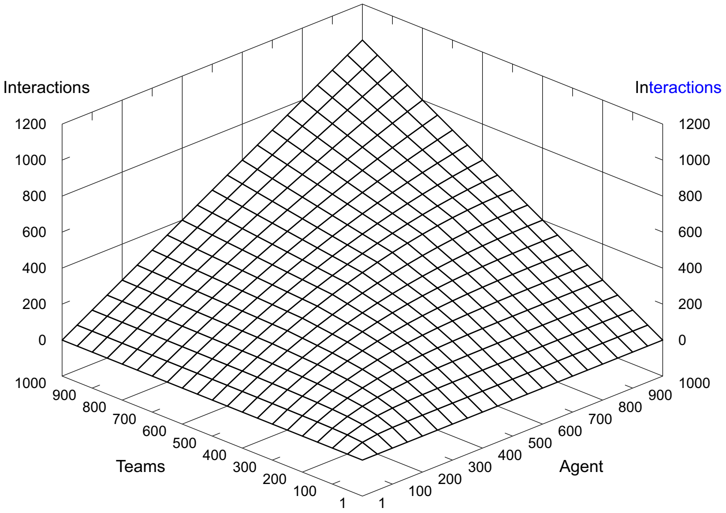

Fig. 10.4 The number of interaction partners for participants in the team formation interaction protocols based on directory and multicast messages. The number of interactions depends on the number of predecessors in the team formation process.

$$f(a,(N,m),A) := \sum_{n \in N} m(n) \cdot \big(1 - h(0 \,|\, a,n,A)\big) \tag{10.14}$$

To recapitulate, the number of teams corresponds to the number of team managers agent a has to interact with, i. e., it specifies the expected interaction effort of agent a.

Figure 10.4 depicts Equation 10.14 for all $1 \leq a \leq A$ when varying the number of teams $1 \leq T \leq A$ for a total of $A = 1,000$ agents. As elaborated in Section 10.1.2, a considerably high number of such partitions exists. Therefore, Figure 10.4 is confined to considering only those partitions with almost equal team sizes. This is in analogy to the almost equal distribution of consumers to providers described in Section 10.1.2 and specified by Equations 10.8 to 10.11.

The total interaction effort of the whole system until the creation of agent a can be derived from Equation 10.14 as follows:

$$F(a,(N,m),A) := \sum_{i=1}^{a} f(i,(N,m),A) \tag{10.15}$$

$$= \sum_{i=1}^{a} \sum_{n \in N} m(n) \cdot \big(1 - h(0 \,|\, i,n,A)\big) \tag{10.16}$$

The graph in Figure 10.5 shows the overall interaction effort for the same team partitions investigated in Figure 10.4. Note that Figures 10.4 and 10.5

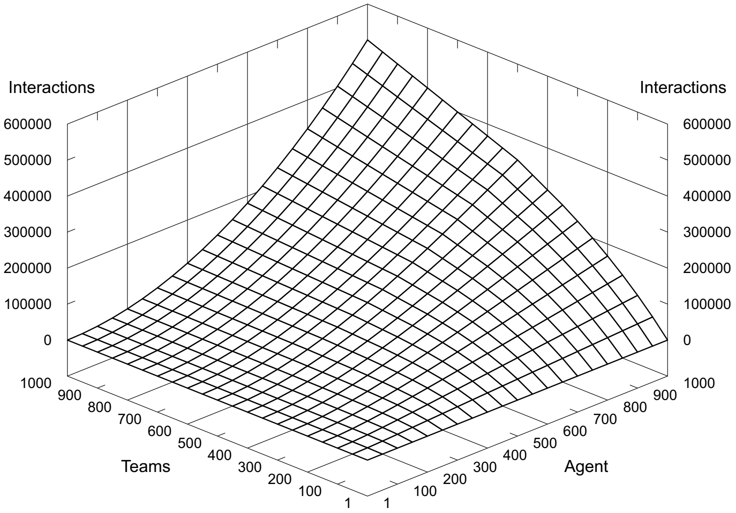

Fig. 10.5 The total number of interactions for team formation based on directory and multicast. Depending on the number of teams, the interaction effort ranges between linear and quadratic.

have a different scale. While the maximum interaction effort for an individual agent is $1,000$, it is $500,500$ for the whole system of $1,000$ agents.

The interaction effort for two extreme cases is worth being investigated in more detail. On the one hand, this holds for the case of all agents pertaining to the same team. On the other hand, this holds for the case of all agents pertaining to individual singleton teams.

Proposition 10.1 (Interaction Effort for Maximum Teams) *Let all* A *agents participating in directory-based team formation have equal descriptors, i. e., only one team exists having the maximum number* A *of members:*

$$M = (N, m) = (\{A\}, \{(A, 1)\})$$

In this case, the individual interaction effort for team formation is constant. Furthermore, the interaction effort for the whole system increases linearly with the number of participants.

Proof: In accordance with the protocol definition, the first member of a team becomes its team manager. Every single agent including the team manager itself has to interact with this team manager because exactly one team exists. The probability that a participant pertains to the team is one because the number n of team members is the total number A of participants:

$$\begin{aligned} f(a,(N,m),A) &= \sum_{n\in N} m(n)\cdot\big(1-h(0\,|\,a,n,A)\big)\\ &= 1\cdot\big(1-h(0\,|\,a,A,A)\big)\\ &= 1\cdot\big(1-0\big)=1 \end{aligned}$$

Consequently, the overall number of interactions is

$$\begin{aligned} F(a,(N,m),A) &= \sum_{i=1}^{a} f(i,(N,m),A)\\ &= \sum_{i=1}^{a} 1 = a \end{aligned}$$

That is, the individual interaction effort is actually constant and the overall interaction effort increases linearly with the number of participants. □

Proposition 10.2 (Interaction Effort for Minimum Teams) *Let all* A *agents participating in directory-based team formation have different descriptors, i. e., the number of teams equals the number* A *of participants:*

$$M=(N,m)=(\{1\},\{(1,A)\})$$

In this case, the individual interaction effort for team formation increases linearly with the number of participants. Furthermore, the interaction effort for the whole system is $\frac{\mathrm{n}^2-\mathrm{n}}{2}$ *with* n *being the number of participants.*

Proof: In accordance with the protocol definition, the first member of a team becomes its team manager. Every participant becomes a team manager because it is member of a singleton team and thus its team manager. The probability that a participant pertains to one team is $\frac{1}{A}$ because the number of teams is the total number A of participants. In draw a, the probability is $\frac{a}{A}$ that a team has already been drawn. For all A teams, the total number of teams drawn is thus a for draw a:

$$\begin{aligned} f(a,(N,m),A) &= \sum_{n\in N} m(n)\cdot\big(1-h(0\,|\,a,n,A)\big)\\ &= A\cdot\big(1-h(0\,|\,a,1,A)\big)\\ &= A\cdot\big(1-(1-h(1\,|\,a,1,A))\big)\\ &= A\cdot\big(1-(1-\frac{a}{A})\big)\\ &= A\cdot\frac{a}{A}=a \end{aligned}$$

Consequently, the overall number of interactions is

$$\begin{aligned} F(a,(N,m),A) &= \sum_{i=1}^{a} f(i,(N,m),A) \\ &= \sum_{i=1}^{a} i = \frac{a^2-a}{2} \end{aligned}$$

That is, the individual interaction effort actually increases linearly with the number of participants and the overall interaction effort is $\frac{n^2-n}{2}$ for n participants. □

10.2.2 Effort of Team Formation by Broker

The broker-based team formation interaction protocol (Section 6.2.2) for autonomous logistics distinguishes from the directory-based in that the central broker administers the team descriptors. Therewith, the broker disburdens new participants from contacting all team managers. Instead, new participants simply contact the broker in order to find a team matching their respective descriptor. Subsequently, at most one other agent is contacted, namely the manager of the respective team. In accordance with the asymptotic interaction complexity of $O(1)$ examined in Section 6.2.2, the interaction effort is thus constant:

$$f(a) := 1 \tag{10.17}$$

Figure 10.6 depicts the graph of Equation 10.17. This allows contrasting it with Equation 10.14 which is depicted in Figure 10.4. Compared to its directory-based counterpart, team formation by broker has a significantly lower interaction effort for all configurations.

The total interaction effort for the whole system is thus:

$$F(a) := \sum_{i=1}^{a} f(i) = \sum_{i=1}^{a} 1 = a \tag{10.18}$$

That is, the interaction effort increases linearly with the number of agents participating in team formation. This is in accordance with the linear asymptotic interaction complexity analysed in Section 6.2.2. The respective graph is depicted in Figure 10.7. Compared to the directory-based protocol, the interaction effort for team formation by broker is significantly lower. To recapitulate, as a drawback the broker constitutes a potential bottleneck in the multiagent system. This, however, is not in the focus of investigation here because advantages and disadvantages of the protocols have already been discussed in Section 6.3.3.

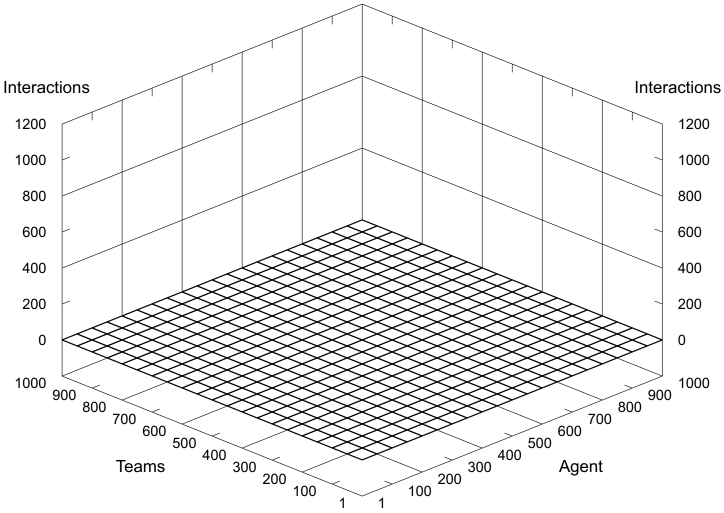

Fig. 10.6 In the broker-based team formation protocol, each participant interacts with the broker and at most one team manager. Hence, the number of interactions is constant and independent from the number of other agents.

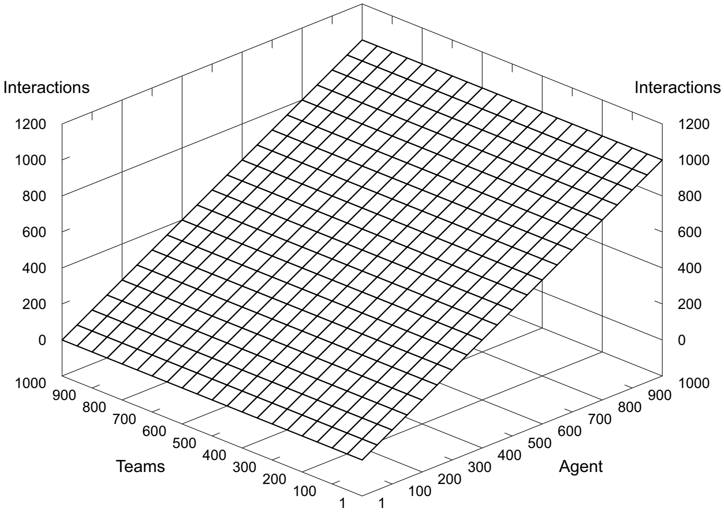

Fig. 10.7 The total number of interactions for the broker-based protocol is obtained by summing up the interactions of all participants. The number of interactions increases linearly with the number of participants.

10.2.3 Limitations for Autonomous Control

As already indicated in Section 3.1.3, a high interaction effort is likely to outweigh the decrease in computational complexity that is gained by decentralisation. Therewith, a high interaction effort is eventually also a limitation for autonomous control. The investigations in the preceding Sections 10.2.1 and 10.2.2 allow quantifying this assumption for the team formation interaction protocols developed in Chapter 6. Therewith, this investigation constitutes an instantiation of the abstract limitations for autonomous control formulated by Philipp, Beer, Windt and Scholz-Reiter (2007, p. 304).

One limitation concerns the interaction effort of individual agents (Equations 10.14 and 10.17). This topic is closely related to flow control in computer networks (Tanenbaum, 2003, p. 192). A limit is reached if an agent must send more messages for team formation than it is able to process in a given period of time. The limit can be determined by intersecting the graphs derived from those depicted in Figures 10.4 and 10.6 with a plane representing the number of messages that can be processed. Configurations in which the graph exceeds this plane constitute a limitation for autonomous control in logistics. For these cases, the application of the respective protocol is neither manageable nor reasonable. It is, however, possible to switch to another protocol, thereby, for instance, decreasing the degree of autonomy granted to the individual agents (Section 6.2.2).

Correspondingly, another limitation can be derived from the interaction effort for the whole system (Equations 10.15 and 10.18). In order to find the respective limit, one must intersect the graphs derived from those depicted in Figures 10.5 and 10.7 with a plane representing the number of messages that can be handled by the system. All configurations that exceed the plane constitute another limitation for autonomous control in logistics. The communication overuse of the whole system is closely related to congestion control in computer networks (Tanenbaum, 2003, pp. 384–396).

The number of messages that can be exchanged by agents depends on the bandwidth of the underlying message transport system. The message transport system is part of the underlying agent platform, i. e., JADE in the case of this research project (Section 8.1.1). Shakshuki and Jun (2004, pp. 210–211) evaluate the message exchange performance of JADE. Their experiment has been conducted with Windows XP running on an Intel Pentium 4 processor with 1.6 GHz and 256 MB RAM. Not surprisingly, the number of messages that can be exchanged per agent decreases with an increasing number of agents. With two agents in total, $2,222$ can be exchanged per agent per second. With 32 agents running, the number of messages per agent reduces to 112 per second. These results hold if all agents populate only one agent container (Figure 8.1). According to their experiments, however, one must distinguish the case in which the agents are distributed to multiple agent containers. Then, messages can no longer be handled locally which results in a decreased efficiency (Shakshuki & Jun, 2004, p. 213). In this case, the number of messages exchanged per agents and second ranges between 78 for two agents and four for 32 agents.

Note, that the time agents need in order to actually process the messages is not considered in the experiments by Shakshuki and Jun. Furthermore, if agent containers are physically distributed over multiple computers also the bandwidth of the underlying computer network has to be considered. Nevertheless, their figures can serve as an indicator for limitations of interaction in autonomous logistics that are due to communication. As an example, consider the configurations depicted in Figure 10.4. In the worst case, the last agent that joins the system must contact $1,000$ other agents for team formation. Given that this agent is the only one communicating at that time, its messages can be delivered within half a second. This holds at least as long all agents populate the same agent container. Otherwise, the message transfer takes about 13 seconds. Whether these transfer times are acceptable depends in both cases on the frequency with which new agents join the system. A frequency that is higher than half a second or 13 seconds, respectively, leads to a communication overload for the agent platform.

Limitations for autonomous control are not restricted to the aspect of communication. Likewise, also monetary limitations can be determined. The interaction effort multiplied with the average number of messages, the average message size, and the price for data transfer reveals the arising costs. A cost limit can then be defined as a plane in the respective graphs in order to indicate which configurations exceed this limit.

To summarise, the thorough investigation of the limitations for team formation in autonomous logistics helps choose the right interaction protocol for a specific purpose. Despite of their own limitations, it is important to mention that the interaction protocols extend the limitations of autonomous control. This is due to the fact that they lay the foundation for benefitting from a decrease in the external interaction effort (Section 10.1.1) and an increase in the efficiency of resource utilisation (Section 10.1.2).

10.3 Process Control by Autonomous Logistics

Following the theoretical investigation of the potential (Section 10.1) and the limitations (Section 10.2) of autonomous control in logistics, this section focuses on the practical application for process control. To recapitulate, the autonomous logistics paradigm delegates process control to software agents which act on behalf of the participating logistics objects. The continuum for deploying these agents ranges between the following extremes (Section 3.2.5):

1. Physically distributed on embedded systems attached to the objects
2. On a central server or server cluster

The first option is particularly appropriate if sensor technology is applied in order to monitor the cargo (Section 3.2.3). Then, decentralised data processing allows evaluating and reacting to sensor measurements locally without the need to communicate all data to the cargo owner. The second option can be applied whenever there is no demand for continuous cargo monitoring. In that case, less investments are required because there is no need to enhance usual logistics objects with embedded systems.

Apart from sensor technology, the major difference between the two alternatives is how localisation is implemented (Section 3.2.2). An agent representative that is not embedded in the logistics object does not have direct access to the localisation unit. Therefore, it must rely on beacon-based localisation (Figure 3.7). This means that it is informed about status updates that occur along the supply chain, e. g., identification at shipping and receiving. Such event notifications are already available today. As an example, think of tracking and tracing in the courier, express, and parcel business (Section 3.2.2). Provided with these event notifications, the responsible software agent can update its schedule accordingly. The difference to software agents on embedded systems is thus the resolution of localisation. An agent with direct access to global navigation satellite systems, can adapt its way through the logistics network more frequently.

In contrast to the frequency of status updates, however, the underlying decision processes of both alternatives do not differ. This allows deploying software agents on a central server cluster for the time being. If embedded systems are by default integrated in logistics objects in the future, it will be possible to deploy the software agents with the same behaviour on embedded systems. Note that deploying software agents on a central server or server cluster is clearly distinct to conventional centralised control because the advantages of reduced complexity and the improved ability of coping with dynamics are retained (Section 3.2.5).

This section addresses the question how actual logistics processes can be controlled by autonomous logistics entities deployed on a central server or server cluster. The particular subject of investigation is the onward carriage process examined in Section 9.2.3. In contrast to the analyses in Section 10.1 and 10.2, the interaction in this real-world process cannot be examined analytically. Instead, the dynamics underlying such processes demand for simulation as means of investigation (Wenzel et al., 2008, p. 15). In particular, multiagent-based simulation is employed because it allows testing the actual agent behaviour (Section 8.2). The simulation model is highly detailed (cf. Krcmar, 2005, p. 21) due to the direct correspondence between agents in simulation and real-world operation.

Section 10.3.1 investigates how the onward carriage process can be controlled with autonomous logistics. A particular focus is on the coverage of important industry requirements derived from the case study in Chapter 9. Subsequently, Section 10.3.2 describes the simulation experiment conducted in order to examine autonomous control in onward carriage. Based on these foundations, the simulation results for different strategies are presented and discussed in Sections 10.3.3 to 10.3.5.

10.3.1 Coverage of Industry Requirements

The onward carriage process examined in Section 9.2.3 deals with allocating storage and transport resources for shipping containers arriving at container terminals. The implementation presented in Chapter 8 lays the foundation for controlling this process autonomously. The implementation is applied as follows. Three types of software agents have been implemented based on the abstract agents available. Namely, these are shipping containers, forwarders, and warehouses (Figure 10.8). Shipping containers are service consumers, forwarders and warehouses constitute service providers.

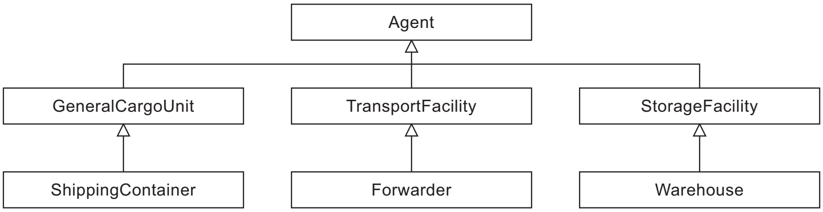

Fig. 10.8 UML class diagram of the inherited agent implementations for shipping containers, forwarders, and warehouses.

The implementations derived for the service providing agents implement the interfaces for descriptors, capacity, and costs (Section 8.1.3). In order to administer the capacity available for utilisation (Definition 5.15) by service consumers, a relational database system is employed. As already elaborated in Section 5.1, the roles taken by logistics service providers are not necessarily limited to single primary logistics functions (Section 2.1). In the process examined here, the warehouse agents are not only responsible for administering storage capacity but also provide capacity for receiving and shipping. The shipping container agent implementation is inherited from the general cargo unit agent (Section 8.1.3). Its behaviour is specified by an implementation of the abstract service consumer behaviour that coordinates the utilisation of logistics resources as described in Section 7.3.3.

Figure 9.12 depicts how dispatch in the onward carriage process is conducted today. The decisions hitherto made by the human dispatcher must be regarded correspondingly in the autonomously controlled process. After release and rejection handling, the first decision made is whether the currently considered shipping container has a high priority. The prioritisation is intended to help human dispatchers focus on high priority containers which demand urgent attention. This confinement, however, is no longer necessary in autonomous logistics because process control is automated by delegating decision-making to the shipping containers themselves.

Further decisions previously made consider the specific properties of shipping containers which constrain the choice of appropriate warehouses. These decisions are reflected in the demand specifications (Definition 7.1) used in the negotiation between service consumers and providers. Likewise, also the check for sufficient capacity is performed by the service providers queried (Algorithm 5.2). Choosing the optimal storage and transport providers (Figures 9.15 and 9.16) is ensured by negotiating with all eligible providers as specified in Algorithm 7.1. The goal of receiving similar goods in the same warehouse if possible (Figure 9.16) is reflected by Algorithm 7.2.

To summarise, the industry requirements specified in Section 9.2.3 are covered by the autonomous logistics approach. In the new approach, the individual shipping containers control the process instead of a human dispatcher. Therefore, dispatch is no longer limited to highly prioritised containers. Furthermore, the frequency changes from dispatch on a daily basis to dispatch based on supply chain events. In total, the system disburdens human dispatchers from standard tasks. Instead, they can now turn their intention to exceptions which go beyond the boundaries of the technical system and which can therefore not be resolved by the autonomous logistics entities (Section 7.3.4).

10.3.2 Simulation Experiment

The general coverage of industrial requirements (Section 10.3.1) shows that the system is in principle capable of satisfying the logistics demands. However, this investigation only considers the aspect of effectiveness. Besides, the efficiency of the approach must be examined, i. e., the performance of process control. For the respective simulation experiment, the shipping container agents are instantiated as follows (Figure 10.9).

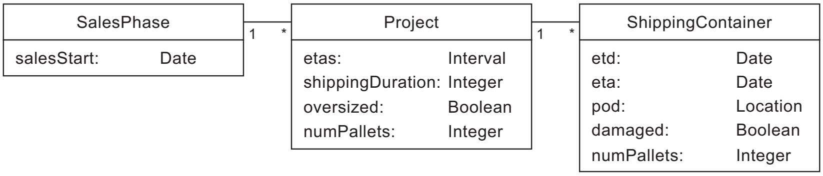

Fig. 10.9 UML class diagram illustrating the interrelationship and parameters of sales phases, projects, and containers.

As elaborated in Section 9.1.2, all articles are sold in weekly sales phases. The sales phase determines the date at which the distribution of the articles starts, i. e., the latest time at which the cargo must be available in the warehouse. The products are organised in so-called projects. Each sales phase consists of about 30 projects, most of them are delivered by shipping container. Depending on their origin in East Asia, the projects have different durations for transport by container vessel. The estimated time of arrival (ETA) for the shipping containers of one project may vary within a certain temporal interval if not all of them are shipped at the same time (Section 9.2.2). Furthermore, the articles within one project of articles may have special properties. In particular, projects with oversized articles cannot be received in high-bay warehouses, containers with damaged articles must be delivered to a selector (Section 9.3.1).

The amount of articles a project consists of is another important property in the examined process. The articles are packed into cardboard boxes. However, these cardboard boxes are re-packed onto pallets when they are received in a warehouse. In order to determine storage capacity required, each project is annotated with the resulting number of pallets. The number of articles and thus pallets is distributed to one or multiple shipping containers. Each shipping container has a specific estimated time of departure (ETD) and arrival which depends on the arrival interval of the respective project. Furthermore, each container has a certain port of discharge (POD) which may be Bremerhaven or Hamburg in the process under examination.

Figure 10.10 shows the probability distribution for the interrelationship between sales phases, projects, containers, and pallets. They are derived from data of eight sales phases from the real-world process in 2006. Each weekly sales phase consists of about 30 sales projects (Section 9.1.2). The number of projects that are delivered by shipping container varies between 15 and 30, the remaining are delivered from European suppliers by truck. The number of containers per project ranges from 1 to 60. Each project consists of up to 4, 200 pallets.

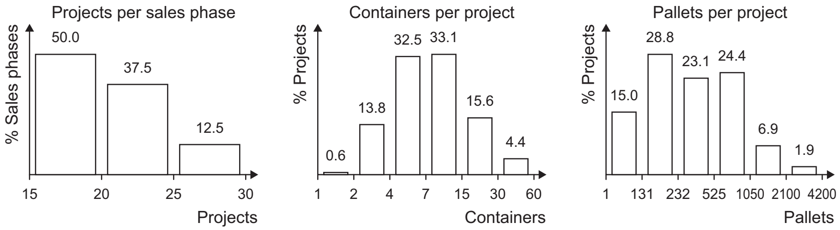

Fig. 10.10 Probability distributions for the projects per sales phase, containers per project, and pallets per project, respectively.

Figure 10.11 depicts the probability distributions for temporal interrelationships of the process. The left hand side shows the number of days the last shipping container of a project arrives before the start of the respective sales phase. The middle diagram of Figure 10.11 shows the interval length of the estimated time of arrival for the containers of one project. The average shipping duration for transporting containers from the Far East to Europe is depicted in the right hand side of the same figure. Note that the probability distribution within the ranges depicted in Figures 10.10 and 10.11 is assumed to be uniform.

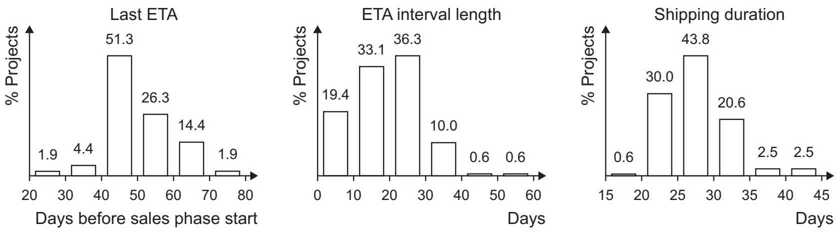

Fig. 10.11 Probability distributions for the last estimated time of arrival (ETA) before the start of the sales phase, the length of the interval of the ETAs, and the shipping duration, respectively.

Figure 10.12 depicts the probabilities for specific states of projects and containers, namely whether the articles of one project are oversized and whether the articles within one container are damaged. If articles are oversized, they cannot be received in some warehouses. If articles are damaged, they must be delivered to a selector for repair. Finally, the right hand side of Figure 10.12 shows the distribution between shipping containers to the ports of Bremerhaven and Hamburg.

Based on the probability distributions (Figures 10.10 to 10.12), synthetical test data for one year has been generated. The 52 sales phases consist of 1, 036 projects with 11, 521 shipping containers in total. These shipping containers populate the logistics network depicted in Figure 10.13. The network includes the ports of Bremerhaven and Hamburg as well as five storage facilities. These storage facilities are distinguished by their receiving capabilities. Warehouses A, B, and C can receive goods that are neither damaged nor oversized. Warehouse D is capable of receiving undamaged oversized cargo. Damaged goods can be delivered to selector E. The costs of the storage service providers are ordered as follows:

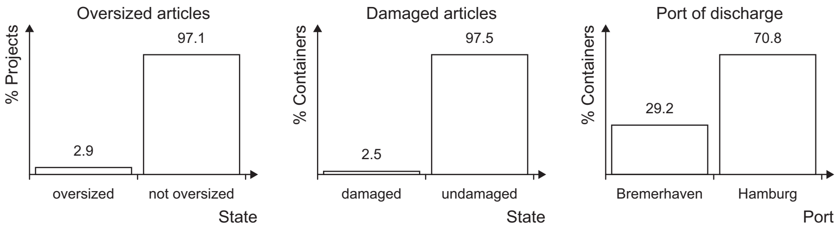

Fig. 10.12 Probability distributions for oversized articles, damaged articles, and the port of discharge, respectively.

$$cost(A) < cost(C) < cost(B) \tag{10.19}$$

Storage facilities D and E are not considered in Equation 10.19. They offer their services only for damaged or oversized goods and do thus not compete with warehouses A, B, and C.

The different locations are connected by several means of transport, namely barge, train, and truck. Their costs are ordered as follows:

$$cost(Barge) < cost(Train) < cost(Truck) \tag{10.20}$$

This order is in accordance with the expectation that employing means of mass transport is cheaper than individual transport of shipping containers. As elaborated in Section 5.1.2, means of mass transport have a minimum utilisation, i. e., they are only cheaper if they are jointly employed by a certain number of shipping containers (Equation 5.14). The respective values for minimum utilisation that should be met for cost efficiency are also given in Figure 10.13.

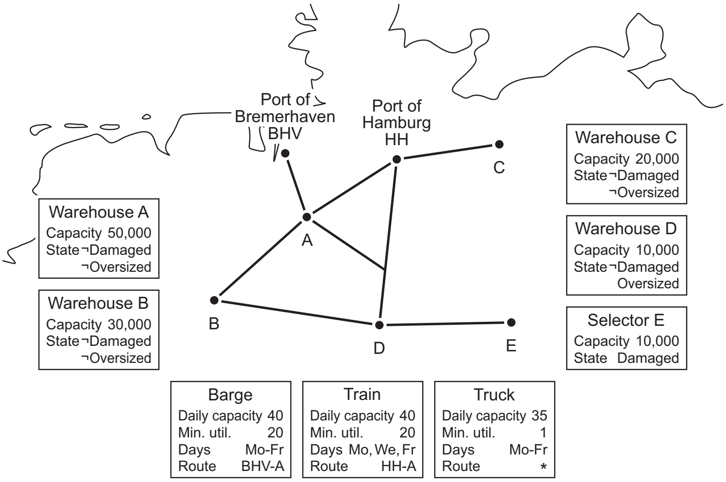

Fig. 10.13 Overview of the logistics network underlying the simulation experiments. The participating logistics service providers are annotated with relevant properties.

The logistics network presented in Figure 10.13 is similar to the original one examined in the case study (Chapter 9). As a difference, the original network comprises some additional storage facilities. However, these extra warehouses are mainly used for goods returning from the outlets. The exact properties of logistics service providers such as location and costs are modified appropriately. In particular, costs are only considered by their qualitative orders (Equations 10.19 and 10.20) because they constitute confidential data. This abstraction, however, does still allow evaluating the system behaviour because it is sufficient to qualitatively compare costs for decision-making. Furthermore, the capacity of the service providers has also been adapted. In the real process, the central high-bay warehouse has a capacity of $200,000$ pallets (Section 9.2.1). However, not all storage positions are available for the process examined. A part is occupied by material flows from domestic suppliers, another by returning goods. Another reason for decreasing its capacity in the simulation is to demonstrate how autonomous logistics deals with limited resources.

10.3.3 Utilisation of Storage Resources

Based on the introduced simulation setup (Section 10.3.2), this section examines the utilisation of storage resources. To recapitulate, the storage facilities differ in their capability of receiving specific goods, their costs, as well as their capacity. Following the order of these attributes, the first question to be answered is whether all shipping containers choose an appropriate warehouse for their cargo. According to the cargo properties in Figure 10.12, some articles are oversized or damaged. In the logistics network, warehouse D is capable of receiving oversized goods. Damaged articles must be delivered to selector E for repair. All other cargo can be received in warehouses A, B, and C. Figure 10.14 shows how the goods delivered by container are distributed to the storage providers.

The time scale of the storage utilisation depicted in Figure 10.14 is as follows. The fill level of each storage facility is given for one year with a resolution of one day. In addition to the 52 weeks under investigation, the graphs additionally show the development for some weeks of the previous year. This time span can be identified as the starting period. In these first weeks, goods are only delivered to the warehouses. By contrast, no goods are shipped because the distribution process starts not before week 0. At the end of the year, the situation is the other way round. Only few cargo is still delivered to the warehouses because sales phases of the subsequent year are not under consideration here. Nevertheless, the goods intended for the current year are still shipped. This means that the warehouse fill levels decrease continuously. In real-world processes, the sales phases of two years follow each other directly. Consequently, there is no such interruption in which storage facilities idle. Hence, the particular interest in this investigation is on the time between starting and end period.

In accordance with their properties, oversized and damaged goods are correctly assigned to the respective storage providers. That is, the negotiations between shipping containers and storage providers actually ensure that damaged goods are delivered to selector E and that oversized goods are received in warehouse D. Apart from choosing the right service provider in terms of the storage capabilities, also the costs of logistics services are an important

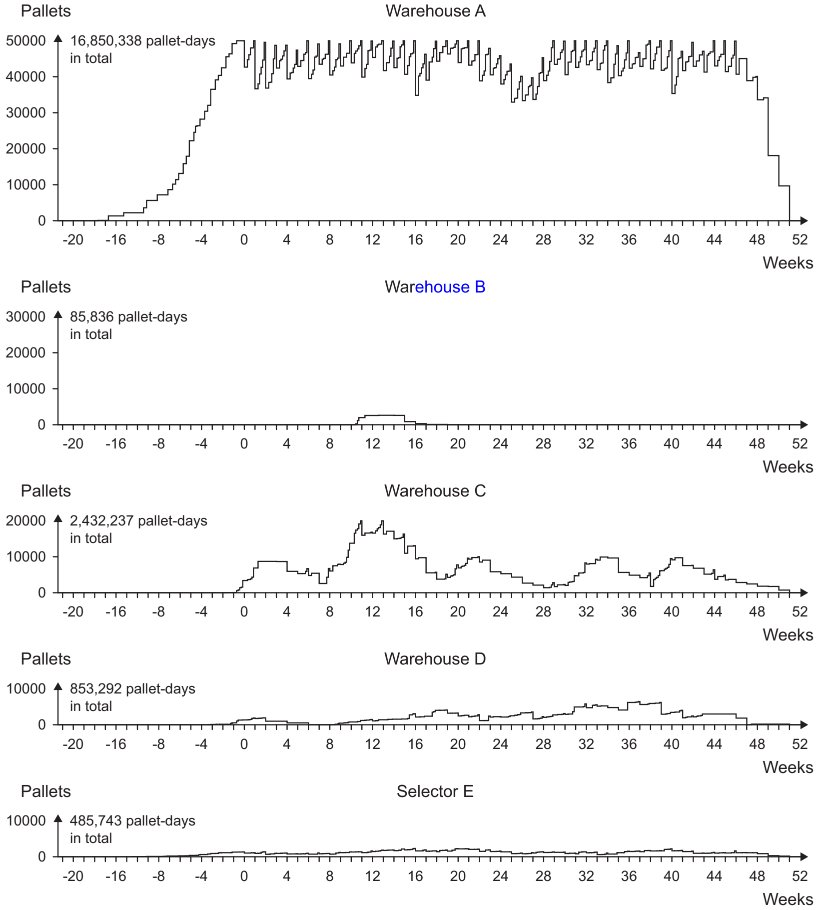

Fig. 10.14 Space utilisation in the storage facilities. Warehouse D and selector E store only oversized and damaged articles, respectively. The other goods are received by warehouses A, B, and C.

factor. Shipping containers without special properties can choose from three options, namely warehouses A, B, and C. With respect to the prices, it is desirable that warehouse A is utilised whenever possible (Equation 10.19). The graphs in Figure 10.14 show that warehouse A is indeed utilised as long as it has sufficient capacity. For the most time of the starting period, warehouse A has sufficient capacity to receive all cargo. As soon as the distribution process starts, a certain amount of cargo leaves the storage facility every week. This results in the sawtooth-like graph (Figure 10.14 top). The cargo shipped is scheduled for sale at the respective date. After shipping these pallets, there is again capacity for receiving new goods.

Storing goods at warehouse C is more expensive than at warehouse A (Equation 10.19). Consequently, the utilisation of warehouse C starts not until the storage resources needed exceed the capacity of warehouse A. During a certain time span, even the joint capacity of warehouses A and C does not suffice. Consequently, some goods are received at warehouse B which is an even more expensive alternative (Equation 10.19). Note that the utilisation of alternative storage facilities does not end immediately as soon as there is again capacity at the primary facility. This can be explained by the fact that goods are generally not transferred between warehouses after they have been received because additional transport costs would arise.

To summarise the findings so far, general cargo units choose the right service providers with respect to their properties, the costs, and the capacity available. As elaborated in Section 9.2.3, however, additional rules have to be considered for process control. In particular, similar goods should be received in the same storage facility in order to decrease costs for subsequent transport. Figure 10.15 compares the fragmentation of projects to storage facilities for different resource allocation strategies:

1. Individual allocation (the strategy examined so far)
2. Optimistic joint allocation
3. Conservative joint allocation

Note that Figure 10.15 only examines storage facilities A, B, and C because intended fragmentation is not considered here. In particular, intended fragmentation means that damaged goods are separated from others and delivered directly to selector E.

Independently from a specific strategy, there is no fragmentation as long as the primary warehouse has sufficient capacity because all shipping containers can employ the primary warehouse, i. e., the cheapest one that matches. Figures 10.14 and 10.15 reveal that the project fragmentation increases when the capacity of the primary storage facility is exhausted. If every shipping container acts individually, the highest degree of fragmentation arises (Figure 10.15 top). This can be explained by the fact that shipping containers simply choose the storage facility that fits their individual demands best. They do not explicitly consider the objective of keeping the project together. On average over all projects, the projects are therefore distributed to up to 1.55 warehouses.

Joint resource allocation (Algorithm 7.2) aims at decreasing project fragmentation. Two strategies can be distinguished, namely optimistic and conservative joint allocation. In the optimistic approach (Section 7.2.2), shipping containers choose a specific storage facility as early as possible. Making decisions early, however, constrains the knowledge that can be incorporated. In particular, only properties of prior team members are known, i. e., the containers do not have any knowledge about future team members. Hence, they cannot estimate how much storage space the whole team of shipping containers will probably need. Nevertheless, optimistic joint allocation already

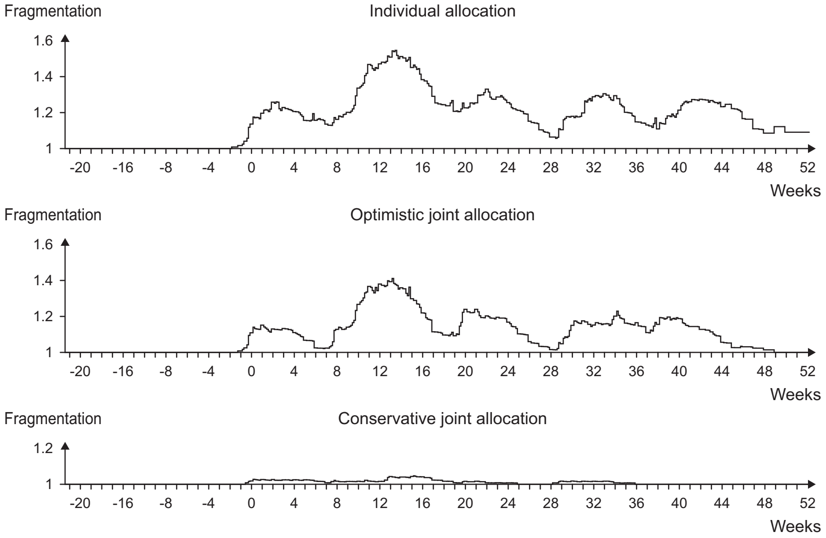

Fig. 10.15 Degree of project fragmentation for different allocation strategies. If storage capacity is allocated individually, projects are most distributed. Optimistic joint allocation reduces the degree of distribution. The conservative approach almost completely prevents fragmentation.

decreases the maximum daily average project fragmentation from 1.55 to 1.41 (Figure 10.15 middle).

The conservative counterpart makes decisions as late as possible (Section 7.2.3). Deferring decision-making enables collecting the demands of as many team members as possible. In this approach, shipping containers jointly request storage capacity as late as possible. Hence, it can be better ensured that there is actually enough space in the storage facility of their choice. Not surprisingly, this procedure significantly reduces project fragmentation to a maximum daily average of 1.05 (Figure 10.15 bottom). The remaining fragmentation can be explained by two facts. On the one hand, large projects must be divided to multiple warehouses due to their size. On the other hand, single shipping containers may join a storage team after the other team members have already requested the storage capacity. It may then still happen, that the storage capacity of the warehouse chosen is exhausted so that it cannot receive additional goods.

These results show that autonomous logistics entities can actually minimise project fragmentation by coordination. However, minimising project fragmentation also has drawbacks. Figure 10.16 compares the percentage utilisation of the primary and alternative storage providers for the allocation strategies. This comparison is based on the total number of pallet-days utilised. The amount of pallet-days for a storage provider can be obtained by summing up the amounts of all utilisations (Definition 5.15) multiplied with their respective interval durations:

$$\sum_{\substack{(Interval, amount)\\ \in Utilisations}} amount \cdot duration(Interval) \tag{10.21}$$

The *duration* mapping in Equation 10.21 is defined based on Equation 5.36.

The first three diagrams in Figure 10.16 compare the three allocation strategies for storage resources combined with an optimistic allocation of transport resources. The comparison shows that the desired utilisation of the primary storage facility decreases from individual over optimistic to conservative allocation by about two percentage points each. The utilisation of warehouse D and selector E is highlighted by dashed lines because the respective cargo cannot be received in the primary warehouse. Apart from rounding inaccuracy, this amount does not depend on the strategy chosen.

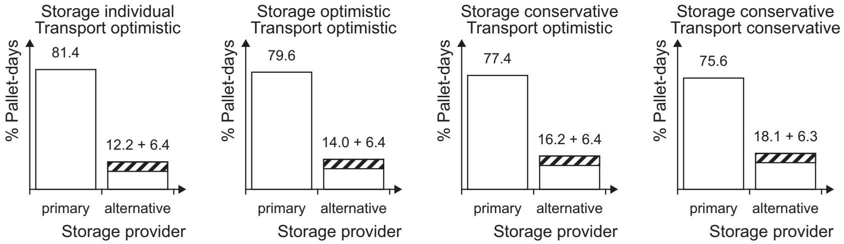

Fig. 10.16 Utilisation of storage providers with different logistics strategies measured in pallet-days. The utilisation of alternative storage providers summarises the pallet-days of B, C, D, and E. The dashed lines highlight the utilisation of storage facilities D and E because the primary warehouse A is not capable of receiving the respective cargo.

The decrease in the utilisation of the primary storage provider can be explained as follows. If resources are allocated individually, every single shipping container can choose a storage provider independently. Optimistic joint allocation constrains this choice. If the first part of a project is already stored in an alternative storage facility, also the other ones are received there. The capacity actually used at the primary storage facility decreases even more if storage resources are allocated conservatively. This can be explained by the fact that resource allocation is deferred in this case. Early allocation first reserves a certain amount of storage capacity and releases unused capacity later. The tight schedule of conservative allocation decreases the possibility for re-use of released capacity.

To summarise the examination of storage utilisation, autonomous logistics entities can actually choose the right storage providers with respect to their properties, the costs, and the capacity available. This results in a desired overall system behaviour in which cheapest matching resources are primarily utilised. By coordinating each other, logistics entities can also effectively prevent project fragmentation to different storage providers. Different strategies for storage resource allocation have been analysed. The choice for a specific strategy is a tradeoff between preventing project fragmentation and utilising the primary storage facility as much as possible. The best strategy for a specific process depends on its respective cost structure.

10.3.4 Utilisation of Transport Resources

So far, the performance evaluation of autonomous logistics entities primarily focused on the utilisation of storage resources. This section broadens the perspective and investigates also the utilisation of transport resources and respective interdependencies. To recapitulate, there are three different means of transport available in the logistics network (Figure 10.13). Barges connecting Bremerhaven to warehouse A, trains connecting Hamburg to warehouse A, as well as trucks for all other routes. From the cost perspective, it is desirable to transport as much containers as possible by means of mass transport (Sections 5.2.2 and 9.3.4), i. e., barge and train.

As elaborated in Section 5.1.2, individual allocation of means of mass transport is not reasonable. Hence, only joint allocation is investigated here:

1. Optimistic joint allocation
2. Conservative joint allocation

For the investigations in the preceding Section 10.3.3, optimistic joint allocation (Section 7.2.2) is the chosen transport resource allocation strategy. Figure 10.17 shows how the means of transport are utilised with this strategy. The graphs show that barge and train are actually the preferred means of transport. They are chosen whenever it is possible. Truck must be employed in order to transport shipping containers to alternative storage facilities because only the primary warehouse is connected by barge and train. That is, the minimum utilisation of trucks is defined by the amount of containers that are received in alternative storage facilities (Figure 10.16). Peaks in the utilisation of trucks in Figure 10.17 actually correlate with an increased utilisation of alternative storage facilities in Figure 10.14. Besides, trucks are employed if the demand for transport resources exceeds the capacity of barges and trains.

A closer investigation of Figure 10.17 reveals a drawback of the optimistic allocation strategy. With this strategy, autonomous logistics entities choose means of mass transport even if they cannot be sure that the minimum utilisation will be met. Although subsequent entities prefer this pre-selected capacity, resources remain sometimes unused. Figure 10.17 shows that this happens sometimes for barges and trains. An alternative is conservative allocation of transport resources (Section 7.2.3). In this approach, resource allocation is

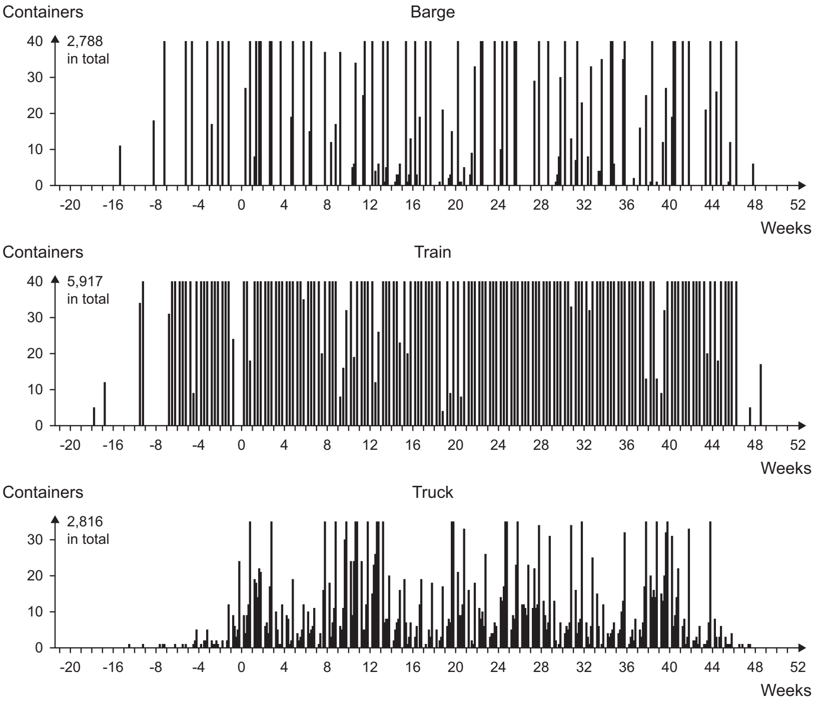

Fig. 10.17 Utilisation for transport resources with optimistic allocation. In accordance with Figure 10.14, individual allocation is applied for storage resources.

deferred until there is sufficient demand for means of mass transport. That is, it can be guaranteed that the minimum utilisation of barges and trains will be met. If there are not enough containers, individual means of transport must be employed. This means that the total number of trucks employed increases. In turn, empty vehicle running is decreased significantly for means of mass transport, i. e., the total costs are still decreased.

Figure 10.18 compares the resource allocation strategies with respect to their utilisation of means of transport. The leftmost diagram is an aggregation of the graphs depicted in Figure 10.17. The diagrams in Figure 10.18 distinguish means of mass transport on the one hand and individual means of transport on the other hand. The optimistically allocated but actually unused capacity of means of mass transport is highlighted by dashed lines. As expected, there is no allocated but unused capacity if transport capacity is allocated conservatively (Figure 10.18 right). The determination of unused capacity is based on the minimum utilisation of the means of transport (Figure 10.13). Apart from one exception, it can be observed that the utilisation of individual means of transport increases from the left to the right. This effect is due to the fact that, with more restrictive storage allocation strategies, the utilisation of alternative storage facilities increases (Figure 10.16). This results in an increased truck utilisation because the alternative storage facilities can only be reached by truck (Figure 10.13).

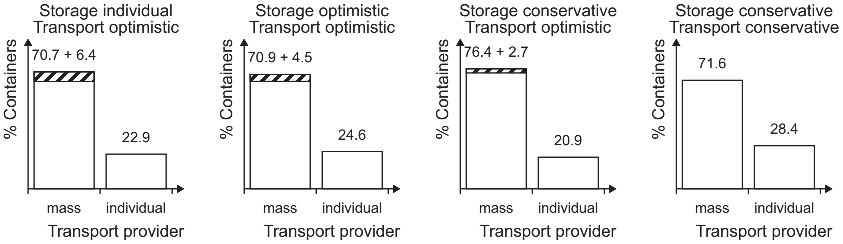

Fig. 10.18 Utilisation of the means of mass transport barge and train in comparison to trucks as individual means of transport. The diagrams compare the utilisation distributions for different allocation strategies. The dashed lines mark capacity that is optimistically allocated but unused.

As an exception from this observation, however, conservative storage allocation combined with optimistic transport allocation achieves the highest utilisation for means of mass transport. This can be explained as follows. In the first two cases in Figure 10.18, individual and optimistic joint storage allocation are applied, respectively. In these strategies, decisions for storage providers are made as early as possible, e. g., as soon as the shipping container is loaded onto a container vessel. Subsequently, also transport resources for onward carriage are chosen early. However, the transport duration of shipping containers differs (Figure 10.11). This first-come, first-served (FCFS) approach means that some shipping containers can allocate their resources much earlier than others. As a consequence, they restrict the options for later containers. This particularly affects the train. On the one hand, the train is the means of mass transport that is used most (Figure 10.17). On the other hand, it is the means of transport with the lowest weekly capacity (Figure 10.13). If conservative joint storage allocation is applied, also the decision for transport resources is deferred. Hence, the individual benefit of joining the network early is abolished and transport resources can be allocated more fairly, i. e., rather priority-based than FCFS. Resulting from this, some shipping containers that are transported by train in the first combination of strategies cannot employ a train. The overall utilisation of trains, however, is increased with the third combination of strategies.

In the rightmost case depicted in Figure 10.18, the utilisation of means of mass transport is decreased again. This is due to the fact that allocated but actually unused transport capacity is completely prevented. That is, trucks are employed if the minimum utilisation for means of mass transport is not met. However, this strategy still leads to a decrease in total costs because also optimistically allocated but actually unused transport capacity must be fully paid for. Nevertheless, it is worth mentioning that conservative transport allocation also effects the utilisation of storage resources (Figure 10.16 right). By deferring the decision for a transport provider even further, the possibility for re-using released storage capacity is further decreased (cf. Section 10.3.3).

Another important criterion for measuring the performance of process control is to which degree shipping containers exploit the free storage time at container terminals. To recapitulate, shipping containers are granted a certain time during which they remain at the terminal for free (Section 5.1.4). This means that one can reduce chargeable storage utilisation in the warehouses. In practice, it depends on factors such as the specific container terminal, shipping company, and contract when demurrage and detention are charged (Section 9.3.1). For the simulation experiments, a free time of 22 days is assumed for all shipping containers. By applying variable temporal intervals (Definition 7.3), autonomous logistics entities can specify that they want to use this time to save storage costs. Figure 10.19 compares how the different strategies succeed in this attempt.

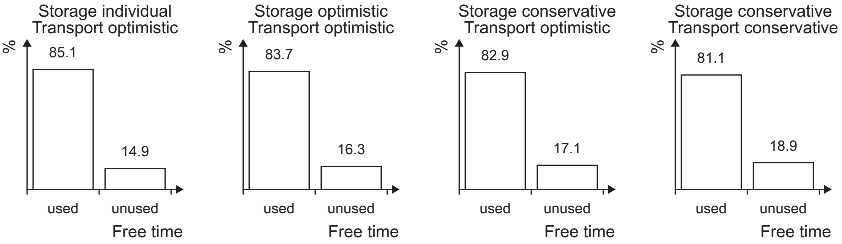

Fig. 10.19 Exploiting the free storage time at the container terminal maximally, i. e., collecting containers as late as possible, decreases storage costs. The diagrams compare how different strategies use this potential.

A free time exploitation of 100% is most unlikely because the exploitation depends on the availability of transport resources. In particular, there are no transports on weekends. Furthermore, most shipping containers are transported by train (Figure 10.17) which is only available on Mondays, Wednesdays, and Fridays (Figure 10.13). The highest exploitation of free times is achieved with individual allocation of storage resources and optimistic joint allocation of transport resources (Figure 10.19 left). The free times are used to 85.1% for this configuration. The diagrams in Figure 10.19 show that the utilisation of free times decreases from 85.1% for individual and optimistic to 81.1% for conservative allocation. This observation can be explained as follows. In the case of individual allocation, shipping containers have the highest degree of freedom in decision-making. That is, they can choose their storage and transport provider without or with only little external constraints. In the strategy combinations compared in Figure 10.19, these constraints increase from the left hand side to the right hand side. With joint allocation of transport resources, shipping containers can no longer decide freely on their own but must coordinate with others. This can particularly mean deviating from desired transport dates and therefore also affect the exploitation of free times at the container terminals.

Finally, it is important to examine how much time dispatching takes if autonomous logistics entities are responsible for process control. All simulation experiments presented in this chapter have been conducted on a computer with eight dual-core AMD Opteron 8218 processors with 2.6 GHz each and 64 GB RAM in total. Note, however, that the number of processors is not fully utilised because the number of software agents acting in parallel is usually limited. For the individual and the optimistic joint allocation strategies, the simulation takes between 15 and 16 minutes. In these cases, the agents are seldom executed on more than one CPU core. This can be explained by the fact, that the shipping container agents act more or less individually. As a consequence, they complete their individual dispatch task and are finished afterwards. If conservative allocation is applied, the agent lifetimes are longer because agents must wait for others before they can choose service providers. Conservative storage allocation combined with optimistic transport allocation takes about 40 minutes. Allocating both storage and transport resources conservatively takes approximately 50 minutes. In these cases, more agents act in parallel. As a consequence, on average three and four CPU cores are utilised, respectively. However, even if dispatch takes 50 minutes in total, the duration for dispatching each of the $11,521$ shipping containers is less than a second. This means that the autonomous logistics entities make their decisions significantly faster than a human dispatcher could do.

To summarise the examination of transport utilisation, autonomous logistics entities can actually choose the right transport providers with respect to the costs and the capacity available. Different strategies can be applied for resource allocation. The optimistic joint allocation strategy allows making decisions early. As a shortcoming of this approach, however, capacity allocated from means of mass transport may be not fully used. This can be prevented by the conservative joint allocation strategy that defers decision-making. It ensures that means of mass transport are only employed if there is sufficient demand to meet a minimum utilisation. Another objective is utilising free times at container terminals which allows saving costs by minimising storage utilisation. The objectives of choosing the individually best storage and transport provider, minimising project fragmentation, minimising allocated but unused capacity, as well as maximally utilising free times at the terminals are interrelated. That is, optimising one property may downgrade another one. Nevertheless, the simulation results presented in Sections 10.3.3 and 10.3.4 show that process control by autonomous logistics is not only effective but also efficient. In particular, the overall system behaviour emerging from the local behaviour of the participating entities utilises logistics resources in a desired way, i. e., cheapest matching resources are utilised as much as possible. Furthermore, global and potentially contradictory objectives can be approached successfully by cooperation despite of the decentralised approach of autonomous logistics.

10.3.5 Comparison to Present Process Control

At present, the original process examined in this evaluation is controlled manually by a human dispatcher (Section 9.2.3). This dispatcher is supported by information systems that provide relevant information. Consequently, these information systems will also be the foundation for decision-making if autonomous logistics entities take over process control. The investigations in the preceding Sections 10.3.3 and 10.3.4 show that the individual behaviour of the autonomous logistics entities emerges to a desired overall system behaviour. Furthermore, it has been shown that automated dispatch is significantly faster than a human dispatcher. Nevertheless, an open question is how the local optimum reached by the autonomous logistics system relates to the result of the human dispatcher. This question must be answered in order to decide whether autonomous logistics can be reasonably applied in order to support the human dispatcher. The foundation for this investigation is the real-world dispatch data from eight sales phases in 2006 that has already been analysed to generate synthetic test data (Section 10.3.2).

The logistics network underlying the simulation experiments so far is depicted in Figure 10.13. As explained in Section 10.3.2, there is an intentional deviation from the original network. More precisely, the capacity of the primary storage facility has been decreased $50,000$ pallets in order to create a more challenging setting for the autonomous logistics entities. In particular, the decreased capacity causes higher demands regarding the choice of alternative warehouses as well as the prevention of fragmentation of similar goods to different warehouses (Section 10.3.3). To recapitulate, the original high-bay warehouse has a capacity of $200,000$ pallets (Section 9.2.1). To permit comparability with the original dispatch results, the capacity is thus assumed to be $200,000$ in the further examination.

The effect of this change is far-reaching. It means that regular cargo can almost always be completely received in the central high-bay warehouses. At least, this can be observed in the real-world dispatch data for the examined time span. The choice for an appropriate warehouse is thus easy for both the human dispatcher and the autonomous logistics system. As a consequence, project fragmentation also does not pose a major problem for this time span. Regarding transport, it is important to utilise means of mass transport. Barge and train, the means of mass transport participating in the process, are connected to the central high-bay warehouse. This means utilising them is not challenging because most shipping containers have to be transported to the respective warehouse. As a consequence, it does not pose a major challenge to both the human dispatcher and the autonomous logistics system to consider four important criteria. At least, this holds for the span of eight weeks for which real-world dispatch data is available for investigation. Remember, however, that it has been shown in Sections 10.3.3 and 10.3.4 that the autonomous logistics system is capable of handling even more demanding situations.

So far, the discussion of the parameters considered does not cover the utilisation of free times at the container terminal. Nevertheless, this is an important parameter as it allows saving storage costs at the warehouses. The evaluation in Section 10.3.4 already shows that the autonomous logistics system is capable of considering this property for synthetical data. To this end, a fixed free time has been assumed for all containers. In reality, considering this property is more demanding because containers have different free times depending on the specific terminal, shipping company, and contract (Section 9.3.1). Consequently, it can be assumed that this objective is only partly considered by human dispatchers because other objectives are more important for effective process control.

The foundation for examining this assumption is real container data from eight sales phases in 2006. The data covers 1,645 shipping containers corresponding to 78,280 pallets. In order to circumvent the starting period discovered in Section 10.3.3, the original data is embedded into the synthetical data employed already in the previous sections. That is, eight sales phases in the synthetical data have been replaced by the original data. To recapitulate, the synthetical data has been generated based on probability distributions derived from the original data. The subsequent dispatch analysis is then performed only with results for the original containers. The network structure is as depicted in Figure 10.13 with the capacity of 200,000 pallets for warehouse A being the only exception. The allocation strategies applied are conservative for both storage and transport resources.

An analysis of the real-world dispatch data reveals that the human dispatcher utilises 45.5% of the free times at the container terminal. In the multiagent-based simulation, the autonomous logistics system can exploit even 68.3% of the free times. This is 22.8 percentage points and 50.2% better than the result of the human dispatcher. In absolute figures, transports are deferred by about five days on average. In total, this makes 392,720 pallet-days of storage that can be saved in the investigated time span of eight sales phases. Extrapolating this to 52 sales phases per year, about 2.6 million pallet-days could be saved annually. On average, this corresponds to a warehouse with about 7,000 storage positions for pallets. This means that there is a high potential for saving costs. This potential can be exploited in two ways. Firstly, it is possible to utilise the free time at the container terminal more efficiently and save storage costs at the warehouses. Secondly, it is possible to negotiate shorter free times with the carriers and thus save transport costs. This is possible because the autonomous logistics system is capable of considering this parameter reliably in the dispatch process. To summarise, autonomous logistics entities can actually support human dispatchers for standard cases.

Furthermore, the autonomous logistics system helps save costs by improved process control.

10.4 Conclusion

The evaluation conducted in this chapter is twofold. It approaches autonomous logistics both analytically and by simulation. The analytical examination reveals that there is a potential for cooperation of the participating logistics entities because the interaction effort for coordination can be significantly decreased. This means that process control by autonomous logistics entities is more efficient if they decide for cooperation and against purely individual action. Furthermore, also the resource utilisation efficiency can be decreased by cooperation. These insights allow deriving an appropriate degree for autonomous control. Cooperation in autonomous logistics can be initiated by different team formation interaction protocols. The effort for team formation depends on the specific interaction protocol chosen. From the arising interaction effort, also limitations for autonomous control can be derived. A thorough investigation of the interaction effort and other relevant protocol properties provides the means to choose an appropriate team formation protocol for specific purposes.

The actual application of autonomous logistics to process control is examined by means of multiagent-based simulation. To this end, a real-world onward carriage process in container logistics is investigated. The findings resulting from the simulation experiment are twofold. On the one hand, it can be shown that process control in onward carriage can be accomplished based on the multiagent-based implementation presented in this thesis. This means that the approach is effective in covering requirements from industry. On the other hand, the simulation results also show the efficiency of process control by autonomous logistics entities, i. e., a desired global system behaviour emerges from the aggregated behaviour of the local entities.

Different strategies can be applied in order to satisfy the logistics objectives. The strategies distinguish in the knowledge that is considered for decision-making and in the point of time at which decisions are made. A comparison reveals that coordinating logistics entities can decrease the fragmentation of similar cargo to different storage facilities. Deferring decision-making is the foundation for incorporating additional knowledge. It can be shown that the fragmentation can be further decreased by deferred decision-making. Likewise, the additional knowledge helps utilising means of mass transport more efficiently. Another criterion that is relevant in the examined process is the utilisation of times during which shipping containers can remain at the container terminal without being charged for storage. The different objectives are interdependent, i. e., optimising one may downgrade another. This makes it particularly challenging for dispatchers to incorporate all objectives adequately. The multiagent-based simulation results show that the autonomous logistics entities performs better than human dispatchers. On the one hand, the autonomous logistics systems makes decisions faster than a human. On the other hand, it is more reliable in considering the logistics objectives. In particular, it has been shown that the autonomous logistics entities utilise free times at the container terminal better than the human dispatcher. This helps saving storage resources and thus money.

To conclude, the simulation of the onward carriage process shows that autonomous control in logistics is actually possible. The dispatch result is a local optimum which ensures that the cheapest matching resources are utilised as much as possible. Besides, additional overall objectives can be reflected by cooperation despite of the decentralised approach of autonomous logistics. This means that autonomous logistics can actually support human dispatchers in their work. In particular, standard situations can be handled efficiently by the systems. In case of exceptional situations, human dispatchers can be informed accordingly. Disburdening humans from standard cases means that they can focus on exceptional cases and spent more time on solving these challenges more efficiently.

References

Dörfler, W. & Peschek, W. (1988). *Einführung in die Mathematik für Informatiker.* Munich, Germany: Carl Hanser Verlag.

Krcmar, H. (2005). *Informationsmanagement.* Heidelberg, Germany: Springer-Verlag.

Lint, J. H. van & Wilson, R. M. (1992). *A Course in Combinatorics.* Cambridge, UK: Cambridge University Press.

Philipp, T., Beer, C. de, Windt, K. & Scholz-Reiter, B. (2007). Evaluation of Autonomous Logistic Processes – Analysis of the Influence of Structural Complexity. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics: The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 303–324). Heidelberg, Germany: Springer-Verlag.

Shakshuki, E. & Jun, Y. (2004). Multi-Agent Development Toolkits: An Evaluation. In R. Orchard, C. Yang & M. Ali (Eds.), *17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 2004)* (pp. 209–218). Ottawa, Canada: Springer-Verlag.

Tanenbaum, A. S. (2003). *Computer Networks* (4th ed.). Upper Saddle River, NJ, USA: Pearson Education.

Wenzel, S., Weiß, M., Collisi-Böhmer, S., Pitsch, H. & Rose, O. (2008). *Qualitätskriterien für die Simulation in Produktion und Logistik.* Heidelberg, Germany: Springer-Verlag.

Zoghbi, A. & Stojmenovic, I. (1998). Fast Algorithms for Generating Integer Partitions. *International Journal of Computer Mathematics*, *70*(2), 319–332.

Chapter 11
Conclusion and Outlook

This thesis has approached the field of autonomous control in logistics in an interdisciplinary way. Taking the requirements from the field of logistics as a starting point, multiagent technology derived from Distributed Artificial Intelligence has been identified as an appropriate means to implement autonomous logistics. A concept for team formation and team action in autonomous logistics has been developed and investigated exemplarily in a real-world process. This final chapter is intended to summarise findings of the preceding chapters and to draw overall conclusions.

Section 11.1 revisits the initial research questions (Section 1.1) and summarises the conclusions drawn in this thesis. Subsequently, Section 11.2 derives directions for future research in the field of autonomous logistics.

11.1 Research Questions Revisited

The introductory Chapter 1 lists four research questions that are dealt with in this thesis. Consequently, the following Chapters 2 to 10 elaborate on several aspects related to these topics. As an overall conclusion, this section summarises relevant findings guided by the initial questions (Section 1.1):

1. What constitutes autonomous control in logistics?
2. How can autonomous control in logistics be operationalised?
3. How important is cooperation for autonomous control in logistics?
4. How can autonomous control be applied to actual logistics processes?

To answer the question what actually constitutes autonomous control in logistics, the requirements from logistics are taken as a reference. Chapter 2 investigates supply network management as well as current and future trends in this field. It can be observed that the size of individual shipments decreases while the total number of shipments and receivers increases. Furthermore, also the number of stakeholders in logistics processes increases. These de-

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics*,
DOI 10.1007/978-3-642-20092-2_11, © Springer-Verlag Berlin Heidelberg 2011

velopments pose challenges for logistics control. In particular, limitations of previous centralistic approaches which are due to the complexity, the dynamics, and the distribution of logistics processes have been identified. For holistic supply network management that integrates all primary logistics functions, it is necessary to overcome these limitations. To this end, autonomous logistics divides supply networks into loosely, but intelligently coupled control systems. As elaborated in Chapter 3, the advantages over previous approaches are as follows. Autonomous logistics entities can react locally on exceptions without the necessity to re-schedule the whole system. Furthermore, it is not necessary to reveal internal decision-making over company boundaries. Finally, process control is even possible for physically distributed systems with limited or even without communication bandwidth.

This leads to the second question, namely how autonomous logistics can actually be operationalised. As a foundation, Chapter 3 contributes a comprehensive survey of the technologies enabling autonomous control. The technologies needed in order to implement autonomous logistics entities comprise identification, localisation, sensor, and communication technology. Although these technologies are necessary for autonomous control, they are not sufficient. Moreover, they can also be applied in order to improve conventional control. The crucial technology for autonomous logistics is decentralised data processing which enhances logistics entities with the capability for local decision-making. Decentralised data processing can be implemented with the intelligent software agent paradigm from Distributed Artificial Intelligence. Chapter 4 motivates this choice by the fact that the requirements for autonomous logistics entities are directly reflected by the definition of software agents. Software agents can be deployed either on embedded systems attached to the logistics objects or on a central server or server cluster. Even in the second alternative, the advantages of reduced computational complexity and the improved ability of coping with dynamics are retained. Chapter 5 identifies key participants in logistics networks. On the one hand, these are service consumers such as general cargo units. On the other hand, there are service providers which offer the primary logistics functions transport, handling, storage, and picking.

The question how autonomous control can be operationalised is closely linked with the third research question which focuses on the importance of cooperation. Chapter 5 gives two motivations why cooperation is necessary. Firstly, autonomous logistics entities can use logistics resources such as means of mass transport jointly. This helps satisfying the logistics objectives imposed by the cargo owner more efficiently. Secondly, a high interaction effort arises if every entity acts individually. Acting jointly prevents that the decrease in computational effort gained by autonomous logistics is outweighed by the increase in interaction effort. Possible scales for autonomous logistics entities are components, articles, sales units, cardboard boxes, pallets, or shipping containers. An appropriate degree for autonomous control is discussed in Chapter 10. The lower bound is defined by the arising interaction effort and an adequate resource utilisation efficiency. The upper bound is related to the synergy by joint action. The operationalisation of cooperation follows the model for cooperation by Wooldridge and Jennings. Chapter 5 identifies the potential for cooperation. Chapter 6 describes how individual autonomous logistics entities can form teams. Chapter 7 deals with joint action of the respective teams, i. e., inter-agent collaboration and intra-agent coordination. The actual implementation of this concept by means of multiagent systems is described in Chapter 8.

The first three research questions approach the field of autonomous logistics theoretically. The last question shifts the focus to the actual application. A particular focus is on the question how autonomous logistics performs in process control. The foundation for this investigation is the case study presented in Chapter 9. The case study examines a real-world onward carriage process in container logistics of a major retailer of consumer products in Germany. Due to the sales strategy of a weekly changing range of products, high demands regarding logistics complexity and dynamics arise. Hitherto, the process is manually controlled by a human dispatcher. The dispatcher is supported by information systems which provide him with relevant information for decision-making. These information systems are an interface to the underlying logistics systems and thus also the foundation for autonomous control. The question is to what extent autonomous logistics entities can take over process control if they are provided with the respective information. The simulation experiment conducted in Chapter 10 reveals that the local behaviour of the autonomous logistics entities does actually emerge to the desired overall system behaviour. In particular, the cheapest matching logistics service providers are employed whenever this is possible. Furthermore, global objectives, such as minimising fragmentation of similar goods to different providers, minimising allocated but actually unused resources, as well as maximising the utilisation of free times at container terminals, can also be successfully considered by coordinated actions. The autonomous logistics system makes decisions faster than a human and considers logistics objectives more reliably. As an example, this has been shown for the utilisation of free times at the container terminal. All decisions made by autonomous logistics entities are based on negotiations and thus comprehensible to humans. That is, autonomous logistics can actually support the human dispatcher in standard cases. Relieved from standard cases, the human dispatcher is able to focus on more challenging exceptional cases that cannot be handled by the autonomous logistics entities.

The contribution of this thesis is not limited to the onward carriage process investigated in Chapter 10. By contrast, it provides general interaction schemes for cooperating autonomous logistics entities. The particular focus of this thesis is on interaction of such entities. Consequently, these agent representatives are not restricted regarding their internal decision-making, i. e., a broad bandwidth of agent implementations can interact based on the protocols developed in this project. The bandwidth for possible internal agent modelling ranges from conventional control methods to autonomously controlled subsystems which are encapsulated by an agent representative. Furthermore, this thesis presents and compares different resource allocation strategies for autonomous logistics. As elaborated in Chapter 8, multiagent-based simulation allows testing the actual behaviour of agents intended for real-world operation in simulation. The effects of individual or combined strategies in a specific process can thus be examined by multiagent-based simulation even before deploying the agents to the actual process.

11.2 Directions for Future Research

The paradigm of autonomous control in logistics is a wide field (Chapter 3). Consequently, one thesis can only approach selected aspects of this topic. Several open questions point out directions for future research. The open questions arising from the thesis at hand pertain to two categories. Section 11.2.1 addresses those questions pertaining to the area of inter-agent collaboration, the main focus of this thesis. Subsequently, Section 11.2.2 shifts the perspective, thereby focusing on modelling the internal decision-making of autonomous logistics entities.

11.2.1 Inter-Agent Collaboration

This research project has developed an approach for decentralised process control by autonomous logistics entities. The application to a real-world process, namely onward carriage in container logistics, has been studied in Chapter 10. Although the participating logistics entities are not known in advance, at least the companies participating in the process are known. That is, the autonomous logistics entities allocate resources from pre-negotiated contracts with service providers. The idea of autonomous logistics, however, is not limited to controlling single processes with pre-negotiated contracts. By contrast, networks of autonomous logistics entities have the potential to serve multiple processes in parallel. Applying autonomous logistics on this scale raises additional challenges for inter-agent collaboration:

1. Integrating additional service providers and consumers dynamically
2. Discovering service providers and consumers in large-scale networks
3. Coordinating logistics entities on different control layers

Integrating additional service providers and consumers from different companies on demand also impacts team formation. This thesis particularly focuses on team formation of similar entities with joint objectives (Chapter 6). In the onward carriage process examined, the entities within one team pertain to the same company while different teams may pertain to different companies. Consider a team of service consumers of one company that negotiate with different service providers (Chapter 7). In general, even large companies may be incapable of satisfying complex logistics demands requested by their consumers. For instance, transport over long distances incorporating several modes of transport may require cooperation even over company boundaries. From the service consumer perspective, the same applies if entities of one company cannot meet minimal lot sizes, e. g., for means of mass transport. Both aspects raise the question how joint action of autonomous logistics entities can be organised over company boundaries.

One answer might be the emergence of fourth-party logistics providers (Section 2.2.1), a concept that is hitherto often criticised as being too visionary. Acting as a uniform contact person to their customers, 4PL integrate services by other providers. Therewith, they are able to satisfy consumer demands without executing logistics functions themselves. This aspect of the paradigm of autonomous logistics might lower the barriers to entry the logistics market. Even small logistics providers offering highly specialised services could thus participate in solving complex logistics tasks.

Integrating a high number of companies leads to another challenge for autonomous logistics, namely service discovery. The question is how one can efficiently discover potential cooperation partners. This aspect is closely related to the findings regarding the interaction effort (Section 10.2) of the team formation interaction protocols (Chapter 6). In some protocols, the interaction effort increases significantly with the number of participants. By contrast, a broker can decrease the interaction complexity. However, it constitutes a single point of failure. This limitation can be addressed by distributing the broker role over multiple agents which internally coordinate with each other (cf. Bellifemine, Caire & Greenwood, 2007, p. 34). However, this solution would still depend on a number of exposed agents. Hence, the question is how the number of responders in the decentralised multicast-based team formation protocol (Section 6.2.3) can be reduced in an adequate way. As already discussed in Section 7.1.2, the regions of relevance proposed by Gehrke (2009, pp. 103–104) are a promising approach for this purpose. Respective conceptual, spatial, and temporal criteria can be derived directly from the service demands (Definition 7.1) specified in Section 7.1.1.

For process coordination, it is important that relevant information about logistics entities is available to authorised participants. If all representatives of logistics objects populate a central server or server cluster, it is possible to ask them directly about their current state. In the case of a physically distributed application, however, it is neither desirable nor reasonable to contact the individual logistics entity in order to learn about its current state. Hribernik, Hans and Thoben (2009) examine how the Internet of Things (Section 3.2.1) can be employed to facilitate autonomous logistics processes. Indeed, their initial investigation shows that the Internet of Things can increase supply network visibility without burdening local logistics entities with status requests. To this end, relevant information can be passed to EPC information services (Figure 3.6) which then answer requests about the current state of the respective entities. This application indicates that the actual integration of the Internet of Things and the autonomous logistics paradigm is another relevant direction for future research.

Autonomous control in logistics can be applied on different layers. The approach introduced in Chapters 5 to 7 focuses on coordinating multiple primary logistics functions. However, there are also approaches to apply the decentralised control paradigm to individual functions. Due to the inherent complexity of transport networks, it is especially promising for the transport primary logistics function (Section 3.1.1). For instance, Wenning, Rekersbrink, Timm-Giel, Görg and Scholz-Reiter (2007) investigate the application of computer network routing algorithms to transport networks. Also this application exhibits a problem of interaction effort. Therefore, a clustering algorithm that groups packages in distribution centres for joint routing has been proposed by Singh, Wenning, Singh and Görg (2007). An important research question is how both layers of autonomous control, i. e., route planning and holistic supply network management, can be combined. A particular subquestion is whether agents and teams from the supply network perspective can be transferred directly to transport network routing.

11.2.2 Intra-Agent Coordination

In this thesis, the focus is on coordinating autonomous logistics entities. Consequently, the discussion of directions for future research covered logistics networks with a high number of participants so far. Besides, it is important to examine the participating autonomous logistics entities themselves. A particular focus in this context lies on the internal modelling of autonomous logistics entities (Section 4.1.2). Following the commitment of agents to their teams (Definition 5.14), two different cases can be distinguished. On the one hand, multiple agents acting as a team may be merged into one agent which then acts on their behalf. On the other hand, the individual participants may act either autonomously or in an association. Then, one or multiple agents act as representatives to the outside world. For joint action, however, all participants coordinate each other internally.

To coordinate agents within one team internally, it is possible to apply autonomous control also on this level. One example is the application of network routing protocols (Section 11.2.1) to model transport service providers. Cooperation has been identified as important for process control by autonomous logistics entities (Sections 5.2 and 10.1). The interaction mechanisms developed in this thesis help logistics entities achieve their individual logistics objectives more efficiently. Apart from individual goals, however, it is also important to achieve global objectives for the whole company. Following the model for cooperation (Section 4.3.2), an important research question is how the plan formation step can be extended to consider joint objectives explicitly. For instance, Pantke (2009) proposes user-configurable key performance indicators for intelligent agents. Autonomous logistics entities should employ these indicators to coordinate their individual actions towards a global optimum.

As an alternative, multiple agents may be merged into one if they have sufficiently similar objectives (cf. Section 10.1.3). Given that the original number of team participants is manageable, it is possible to apply conventional control methods within the scope of the original participants. This means that the encapsulation by software agents allows resorting to control methods from operational research where they can be applied appropriately with a manageable number of objects and parameters (Section 2.3). The actual combination of autonomous and conventional control is thus another important direction for future research.

Intelligent agents as representatives of logistics entities are not omniscient. Therefore, it is often important to exchange oneself with others in order to broaden the perspective on the real world. It is obvious that not all information has the same value. As a consequence, one must measure which information is most useful in the current context (Gehrke, 2009, pp. 99–101). This consideration can be motivated by the fact that information is not necessarily free of charge (Langer et al., 2006, p. 285). Furthermore, agents have restricted resources and must thus confine themselves to relevant information (Section 3.2.5). This can also be reflected by aggregating information, e. g., by learning from previous experience of oneself or others. As an example, a storage service provider may notice that bookings are regularly cancelled by consumers. It might therefore choose to overbook its services to improve the degree of actual utilisation. Transport service providers may recognise a potential for regular scheduled service on certain routes. Other strategies to keep knowledge bases manageable include forgetting of irrelevant information (Werner, Schuldt & Daschkovska, 2007, p. 15). These considerations are particularly important if control is not only logically but actually physically distributed to embedded systems (Section 3.2.5).

References

Bellifemine, F., Caire, G. & Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE.* Chichester, UK: John Wiley & Sons.

Gehrke, J. D. (2009). Evaluating Situation Awareness of Autonomous Systems. In R. Madhavan, E. Tunstel & E. Messina (Eds.), *Performance Evaluation and Benchmarking of Intelligent Systems* (pp. 93–111). Heidelberg, Germany: Springer-Verlag.

Hribernik, K., Hans, C. & Thoben, K.-D. (2009). The Application of the EPCglobal Framework Architecture to Autonomous Controlled Logistics. In H.-J. Kreowski, B. Scholz-Reiter & K.-D. Thoben (Eds.), *2nd International Conference on Dynamics in Logistics (LDIC 2009).* Bremen, Germany: Springer-Verlag.

Kreowski, H.-J., Scholz-Reiter, B. & Thoben, K.-D. (Eds.). (2009). *2nd International Conference on Dynamics in Logistics (LDIC 2009).* Bremen, Germany: Springer-Verlag.

Langer, H., Gehrke, J. D., Hammer, J., Lorenz, M., Timm, I. J. & Herzog, O. (2006). A Framework for Distributed Knowledge Management in Autonomous Logistics Processes. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, *10*(4), 277–290.

Pantke, F. (2009). Intelligent Agent Control and Coordination with User-Configurable Key Performance Indicators. In H.-J. Kreowski, B. Scholz-Reiter & K.-D. Thoben (Eds.), *2nd International Conference on Dynamics in Logistics (LDIC 2009).* Bremen, Germany: Springer-Verlag.

Singh, G., Wenning, B.-L., Singh, A. & Görg, C. (2007). Clustering in Autonomous Cooperating Logistic Processes. In H.-D. Haasis, H.-J. Kreowski & B. Scholz-Reiter (Eds.), *1st International Conference on Dynamics in Logistics (LDIC 2007)* (pp. 349–257). Bremen, Germany: Springer-Verlag.

Wenning, B.-L., Rekersbrink, H., Timm-Giel, A., Görg, C. & Scholz-Reiter, B. (2007). Autonomous Control by Means of Distributed Routing. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics: The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 325–335). Heidelberg, Germany: Springer-Verlag.

Werner, S., Schuldt, A. & Daschkovska, K. (2007). Agent-Based Container Security Systems: An Interdisciplinary Perspective. In R. Koschke, O. Herzog, K.-H. Rödiger & M. Ronthaler (Eds.), *37th Annual Conference of the German Society for Computer Science (GI 2007)* (pp. 11–16). Bremen, Germany: Gesellschaft für Informatik.

Wooldridge, M. & Jennings, N. R. (1999). The Cooperative Problem Solving Process. *Journal of Logic & Computation*, *9*(4), 563–592.

Index

A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics,*
DOI 10.1007/978-3-642-20092-2, © Springer-Verlag Berlin Heidelberg 2011