HANS MOONEN

Multi-Agent Systems for Transportation Planning and Coordination

Multi-Agent Systems
for Transportation Planning and Coordination

Multi-Agent Systems for Transportation Planning and Coordination

Multi-agent systemen voor transport planning en coördinatie

Proefschrift

ter verkrijging van de graad van doctor
aan de Erasmus Universiteit Rotterdam
op gezag van de rector magnificus
Prof.dr. S.W.J. Lamberts
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op
donderdag 25 juni 2009 om 16:00 uur

door
Johannes Maria Moonen
geboren te Nijmegen

Promotiecommissie
Promotoren:
Prof.dr. J. van Hillegersberg
Prof.dr. S.L. van de Velde

Overige leden:
Prof.dr.ir. J.A.E.E. van Nunen
Prof.dr. M.B.M. de Koster
Prof.dr. K. Kumar

Erasmus Research Institute of Management – ERIM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
Erasmus University Rotterdam
Internet: http://www.erim.eur.nl

ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

ERIM PhD Series in Research in Management, 177
Reference number ERIM: EPS-2009-177-LIS
ISBN 978-90-5892-216-8

Design: B&T Ontwerp en advies www.b-en-t.nl
Photography: Hans Moonen
Print: Haveka www.haveka.nl

Acknowledgements

Somehow, it is quite unfair: The part of a dissertation that takes the least amount of time to write, will be the part most read. Still, it is, in a way, the most enjoyable part to write. After all, it can only be written at the end of the journey, and it provides a place to say thanks to some of the special people along the way.

Back in my days at Eindhoven University of Technology I never expected to begin down the long road of writing a dissertation; neither did most of my friends expect me to. However, working on my master's thesis under the inspiring supervision of professors Piet van der Vlist and Graham Sharman, and often interacting with professor Hans Wortmann, Sipke Baarsma, Roel van den Berg and Arian Zwegers at BaaN, the idea of starting a PhD began to materialise. Sjaak Brinkkemper was the link to Erasmus University; he introduced me to my promotors Jos van Hillegersberg and Steef van de Velde. Jos and Steef, I am very grateful to Sjaak for establishing that link. The two of you showed belief in my capabilities back in the summer of 2002 – I am glad that you did not hire me solely for having the right hobbies (tennis & skiing). Over the years, we've had countless meetings and just as many very intense discussions, gone to a series of conferences all over the world, and indeed gone skiing and played tennis together. It has been a very pleasant time, and I realise that I would not be where I am now if I had not had your help! Thanks!

A very special thanks goes to Jo van Nunen. Jo, we've had a special connection since the beginning. I was, however, never able to figure out the initial origin of the connection: Was it because of our joint engineering background, because we share a slightly southern accent, or because I had a sense of humour similar to yours? Regardless, I would like to extend my deepest gratitude. You have been, and always will be, a real inspiration to me for your ability to combine research with practise, and for your endless desire to really make a difference in the field of logistics and supply chains. I am looking forward to many more inspiring conversations in the future, and of course also to the always accompanying, though not always as politically correct, jokes.

René de Koster and Kuldeep Kumar complete my small committee. Many thanks for the good feedback that helped improve this dissertation, and for the stimulating conversations we had in the past. René, some of my best memories of the past years of academic life go back to the times in Riezlern. Many thanks for organising that. Kuldeep, I am not sure about the karaoke, but I am nevertheless glad that you can attend my defence.

A further word of thanks goes to the professors that complete my large committee: Richard Welke, from Georgia State University, and Matti Rossi, from Helsinki School of Economics. Thank you for taking up this task. A very special thanks (again) to my old-TUE professor, Graham Sharman, who applied for a place on my defence committee. Graham, you have been an inspiration, and I appreciate all the critical comments you have given me over the years.

My research took place within several government-funded research projects. My second day at Erasmus was spent directly at TNO in Delft, where I got involved in the Connekt IA project. It was truly an explorative project, and helped me shape my further research plans. Thierry, Justus, Marco, Babiche, and of course Jos and Steef – I have good memories of our discussions, joint work, and our unforgettable lunch sessions.

Bastiaan, you were the connector who got me, and Erasmus, involved in the APPROACH stream of work. You had an endless drive to really change things in the port. Furthermore, I look back fondly on the series of nice meals we had over the years – often in de Oude Haven – which was a tradition we began during the time you were still at Erasmus. The APPROACH work gave me the chance to work with many others, and I would especially like to extend a word of thanks to Ian and Marco from INITI8, and Albert from the University of Twente. Albert, we had many interesting agent-based conversations, and wrote several papers together, parts of these contributed to my Chapter 6. Thanks for the nice collaboration; I look back with a smile.

The largest part of the research in this dissertation was performed in the DEAL project, a project initiated by Almende. We soon developed into a real research community, and, despite our different backgrounds and perspectives, I think I can say that we did pretty well in understanding each other. My co-researchers from VU, CWI and Almende – thanks, I look back at a series of inspiring DEAL meetings. Tamas and Elfi, I have very good memories of our joint train and car rides to Amsterdam. Martijn, Viara, and Elfi, we came from rather different perspectives, but brought all of these together. It all accumulated into a nice stream of work, and it is still sometimes cited.

As part of DEAL, Arthur, Jos and I started our own subproject together with Post-Kogeko. Richard, Fred, Ben, Alberdine and Frans, I realise that at times we did not speak the exact same language, but I look back with pride on what we jointly achieved: We turned theory into practise, and vice versa.

The e-Dispuut deserves a special place here as well. The e-Dispuut was an environment in which we could present and reflect on our work among fellow PhD colleagues. Michel (University of Twente) and Jeffrey (TU Delft), I look back on many pleasant memories, and look forward to our still-to-be-planned joint BBQ.

Working for the Commissie Van Laarhoven was an idea Jo came up with to the end of my research. Jo, many thanks for having done so – those were interesting times. Everyone at Connekt, thank you. A special thanks to Hanny, and (of course) Machteld. Machteld, we drove around the entire country, and you not only shared a passion for logistics, but also for eating dropjes while sitting in the car.

Ann Jensby, thank you for helping me to turn my Denglish into something better readable.

Thanks also to my new colleagues at Logica. You made me feel at home immediately, and showed me that practical research can really matter and make a difference. I am sure we will have interesting and challenging times ahead. Wim, it took me more time than initially expected to completely finalize the last version – thanks for giving me the freedom to finish it. Jos, thank you for challenging me to continue in academia for one day a week – the first months in Twente have been inspiring as such.

Speaking about colleagues brings me to the group at Erasmus that I spent more than five years with, first in the old building, later in the new. Department One (Decision and Information Systems, for non-insiders): those were pleasant and inspiring years. Thanks to everyone, but I would also like to mention several persons in particular. First, Naima, Chantal, Tineke, Carmen, Olga and later Annemieke, Cheryl and Ingrid, without your help, doing research and writing a dissertation would have been an even larger challenge. Thanks also to the guys at IT Support, especially Frank and Richard. Also, to my old roommates: Arnoud, Jordan, Sarita, and Ting: I look back on many interesting conversations and some very intense discussions, or should I say "debates". It was a true pleasure to have shared the same room. Otto and Jelle, the two of you were a perfect team for making those fresh PhD candidates sharp, and keeping them that way. I recall a series of long Friday afternoons in De Smidse and in De Oude Haven… and I definitely look back with a smile.

Perhaps closest to my experience of going through the research process was my generation of PhD Candidates at RSM: Hans, Elfi, Bas, Niels, Diederik, Umut, Niels, Ting, Chintan, Jordan, and Marcel. Hans and Elfi, I still do not understand why none of the others mentioned here ever joined us to Riezlern. Bas, Niels, Diederik and Niels: I look back on all the cappuccinos we drank together in the breaks, or after lunch. Marcel and Niels, I miss the lunch hikes – a tradition that really started when we moved to the new building which is closer to the Maas. Ting and Elfi, thanks for being my two paranimfen at the defense, could not have wished for a better support team. Bas, we were a great team renovating kitchens, making photos and helping in organizing weddings. The time we spent as each other's neighbours passed too quickly; the same holds for Ting and Hailiang.

Arthur, you deserve a very special place here, and you know that. When you started your master's project, I could not imagine where it would bring you and me. We have worked closely together for a very long time; first with the work on your thesis, which continued when you joined the Department and then developed into a spectacular piece of work, and later also at Logica. We worked together for a long time, but it was still too short. I owe you something. Thanks for having become such a good friend. It is a pity that you will miss my defence due to the fact that you will be starting your new career. I am looking forward to fly along with you somewhere in the future.

Friends and family, from myself, and also from Rinske: I know that I was very busy over the last few years, and I know that I spent too little time with you. Please be assured that I have not forgotten any of you. Thank you all for all support, and interest you have shown over the years.

Sports functioned as a nice escape from the hectic of research. Thanks to all who went mountainbiking with me (in particular Richard, Arthur, and Martijn), and to everyone who played tennis (in the competition) with me in Rotterdam, and at my old club in Overasselt. Jos, thanks to you as well in this respect – we still need to play a doubles tournament together...

Rob, pap en mam. Mam, vaak vroeg jij je af: "*Ben je nou nog niet klaar met het schrijven van dat boek. Je weet toch al zo lang dat het af moet*." Dat wist ik inderdaad, echter tussen weten en realiseren ligt een hele wereld, en een ontzettend lange weg. Ik ben echter heel blij de finishlijn nu eindelijk te kunnen zien. Dankjewel voor alle support die je altijd hebt gegeven. Pap, reuze bedankt voor het keer-op-keer lezen van mijn concepten, en andere artikeltjes. Ook voor het blijven rondkijken naar zaken die aan mijn onderzoek relateren, en de vele andere interessante vader-zoon discussies die we de afgelopen jaren gevoerd hebben. Rob, jouw eindeloze energie en enthousiasme om het tennisracket op te pakken, en andere leuke dingen te doen en organiseren, zijn zeker een inspiratie geweest. Keep up that spirit!

Rinske, de meest bijzondere plaats verdien jij. Wij samen zijn een resultaat van mijn onderzoek. En al heel lang weet ik dat "ons" als resultaat tegen geen enkel proefschrift opweegt. Jij was er als ik vol enthousiasme over bevindingen of resultaten vertelde, als ik het niet meer zag zitten, of als ik afleiding zocht. Jij was er, jij bent er, en ik weet zeker dat dat nog heel lang zo gaat zijn!

Hans Moonen

Utrecht, April 2009

Table of contents

ACKNOWLEDGEMENTS VII

TABLE OF CONTENTS XI

LIST OF FIGURES XV

LIST OF TABLES XVII

LIST OF EQUATIONS XIX

CHAPTER 1 INTRODUCTION 1

1|1 Research motivation 1

1|2 Research question 3

1|3 Research methodology 4

1|4 Scientific and managerial contribution 7

1|5 Structure and outcomes 9

CHAPTER 2 LITERATURE REVIEW 11

2|1 Introduction 11

2|2 Domain challenges in SCM and transportation 11

2|3 Enterprise information systems 14

2|4 Inter-organisational systems 18

2|5 Multi-agent systems 21

2|6 Implementation and adoption 29

2|7 Design of multi-agent systems 35

2|8 Discussion 37

CHAPTER 3 MAS & INTER-ORGANISATIONAL PROCESSES .. 39

3|1 Introduction 39

3|2 A framework for MAS application in supply chains 39

3|3 Industry workshop 43

3|4 Discussion 45

CHAPTER 4 THE SCIENCE OF PROTOTYPE EVALUATION 47

4|1 INTRODUCTION 47

4|2 METHODOLOGICAL PERSPECTIVES AND DIFFERENCES 47

4|3 PROTOTYPE EVALUATION IN RESEARCH 50

4|4 EVALUATING AGENT-BASED INTER-ORGANISATIONAL SYSTEMS 54

4|5 CONCLUSIONS 55

CHAPTER 5 REAL-TIME TRUCK PLANNING 57

5|1 INTRODUCTION 57

5|2 CASE DESCRIPTION 58

5|3 DESIGN 65

5|4 EVALUATION I: DEVELOPMENT AND PROTOTYPING 85

5|5 EVALUATION II: SIMULATION 86

5|6 EVALUATION III: EXPERT EVALUATION 93

5|7 EVALUATION IV: FIELD-TEST 98

5|8 DISCUSSION 110

CHAPTER 6 BARGE ROTATION PLANNING 113

6|1 INTRODUCTION 113

6|2 MANUAL PLANNING OF BARGE ROTATIONS 114

6|3 IN-DEPTH PROBLEM DESCRIPTION 122

6|4 DESIGN DIRECTIONS 125

6|5 COORDINATION THROUGH WAITING PROFILES 127

6|6 EXPERT EVALUATION 129

6|7 DISCUSSION 132

CHAPTER 7 TOWARDS IMPLEMENTATION 137

7|1 INTRODUCTION 137

7|2 EXPERT FEEDBACK ON IMPLEMENTATION ISSUES 137

7|3 DISCUSSION OF INSIGHTS FROM LITERATURE 142

7|4 CONCLUSIONS 145
CHAPTER 8 CONCLUSIONS AND DISCUSSION 147
8|1 INTRODUCTION 147
8|2 RESULTS 147
8|3 DISCUSSION OF IMPLICATIONS 150
8|4 REFLECTION AND FUTURE WORK 152
8|5 CLOSING WORDS 156
APPENDIX 159
APPENDIX A LITERATURE STUDY 161
APPENDIX B POTENTIAL MAS BENEFITS 162
APPENDIX C UML DIAGRAMS POST-KOGEKO 163
APPENDIX D OVERVIEW FEEDBACK SESSIONS 167
APPENDIX E TYPES OF JADE BEHAVIOURS 169
APPENDIX F XML FILES UI 170
APPENDIX G SOURCECODE TRUCKAGENT 172
APPENDIX H SOURCECODE LISTENBEHAVIOUR 185
APPENDIX I SOURCECODE INITIATORBEHAVIOUR 187
APPENDIX J DESIGN PROBLEMS 190
J|1 SUPERTRUCKEATS UP ORDERS 190
J|2 TRUCKAGENTS GO HOME 190
J|3 ORDERS THAT CANNOT BE SERVED ON TIME ANYMORE 191
J|4 IDLE TRUCKS 192
J|5 HUMAN DECISION MAKING 193
APPENDIX K AGENT COMMUNICATION 194
APPENDIX L EXPERT FEEDBACK 196
APPENDIX M FEEDBACK FIELDTEST 205
APPENDIX N CHAIN WORKSHOP PARTICIPANTS 208

APPENDIX O BARGE ROTATION PLANNING 209

LIST OF ABBREVIATIONS 211

BIBLIOGRAPHY 214

SUMMARY 231

SAMENVATTING (SUMMARY IN DUTCH) 234

ABOUT THE AUTHOR 237

ERIM PHD SERIES IN MANAGEMENT 239

NOTES 253

List of Figures

Figure 1.1 – Building blocks that shape our research ... 4
Figure 1.2 – Regulative design cycle (Van Strien, 1986) ... 5
Figure 1.3 – Research progress depicted in time ... 8
Figure 1.4 – Structure of chapters ... 9
Figure 2.1 – The adoption of networked systems (Van Hillegersberg, 2006) ... 32
Figure 3.1 – Classification of multi agent-based collaboration structures ... 42
Figure 4.1 – Different types of IS projects (Fitzgerald, 1998) ... 50
Figure 4.2 – Different phases in IS development (Kushniruk, 2002) ... 51
Figure 4.3 – Spectrum of prototype validation methods (Kushniruk, 2002) ... 53
Figure 5.1 – Standard physical flow of containers in the Post-Kogeko container unit ... 59
Figure 5.2 – Post-Kogeko truck carrying a CoolBoxx container ... 60
Figure 5.3 – Screenshot CarrierWeb system ... 63
Figure 5.4 – Continuous interaction with Post-Kogeko employees ... 66
Figure 5.5 – Agents in the system and their characteristics ... 67
Figure 5.6 – Flowchart – Agent Head Automata of a TruckAgent (initiator role) ... 69
Figure 5.7 – The JADE Remote Management Agent GUI ... 71
Figure 5.8 – The JADE DF GUI ... 71
Figure 5.9 – The JADE agent sniffer ... 73
Figure 5.10 – The JADE introspector agent ... 73
Figure 5.11 – Browser based UI, developed with GWT ... 74
Figure 5.12 – Evolution through time of versions of the prototype ... 75
Figure 5.13 – Sequence diagram – FindOrder process ... 78
Figure 5.14 – The user interface, showing the log dialog ... 79
Figure 5.15 – Communication and logging in the simulator setting ... 88
Figure 5.16 – The expert workshops ... 94
Figure 5.17 – The different sessions related to the field-test and the evaluation ... 103
Figure 5.18 – Basic chain interaction diagram ... 109
Figure 6.1 – Plan board container terminal, showing available quay space ... 117
Figure 6.2 – Computer input screen barge operator planner ... 117
Figure 6.3 – Computer input screen container terminal operator planner ... 118
Figure 6.4 – Percentage of barges arriving too late at the terminal ... 119
Figure 6.5 – Planning moments depicted in time ... 123
Figure 6.6 – Inter-organisational processes spanning barge- and terminal-operators ... 124
Figure 6.7 – Different architectures to be researched ... 126
Figure 6.8 – Order acceptance and pre-planning of barge visits ... 127

Figure 6.9 – Screenshot bargeplanning game 131
Figure 6.10 – Coordination mechanisms in IOS 135
Figure 8.1 – Some design decisions in MAS system design (Douma *et al.*, 2008) 155
Figure 8.2 – Uncertainty in planning through time 156
Figure C.1 – Activity diagram – Assign second (or later) order 163
Figure C.2 – Activity-diagram – Execution of an order 164
Figure C.3 – UML class diagram showing entities in the container planning 165
Figure C.4 – AUML Class Diagram – TruckAgent (initiator role) 166
Figure J.1 – Trapezoid function 192
Figure J.2 – Customer time window scoring function 192
Figure J.3 – Too late, but now prioritized function (SFuzzySet) 192
Figure K.1 – Should Be, TruckAgent-DF-OrderAgent communication sequence 194
Figure K.2 – Wrong TruckAgent-DF-OrderAgent communication sequence 195
Figure O.1 – Illustration of the working of the agent model (Douma *et al.*, 2008) 209
Figure O.2 – Example of a waiting profile sent to a BOA by a TOA 210
Figure O.3 – Inner mechanism of the MAS utilizing waiting profiles 210

List of Tables

Table 1.1 – Sub research questions, methods and chapters 6
Table 1.2 – Overview of publications contributing to chapters and projects........ 10
Table 2.1 – Opportunities for savings for LSPs (Bold and Olsson, 2005) 13
Table 2.2 – Different types of inter-organisational systems (Kumar and Van Dissel, 1996) 19
Table 2.3 – Agent characteristics and references citing these 23
Table 2.4 – Pros & cons of centralized and distributed planning approaches 26
Table 2.5 – Critical success factors in the implementation of information systems 31
Table 2.6 – Factors that hinder adoption of MAS in industry 34
Table 2.7 – Paradigm shifts identified 38
Table 3.1 – Workshop attendance 43
Table 4.1 – Evaluation methods for MAS prototype evaluation 56
Table 5.1 – Matrix showing order scorelist for TruckAgent 68
Table 5.2 – Machines used in the development trajectory 74
Table 5.3 – Agents in the different versions of the prototype 76
Table 5.4 – Overview of agents and their behaviours (* sim only; # manual only) 80
Table 5.5 – Five specific engineering decisions in the prototype building process........ 83
Table 5.6 – Four simulation scenarios 88
Table 5.7 – Parameter settings in four different experiments........ 90
Table 5.8 – Mean values hours and trucks [80/20 experiment]........ 90
Table 5.9 – Mean values hours and trucks [corrected 80/20 experiment] 90
Table 5.10 – Mean values deliveries [corrected 80/20 experiment]........ 90
Table 5.11 – Comparison of results from the same seven days [scenario C] 90
Table 5.12 – Overview of workshop participants........ 94
Table 5.13 – Questions asked during the workshops 95
Table 5.14 – Selection of remarks from group discussion roadplanning workshops 96
Table 5.15 – Response times on FindOrder event (all orders are still available) 102
Table 5.16 – Feedback on the prototype, field-test planner I 106
Table 5.17 – Feedback on the prototype, field-test planner II........ 107
Table 5.18 – Conclusions field-test at Post-Kogeko, Spring 2008........ 108
Table 5.19 – List of identified issues that result from wrong information exchange 110
Table 6.1 – Barge operators as used in the workshop 116
Table 6.2 – Container terminal operators as used in the workshop........ 116
Table 6.3 – Workshop output: initial manual plan for the Rhine I........ 120
Table 6.4 – Workshop output: confirmed manual plan for the Rhine I........ 120
Table 6.5 – Total double book time for the manual planning results 121

Table 6.6 – Call sizes dependent on type of barge traffic 125
Table 7.1 – Model to cluster feedback 138
Table 7.2 – Adoption and implementation aspects from expert feedback 141
Table 8.1 – Key findings from this dissertation 149
Table A.1 – List of journals covered [2000-2006] 161
Table A.2 – Academic search engines used 161
Table A.3 – Key words used 161
Table B.1 – Potential benefits MAS architecture for Enterprise IS (Lea *et al.*, 2005) 162
Table D.1 – Feedback sessions throughout the design process 167
Table E.1 – Overview of different types of Behaviours within Jade 169
Table F.1 – Example of log.xml file 170
Table F.2 – Example of orders.xml file 170
Table F.3 – Example of trucks.xml file 171
Table L.1 – Index of tables with expert feedback 196
Table L.2– Advantages road transport prototype 197
Table L.3 – Disadvantages road transport prototype 197
Table L.4 – Feedback on how to support the planner 199
Table L.5 – Advantages barge planning prototype 199
Table L.6 – Disadvantages barge planning prototype 200
Table L.7 – Advantages of the exchange of waiting profiles for barges 201
Table L.8 – Disadvantages of the exchange of waiting profiles for barges 201
Table L.9 – Success factors that drive adoption 202
Table L.10 – Factures that explain adoption failure 202
Table L.11 – Potential other applications 202
Table L.12 – Results from discussion in the first group road transport workshop 203
Table L.13 – Results from discussion in the second group road transport workshop 203
Table M.1 – Overview of feedback received 205
Table N.1 – Overview of participants in the chain implication session 208
Table O.1 – Four different coordination mechanisms for pre-planning 209
Table O.2 – Results from the three different approaches 210

List of Equations

Equation 5.1 – Central formula CalculateScore (TruckAgent) 76
Equation 5.2 – Amount of trucks depends on distance and processing time 88
Equation 5.3 – Estimated amount of CalculateScore calls 101

Chapter 1 Introduction

1.1 Research motivation

Over the past decade, mobile phones have changed the way people plan and organise their social and business activities. In the past, very detailed, pre-determined plans had to be made, for example: "We will meet in front of the metro station at 8 o'clock [sharp]". Nowadays, plans are often more roughly coordinated, such as: "We will see each other downtown this evening". These are combined with last minute coordination over the phone, such as: "Hey, I am standing in front of the restaurant now, but it is very busy here, Where are you? Where shall we meet? Shall we go somewhere else?"

In contrast, planning in supply chains has remained largely unchanged over the last twenty years. Generally speaking, each link in the chain plans their own activities, and utilise forecasting techniques to predict what is likely to happen. The example of the mobile phone illustrates the possible impact a new technology can have on planning and coordination. It is suddenly possible to utilise real-time information, adapt plans dynamically when circumstances change, and coordinate activities directly with other parties. Therefore, is it not time to start approaching planning and coordination in supply chains differently? Could we not now utilise much more information, make real-time decisions, and coordinate with other parties in the supply chain?

The world is changing at a rapid pace. It becomes more flat and globalised every day (Friedman, 2005). Companies nowadays have the world as their playing field – not in the least due to the Internet. Competition has become fiercer than ever before (Fingar, 2006), and as such, companies have to continually increase their performance (Treacy and Wiersema, 1995), and become more agile and responsive to change in order to remain competitive (Verwaal, 2005). Furthermore, as competition increases and product life cycles shorten, traditional static supply chains are no longer sufficient (Lee, 2004).

The Supply Chain Management (SCM) paradigm was born in response to the challenges of increased competition, globalisation, longer supply chains, and fast market demand changes. As Lambert (2000) states: "SCM represents one of the most significant paradigm shifts of modern business management by recognising that individual businesses no longer compete as solely autonomous entities, but rather as supply chains (consisting of individual businesses, working together)." Collaboration within the supply chain can, for example, reduce chain-wide inventories (Chen *et al.*, 2005; Van Der Vlist, 2007). As tasks increasingly become an accumulation of individual contributions and are more contingent upon each other, actors need to connect and share information and knowledge. Collaboration, complex coordination, and information sharing between organisations require effective support through information systems. Sharman (2003) sketches the vision of chain-wide collaboration to achieve cost reductions in supply chains, going beyond the first attempts to electronically enable supply chains which "in principal all came down to instruments that helped to reduce transaction costs". One particular party that could greatly benefit from increased collaboration is the Logistics Service Provider (LSP). The LSP is by nature an intermediary positioned between other parties (Lai *et al.*, 2004). Practise shows however, that LSPs have difficulties harvesting the fruits of collaboration. As a result, margins are low, operations could be improved upon, and innovation lags behind (Chapman *et al.*, 2003; Bold and Olsson, 2005; Langley Jr and Allen, 2006). For instance, the percentage of empty-truck-kilometres is considerable. De Ridder (2003) estimated it at 25-50% of total truck kilometres – a percentage unacceptable high from a sustainability perspective. For more on this topic, see also Brinckman and Ungerman (2008).

In the 1960s, information technologies began being introduced to the corporate world. Enterprise information systems have become crucial to most companies' daily operations (Brynjolfsson and Hitt, 1996). Nevertheless, serious doubts exist about whether the current enterprise systems are suited for today's SCM challenges (Davenport and Brooks, 2004). Up until now, it has turned out to be a rather complex task to support the SCM concepts with information systems. Enterprise Resource Planning systems (ERPs) are generally "tightly integrated and monolithic systems that reflect and respect traditional company boundaries" (Sharman, 2003). A particular challenge for these systems is to enable flexibility and interconnectivity between information systems throughout the supply chain. Indeed, Lee and Myers (2004) have demonstrated that ERP likely results in a lock-in into relatively inflexible business processes. Van Hillegersberg *et al.* (2006) specifically illustrate the difficulties of ERP-type styled centralised systems, or marketplaces/hubs, in going inter-organisational.

Over the past two decades researchers have been working on a different type of information system architecture, namely multi-agent systems (MAS) (Wooldridge and Jennings, 1995). Different than centralised information systems, such as ERP, MAS consists of many autonomously interacting agents. These interacting agents are small software programs that have a certain level of intelligence and individual behaviour (Schleiffer, 2005). Communication and coordination (between agents) are the essential elements in such systems.

Specifically the transportation domain is an interesting candidate for the application of multi-agent systems (Fischer *et al.*, 1996; Luck *et al.*, 2004; Davidsson *et al.*, 2005; Moyaux *et al.*, 2006). Especially the inter-organisational nature of this domain makes agents logical candidates, as the heavy interdependence on chain partners troubles the implementation and utilisation of centralised systems as we have just discussed. Fischer *et al.* (1996) show that MAS has the potential to perform similar to traditional centralised (OR) mechanisms, however MAS design may provide fundamental advantages such as increased flexibility, and real-time capabilities.

Nevertheless, MAS have not been widely adopted in industry yet (Caridi and Cavalieri, 2004). Many papers, even some describing "practical applications" – see Moyaux *et al.* (2006) – solely describe academic experiments without concrete implementations. Chmiel *et al.* (2005) conclude that most of the current multi-agent system research is far from realistic because the setting is oversimplified, and designs generally only include a very limited number of agents. They plead to researchers to “start designing and implementing large software systems, consisting of hundreds of agents, and study their behaviour”. Nwana and Ndumu (1999) make a similar case.

The organisational information processing theory, introduced by Galbraith (1974), tells us that businesses can tackle uncertainty by either reducing the need for information processing (for example through better forecasting), or by increasing the capacity to process information. Premkumar *et al.* (2005) reason that this also holds for inter-organisational relationships, in which increased information exchange can be a key enabler for improving firm and supply chain performance. Multi-agent systems are based on information exchange, communication flows, and negotiations between different agents/entities. As such, MAS could be an instrument to establish (future) supply chain systems that focus on solving problems through synchronization of activities – by means of communication and negotiation – rather than optimisation in isolation, as tends to be the case in centralised architectures.

We conclude that there is a gap between research and practise concerning inter-organisational multi-agent system application. Back to the example we started with: Mobile phones changed the way we coordinate in daily life. Mobile phones have introduced more flexibility, and have led to less unnecessary waiting, and fewer frustrations in daily life, and as such, have increased quality of life (Plant, 2002; Levinson, 2004). A new technology created a different way of coordination. Can concepts developed within multi-agent system research enable similar change in supply chains, more specifically in transportation? More information than ever before exists, and computing and communication devices are literally everywhere (Wooldridge, 2005). Will the introduction of agents result in better performing inter-organisational systems? Will this lead to systems that are easier to implement?

1|2 Research question

Following from the discussion above, the central explorative research question that we pursue throughout this dissertation is formulated as follows:

How can multi-agent systems be successfully applied to design and implement better performing inter-organisational systems for transportation?

This research question has been turned into a conceptual research framework, depicted in Figure 1.1, in the form of a so-called multi-tier influence diagram. This is the most commonly used research model approach in information systems today (Palvia *et al.*, 2006). At its core, this question concerns an explanatory relationship between domain challenges in logistics, requirements for inter-organisational systems and the successful multi-agent system design and implementation, which together lead to better systems.

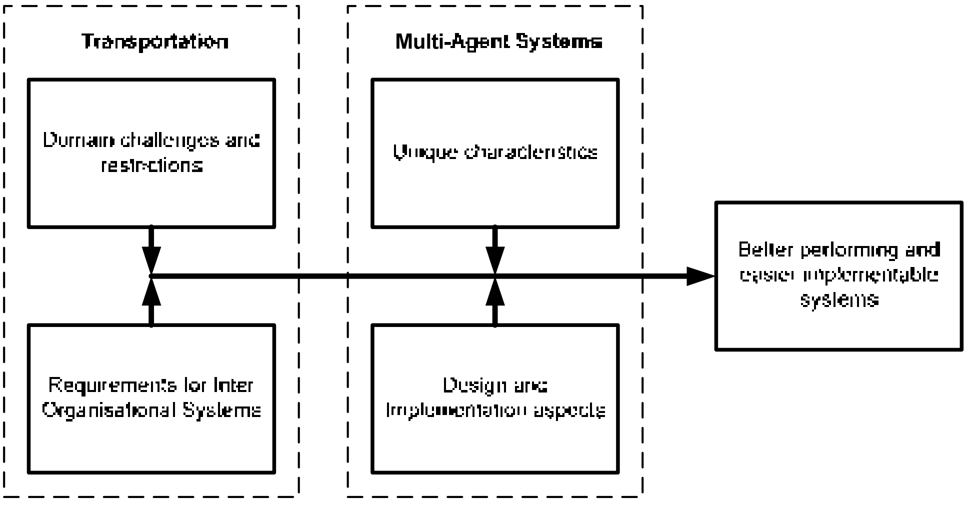


Figure 1.1 – Building blocks that shape our research

1|3 Research methodology

1|3|1 The role of research methodology in this dissertation

Most (PhD) research in social sciences nowadays takes a theory-testing perspective (Van Aken, 2004). In our research, we explore the boundaries of novel inter-organisational information system architectures. We follow a design science approach that takes the perspective of new artefact and theory development by means of explorative methodologies, embedding a crucial role for empirical work. This basically pleads for repetitive research cycles, as Booth *et al.* (1995) has described it: “A practical problem motivates a research question, which defines the research problem, and results in research answers; these in turn help to solve the practical problem. Surprisingly, however, this is not the standard (anymore) in the Information Systems (IS) and Operations Research (OR) / Operations Management (OM) research (Vessey *et al.*, 2002; Van Aken, 2004; Denning, 2005).

The (academic) information systems community does not perceive technical research to be very important anymore, as Vessey *et al.* (2002) conclude. Neither does its design science origin, as Iivari (2007) observes: mainstream IS research lost sight of its (design science) origin over the past twenty-five years and now mainly concentrates on theory-based research aimed at making prescriptions. Iivari claims “*this “theory-with-practical-implications” research has over the past years seriously failed to produce results that are of real interest in practise*”. A series of scholars perceive design research as essential in making research more relevant (March and Smith, 1995; Romme, 2003; Hevner *et al.*, 2004; Van Aken, 2004; Verschuren and Hartog, 2005; Cross, 2007). The different phases in any design research trajectory are documented by Van Strien (1986), and visualised in

Figure 1.2. Although such a design cycle suggests that it functions as a one-time only cycle, it is normal to go back through the cycle several times.

A related methodology to design research is action research, which comes close to scientific consulting. "*An action researcher takes a real problem as a starting point, gets informed through literature, works on improvements in the field, and then links back to literature again*" (Gummesson, 2000). Important to realise is that action research goes beyond consulting since it is initiated from the question: "*Why do certain things work, or why do they not work?*" A detailed analysis and synthesis phase are very important elements here. Slowly, action research is becoming more accepted in IS and the social sciences in general – see Baskerville (1999) & (2004) and Lindgren *et al.* (2004). Romme (2004) states that action research and design research should always go hand-in-hand. However, there is an important difference as well (Van Aken, 2004; Iivari, 2007). Action research generally addresses improvement problems rather than construction problems; it thus focusses on "*treating social illness*". Technological change may be part of the "*treatment*", but the focus is on adopting these technologies rather than building it.

Nunamaker and Chen (1990) make clear that innovative novel systems development is, and should be, a research method as such, focussed on the construction of new IS artefacts. Prototyping and product development are instruments to test new theoretical concepts.

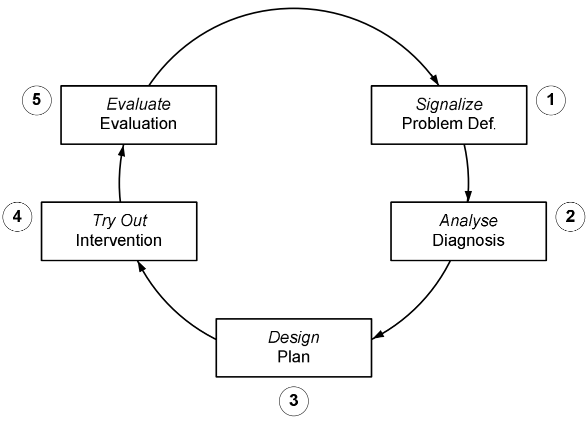


Figure 1.2 – Regulative design cycle (Van Strien, 1986)

1|3|2 Methodologies utilised and sub-research questions

We use a multi-method research methodology, in which we include elements of different methodological streams, that specifically concentrates on the design and evaluation of new concepts. A mixture of methodologies makes it possible to combine strengths and fill in weaknesses of single methodologies. See for example the analysis, overview, and suggestions by Palvia *et al.* (2006) and Nwana and Ndumu (1999).

The general research question defined in section 1|2 leads to a number of more detailed sub-questions listed in Table 1.1. The table also briefly describes the research method(s) for each question, and lists the chapters that correspond to these. Each chapter includes an introduction in which we briefly discuss the used methodology and approach followed.

At the core of this dissertation are two design cases. One case concerns the design of a multi-agent system for real-time assignment of container trucks; the other covers a barge-rotation planning problem in the port of Rotterdam. Hence, the replicability and generalisability are an issue due to the diversity of the cases – see Benbasat *et al.* (1987), Eisenhardt (1989), Lee (1989), and Yin (2003). Nevertheless, the cases are valuable in understanding how multi-agent system concepts can be used in inter-organisational system design and implementation. After all, a case is a logical evaluation in a trajectory of design (Hevner *et al.*, 2004). A literature review is an inevitable element of any scientific research. In our review, we follow the guidelines provided by Blumberg *et al.* (2005) and Webster and Watson (2002).

Table 1.1 – Sub research questions, methods and chapters

#	Question #	Research Method	CH
1	What are the domain challenges in transportation?	Literature review	2
2	What are the requirements for inter-organisational systems for transportation?	Literature review	2
3	What are the unique characteristics of multi-agent systems?	Literature review	2
		Design and feedback cycle	5 + 6
4	What are the design and implementation aspects of multi-agent systems?	Literature review	2
		Design and feedback cycle	5 + 6
5	Can multi-agent systems contribute to better performing, and easier-to-implement systems for transportation?	Explorative & industry evaluation	3
		Design research cycle – focussed on enterprise centric system (multiple validation methods)	5
		Design research cycle – focussed on network centric system (multiple validation methods)	6
		Analysis & discussion	7 + 8

1|3|3 Time scheme

An overview of the research progress over time is given in Figure 1.3. It illustrates the order in which our work progressed, and shows the relations between the different elements.

1|4 Scientific and managerial contribution

The scientific contribution of this dissertation is foremost the integral perspective it takes. It integrates knowledge, concepts, theories and ideas from different scientific communities – more specifically the fields of Information Systems (IS), Computer Science (CS), Artificial Intelligence (AI), Operations Research (OR) and Operations Management (OM) – and combines these with real requirements from practise, through the two explored cases. This results in a comprehensive exploration of what multi-agent system concepts can contribute to inter-organisational systems. Both prototypes are examples of the type of applied research March *et al.* (2000) ask for: multi-agent systems designed to operate in networked environments. They pointed at prototypes as extremely valuable instruments to assess the features of concepts and ideas, combined with the fundamental problems encountered in real-world environments.

From a managerial point-of-view, this dissertation contributes to the knowledge and insights about directions for future inter-organisational system developments, specifically focussed on the transportation domain. It describes how the practise of planning is changing by the increased availability and utilisation of real-time information. It furthermore shows the large potential supply-chain-wide coordination offers, and how these two changing factors combined ask for a different breed of enterprise information systems. The work presented here has the potential to influence the design of future inter-organisational systems. The proof-of-concept multi-agent systems developed may be a basis for real business applications and implementations in the future.

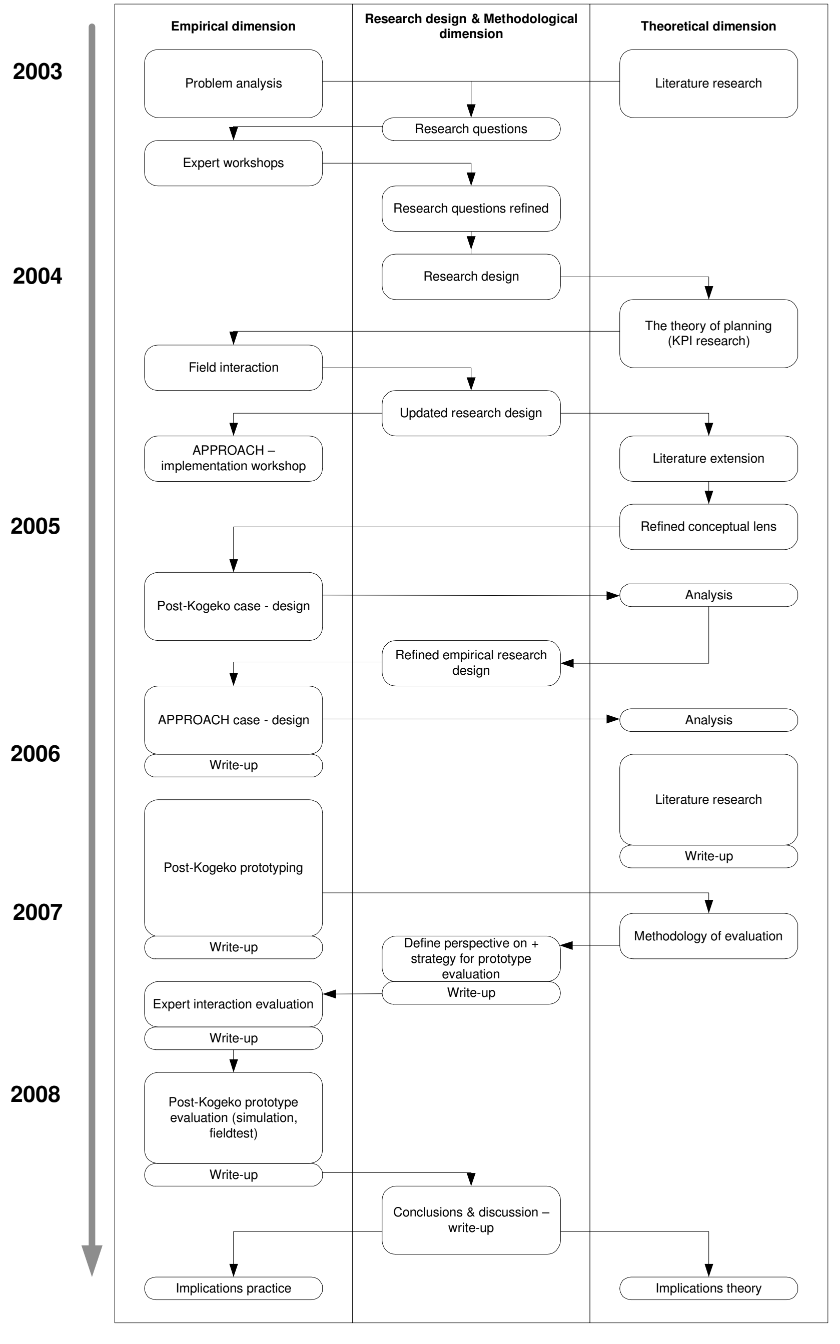


Figure 1.3 – Research progress depicted in time

1|5 Structure and outcomes

Figure 1.4 gives the structure we follow in this dissertation. Chapter 2 covers an in-depth literature survey, which concentrated on several of the sub-research questions we formulated. Chapter 1 describes an industry evaluation we performed at the beginning of our research to understand the state-of-the-art of agent systems and concepts in industry. This work helped shape our further research plans.

Chapter 1contains a methodological discussion of prototype evaluation approaches. Furthermore it formulates a novel perspective on the evaluation of inter-organisational research prototypes, which we utilised in the two case chapters that follow.

Chapter 1 and Chapter 6 describe the two design cases we worked on: the design of a multi-agent system for real-time truck assignments, and the design of a scheduling system for barge rotations in the Port of Rotterdam. In Chapter 7, we cluster the research findings with respect to future implementations of multi-agent systems. Chapter 1 is used for synthesis and reports on our larger findings. It furthermore defines a research agenda and a list of practical recommendations.

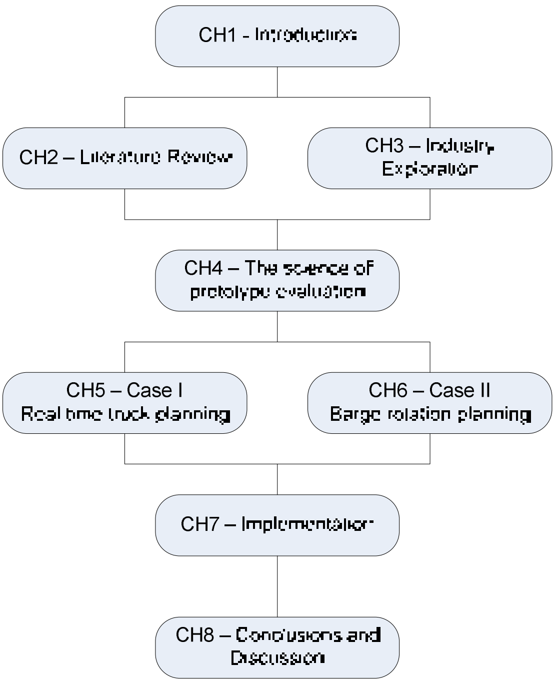


Figure 1.4 – Structure of chapters

The research in this dissertation was performed within three government funded research projects: the Connekt Intelligent Agent project [Connekt number 224 ”Verkenning Toepassingsmogelijkheden Intelligent Agents”], which took place in 2003; the DEAL project [EETK01141 “Distributed Engine for Advanced Logistics”], which ran from 2003-2007; and the Transumo Diploma project [project number GL05028], which ran from 2005 onwards (and will finish in 2009). An overview of the scientific publications that contribute to the different chapters in this dissertation can be found in Table 1.2, and the corresponding research projects also mentioned. Thanks to all fellow co-authors for the cooperative work.

Table 1.2 – Overview of publications contributing to chapters and projects

Ch	Previous publications	Research project
2	(Moonen *et al.*, 2003), (Krauth *et al.*, 2005), (Krauth *et al.*, 2005) (Moonen *et al.*, 2006), (Lang *et al.*, 2008), (Van Baalen *et al.*, 2009 (expected))	DEAL / Diploma
3	(Becker *et al.*, 2003), (Van Hillegersberg *et al.*, 2004)	Connekt IA
5	(Oink, 2005), (Moonen *et al.*, 2005), (Moonen *et al.*, 2007), (Srour *et al.*, 2008), (Moonen *et al.*, 2008)	DEAL
6	(Moonen and Rakt, 2005), (Moonen *et al.*, 2007), (Douma *et al.*, 2008), (Moonen *et al.*, 2008)	Diploma
7	(Moonen *et al.*, 2008)	Diploma

Chapter 2 Literature review

2|1 Introduction

This chapter is a review of relevant literature. It details the structure of the problem domain, the state-of-art in concepts and theories, and identifies research and practical challenges. This review contributed to the Transumo Diploma project – see Lang *et al.* (2008). In Appendix A, an overview is given of the journals covered, the search engines used and the key words searched for.

In section 2|2, we consider the first sub-research question: "*What are the domain challenges in transportation?*" The following two sections (2|3 and 2|4) were initiated from the second research question, namely: "*What are the requirements inter-organisational and enterprise systems for transportation face?*" The next section (2|5)) discusses literature around research question number three: "*What are the unique characteristics of multi-agent systems?*" The fourth question on design and implementation is split into two parts; see 2|6 and 2|7. The concluding discussion section (2|8) gives an overview of our conclusions and thoughts with respect to potential future work.

2|2 Domain challenges in SCM and transportation

2|2|1 Supply chain management

"Supply chain management (SCM) encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities. Importantly, it also includes coordination and collaboration with channel partners, which can be suppliers, intermediaries, third party service providers, and customers. In essence, supply chain management integrates supply and demand management within and across companies", according to the Council of SCM Professionals. Logistics management, as such, is part of the wider SCM function. "*The term SCM [is] used to explain the planning and control of materials and information flows as well as the logistics activities not only internally within a company but also externally between companies*" (Chen and Paulraj, 2004). Lambert and Cooper (2000) claim that "*SCM represents one of the most significant paradigm shifts of modern business management by recognizing that individual businesses no longer compete as solely autonomous entities, but rather as supply chains (consisting of individual businesses, working together).*" As such, SCM has become an integral part of each and every function and entity within a supply chain. Important differences with the traditional perspective on logistics are an increased utilisation of information throughout (inter-organisational) decision-making processes, an inter-enterprise focus, and an end-customer orientation for all parties in the chain (Kopczak and Johnson, 2003).

2.2.2 Planning

Planning within the logistics domain is the process of anticipating and preparing for future events, generally customer demands, variations in supply, and other internal or external variations – see also Daganzo (2005). It is performed in order to reduce uncertainties in fulfilling customer demands and to reduce lead-times. Galbraith (1974) identified two strategies to reduce task uncertainty in business processes: the reduction of the need for information processing, and the increase of the capacity to process information. Raman (1995) showed that logistical information- and decision-support systems generally focused on reducing the need for information processing.

Companies could increase the capacity to process information, for example by information coupling in their supply chains. Instead of predicting and anticipating what is likely to happen, companies could utilise real-time information from up- and/or downstream on the supply chain to monitor what really happens and react accordingly (Sheombar, 1997). Up-to-date supply chain information is becoming increasingly important (Sriram *et al.*, 2000). The MIT Beergame illustrates the importance of information exchange in supply chains – see Lee *et al.* (1997). Collaborative planning with partners in the supply chain is often suggested as an instrument to cope with uncertainty and to improve the overall supply chain (Lambert and Cooper, 2000; Narayanan and Raman, 2000; Holmstrom *et al.*, 2002; Moyaux *et al.*, 2005), and its robustness (Chen, 1999).

On the one hand, the needs for information processing are influenced by uncertainties that exist in an organisation (Premkumar *et al.*, 2005). On the other hand, an improved integration of information and processes leads to less uncertainty and thus to more stable planning and control (Bretzke, 2003). Bretzke proves that better supply chain information reduces the need for forecasting.

2.2.3 The challenges in the transportation domain

Transport is an essential element in any supply chain. Goods have to be moved between production stages, and eventually find their way towards consumers. Producers, brand-owners, and retailers generally do not consider transportation their core competence. They often outsource this work to specialised firms, referred to as Third Party Logistics (3PL) or Logistics Service Providers (LSP) (Christopher, 1999). Transportation is a main activity for an LSP, but LSPs often offer additional services, such as warehousing, customer service, and inventory management (Sink *et al.*, 1996; Vaidyanathan, 2005).

Logistics service provision is an industry under great pressure. Margins are small. Many LSPs try to reduce costs by scaling up or expanding their activities outside their home country (Lemoine and Dagnaes, 2003). Other innovations are technical innovations such as new communication systems (e.g., RFID, GPS) or offering multi-modal solutions (Chapman *et al.*, 2003). Bold and Olsson (2005) identify a long list of troubles and shortcomings. Especially waiting time, order changes, and order reception are identified as key issues for improvement. Largest savings can be achieved in the "*during transport*" phase. A more detailed overview of the issues identified is provided in Table 2.1. The important factors in the "during transport" phase are supported by findings from Van Donselaar *et al.* (1998). Their findings are not very surprising, as they point at the attractiveness of long trips, the benefits of a low percentage of empty miles (of total miles driven), and the benefits of combining (international) shipments. Major, more macro-level, challenges to the sector are legislation and regulations, and the troubles LSPs have with IT developments and implementations (in-house, and in the network) (Verwaal, 2005).

LSPs are generally not seen as very strategic supply chain partners. One factor that illustrates this is the fact that the most important factor for selecting an LSP still is price, the quality of its logistics services ranks only second (Stewart, 1995; Menon *et al.*, 1998; Fowkes *et al.*, 2004; Langley Jr and Allen, 2006; Moore *et al.*, 2006). Other important factors are speed and reliability, loss and damage rate, and on-time delivery. Second, it is also illustrated by the fact that only as little as 25% of the shippers use electronic data integration with their LSP (Moore *et al.*, 2006). Another 56% of firms use technology to correspond with their LSP, but in labour-intensive ways: through means such as e-mail and Internet portals. As a result, the LSP is an intermediary with only a small amount of space to decide on how it fulfils its tasks (Lai *et al.*, 2004). It is poorly integrated with the up- and downstream of the supply chain, and hence has little opportunities to optimise streams, as parties throughout the supply chain too often attain local optima (Moore *et al.*, 2006). LSPs, in turn, try to become more dedicated to, and integrated with, their (large) customers (Bromley, 2001; Hertz and Alfredsson, 2003), in order to improve their position in the supply chain.

Table 2.1 – Opportunities for savings for LSPs (Bold and Olsson, 2005)

Phase	Savings
Pre transportation	Labour costs Reduction in losses due to manual processes Savings on high cost of expedited freight
During transportation	Empty haulage Reduce waiting times Less penalty costs Lower overtime costs
Post transportation	Renegotiations Automated communication Invoicing and reconciliation Demurrage

Fourth Party Logistics (4PL) is a term coined in the late 1990s by Accenture. A 4PL is, by their definition, "*an integrator that assembles the resources, capabilities, and technology of its own organisation and other organisations to design, build and run comprehensive supply chain solutions*". ICT is envisioned to have an important role herein. Over the years, many 3PLs have tried to become 4PLs, but most of them failed in their ambitions. Several reasons for this failure are given by Hertz and Alfredsson (2003). First, customers require neutrality from their supply chain manager. 4PL's with a background as a 3PL have a legacy of resources (e.g., warehouses and wheels). Second, 3PL's often lack the more advanced knowledge and capabilities needed for a 4PL. Third, up- and downstream supply chain partners are often not (yet) ready for a 4PL structure, which often includes a transfer of decision authorities (e.g., the 4PL deciding about shipment dates). Berglund *et al.* (1999) pointed to a future 4PL role for information-oriented outsiders such as information technology or consultancy firms.

2|3 Enterprise information systems

2|3|1 Computers in business

The Second World War (WWII) aggressively accelerated the development of computing (-technology). Computers were first developed and deployed for military purposes, such as calculations on artillery firing tables and the design of the bomb, but also to manage scarce resources as efficiently as possible (Haley, 2002). Computing hit the business stage a decade later, in the 1950s. This development went hand-in-hand with the first business application of Operations Research (OR) principles, which also traces its roots back to WWII (Mahoney, 1988). Over the years, government spending kept dominating and driving developments in computer technology (Mahoney, 1988).

The pace of change in enterprise information systems application in industry is a paradoxical one. On the one hand, developments in hardware and software seem to progress at rocket speed – see for example Coltman *et al.'s.* (2001) description of the pace of Internet adoption, reread Bill Gates' 1995 vision of the Internet and network services in light of the current situation (Gates, 1995), or read the history of information technology in the Netherlands (Van Den Bogaard *et al.*, 2008) and see how fast technology has evolved. Technology matures and develops – computer power still doubles every eighteen months (Moore, 1965). What is state-of-the-art today is ready for the museum tomorrow. As Milojicic (2004) put it: "*What you have on your desk now, is more powerful than all power of the world's supercomputers together 30 years ago. Imagine what another 30 years of developments will bring us?*" Nevertheless, looking at the underlying processes one could make an opposite observation. Fundamental change, in information systems, takes a long time. Real-time systems for example were already reported on as early as 1970 (Zani, 1970). Also Enterprise Resource Planning (ERP) (Haigh, 2001) and Business Intelligence (BI) (Luhn, 1958) took a long time to get from idea to practice. An interesting example is the LEO, which is recognised (Baskerville, 2003; Glass, 2005) as the first business software application ever, which was first booted in 1951. Although technology might have accelerated at rocket speed ever since, many of today's system implementations still aim at achieving objectives similar to the ones LEO delivered in the early 1950s.

2|3|2 Enterprise systems changed business

The first computing applications in business in the 1950s and 1960s were mainly used for simple calculations and data storage. When hard- and software capabilities evolved, Material Requirements Planning (MRP) and Manufacturing Resource Planning (MRP-II) applications became available throughout the 1970s (Van Busschbach *et al.*, 2002), mainly to support the business need for better-coordinated material flows. In the late 1980s (Kumar and Hillegersberg, 2000), the first Enterprise Resource Planning (ERP) applications were introduced. These systems were mainly designed to solve the fragmentation of information in large business organisations and standardise processes (Davenport, 1998). ERPs initial focus was to execute and integrate functionality to support finance, accounting, manufacturing, order entry, and human resources (Davenport and Brooks, 2004), as such it brought operational improvements (Cotteleer and Bendoly, 2006). ERP influenced enterprise performance especially in environments with multiple units that were interdependent and had relatively standardized processes (Gattiker and Goodhue, 2005). Gattiker and Goodhue (2005) and Cotteleer and Bendoly (2006) point at learning effects taking place over years; in other words, it takes time to leverage the full potential of the system. An overview of ERP benefits distilled from 233 implementations by Shang and Shedoon (2002) illustrates that ERPs are much more than just an IT system; they impact all aspects of an organisation.

ERPs are generic systems, designed with "best practices" in mind. "Best", however, is vendor-defined, and tends to be hard-coded in. It is standard software, which needs to be customised (Ragowsky, 2002). Customising an ERP – which means changing or extending internal code, or interfacing with legacy systems – adds complexity, costs, and complicates upgrades and integration with business partners. Not surprisingly, a good customisation has an important influence on the performance of enterprises (Shang and Seddon, 2002; Gattiker and Goodhue, 2005).

In the 1990s companies also started looking for enterprise software that could reach beyond the enterprise's borders (Van Busschbach *et al.*, 2002). Electronic Data Interchange (EDI) technologies were later followed by more flexible XML (eXtended Markup Language) technologies that could leverage the standard Internet infrastructure (Davenport and Brooks, 2004).

An entire category of "*supply chain management*" software is sold as an extension to ERP. Surprisingly, this category mainly covers software with an intra-enterprise focus (Davenport and Brooks, 2004), as it (often) lacks a view of the wider supply chain. Examples of tools within this category are: (1) supply planning tools, (2) demand planning tools, (3) plant scheduling tools, and (4) logistics systems to support warehouse management and order management (Davenport and Brooks, 2004). Advanced Planning and Scheduling (APS) solutions can be found throughout these categories (Sridharan *et al.*, 2005). Also, Supply Chain Event Management (SCEM) systems are part of this category; SCEM systems monitor their environment and trigger alerts or human-involved problem solving workflows (Bretzke, 2003). Hence, despite its name, SCM software hardly supports SCM activities or processes throughout the wider supply chain, and mainly concentrates on planning and scheduling within the four walls of the enterprise.

2|3|3 Trends in enterprise information systems

ERPs have drawbacks. One of the largest problems is the lock-in into rigid business processes and a reduction in flexibility (Levy and Powell, 1998; Hagel III and Brown, 2001; Ragowsky, 2002; Sharman, 2003). Another frequently mentioned problem is the fact that ERPs are not designed for inter-organisational usage (Wortmann and Szirbik, 2001; Akkermans and van Helden, 2002; Sharman, 2003; Davenport and Brooks, 2004). Three (additional) shortcomings of ERP are (Klapwijk, 2004): (3) A lack of open standards at the functional interface level; (4) No best-of-class within functional areas; and (5) Time-consuming implementation trajectories. Klapwijk points to the "*top-down*" architecture of such systems as their most important drawback. Geoffrion and Krishnan (2001) signal that despite its name, ERP never delivered its P.

Triggered by a changing world, in which globalization leads to networked organisations (Heng, 2003), enterprise software needs to become (more) inter-organisational (Anussornnitisarn and Nof, 2003). Hagel and Brown (2001) state that "*that is where the limitations of existing IT architectures are most apparent and onerous; applications on the edge of one's enterprise can benefit by definition from sharing*". Traditional enterprise information systems do not sufficiently cover inter-organisational coordination processes (Sharman, 2003; Lee and Myers, 2004; Van Hillegersberg *et al.*, 2006).

Next to the factor (inter-organisational) "*scope*" discussed above, the factor "*time*" is another important dimension that is changing. We observe the need for real-time systems. ERP systems are designed around an optimisation engine that typically runs once a day (or night). Nowadays, information is available everywhere and at any time, which shapes possibilities for real-time usage of this information (Klapwijk, 2004). Examples of sensor systems include RFID (Radio Frequency Identification) technology and GPS (Global Positioning System) positioning (McFarlane and Sheffi, 2003; Bose and Pal, 2005). Future generations of RFIDs can be equipped with processors to execute software code (Bose and Pal, 2005).

"*Clear communications and quick responses to those communications, are key elements of successful supply chain management*" (Sridharan *et al.*, 2005). Indeed, real-time chain-wide information increases the performance of individual enterprises – see among others Green *et al.* (2007). The use of real-time information extends beyond sole planning purposes, as Anderson-Lehman *et al.* (2004) describe in their example of the use of real-time business intelligence techniques at Continental Airlines.

The mentioned examples illustrate a third factor of change: namely, the "*information explosion*" currently taking place – see also Iastrebova (2006).

A fourth factor of change is the need for "*more flexible or agile systems*"; systems should be easier changeable or adjustable to changes in the supply chain (Sharman, 2003; Davenport *et al.*, 2004; Lee *et al.*, 2004; Lee, 2004; Tolido, 2006). Virtual organisations are an illustration of environments that require more flexibility, in that the temporary linkages require different automation (Weber, 2002). One way to achieve such agility is through different enterprise system architectures. Van Heck and Vervest (2007) envision a "*business operating system*". The business OS is a layer on top of the transactional layer which steers the operational (physical) layer of an organisation. A Business OS makes it possible to abstract processes from individual implementation details. Unfortunately, they solely present the vision, not at all hinting in the direction of specific technologies, next steps in research, or changes needed. Davenport and Brooks (2004) hint that such a transformation "*has to be measured in years or even decades*".

2.3.4 The competitive value of IT

IT has fundamentally changed the way we conduct business. Once software for transaction processing is installed, the marginal costs of processing additional transactions fall rapidly towards zero (Farrell, 2003). The same holds for software itself: once it is developed, it can be sold over and over again. Porter (2001) and Carr (2003) both augmented that IT no longer matters for organisations, rather, it has become a commodity. Its competitive additional value is decreasing: best practices are easy to copy as competitors simply buy the same system. One could argue about whether this situation has been achieved already, or will ever be achieved; innovative IT applications can still be true differentiators. Clemons and Row (1991) have already observed that "*the strategic advantage of well-developed and implemented IT lays in its ability to tap into the "unique resources" of the innovating firm, so that competitors do not fully benefit from imitation*". More recently, Siau (2003), Ordanini (2005) and Marquis (2006) came to similar findings. Although late adopters can benefit from copying technologies, they lack the innovative mindset or atmosphere that keeps them at the front of the pack (Siau, 2003); that innovative mindset is hard to copy (Ordanini, 2005). McAfee and Brynjolfsson (2008) recently showed that IT, in fact, sharpened differences between companies, increased competition, and leads to innovation; not in the least due to the fact that it is not that easy to successfully deploy enterprise systems.

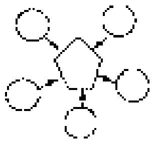
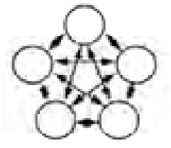
2.4 Inter-organisational systems

2.4.1 What an IOS is – definitions and typologies

"*A typical inter-organisational system (IOS) is an information system that links one or more firms to their customers [and/] or their suppliers and facilitates the exchange of products and services*" (Bakos, 1991). Such systems support the integration of inter-organisational processes (Hammer, 2001). As such, the field of IOS is wider than only the systems that enable information exchange between supply chain partners. It also includes the deployment of systems that help in collaborative product development, process control or knowledge sharing. Important to add is that an IOS is defined as "*a network-based IS that extends beyond traditional enterprise boundaries*" (Hong, 2002), therefore, it is essentially different from traditional enterprise-centric information systems. Kumar and Van Dissel (1996) introduced a frequently cited division in three different typologies of systems with different architectures. This division is given in Table 2.2. The architectural types presented can still be widely recognized in today's systems. B2B marketplaces and industry-wide coordination platforms are an example of *pooled interdependency*, whereas *reciprocal interdependency* relates to the less structured ways of inter-firm coordination that can be widely recognized in industry nowadays. Firm to firm integration, in the traditional sense, is an example of *sequential interdependency*.

Another split is made by Hong (2002). Hong's first dimension is the role linkage: horizontal versus vertical. Horizontal linkage involves parties performing common value activities – see also Cruijssen *et al.* (2007). Vertical linkage is linkage throughout the supply chain, with individual participants all performing different activities. Hong's second dimension is the process level support: either operational or strategic. Hong makes clear that "*unlike strategically oriented IOS, [an] IOS for operational support causes the participants' operations to be integrated, creating exit barriers*".

Table 2.2 – Different types of inter-organisational systems (Kumar and Van Dissel, 1996)

Type of interdependence	Pooled interdependency	Sequential interdependency	Reciprocal interdependency
Configuration			
Coordination mechanisms	Standards & Rules	Standards, Rules, Schedules, & Plans	Standards, Rules, Schedules, Plans, & Mutual Adjustments
Technologies	Mediating	Long-linked	Intensive
Structurability	High	Medium	Low
Potential for conflict	Low	Medium	High
Type of IOS	Pooled information resource IOS	Value/supply chain IOS	Networked IOS
Motives	Risk sharing and economies of scale. Market type of transactions – no integration.	Chain integration leads to reduction of SC uncertainties (gaining cost, cycle time, quality).	Process- rather than transaction support. Often temporary of nature.

2.4.2 The economic rationale of IOS

Coase (1937) envisioned that when transaction costs decrease, the need for corporations to keep all business functions in-house reduce. Over the years, transaction costs have gone down, and firms have become more specialized. IOS helps in reducing these costs to a larger extent (Malone *et al.*, 1987; Bakos, 1991). Three unique characteristics often act as incentives for IOS development (Bakos, 1991), namely: (1) It decreases the costs of exchanging and acquiring information by participating firms; (2) The benefits for the IOS innovator increase as the number of firms joining the network increases; and (3) Considerable switching costs occur when a firm shifts from one IOS to another. In the eighties, Malone *et al.* (1987) reflected on the impact information technologies were likely to have on industrial structures. They split markets and hierarchies, and forecasted an overall shift towards a proportionately larger use of markets – rather than hierarchies – to coordinate economic activity. Bakos (1997) explains that electronic markets might decrease information-asymmetry by reducing search costs. This is especially negative for sellers as their profits decline as search costs are reduced. Furthermore, providing perfect information in chains can (also) be a major point of struggle (Van Der Heijden *et al.*, 1995).

2|4|3 How IOS is applied in industry

Nowadays, most larger companies have some type of inter-organisational process and system in place (Kulmala *et al.*, 2005). Most are IOS that support operational information sharing (Bowersox *et al.*, 2003). Integration, on the tactical and strategic level, is more complicated to achieve. For that, processes often need to change radically, and trust between partners is important. However, the focus should never be solely on the quantity of information exchanged, improving the quality of the information would often be a greater benefit (Gosain *et al.*, 2004). Premkumar *et al.* (2005) relate the information processing theory of Galbraith (1974) (which we briefly discussed in section 2|2|2) to inter-organisational relationships. They show that information processing needs are heavily dependent on the type of products delivered. Per typology – they identify six – the needs differ, and therewith the type of technology support and economic rational.

Despite Malone *et al.'s* (1987) predictions, the past two decades have not shown a massive uptake of market-type technology in the business-to-business domain. On the contrary: The largest success stories come from hierarchical systems. Dell, for example, is a typical hierarchical supply chain master (Li *et al.*, 2006) that shares (demand forecast) information with many of its suppliers. It also shares data about its defect rates, engineering changes, and product enhancements. Can companies truly benefit from frequent supplier changes enabled through electronic markets (Li *et al.*, 2006)? Such changes might result in lower prices, but what about overall quality, and unique competitive features? Is not one of the great advantages of IOS that these systems imply a level of cooperation and coordination beyond the traditional arms-length relationship that exists between organisations acting as free-agents in markets, as Kumar and Van Dissel (1996) state it? Established supply chain integration with partners is an important strategic weapon (Rai *et al.*, 2006), and not in the least because it is very difficult to copy. Furthermore, we should be aware that IOS does not solely concern the technical aspects of information systems. IOS is, foremost, a human activity system that is "*subject to all risks and foibles of joint human endeavour*" (Kumar and Van Dissel, 1996). In their review on transportation exchanges, Alt and Klein (1998) confirmed many of the theories discussed above as factors leading to the limited success of these exchanges. Most markets they studied never achieved liquidity, for the simple reason that too often only cargo was offered that could not have been sold through traditional channels. Furthermore, the exchange solely offered information on loads, but little additional services and process support. Especially the larger companies benefited from information-asymmetry in traditional ways of working. The adoption was so limited that the desired "network-effect" was not reached, except for TeleRoute in France and COMIS in Germany.

Last but not least, they pointed out the expensive proprietary technologies. The success of TeleRoute can be partly explained by the Minitel backbone, which was all around in France at that time.

The situation in the transportation of physical goods has several characteristics of a pure market (De Toni *et al.*, 1994; Graham *et al.*, 1994; Lewis and Talalayevsky, 2000). The prime decision variable for shippers is cost, which is one of the important causes for the inefficiencies in the transportation of physical goods, along with a lack of integral management and unavailable information (Alt and Klein, 1998). "*Shippers tend to avoid close integration with LSPs, whereas LSPs claim to be true strategic partners but remain unable to provide the service required*" (Makukha and Gray, 2004).

Although IT investments in better coordination in the value chain can have a positive impact on market performance (Ross, 2002), coordinating activities with suppliers and customers is very complex – a high level of understanding and trust between parties is needed (Raman, 1995; Kumar *et al.*, 1998). Furthermore, note that the state-of-art in IOS for many (smaller) companies is often still the phone and fax-machine (Stefansson, 2002; Bharati and Chaudhury, 2006). Kemppainen and Vepsäläinen (2003) reflect that it is neither feasible nor profitable to have strong collaboration with all supply chain partners.

2|4|4 Establishing IOS

The first years of inter-organisational systems were mainly driven by EDI (electronic data interchange) technologies. However, implementing and operating these technologies is relatively expensive. The history of EDI trajectories in the Port of Rotterdam, is documented by Van Baalen *et al.* (2000).

The introduction of Internet-based technologies (e.g., XML) resulted in lower operating costs, since standard network infrastructures can be used, however, implementation costs are still present. As a result, the amount of information linkages in chains nowadays are still limited. Implementations tend to solely link large trading partners. Sriram *et al.* (2000) and Mukhopadhyay and Kekre (2002) observed that forced-implementations of EDI show better results than those that started voluntary. Forced implementation often results in a significant amount of additional business between those partners (Mukhopadhyay and Kekre, 2002). Not surprisingly, internal IT sophistication forms the basis for external IT integration (Wang *et al.*, 2005).

2|5 Multi-agent systems

2|5|1 What agents are

Intelligent agents began to appear in computer science and artificial intelligence (AI) literature in the late 1980s as an outgrowth of work within the objected orientation and distributed AI fields (Jennings *et al.*, 1998). Despite over two decades of history, a definition for the term agent still remains debated. Schleiffer (2005) states that "*intelligent agent technology is the articulation of human decision-making behaviour in the form of a computer program*". While this definition is elegant, it is not sufficient in that it does not explicitly specify the characteristics of human behaviour agents seek to emulate. One of the most cited agent definitions was published by Wooldridge and Jennings (1995). They put forward four distinct characteristics, namely: autonomy, social ability, reactivity, and pro-activeness. These characteristics are widely accepted as they are at the heart of what agents represent – human decision-making processes. This set of four properties has been expanded on significantly over the years and across multiple fields.Table 2.3 presents a list of agent characteristics as we find them cited throughout literature. It illustrates that agents are more than "just programs" (Anumba *et al.*, 2001).

Most early publications on agents cover work on single agent systems, which are agents that gather information on behalf of a user, or do specific tasks for them. Examples are the articles by Maes (1994), Guttman *et al.* (2001), Klusch (2001), Wooldridge (2001), and Trappey *et al.* (2004).

2|5|2 Multi-agent systems

Systems consisting of multiple agents interacting with each other and their environment are known as multi-agent systems. In such systems, not all agents are equal: each agent can have unique capabilities and objectives, representing its real-world counterpart. A multi-agent system is an assembly of different agents, with different roles, capabilities and goals – for different categories of agents, see Papazoglou (2001).

In a multi-agent system the agent construct becomes more than just an entity performing local tasks. The agent must also possess the ability to communicate and coordinate. The important characteristics of a multi-agent system are: (1) Agents need each other for completeness of information and problem solving; (2) No global control system; (3) Data is decentralised; and (4) Asynchronous computation (Rudowsky, 2004). Caridi and Cavalieri (2004) and Moyaux *et al.* (2006) add: (5) Modularity; (6) The possibility to embed multi-objective functions; and (7) The fact that design can be a step wise process, as additional benefits of MAS. Wooldridge (2005) lists the three main potentials offered by multi-agent systems: First, a MAS system resembles the organisation of the business itself, making it easier for programmers and analysts to understand its function and behaviour. Second, problem solving in the system is based on problem solving in the organisation (decentralised: no "agent" owns the whole system). Third, because agents are autonomous and always active, the system is responsive to changes and problems.

Table 2.3 – Agent characteristics and references citing these

Characteristic	Sources
Autonomy	(Castelfranchi, 1995), (Wooldridge and Jennings, 1995), (Franklin and Graesser, 1996), (Khoo *et al.*, 1998), (Wooldridge, 1999), (Fox *et al.*, 2000), (Jennings, 2000), (Bonabeau and Meyer, 2001), (Luck and d'Inverno, 2001), (Nissen, 2001), (Petersen *et al.*, 2001), (Luck *et al.*, 2003), (Rudowsky, 2004), (Schleiffer, 2005), (Samuelson and Macal, 2006)
Social ability	(Genesereth and Ketchpel, 1994), (Wooldridge and Jennings, 1995), (Wooldridge, 1999), (Fox *et al.*, 2000), (Bonabeau and Meyer, 2001), (Nissen, 2001), (Petersen *et al.*, 2001), (Rudowsky, 2004), (Schleiffer, 2005)
Communication	(Fox *et al.*, 2000), (Luck and d'Inverno, 2001), (Nissen, 2001), (Luck *et al.*, 2003), (Marik and McFarlane, 2005)
Negotiation	(Luck *et al.*, 2003), (Marik and McFarlane, 2005)
Cooperation	(Luck and d'Inverno, 2001), (Marik and McFarlane, 2005), (Schleiffer, 2005)
Reactivity	(Wooldridge and Jennings, 1995), (Wooldridge, 1999), (Luck and d'Inverno, 2001), (Rudowsky, 2004)
Pro-activeness/Goal oriented	(Wooldridge and Jennings, 1995), (Franklin and Graesser, 1996), (Wooldridge, 1999), (Fox *et al.*, 2000), (Jennings, 2000), (Luck and d'Inverno, 2001), (Nissen, 2001), (Petersen *et al.*, 2001), (Rudowsky, 2004)
Situatedness (both time and space)	(Franklin and Graesser, 1996), (Wooldridge, 1999), (Jennings, 2000), (Luck and d'Inverno, 2001), (Schleiffer, 2005)
Decision-making	(Wooldridge, 1999), (Schleiffer, 2005), (Samuelson and Macal, 2006)
Ability to influence environment	(Franklin and Graesser, 1996), (Wooldridge, 1999)
Reasoning/Problem solving	(Jennings, 2000), (Luck *et al.*, 2003), (Schleiffer, 2005)
Learning	(Wooldridge, 1999), (Luck *et al.*, 2003),
Robustness	(Wooldridge, 1999), (Bonabeau and Meyer, 2001), (Schleiffer, 2005)
Coherence in sensing environment	(Franklin and Graesser, 1996), (Wooldridge, 1999), (Schleiffer, 2005)

The methodologies implemented to achieve communication and coordination are among the defining features of a MAS (Odell *et al.*, 2002): "*designing an agent-based system is not just about designing the agents; it is also about designing the agent environment and interaction*." Agent communication is described by both the language and the method by which they exchange messages. Agent coordination – or "interaction" – refers to the mechanism by which agents organise themselves to work on the full system's problem.

Agent communication is a field of study unto itself, situated at the crossroads of linguistics, cognitive science, artificial intelligence, formal logic, and computer science. The field of communication is dominated by both language semantics and dialogue protocols. Language semantics refer to the meaning that is expressed in a language or code. A dialogue protocol, additionally, specifies a set of rules that regulate the dialogue between two or more communicating agents (Endriss *et al.*, 2003). Coordination among agents in a multi-agent system is a critically important process to ensure that the system acts in a coherent manner (Nwana *et al.*, 1996). For an overview of developments in coordination schemes, we refer the reader to Durfee *et al.* (1989), Jennings *et al.* (1998) and Caridi and Cavalieri (2004). Agents in a multi-agent system coordinate with each other in order to come up with a solution to the full problem (Lesser and Corkill, 1987). As such, the interaction patterns including the sequence become important, and are at the basis for modelling a MAS system (Da Silva and De Lucena, 2007). The important role of the FIPA (the Foundation for Intelligent Physical Agents) and its standards is mentioned by Willmott *et al.* (2004). The FIPA was founded in 1996 to promote interoperability between heterogeneous software agents and agent-based systems, and FIPA standardized 25 different interactions. The FIPA is now a committee of the IEEE organisation.

De Weerdt *et al.* (2006) formally define a multi-agent planning problem as following: "*Given a description of the initial state, a set of global goals, a set of (at least two) agents, and for each agent a set of its capabilities and its private goals, find a plan for each agent that achieves its private goals, such that these plans together are coordinated and the global goals are met as well.*"

MAS has roots in computer science with a solution focus on "*coordination through communication*" more than on the best possible algorithms. Shim *et al.* (2002), Schneeweiss (2003), Luck *et al.* (2004) and Van Dijke (2008) see an opportunity for collaboration between the MAS and OR research fields, especially to increase the optimisation characteristics of MAS. The survey paper on distributed planning by desJardins *et al.* (1999) illustrates how separated the different research communities actually are. The paper is written from an Artificial Intelligence perspective, and it uses totally different vocabulary and terminology than papers from other communities. Furthermore, there is hardly any referencing to publications coming from the CS, OR and IS communities. The same holds for most papers published in these respective domains.

2|5|3 Reasons to use agents in logistics

Most logistical planning systems are of a centralised nature, see section 2|3|2. Centralised information processing in supply chains results in a minimum amount of transactions, and one single point for optimisation (Lewis and Talalayevsky, 2004).

However, centralised solutions have difficulties coping with a high degree of complexity and change, which requires the solution to be robust to disruption and reconfigurable when necessary (Marik and McFarlane, 2005). Decentralised solutions, of which multi-agent systems are an example, seem to be suitable in situations where a classical centralised solution is less than suitable and where the distribution of information and decision-making is necessary. Marik and McFarlane specifically mention three possible characteristics that are in favour of a decentralised solution approach. First among these is a centralised solution's (theoretical) infeasibility. At any time, each possible decision-making node has only part of the information required to make the decision. Impracticality is the second characteristic. Even if all information is potentially available to each decision-making node, practical constraints (time, cost, and quality) on making information centrally available or on performing synchronised, centralised decision-making may inhibit a centrally based decision. The third characteristic is inadvisability. Even if centralised decision-making is feasible and practical, it might still be inadvisable. For example, one of the benefits of distributed planning is that more computer power can more easily be involved in the decision-making process (Mönch *et al.*, 2006). An interesting pro-centralised systems paper has been written by Lewis and Talalayevsky (2004). Lacking in their assessment is the (important) fact that the costs of transactions rapidly fall with further advances in technology. In Table 2.4 we have compiled a list of advantages and disadvantages of both paradigms, also incorporating the issues identified by Parunak (1996). Note that most of the SCM software in industry now is typically of a centralised nature. This is especially illogical for operational-level decision-making (Singh *et al.*, 2007). The problem characteristics in logistics closely match those of an ideal multi-agent system, according to Davidsson *et al.* (2005).

Fischer *et al.* (1996) succinctly identify four main reasons why applying multi-agent systems to transportation planning problems is appealing. First, transportation is an inherently distributed task. Trucks and jobs are not only geographically distributed, but also maintain a level of autonomy in the field. Second, transportation must cope with dynamic events. Agent architectures have the capability to handle such dynamics. Third, in order to use classical methods for transportation planning, a large amount of information must be maintained centrally. Fourth, transport firms engage in a high level of negotiation and cooperation in performing their daily tasks. MAS have the capability to include such cooperative capabilities, while optimisation-based algorithms do not.

Table 2.4 – Pros & cons of centralized and distributed planning approaches

Centralised planning		Distributed planning	
+	Plan (theoretical) optimal Overseeing the entire system Computational stability Limited number of connections (and transactions) to parties	+	Plan (theoretical) optimal Overseeing the entire system Computational stability Limited number of connections (and transactions) to parties
-	Relatively long calculation times (although heuristics can be very fast) Theoretical infeasibility (required information is distributed – and should be made available centralised) Difficult to include events Difficult to get an inter-enterprise perspective Single point-of-failure	-	Relatively long calculation times (although heuristics can be very fast) Theoretical infeasibility (required information is distributed – and should be made available centralised) Difficult to include events Difficult to get an inter-enterprise perspective Single point-of-failure

2|5|4 Application domains for agents

Where have multi-agent systems been really implemented? Roth (2004) claims that 30 years after their first inception, the only widespread incarnation of mobile software agents is in the malware domain: computer viruses, spyware, Trojan horses, et cetera. This pessimistic conclusion can easily be falsified: modern computer games, for example, are a domain in which one can find widespread MAS application (Luck *et al.*, 2004). Agent technologies are a way to add intelligence to computer games and a means to let the system learn from the behaviour of the user. Another domain is telecommunication networks (Luck *et al.*, 2004). Agents have tasks as diverse as load-balancing, selling & buying of network capacity, routing, self-healing, et cetera. Important in this domain are the real-time behaviour and quickly coordinated interactions.

Application domains that are likely to benefit from multi-agent systems have the following characteristics (Sierra, 2004): (1) very fast interactions; (2) interactions are repeated with either (a) high communication overheads, or (b) a limited domain so that learning done by the agent about user behaviour is effective; (3) each trade is of relatively small value; (4) the process is repeated over long periods of time; and (5) the product traded is relatively easy to specify. Chen (1999) suggests that environments with a large amount of variance and continuous change are interesting. Mentioning similar characteristics, several scholars foresee future applications in domains such as automated marketplace trading, defence simulation and training, industrial control systems, simulation modelling, smart sensor networks, enterprise system integration, event management systems, and planning and scheduling in logistics and SCM (Maes *et al.*, 1999; Shen and Norrie, 1999; Luck *et al.*, 2004; Marik and McFarlane, 2005; Belecheanu *et al.*, 2006; Zimmermann *et al.*, 2006).

A future replacement of ERP in enterprises through multi-agent systems is envisioned by Lea *et al.* (2005). Their paper introduces a conceptual design, but does not present any plans for prototyping or concept evaluation. The paper encompasses a long list of advantages multi-agent architectures have over ERP architectures. The list – which we included in Table B.1 in Appendix A – can be an instrument when assessing future systems or prototypes. This is not in the least due to the fact that several factors mentioned are opposite of other findings from literature. Many MAS in SCM papers written to date do not deal with chain relations, coordination, and negotiation. The Trading Agent Competition (TAC), for example, solely deals with constructing smart "agents" that receive information, process it, and react accordingly – see for example Collins *et al.* (2009). Another example are the agents in the beergame article by Kimbrough *et al.* (2002), which position agents as software-based decision makers, resulting in more rational, hence programmed, decisions than humans make.

2.5.5 Out of the lab

The number of papers on concrete agent applications in industry is limited. Many articles that claim to be application papers are, in fact, theoretical design attempts. The papers by Spieck *et al.* (1995), Khoo *et al.* (1998), Adler and Blue (2002), Blake (2002), Huhns *et al.* (2002), Frey *et al.* (2003), Trappey *et al.* (2004), Lima *et al.* (2006) all have in common that they introduce a theoretical design. However, they do not validate nor test these, or detail plans on how to approach this. Our observation, that most agent research has a predominantly theoretic design orientation, is shared by Wareham *et al.* (2005). Little empirical or experimental research is done to really apply agent concepts in practise.

Although the (scientific) field is dominated by these "*claim-to-be-practice-oriented*" articles, a few examples of MAS application in industry are documented. Thomas and Seibel (1999) reported on an application of MAS in the cargo operations of Southwest Airlines. Dynamic agent-based routing of parcels in the Southwest network resulted in considerable operational savings compared to the previous manual way of planning: parcels took shorter routes, with less shifting to other planes, and arrived earlier at their destinations. Each parcel is equipped with an agent, which found its way through the network by negotiating with airplane agents and other parcel agents. The article reports on a simulation study, and mentions plans for a real-life implementation of the concepts. Unfortunately we could not find any follow-up traces online nor in journals. Gambardella *et al.* (2003) report on the utilisation of multi-agent systems in the dynamic planning of delivery trucks at Pina Petroli, a petrol firm in Switzerland. The MAS is used to direct a fleet of trucks in distributing heating oil to residential customers in Switzerland. The scheduling is complex due to variability in trucks, unpredictable and complex customer orders, and unexpected traffic and weather conditions. The MAS works as a decision support system, where human planners can either accept the suggested solution or make adjustments. Similarly, Himoff *et al.* (2006) describe a prototype application build by Magenta Technologies. Business initiatives to apply agents at Unilever, Hewlett-Packard, and Enron, are mentioned by Bonabeau and Meyer (2001). An initiative at Procter & Gamble is mentioned by Anthes (2003). Multi-agent systems play an essential role in the DARPA advanced logistics project, documented by Adali and Pigaty (2003).

TNO recently concluded "*that many ideas on intelligent agents exist, but applications and tests in the field of logistics are hard to find*" (Van Rijswijck and Davydenko, 2007). Furthermore, they state that "*not much attention is spent on performance of agent systems. It appeared that the technology has not gone further than university research laboratories and some pilot projects*."

Several software firms are active in the multi-agent system domain, most of them with roots in research and academia. Examples include: Magenta Technology, NuTech Solutions (which acquired the former BiosGroup), Whitestein Technologies, Tryllian, INITI8, MP Objects and Almende / Deal Services. Their success seems to be limited, except for the first two companies mentioned, which claim to have several customers already. Despite the whitepapers on their websites, it remains unclear how successful these companies really are. A more complete list of companies and technologies can be found at the website of AgentLink (`www.agentlink.org`). Note that a serious amount of listed companies and technology toolkits have disappeared over the last few years, and that several firms that list themselves as software companies should, in reality, be listed as research labs.

Multi-agent systems do have much in common with the Service Oriented Architectures (SOA) domain (Davies *et al.*, 2004; Papazoglou and Heuvel, 2006; Sonntag, 2006), which is built upon the WebServices technology stack. WebServices (Ma, 2005) enable machine-to-machine interaction over a network, generally the Internet, thereby communicating through XML messages that follow the SOAP protocol (Simple Object Access Protocol) and carry a machine-readable service description in the form of a WSDL (Web Services Description Language) message. WebServices have gained a great amount of industry support. Due to standardization, many technologies have started to appear, all clustered under the umbrella of SOA. Examples are BPEL and BPML, two business process orchestrating languages that are executable (for other WebServices) (Peterson, 2003). Being an umbrella of different technologies, services and concepts, SOA has become a buzz word in industry. It is perceived to be a new type of super-glue that can bring existing (heterogeneous) IT environments to the next level by interconnecting existing components and establishing new services on top of these (Tolido, 2006). One of the thoughts behind SOA is to construct (new) systems by connecting smaller (existing) components or systems. Vendors such as BEA, Cordys, IBM, Microsoft, Oracle and SAP have developed SOA technology platforms, and the large ICT consultancy firms, such as Accenture, AtosOrigin, CapGemini, IBM, and Logica actively push SOA in their consulting practices.

However, there are several important differences between SOA and MAS: service oriented architectures do not necessarily encompass the characteristics of autonomy, social ability, reactivity, and pro-activeness, which are important building blocks for multi-agent systems. These characteristics can be potentially of great value when applied in an industrial context, as Marik and McFarlane (2005) reason.

2|6 Implementation and adoption

2|6|1 Implementation of enterprise systems

Literature on the implementation of complex enterprise systems can be divided into literature covering single-enterprise implementations (often ERP) and literature covering inter-organisational systems. ERP implementations have a huge impact on organisations. Often organisations need to "*learn to function in radically different ways*" (Robey *et al.*, 2002). An ERP implementation is an organisational change process, in which technology is an enabler. The change process does not end when the system is implemented (Biehl, 2007). Employees absorb knowledge, and develop new ways of working. One of the aspects that makes an ERP implementation a tough job is customisation (Ragowsky, 2002). ERPs are standardized software packages, with thousands of functions, features and screens. ERPs are largely based on best practices: a standard representation of how a certain process should be conducted in a company. Not surprisingly, most processes in practise are not fully (software-vendor specified) best practice compatible. Hence, this demands not only customisation of the software, but also changes in the way of working. In Davenport's (1998) terms: "*putting the enterprise in the system*".

Sridharan *et al.* (2005) researched implementations of APS (advanced planning and scheduling) packages and came to similar conclusions. APS packages are often hard to implement, foremost because of their complexity. They warn that great care is needed when changing "standard templates" – which is needed in most cases. Before clients switch to a new system, or changed templates, rigorous and adequate testing is needed to see if client's requirements are truly met. Goldratt (2000) states that implementation consultants must identify the hidden assumptions in how people were doing their jobs prior to implementing the software, and find out which of these assumptions no longer hold with the software in place. For example, if your (ERP) system enables you to run financial reports once a day instead of once a month, but you still only run them once a month, you won't achieve the benefit you could be getting.

IOS literature adds additional dimensions such as inter-organisational trust, inter-organisational business process redesign, and shared standards. For background reading, we refer the reader to, among others, Kumar and Van Dissel (1998) and Ibrahim (2006). From IOS and ERP implementation literature, we have compiled a list of success factors for the implementation of inter-organisational systems, which is presented in Table 2.5.

Why do companies choose certain systems? Shang and Seddon (2002) found that the primary selection criteria in their sample of 233 enterprise system implementations were (in order of frequency of citation): (1) business fit; (2) ease of implementation; (3) vendor services and support; (4) special industry or application capabilities; (5) product affordability; and (6) compatibility with other systems. Often, external parties, more specifically industry analysts and consultancy firms, have an important role in an enterprise's buying decision (Moonen, 2003). Furthermore, one should realise that excess inertia exists, such as high switching costs. Such switching costs restrain a switch to more open and better standards or solutions (Zhu *et al.*, 2006).

2.6.2 Adoption of enterprise systems

Successful implementation of a system does not necessarily lead to actual adoption of it (Devaraj and Kohli, 2003). Too often systems in place are not used as intended. The frequently cited survey by DeLone and McLean (1992) reveals that the six most important factors for adoption are: (1) System quality, (2) Information quality, (3) Use, (4) User satisfaction, (5) Individual impact, and (6) Organisational impact. With respect to the use of technology, Venkatesh *et al.* (2003) compared a large body of implementation and adoption theories, and conclude that there are three main dimensions. These are in line with earlier findings from (Henderson and Venkatraman, 1993), namely: (a) Performance expectancy; (b) Effort expectancy; (c) Social influence. These factors primarily measure usage intention, instead of actual usage. Actual usage has everything to do with human-system interaction (Doll and Torkzadeh, 1998); hence the importance of a good user interface.

Van Hillegersberg (2006) introduces a framework for the adoption of networked systems by organisations, based on Kurnia and Johnston (2000) – see Figure 2.1. The important difference in the discussed work is that this model also implies alignment of systems and processes between companies instead of sole alignment internally.

Table 2.5 – Critical success factors in the implementation of information systems

Critical success factor	Sources
Top management support	(Akkermans and van Helden, 2002),(Brown and Vessey, 2003), (Ngai and Gunasekaran, 2004), (Lee *et al.*, 2005), (Jones *et al.*, 2006), (Lu *et al.*, 2006), (Zhu *et al.*, 2006), (Biehl, 2007)
External pressure to implement	(Kumar *et al.*, 1998), (Sriram *et al.*, 2000), (Mukhopadhyay and Kekre, 2002)
Cross-organisational implementation team	(Akkermans and van Helden, 2002), (Brown and Vessey, 2003),(Lu *et al.*, 2006), (Biehl, 2007)
Inter-organisational BPR (Business Process Redesign)	(Li and Williams, 1999), (Goldratt, 2000), (Akkermans and van Helden, 2002), (Robey *et al.*, 2002), (Nahm *et al.*, 2003), (Wang and Tai, 2003), (Ngai and Gunasekaran, 2004), (Lee *et al.*, 2005), (Lu *et al.*, 2006), (Biehl, 2007)
Own house in order	(Li and Williams, 1999), (Frohlich, 2002), (Wang and Tai, 2003), (Ngai and Gunasekaran, 2004), (Lee *et al.*, 2005), (Lu *et al.*, 2006)
Strong integration with internal systems	(Wang and Tai, 2003), (Lu *et al.*, 2006)
Shared standards	(Lee *et al.*, 2005), (Lu *et al.*, 2006)
Third parties fill gaps in expertise	(Brown and Vessey, 2003)
Vendor support	(Akkermans and van Helden, 2002)
Careful package selection	(Akkermans and van Helden, 2002)
Education and training	(Goldratt, 2000), (Robey *et al.*, 2002), (Nahm *et al.*, 2003), (Ngai and Gunasekaran, 2004), (Jones *et al.*, 2006)
Trust needed	(Kumar *et al.*, 1998), (Li and Williams, 1999), (Meijer *et al.*, 2006)
Project urgency	(Goldratt, 2000), (Biehl, 2007)

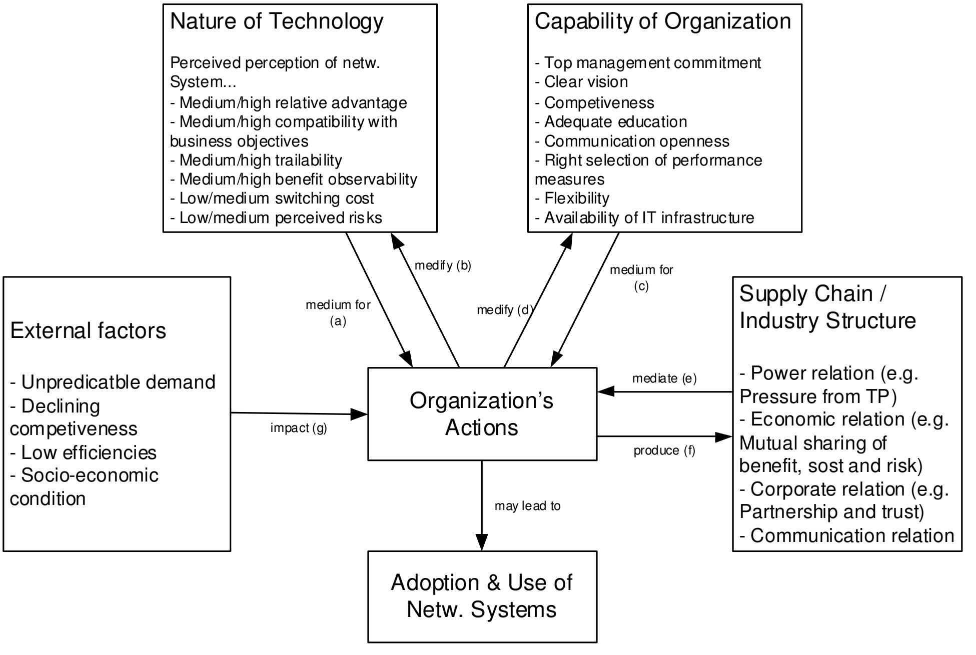


Figure 2.1 – The adoption of networked systems (Van Hillegersberg, 2006)

2.6.3 Factors that hinder multi-agent system adoption

Actual agent implementation in industry is limited, as we noticed before. Caridi and Cavalieri (2004) state that the few actual applications that are referenced in literature are all outcomes of research programs. Industrial companies and software houses seem to be not yet receptive of the agent paradigm.

This phenomenon has been studied by several scholars. In Table 2.6, we give an overview of factors that hinder MAS adoption in industry, as identified. Although Lea *et al.* (2005) foresee lower total cost for implementing and operating multi-agent systems, serious doubts exist by other scholars concerning the factor "c*ost*", at least in the short run. Cost has also a relationship with most other factors. For example, the factor "*accuracy and correctness of the results*", is mentioned as an important factor in several sources. Most agent work found in literature is only tested and evaluated to a limited extent. Another related factor is what we have named "*the legacy of legacy design & techniques*". Most software architects, developers and consultants are so acquainted to the traditional way of system development: centralised monolithic systems. Their skills are focused on established technologies (Bauer and Müller, 2004). "*Security*" of agents is seen as a threat, also, specifically the issue that more points of possible failure are created. "*Legal and ethical issues*" are perceived to be important, especially also when more "*intelligence*" is added to the system, and outcomes are not directly traceable back to a clear set of decision rules. "*Scalability*" of the system, "*acceptance by* *users*", and the importance of a "*central role for human decision makers*" are also factors. Mentioned by many scholars is the factor we refer to as "*stuck in academic prototyping*". Petrie and Bussler (2003) suggest that the agent research community should reinvent itself, and integrate its concepts and ideas with the SOA domain. Specifically, they state that "*ignoring industrial technologies leads only to published papers, while ignoring well-studied advanced distributed computing principles can lead to slow industrial progress due to the necessity for re-invention based on experience*". Parunack (2000) pleads for a clear interaction between researchers and business developers to create a convergence of MAS concepts with industrial applications.

Three other factors turn out to be important. The factor "*standards*" is mentioned, just like "*misapplication*", which basically relates to the fact that agents cannot solve all problems. Fonseca *et al.* (2001) point to the FIPA standards as an important enabler for real implementations of multi-agent systems, however, Marik and McFarlane (2005) and Belecheanu *et al.* (2006) wonder whether these standards are actually so standard. A last factor is the need for "*professional development methods*". Jennings and Wooldridge (1998) make clear that multi-agent systems need different design methodologies. Literally: "*it is more than just throwing together a number of agents and let the system run*". Pena *et al.* (2006) plead for a focus on product software, instead of the current practice of non-standardized single MAS (prototypes), which lead to the generation of legacy from the start. The interaction between, and roles of, human decision-makers versus intelligence within the agents is an important and complex design factor for any MAS (Hess *et al.*, 2000; Nissen, 2001). Design methodologies and professional multi-agent development environments (Bauer and Müller, 2004; Belecheanu *et al.*, 2006) need to be further developed and made part of the education of tomorrow's software engineers (Parunak, 2000; Marik and McFarlane, 2005; Wagner *et al.*, 2005). March *et al.* (2000) and Luck *et al.* (2004) make a call to action to industrial researchers to start experimenting with agent prototype applications in (and with) industry, especially prototypes that span inter-organisational boundaries.

Table 2.6 – Factors that hinder adoption of MAS in industry

Factor	Sources
Cost	(Caridi and Cavalieri, 2004), (Rudowsky, 2004), (Marik and McFarlane, 2005)
Security	(Roth, 2004), (Rudowsky, 2004), (Belecheanu *et al.*, 2006), (Sonntag, 2006)
Legal / ethical issues (i.e., when more "intelligence" is added to a system)	(Sandholm, 1999), (Rudowsky, 2004)
Accuracy and correctness of the results / Guarantees of operational performance	(Jennings *et al.*, 1998),(Caridi and Cavalieri, 2004), (Rudowsky, 2004), (Marik and McFarlane, 2005), (Davidsson *et al.*, 2005), (Belecheanu *et al.*, 2006)
Scalability	(Jennings *et al.*, 1998),(Roth, 2004), (Marik and McFarlane, 2005)
Acceptance by users	(Sandholm, 1999), (Rudowsky, 2004), (Belecheanu *et al.*, 2006)
Central role human decision-makers	(Hess *et al.*, 2000), (Nissen, 2001), (Wagner *et al.*, 2005), (Nissen and Sengupta, 2006)
Professional development methods	(Jennings *et al.*, 1998), (Parunak, 1999), (Hess *et al.*, 2000), (Parunak, 2000), (Bauer and Müller, 2004), (Belecheanu *et al.*, 2006), (Marik and McFarlane, 2005), (Wagner *et al.*, 2005), (Peña *et al.*, 2006)
Standards	(Marik and McFarlane, 2005), (Belecheanu *et al.*, 2006)
The legacy of legacy design & techniques (traditional focus on centralised control systems)	(Caridi and Cavalieri, 2004), (Marik and McFarlane, 2005), (Wagner *et al.*, 2005)
Misapplication (cannot solve all problems)	(Jennings *et al.*, 1998),(Marik and McFarlane, 2005), (Wagner *et al.*, 2005)
Stuck in academic prototyping	(Jennings *et al.*, 1998), (Parunak, 1999), (March *et al.*, 2000), (Parunak, 2000), (Petrie and Bussler, 2003), (Bauer and Müller, 2004), (Caridi and Cavalieri, 2004),(Davidsson *et al.*, 2005), (Wareham *et al.*, 2005),(Wagner *et al.*, 2005)

2.7 Design of multi-agent systems

2.7.1 Where agents differ from object orientation

From a software engineering design perspective, it is good to understand where agent-based approaches differ from traditional Object Orientation (OO) development methods. Most industrial methodologies for software development are based on OMG's Unified Modelling Language (UML) accompanied by process frameworks such as the Rational Unified Process (RUP) (Bauer and Müller, 2004). Jennings (2001) listed the most compelling differences between agent-based and OO: (1) Objects are generally passive in nature – they need to be send a message before they become active; (2) Objects do encapsulate state and behaviour realisation, they do not encapsulate behaviour activation (action choice) – more specifically, an agent can have behaviours which are reactive, proactive, and/or social in nature; (3) OO fails to provide an adequate set of concepts and mechanisms for modelling complex systems; (4) OO approaches provide minimal support for specifying and managing organisational relationships; and (5) Agents have at least one thread of control but may have more, whereas Objects have solely one thread of control (Wooldridge, 1999). Nevertheless, as Jennings explains, one can construct agent-based systems utilising OO techniques and environments. Multi-agent system development methods might be not that different from current methods, since many object-oriented analyses start from precisely this perspective: "*we view the world as a set of autonomous agents that collaborate to perform some higher level function*" (Jennings and Bussmann, 2003). Similarly, Scholz-Reiter and Höhns (2002) state that agents are "*a powerful, natural metaphor for conceptualizing, designing and implementing complex, distributed applications*". For this reason, the main concepts and tenets of the approach should be readily acceptable to software engineering practitioners. After all, agent-based systems are computer programs and all programs have the same set of computable functions (Nwana and Ndumu, 1999). As such, agent techniques represent not so much a revolution, but merely a natural progression of current software engineering thinking.

2.7.2 Methodologies for engineering multi-agent systems

The state-of-art in methodologies and notations for the development of agent-based systems was surveyed by Bauer and Müller (2004), who identified three roots. First are approaches based on knowledge engineering principles. These generally lack support for specific agent-related constructs. Second are pure agent-oriented approaches which provide rich support for modelling artefacts such as goals, intentions and organisations. Third, several specific methodologies were developed that extend state-of-the-art OO approaches, with agent-oriented features, and Agent UML is an example of the latter (Cervenka *et al.*, 2005). For a discussion and comparison of agent-oriented methodologies, we refer the reader to Shehory and Sturm (2001), Wooldridge (2001), Tran *et al.* (2005) and Mes *et al.* (2007). Most methodologies consist of the following steps (Mes *et al.*, 2007):

(1) Decomposition of the system into multiple functionalities; (2) Allocation of functionalities to agents; (3) Establishing interaction protocols between agents; and (4) Designing the decision-making capabilities of agents.

Mes *et al.* (2007) and Pokahr *et al.* (2008), plead to make simulation an integral part of any MAS design in order to compare different alternative designs, and to experiment with different mechanisms. Simulation might deliver new insights for engine design by revealing rule-sets for specific situations to be applied in the engine.

2.7.3 The role of humans in software systems

One important success factor, which we touched upon before, is the role of humans in the system. In most agent systems designs, human agents play an important role. They either perform specific tasks in the system or monitor the state of the system. Nissen and Sengupta (2006) state that "*when tasks become particularly complex, novel or risky, humans should be the decision makers, supported by smart software support systems*". They plead to automate the simpler (operational) tasks, making humans focus on more strategic processes. Humans monitor operational transactions and can step in and correct decisions when needed. The role of software differs with the type of decision support provided. The more strategic tasks, which have a high(er) impact, are supported through information gathering and decision preparation, not so much through automation. More operational tasks generally tend to be more standard and thus easier to automate.

Automation generally takes away the emotional element in decision-making. Kimbrough *et al.* (2002) demonstrate in their paper, "*agents playing the beergame*", that replacing human decision steps might be beneficial also for supply chains, since the behaviour of the chain as a whole becomes more rational and less emotional. They demonstrated that their software agents did better than humans, consistently throughout several scenarios.

Wooldridge (2005) already mentioned that it can be difficult for the planner to accept the decision of its agent if it is not clear how the agent derived to its decision, or when the planner is uncertain about the extent to which his interests are represented by the agent. Likely, however, communication processes are better understandable to planners than traditional centralised architectures with a "black box" optimisation engine. Human-computer interaction is a complex and broad topic of research by itself. We do not try to provide an extensive overview here, but would like to refer the reader to, among others, Krauth (2008).

2.8 Discussion

2.8.1 Summary

Transportation is a logistical activity that deals with transporting goods from one party to another, often performed by an independent third-party. The main selection criterion for selecting a service provider is cost. The sector is facing huge challenges. Margins are going down, competition is increasing, and environmental issues and legislation are creating additional challenges. Information technologies are a candidate to reduce costs; information flows can partly replace physical flows by cutting out inefficiencies. These inefficiencies generally concern the handling of uncertainties within the wider supply chain context.

Logistical costs have been reduced substantially in intra-company logistics; the next frontier for cost reductions is the inter-organisational domain. Transportation is an in-between-companies activity, and, as such, it is a logical candidate for the application of inter-organisational systems. However, many early attempts to construct IOS for transportation failed painfully. These systems were generally perceived to be too costly, too static and inflexible. Sharman (2003) reasons that the ERP backbones that firms currently have might, in fact, be the major hindering factor to introducing flexibility and interconnectivity between information systems throughout the supply chain. Van Heck and Vervest (2007) observe that the time is ripe for another level in enterprise systems. They position the "*business operating system*" as the next level, a business layer that orchestrates underlying technologies. However, they do not go into any detail on how this would or could look like.

Multi-agent systems (MAS) might be an instrument to realise this. MAS are systems constructed of multiple interacting agents within a particular environment. Generally speaking: data are decentralised, a global control system is lacking, and computation is asynchronous. The focus in MAS is on "*coordination through communication*". In principle, it hardly matters whether this communication is intra- or inter-enterprise in nature.

The domain of transportation seems to be an ideal domain for multi-agent systems application (Fischer *et al.*, 1996; Caridi and Cavalieri, 2004; Davidsson *et al.*, 2005). Planning in transportation has to cope with large amounts of distributed and dynamic data. Furthermore, transportation firms are, by nature, engaged in a high-level of negotiation and cooperation in the performing of their tasks. Nevertheless, the amount of concrete MAS implementations in industry is limited. Agents remain largely an academic topic and instrument, and most of the papers researchers have produced to date lack a sense of realism.

Developments in transportation technologies, such as real-time data generated through GPS and RFID, shape new possibilities for control. The SOA domain already possesses some characteristics of multi-agent concepts, and has potential to further incorporate MAS concepts. MAS might be an interesting alternative to traditional system design, as many traditional designs resulted in inflexible hard-wired costly linkages, which form a major contrast with the business practice of cost-oriented switching and limited integration. One of the technology's main potentials is the different way of modelling and implementing systems. Not as pre-shaped best-practice oriented as ERP systems are, but modularized application and company-oriented, ground-up modelling. Furthermore MAS make it possible to integrate the role of humans differently in the system. A MAS is, in fact, a clustering of human and computerised roles. A perfect MAS design methodology does not yet exist. Experimentation and simulation are recommended instruments in development trajectories (Mes *et al.*, 2007; Pokahr *et al.*, 2008). At the same time, we should not neglect lessons learned from previous generations of systems (Brown and Vessey, 2003).

2.8.2 Fundamental paradigm shifts

Throughout the process of performing this literature review, we have identified several paradigm shifts which are currently taking place, and have implications to our research. These shifts are listed in Table 2.7.

Table 2.7 – Paradigm shifts identified

#	Paradigm shift as identified
1	The trend in (and requirements for) enterprise systems shows a shift from (offline) batch-wise intra-enterprise to real-time inter-enterprise. This results in a fundamental challenge for systems engineering and development.
2	Increasingly, distributed system architectures are becoming an alternative for centralised system architectures. Distributed architectures seem to have a more natural fit with the business environment.
3	Process uncertainty in a company can be reduced through chain-wide information integration and chain collaboration – see, among others, Galbraith (1974).
4	Over the past decades, IT has lead to inflexible non-adaptive hard-wired processes; a new generation of IT systems should be constructed more adaptive to change.
5	Information systems have become ubiquitous; information is everywhere. As a result, decisions can be made at anytime, anywhere.
6	Although we recognize the first transfer of MAS concepts to industrial practice, MAS is still largely an academic topic.

Chapter 3 MAS & inter-organisational processes

3|1 Introduction

The application of multi-agent systems within supply chains has received limited attention in the literature, as we noted in the previous chapter. In this chapter, we report on an explorative research project, which we undertook in the beginning of our research. The chapter does not detail one specific research question. This research was part of a Dutch government funded Connekt research project, entitled "*Applications of Software Agents to Supply Chains*" which ran throughout 2003. The project was a joint effort by RSM Erasmus University, TNO and A&S Management.

This explorative research uses a mixed method approach. We conducted desk research, interviews, and a workshop with technology vendors, researchers and potential users. The workshop consisted of a mixture of plenary sessions and group sessions, and focused on inter-organisational processes in the Business-to-Business domain.

The main objective of the research in this chapter was to identify opportunities for multi-agent systems to support inter-organisational processes in current and future supply chains. In order to investigate this, we introduce a framework (in section 3|2) that we developed to study the use of agents in supply chains. The industry workshop (section 3|3) utilised the framework and provided insights on where industry foresees the largest potential for agent applications. The chapter concludes with discussion in section 3|4.

3|2 A framework for MAS application in supply chains

Literature did not provide us any concrete classification or frameworks to identify specific inter-organisational processes in which (multi-agent) systems can be expected to deliver an added value. Therefore, we develop a framework that starts from a supply chain coordination perspective combined with a process approach. To this end, we define so-called "*operating environments*": specific situations in which supply chain partners coordinate their processes. The framework has two dimensions.

The first dimension is the type of coordination. Hong and Kim (1998) classify Inter Organisational Systems (IOS) for inter-company coordination (or collaboration) into three main types: vertical, horizontal and cross IOS. A *vertical IOS* links organisations that play different roles in a value chain. It generally supports the value chain of an IOS participant. A *horizontal IOS* links a homogeneous group of organisations in order to foster their mutual cooperation. It typically reflects a market coalition or partnership – for a good overview of horizontal cooperation in transport and logistics we refer to Cruijssen *et al.* (2007). A *cross IOS* is an IOS that is both horizontally and vertically linked.

Cross IOS enable benefits resulting from vertical cooperation combined with resource-oriented incentives of horizontal cooperation.

The second dimension is the supply chain process approach used by Scholz-Reiter & Höhns (2002) – who use the definition of supply chain processes from the SCOR-model (SCC, 2003). The SCOR-model is used in consulting and scientific research to describe and structure supply chain processes in detail. Scholz-Reiter & Höhns' approach is purely conceptual and does not present practical examples or a detail description of the supply chain processes to be supported by agents.

The classification we develop is based on these two dimensions: the supply chain linkages and the SCOR delivery and source processes. We focus solely on the SCOR Source (S) and Deliver (D) processes, since these are the border-spanning processes, which connect to processes of chain partners. Combining the two dimensions results in a framework (see Figure 3.1) in which we distinguish seven types of agent applications for B2B, namely:

Agent-based horizontal sourcing – Horizontal sourcing involves the inter-organisational coordination of demand by parties that perform similar activities. Companies may want to source together, because they jointly can achieve cost reductions, for example, by bundling and coordinating purchasing orders, which results in a larger total demand than they can realise independently. Such collaboration is easiest in commodity markets, where products are highly standardized. Horizontal sourcing seems less likely in specialized markets, because the necessary negotiation on the product characteristics quickly becomes too complicated. Agents applied in horizontal sourcing can search for which partners want to join in a collective purchasing order – to gain buying power, to negotiate terms (minimum quality, maximum prize, et cetera) and choose a broker to interact with the supplier on behalf of the customers.

Agent-based horizontal delivery – Horizontal delivery concerns the coordination of (consolidated) deliveries. Here, parties work together to obtain a higher performance in transportation or resource utilisation (e.g., warehousing). For instance, carriers exchange shipments based on several criteria (size, weight, and destination) to achieve full-truck-loads (FTLs) or a higher frequency of delivery. Moreover, businesses may enhance their service offerings by connecting to the service offerings of other companies in the same sector (Blake, 2002). A multi-agent environment can offer functionality to realise coordination between competitors through negotiation. Partial transparency is possible.

Agent-based vertical sourcing – In vertical sourcing, a buyer identifies potential suppliers, and negotiates product price, service and quality elements. Vertical sourcing entails the complexity of many relevant negotiation elements. When a customer concurrently negotiates with several suppliers, the selection of a supplier can become highly complex, especially when terms and variables change. The bidding process needs real-time information processing functionality to be able to negotiate simultaneously with multiple suppliers. E-markets for commodities are aimed at supporting this sourcing process. For non-commodity products, a strong argument can be made that conventional IT is inadequate to acquire non-price information associated with differentiated products and services. The capability of agents to search the network, and employ intelligence to represent their users may be used in sourcing for such differentiated products (Nissen, 2001). Tucker and Jones (2000) suggest that multi-agent systems have the potential to fully automate the buyer-seller negotiation process. By taking away many manual steps (and corresponding waiting times), and automating responses and negotiation, multi-agent systems can help shorten sourcing-cycles. Another advantage is that it becomes possible to evaluate (many) more potential suppliers than in traditional sourcing (Tucker and Jones, 2000).

Agent-based vertical delivery – Vertical delivery concerns the coordination of deliveries of one supplier in accordance with the requirements of its customers. Furthermore, it involves managing the interdependencies of the supplier and its customers. Delivery could include response to a CFP (Call-for-Proposals) of customers, smart product configuration functionalities, available-to-promise, carrier selection, et cetera. Traditional IT can support inter-organisational coordination in supply chains that have stable and long-term relationships. In agile supply chains however, agent-based vertical delivery systems can bring flexibility to adapt to the changing characteristics and requirements of customers. For example, an agent-based customer responsive delivery system bargains for time slots at docks of a distribution centre of a customer and manages its own transportation process according to these requirements.

Agent-based cross-organisational sourcing – Cross-organisational sourcing involves inter-organisational horizontal coordination of demand combined with inter-organisational vertical coordination of deliveries and customer requirements. This type of sourcing mainly exists if the sourcing parties (customers) are confronted with a bottleneck or resource/product scarcity at the delivery side. Coordination of replenishment and timing of sourcing are then necessary. In this case, the customers can coordinate their demand in terms of quantity and timing at the sourcing side. In addition, they could coordinate the resulting order plan(s) with their supplier. This type of cross-organisational coordination allows the supplier to produce efficiently. An example of cross-organisational sourcing is the intended use of agents at Procter & Gamble (Anthes, 2003).

Agent-based cross-organisational delivery – Cross-organisational delivery relates to the coordination of deliveries coming from multiple suppliers, where the deliveries need to be delivered at the same time and place. The customer's order consists of such particular specifications that several parties need (or want) to work together to fulfil this order. When the parties have reached agreement on order specifications, the suppliers need to (horizontally) coordinate the delivery activities. To offer the best services and to achieve high efficiency and effectiveness, the suppliers may need to coordinate their final deliveries with the customer concerning location and time.

Agent-based cross-organisational network – The cross-organisational network relates to the inter-organisational coordination of activities of many different actors that are closely coupled and highly interconnected in a network. These interdependencies include both vertical and horizontal collaboration between the parties. A decision made by a single actor may have consequences for several others in the network. This leads to the need for an extensive planning and coordination that reckons with the interests and characteristics of all actors. In general, the benefits discussed in the previous types are combined in this form of network collaboration. Multi-agent systems can be used in cross-organisational networks in which many actors plan their activities simultaneously, because a multi-agent environment enables decentralised and asynchronous coordination processes.

We initially refined the framework by using it to classify a set of research and commercial B2B agent applications reported on in sources such as whitepapers, the Internet and scientific conferences. Using the framework, we could position the examples found and create an overview of the current state of the field. We refer the reader to Becker *et al.* (2003) and Van Hillegersberg *et al.* (2004) for more details. In the next section, we report on the use of the framework as the basis for a workshop. The objective was the identification of perceived benefits of multi-agent systems for practice.

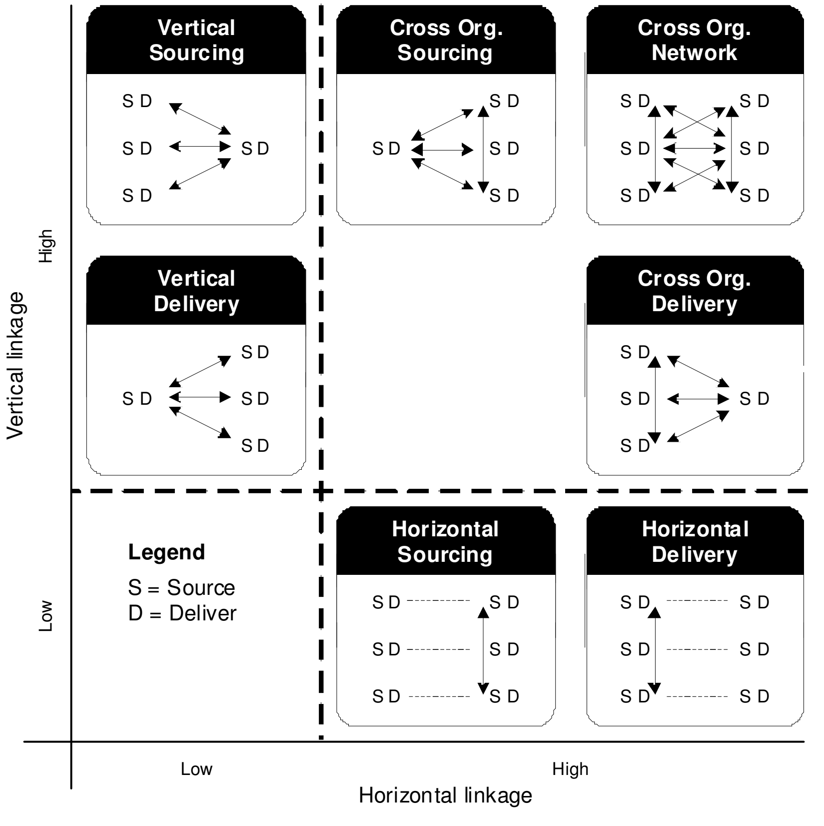


Figure 3.1 – Classification of multi agent-based collaboration structures

3|3 Industry workshop

3|3|1 Setting

To identify promising applications of multi-agent systems, we organised a seminar entitled "*Intelligent agents in supply chains: an exploration*" in August 2003. The seminar brought together academics, consultants, logistic managers, and IT-specialists.

A total of 120 company representatives were invited to attend the seminar. Forty participants of thirty-three different companies/institutions participated. The attendees were a balanced mix of groups invited – see Table 3.1. The seminar was divided into three parts: a plenary session in the morning, covering agent theory and sample applications, four parallel sessions/workshops to identify applications, and a concluding plenary session in the afternoon to exchange findings and conclusions.

3|3|2 Results

Participants were asked to label areas where they perceived largest potential. A total of 22 labels were put on the various areas of the framework. 22% of the labels were put in vertical sourcing and/or delivery relations in the supply chains. The participants expected that multi-agent systems would contribute to further integration or collaboration of actors in their supply chain. 35% of the labels were put in the cross organisational delivery and/or sourcing areas. Several of these participants mentioned perceiving horizontal or vertical sourcing as a first start, and that within five years, multi-agent systems would develop towards agent-based cross-organisational applications. 30% of the labels were put on the cross-organisational network area. Overall, most of the potential of multi-agent systems was expected in more complex cross-organisational areas (65%). We observed a discrepancy in results between researchers, IT and consultants, and participants from LSPs and shippers. IT providers, researchers and consultants were more optimistic about the (pace of) adoption and implementation of complex cross-organisational applications.

Table 3.1 – Workshop attendance

Industry	# of companies	# of participants
Logistic Service Providers (LSPs)	9	11
Shippers	8	8
ICT	5	7
Consulting	4	4
Research institutes/universities	7	10
Total	33	40

Measuring solely the results from LSPs and shippers shows that 33% of participants perceived vertical sourcing or delivery as having the most potential. Applications in horizontal sourcing or delivery and agent-based cross-organisational network are less mentioned (e.g., 13% and 20%). The number of the participants that expect future applications in the cross-organisational sourcing or delivery field account for only 33%, substantially less than the overall 65%. We asked the participants to describe the agent application they had in mind when they put the label in the framework. These results are summarized below.

3.3.3 Specific results for horizontal sourcing

Fashion industry representatives suggested an agent based IOS to enable joint sourcing of transport. Most stores are located in heavily congested city centres. Daily, trucks from different retailers queue in the same shopping streets. Joint sourcing would improve efficiency in transportation. A multi-agent system could gather store data, and take care of the coordination and information exchange of shipments between the suppliers and select the best LSP. Eventually the agent application could take over transport planning and give transportation orders to the LSPs (evolution to cross-organisational sourcing).

3.3.4 Specific results for horizontal delivery

One LSP gave an example of timely and complex coordination processes with its sub-contractors. Daily, this LSP uses more than one hundred sub-contractors, all having different IT systems. EDI, phone and fax are used in the communication and coordination process. Initiatives to use Internet applications had failed. The LSP expected added value for agents in realising connectivity and information sharing with sub-contractors.

3.3.5 Specific results for vertical sourcing

A food manufacturer expected a role for multi-agent systems in contributing to better global sourcing processes. In their view, agents do not need much intelligence; added value is included in realising connectivity and real-time data collection. Future development should focus on adding intelligence to support the sourcing process by giving product alternatives and sourcing advice.

3.3.6 Specific results for vertical delivery

One of the retailers present currently organises its logistics from two distribution centres (DC). They lack information on the exact time of arrival of suppliers. To optimise inbound and outbound processes, information sharing with suppliers, specifically on arrivals, is desired. Multi-agent systems are envisioned to provide connectivity and data-collection.

Participants of the copier industry perceived potential in the delivery of supplies. The actual use of copiers in the field is unknown. If an agent is able to identify when a machine is running out of supplies, the manufacturer can anticipate by sending supplies. This way, clients can reduce their stock of expensive toner, which is perceived to be an added value.

3|3|7 Specific results for cross-organisational sourcing & delivery

Another LSP identified a potential multi-agent system to support transportation. Their planning process is complex when demand fluctuates (e.g., promotions, season), goods are heterogeneous, the volumes vary, and orders arrive late. Agents can fulfil the need of information collecting/sharing. The added value of multi-agent systems leads to a decrease of coordination and transportation costs.

3|3|8 Specific results for cross-organisational networking

Several participating companies expected that within five years, agent applications would be acting in an agent-based cross- organisational network. Unfortunatel,y no examples of possible potential applications were concretely mentioned.

3|3|9 Workshop wrap-up

In general, most participants were positive about the added value and the possible future applications multi-agent systems could have in their respective supply chains. A variety of potential MAS applications were identified. Researchers, Consultants and IT vendors mainly envisioned potential in cross organisational application areas, while shippers and logistics/transport companies perceived the most potential in vertical or horizontal applications. Several benefits of applying agents in these areas were identified, such as systems integration, optimizing planning, decentralised decision-making and supply chain visibility.

3|4 Discussion

3|4|1 Summary and conclusions

In this chapter we reported on our first attempts to identify potential opportunities for multi-agent systems to support current and future supply chains/networks. In order to study and discuss this in a structural manner, we developed a framework based on the SCOR-model and extended it to include horizontal, vertical and cross-organisational processes. The framework was used as the basis to identify promising areas in a workshop with industry.

We present two sets of conclusions, both on the identified promising application areas and on future research such as other applications of our framework. Industry participants applied the framework to identify and describe potential applications of multi-agent systems in their business. The participants working in industry had a preference for the horizontal and vertical coordination areas. Researchers, IT vendors and consultants perceived th most potential to be in cross-organisational applications.

The framework developed in this research could serve as a tool to discover and position potential agent applications and assess their potential benefits. However, the framework currently focuses on functional aspects of agent applications solely. It could be extended to allow for the identification of critical success factors and to help in anticipating the value of the IOS. But, questions such as: "*What is the connection with supply chain aspects such as power, trust and competition?*" also need more thought. Will those hinder adoption (as they did with previous attempts to build inter-organisational systems), or can agents make a difference – for example through their distributed structure? Several workshop participants noted that these aspects are important. Other barriers mentioned were personnel acceptance of new ways of working and the use of new technology, and the willingness of supply chain partners to collaborate and adopt a common system. Moreover, adoption depends on the current status of internal ICT systems. In addition, the need for clear business cases was expressed, including mechanisms to share costs and benefits of the agent system. These aspects were especially mentioned for the more advanced scenarios, like cross-organisational networking. Furthermore, in future research more detail could be added to the framework by including a more detailed description of B2B sourcing and delivery processes – the more detailed SCOR descriptions could be a basis. In the remainder of this dissertation, we look in more detail at several of these issues.

3|4|2 Synthesis

The research and workshop this chapter reports on took place in 2003. At the time, we were exploring the concepts of multi-agent systems and the potential for inter-organisational application. The framework we developed was helpful in signalling and analysing several potential scenarios in a workshop with professionals.

The framework proved to be a useful way to present a new technology and concepts in practical terms that professionals recognize. We neither mentioned specific technologies, nor discussed different theoretical streams within agent research. We spoke about multi-agent systems merely as groups of (smart) agents with specific roles and behaviours.

Actual clustering of applications turned out to be complex, especially in exploring the boundaries between sourcing and delivery (depends on the party that initiates the activity), and the horizontal/vertical/cross division. The framework lacks the explicit time aspect; in the workshop the participants mentioned very diverse application areas ranging from strategic, to transactional activities.

The workshop gave our research a start. We found that multi-agent systems have the potential to automate diverse supply chain processes, and grasped an interest from practioners for further industrial involvement throughout the remainder of our research.

Chapter 4 The science of prototype evaluation

4|1 Introduction

Already for many years researchers are working on multi-agent systems; see Chapter 2. Despite this, the rate of adoption in industry is still limited (Van Rijswijck and Davydenko, 2007). Davidsson *et al.* (2005) show that most agent research is conceptual; not much is truly developed, properly evaluated, let alone implemented. Luck *et al.* (2003) make a strong call to action to create "*working prototypes of commercial agent systems [to] be developed for specific industry sectors and made available for commercial use.*" Specifically, they point out the need to build prototypes that span organisational boundaries, and document cases of early adoption and prototype evaluation in order to analyse the reasons for success or failure.

Industry shows a large interest in multi-agent application and experimentation, as we found out in interactions with practioners. McBurney and Luck (2007) claim that "*we are at the point [now] where we can build open and dynamic systems, which underpin nearly all views of future computing, but we haven't yet done so to any great extent.*"

Such call to actions trigger us to start experimenting with these concepts and technologies; what we do in the chapters that follow (specifically Chapter 1 and Chapter 6). However, like in any design research, validation and evaluation are important steps. Nwana and Ndumu (1999) make a specific call to the MAS research field to develop methods, tools and technologies for evaluation. Many different opinions exist among scholars about evaluations: it is not crystal clear how to properly evaluate an (inter-organisational) agent prototype with all its complexities. Therefore we formulate an additional research question "*How to evaluate a research prototype, being a novel multi-agent system?*". This question has a more general as well as a specific component.

In this chapter we discuss the evaluation of multi-agent system prototypes. First, we discuss in section 4|2 different methodological perspectives. In section 4|3 we discuss the place of prototype evaluation within different research domains. This is followed by the why and how questions, and the different forms of prototype evaluation of agent prototypes as we traced them within literature. Section 4|4 then points at the need for a multi-method evaluation approach, and presents a framework for prototype evaluation. The chapter ends with a concluding section (4|5).

4|2 Methodological perspectives and differences

4|2|1 The differences between qualitative and quantitative research

Both qualitative and quantitative researchers in the social sciences "*think they know something about society, and they use a variety of forms, media and means to communicate their ideas and findings*"

(Denzin and Lincoln, 2005). Of course, they can both be right, but there are essential differences in the research methodologies they choose and the approaches they follow to find the(ir) truth. Becker (1996) identifies several differences, of which we will discuss three. These differences mainly deal with the politics of research, and the question of who has the power to legislate correct solutions to social problems.

The first difference is the one between "*positivism and postpostivism"*. In the positivist tradition, there is a reality out there to be studied, captured and understood, whereas the postpostivists argue that reality never can be fully comprehended, it can only be approximated. A second difference described is referred to as "*capturing the individual's point of view*". Qualitative researchers utilise techniques such as interviewing and observation, whereas quantitative researchers rely on more remote, inferential empirical methods and materials. They often perceive qualitative interpretive methods as unreliable, impressionistic and not objective. A third difference concerns the "*examination of the constraints of everyday life*". Qualitative researchers see the world in action and embed their findings in it, whereas quantitative researchers abstract from this world and seldom study it directly. They study probabilities derived from the study of large numbers of selected cases.

4|2|2 Dealing with different research domains

This dissertation deals with a topic that spans several disciplines; more specifically, this dissertation spans Information Systems, Computer Science, Artificial Intelligence, Operations Research, and Operations Management. All these disciplines have their own methodological approaches and preferences, and they also have their own perspectives on design research and evaluation. For an overview of methodologies, but also a discussion between positivist and interpretivist points-of-view in the field of management information systems research, we refer the reader to Currie and Galliers (1999) and Vessey *et al.* (2002). This difference between and within the disciplines is something we should be aware of. Clearly, there is not one single best way to design and evaluate.

4|2|3 The position of design research in the information systems discipline

Benbasat and Zmud (1999) discuss the need for relevance in (information systems) research and plead that "*theories, concepts, and findings from IS research could be used by practitioners to legitimate, rationalize, and justify courses of action taken*." They continue by pointing out that "*it is important that authors (of research papers) develop frames of reference which are intuitively meaningful to practitioners*." At the same time, we should realise that, as Kaplan and Duchon (1988) formulate it,"*(American) information systems research generally is characterized by a methodology of formulating hypotheses that are tested through controlled experiment or statistical analysis. The assumption underlying this methodological approach is that research designs should be based on the positivist model of controlling (or at least measuring) variables and testing pre-specified hypotheses*." They specifically reference (Meehl, 1978) who "*argues that science does not, and* *cannot, proceed by incremental gains achieved through statistical significance testing of hypotheses.*" Arnott *et al.* (2005) show that only less than 10% of the research on decision support systems (as a sub-domain of the IS field) is of high relevance to practice.

In section 1|3 we briefly discussed design research as a research methodological choice that might bring that relevance and practical impact – see more specifically Hevner *et al.* (2004) and Baskerville (1999). One of the other articles we reference in Section 1|3 is March and Smith (1995), which discusses the difference between design and natural science approaches in IS. They state that "*rather than producing general theoretical knowledge, design scientists produce and apply knowledge of tasks or situations in order to create effective artefacts. If science is [an] activity that produces "credentialed knowledge", then, following Simon (1981), design science is an important part of it.*" March and Smith explain that "*design science consists of two basic activities, build and evaluate. Building is the process of constructing an artefact for a specific purpose; evaluation is the process of determining how well the artefact performs. Significant difficulties in design science result from the fact that artefact performance is related to the environment in which it operates. Incomplete understanding of that environment can result in inappropriately designed artefacts or artefacts that result in undesirable side-effects*." They make the observation that "*in much of the computer science literature it is realised that constructs, models, and methods that work "on paper" will not necessarily work in real world contexts. Consequently, instantiations provide the real proof.*" Straub (1989) makes a conceptual split between exploratory and explanatory research; the article concentrates merely on the latter. Whereas exploratory research utilises qualitative and non-empirical techniques for theory-building, explanatory research concentrates on theory-testing through quantitative empirical techniques. In fact it is a circle: exploratory research leads to new insights, to be tested by explanatory research, which lead to refinements to be used in exploratory research. Prototyping and product development are instruments to test new theoretical concepts – see also Nunamaker and Chen (1990).

Fitzgerald (1998) explains that there basically are two categories of information systems projects – see also Figure 4.1. First, there are efficiency projects, which aim at improving existing processes in a more efficient way, for example through automation. Second, there are so-called effectiveness projects, which concentrate on utilising IS to enable new ways of working. It becomes clear that the different categories need different ways of evaluation. Evaluating efficiency projects is easiest, since their benefits are generally easy to describe and measure. An effectiveness project often results in multiple benefits. Often these are external - or secondary benefits that are less easy to predict and measure. However, one can wonder whether most projects are actually so clear cut that one can easily label a project as either an efficiency or an effectiveness project.

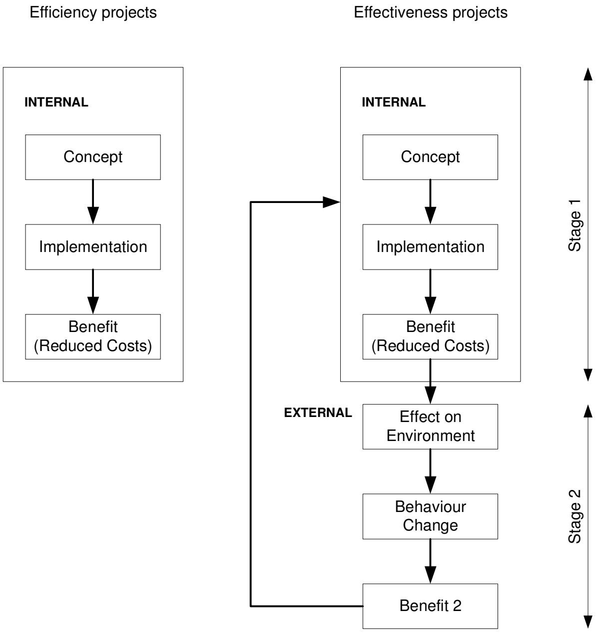


Figure 4.1 – Different types of IS projects (Fitzgerald, 1998)

4.3 Prototype evaluation in research

4.3.1 Different phases in information systems design

Development of (new) information systems goes through different phases. Kushniruk (2002) gives a nice overview of the different steps in the design process, with corresponding evaluation methods. The different steps are: planning, analysis, design, implementation and support. This is visualised in Figure 4.2. Evaluation in iterative design and testing is nowadays an integral part of the newer design methodologies, as Kushniruk explains. He literally states that: "*iteration in this context involves repeating or looping through the same development activities, successively refining the system in each cycle. Typically this involves initial development followed by evaluation and feedback into system design, leading to further cycles of evaluation and redesign until a satisfactory system arises. In contrast to traditional software development cycles, analysis is not finalised at the beginning of system development, but recurs throughout the process*".

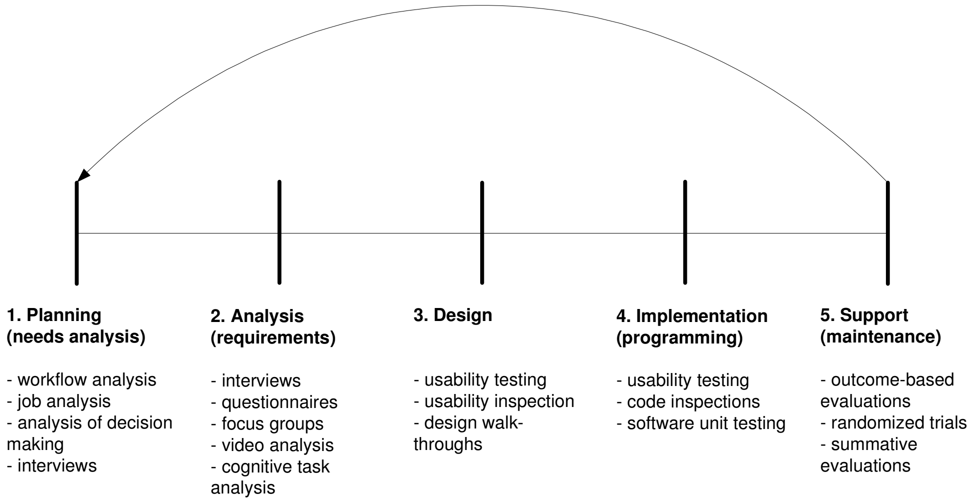


Figure 4.2 – Different phases in IS development (Kushniruk, 2002)

4|3|2 Why we evaluate prototypes

An IS prototype is "*an early version of a system that exhibits the essential features of the later operational system. Some information systems prototypes may evolve into the actual production system whereas others are used only for experimentation and may eventually be replaced by the production system*" (Alavi, 1984). Alavi lists several benefits of using prototypes, essentially as instruments in a development process. Of these benefits a prototype is something real, a prototype provides a common base line, and users of prototypes can be enthusiastic. He also lists two important drawbacks of prototypes: they are difficult to manage and control, and it is difficult to prototype large information systems.

Prototyping can take place for different reasons, as Fitzgerald (1998) explains: for a technical design try-out; or a try-out of novel concepts. Documenting the development of an interactive web application, Levi and Conrad (1996) show that prototyping can be a useful methodology. As such, a prototype helps to get acquainted with new technologies and approaches, to learn from these experiences, and is an ideal tool to gather (end-)user feedback. *"Evaluation means assessing the performance or value of a system, process (technique, procedure,…), product, or policy. As such, evaluation is accepted as a critical necessity in science, technology, and many other areas, including social applications."* (Saracevic, 1995).

4|3|3 How we evaluate prototypes

How one evaluates prototypes depends largely on his/her philosophy of science (Klein and Herskovitz, 2007). "*Rather than focus on searching for inconsistencies between the prototype and* *the software consumer's mental model – which would invalidate the prototype – in the manner of Popperian falsifiers, the systems developer and software consumer adopt the stance of Quinean revisers to save the prototype from rejection by removing inconsistencies via adjustments. Such a stance has pragmatic implications for both research and practice*." Klein and Herskovitz place prototype validation within the wider contexts of theory formulation, knowledge acquisition and evidence evaluation. "*Prototype validation is not viewed as an arbitrary activity in IS that happens to work, but is deemed as an instance of a well-reasoned and well developed philosophy of science."*

Evaluation of a prototype could/should take place at six different levels, as Saracevic (1995) illustrates: (1) Engineering level; (2) Input level; (3) Processing level; (4) Output level; (5) Use or user level; and (6) Social level. Saracevic provides evidence that most prototype evaluations solely focus on one (or two) of these levels. He literally states: "*The point is, there is much more to evaluation [of an information retrieval system] than evaluation of a variety of algorithms and procedures*." A similar multi-method evaluation approach is suggested by Vokurka *et al.* (1996), who discuss a long list of qualitative and quantitative methods to validate and evaluate prototype expert systems. Interviewing and data analysis are helpful instruments for evaluations of novel systems and exploratory research in general, according to Benbasat *et al.* (1987). They provide the example of an evaluation of prototypes in the expert systems domain.

A spectrum of different evaluation methods for prototypes is given by Kushniruk (2002). As Figure 4.3 shows, lab-experiments and controlled simulations can be found at one end of the spectrum, whereas field-tests with observations are at the other end. Kushniruk specifically states that "*there have been a number of arguments made that a high degree of variable control may be neither feasible nor desirable when testing systems in real-world contexts, in particular when attempting to achieve a greater degree of generalizability to complex real-world situations*." An increasing number of researchers promote the use of naturalistic study in evaluating and validating decision-making in complex domains, according to Kushniruk. "*Proponents of naturalistic approaches to evaluation have argued that much of the research from "classical" controlled experimental studies [on decision-making] has not led to results that are generalizable to real-world situations in complex domains.*" Kaplan and Duchon (1988) for example, describe information systems as social systems. Study of these "*systems involves so many uncontrolled – and unidentified – variables, methods for studying closed systems do not apply as well in natural settings as in controlled ones*." They advise against the utilisation of evaluation methods that focus too much on control factors. Their paper describes an evaluation case study in which a combination of methods was utilised by a group of quantitative and qualitative researchers. Qualitative methods included open-ended interviewing, observation, participant observation, and analysis of responses to open-ended items on a survey questionnaire. The book by Van Dijkum *et al.* (1999) discusses simulation as a research methodology extensively; they warn, of the subjective assumptions simulations are often fed with, among other warnings, and they propose to include expert-evaluated meta-models in simulations.

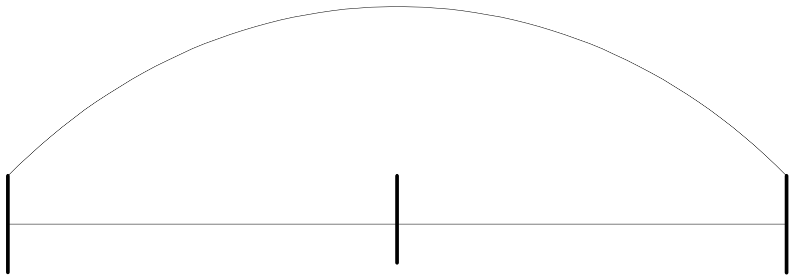


Figure 4.3 – Spectrum of prototype validation methods (Kushniruk, 2002)

4|3|4 Examples of evaluation of agent prototypes

In the literature we found relatively little real validation and evaluation of multi-agent systems prototypes, see section 2|6|3. In principle, two forms of prototype evaluation can be recognized: (1) the agent system as a tool to research other environments, and (2) the agent system designed as an information system aimed at solving a certain problem. An example of the first category is the multi-agent system discussed in the recent dissertation by Boer-Sorban (2008) who utilises agent concepts to research behaviour in financial markets. Our focus is on the second form.

Several scholars report on MAS prototypes that they evaluated with simulations in a controlled laboratory environment. Their prime focus was the validation of the algorithms and outcomes, and to compare these against more traditional approaches. Sousa *et al.* (2004) describe a MAS prototype for machine scheduling. They perform solely quantitative analysis and simulations to compare their algorithm against known approaches and discuss the advantages and disadvantages of their approach. Similar work is done by Mönch *et al.* (2006), who worked on an agent prototype for scheduling wafer fabs in high-tech electronics production. They compare their simulation outcomes against more traditional dispatching rules, such as the first-in-first-out (FIFO) rule. Mes *et al.* (2007) performed an extensive comparison of agent-based scheduling approaches versus look-ahead heuristics in a real-time logistics setting. Their outcomes illustrate that agents are competitive with existing algorithms, furthermore, they reference several sources reporting similar findings.

A different prototype validation approach is documented by Cheeseman *et al.* (2005) who report on a proof-of-concept system they developed for adaptive manufacturing scheduling. The proof-of-concept is ready, but the paper only reports on some preliminary (qualitative) testing of the system to assure its correct technical functioning. In the section on future work in the paper, they introduce their plans for a testing trajectory, which would be a combination of quantitative tests – to see how well the algorithms compare against traditional approaches – and qualitative field tests, to try out the concepts and the prototype in real practice. They expect the latter trajectory to contribute especially to improvements in the prototype, which can build upon the knowledge gathered through the tests and evaluations. We were not able to trace any follow-up work to this paper.

Similarly, the paper by Lea *et al.* (2005), which we already discussed in section 2|5|4, encompasses a long list of advantages multi-agent architectures would have over traditional ERPs. The list of (envisioned) benefits, included in Table B.1, goes further than solely factors that are easily quantifiable. It shows that future research should not solely concentrate on simulations of the mechanism, rather also concentrate on many more factors.

4|3|5 Insights from interpretative work

Other streams in evaluation include more interpretative methods such as expert evaluations or Delphi studies, qualitative analysis, and gaming as a research instrument. Akkermans *et al.* (2003) employed a Delphi study to investigate the impact ERP systems have on supply chain management practices across Europe. They utilised 23 experts from across industries, to identify important issues. They critically discuss the feedback gathered, and link this to insights from existing literature.

Since the appearance of the Unified Theory of Acceptance and Use of Technology (Venkatesh *et al.*, 2003), UTAUT is often used in multi-method evaluations. Garfield (2005), for example, evaluates the use and adoption of Tablet PCs in and across different professional environments. Doing so, she identifies a list of factors which explain use and adoption. She utilised targeted interviewing and observation. Welmers (2005) utilised the model to explore the field of digital radio broadcasting in The Netherlands.

Gaming can also be a research instrument to evaluate behaviour and understand interactions, as Hoogewegen *et al.* (2006) illustrate. Their paper describes the theoretical background and the design of a game they developed. Furthermore, it explains how the game was played and how students and professionals evaluated it.

4|4 Evaluating agent-based inter-organisational systems

4|4|1 The need for a multi-method approach

Prototyping and evaluating new systems in a domain such as MAS raises many questions. It is a complex domain. MAS come with many open issues regarding the concepts as well as the technology toolsets. Also, the inter-organisational nature of many applications adds to the complexity. An IOS brings a multi-actor perspective, but how should we evaluate such a system properly? As Table B.1 demonstrated, agents hold great potential for future enterprise applications. However, these systems differ on almost all the characteristics listed. How should this be evaluated?

In our extensive literature review, we found a long list of critical success factors for the implementation of intra- and inter-organisational systems – see Table 2.5. Similarly, we identified a list of factors that hinder adoption of multi-agent systems. See Table 2.6.

Overseeing the diversity of the domain and the many different factors that play a role in implementation of systems, as well as adoption of agent systems, we recognise that one can never evaluate a system on solely one factor, nor is it possible to utilise a single method. Van Hillegersberg (1997) studied Object Orientation (OO) in information systems development. He found that, in order to properly study all of the diverse aspects that came with OO, a multi-method approach was needed. Considering this, along with the previously discussed work on the methodology of prototype evaluation – including the ideas by Saracevic (1995), Vokurka *et al.* (1996), and Kaplan and Duchon (1988) – we recognize the need for a multi-method prototype evaluation approach.

4|4|2 A framework for evaluating agent-based IOS prototypes

Table 4.1 gives an overview of different evaluation methods along with a short description. Furthermore, we include a classification that matches the level of control, after (Kushniruk, 2002), and the level of evaluation, after the division made by (Saracevic, 1995) as we discussed in section 4|3|3.

4|5 Conclusions

In this chapter, we discuss the different (research methodological) philosophical perspectives that exist across different research domains and how these influence prototype evaluation. We synthesised a number of articles on prototype evaluation, with some covering the evaluation of multi-agent systems (prototypes). Many of these evaluations go in-depth utilising one evaluation mechanism, and thereby neglecting other evaluation methods. Following several of the conclusions from our literature review (in Chapter 2), we signal that one should never evaluate a system on solely one factor, nor utilise a single method. The application of agent systems in practise comes with many different aspects, and demands a proper evaluation of many different factors.

Our main conclusion is that the evaluation of novel software prototypes in a complex (social) environment demands a multi-method validation and evaluation approach. The table, given in Table 4.1, can be an instrument to utilise when evaluating prototypes.

Table 4.1 – Evaluation methods for MAS prototype evaluation

Method	Level of evaluation	Level of control	Rigor / relevance	Short description
Simulation	Processing level	High	Rigor high Relevance low	Evaluate the performance of the prototype in a controlled manner – generally, in comparison with other mechanisms.
Technical evaluation	Engineering level	High	Rigor low Relevance high	Evaluate the technical (correct) working of the system. Generally used by the developer when constructing the system, but also useful when the prototype is finished.
Prototype as design & feedback instrument	Engineering, Processing, Use or user, & Social level	Low	Rigor low Relevance high	A continuous evaluation throughout the design & build phase. Ex-ante evaluation leads to new insights which will be fed into a new version of the prototype.
Field experiment	Input, Processing, Output, Use or user, & Social level	Medium	Rigor low Relevance high	A try-out of the prototype in the field. Does the targeted system achieve what was envisioned? Generally leads to suggestions for improvements.
Interviewing	Use or user, & Social level	Low	Rigor low Relevance high	Coupled to another evaluation method – for example, the field experiment. A method to investigate perceptions, and user experiences/intentions.
Observation	Use or user, & Social level	Low	Rigor low Relevance medium	Coupled to another evaluation method – for example the field experiment. A method to investigate the user experience and user behaviour.
Survey	Use or user, & Social level	Medium	Rigor medium Relevance medium	Coupled to another evaluation method – for example, the field experiment. A method to gather general and broad insights about the system.
Expert opinions / Delphi test	Engineering, Input, Processing, Output, Use or user, & Social level	Low	Rigor medium Relevance high	An evaluation method in which a group of experts is used to discuss and validate a prototype. This is an accepted method to gather meaning, experiences and insights from (domain) experts.
Gaming	Use or user level	Medium	Rigor medium Relevance medium	Use of a gaming environment to experiment with a prototype. Useful when trying out new concepts or prototypes in complex environments.

Chapter 5 Real-time truck planning

5|1 Introduction

The practical application we describe in this chapter concerns an agent-based real-time order assignment system that assigns trucks to containers. Although initiated from the perspective of a single firm (the LSP), there is a substantial inter-organisational element. Container transport is performed for a customer, and involves picking up containers from terminals. Hence, it is inter-organisational by definition. Coordination is required to complete the daily operations.

Post-Kogeko, the LSP, was one of the partners within the DEAL research project. Early in 2005, we started an explorative research project to investigate the possibilities for agent application at Post-Kogeko's container unit. An initial design was made in the MSc thesis by Oink (2005). This initial design was a starting point for research and prototype development throughout the following years. In this chapter, we present the design, the different prototypes, and discuss diverse evaluation methods to validate the prototypes. Question five lead to this point by asking "*Can multi-agent systems contribute to better performing, and easier implementable systems for transportation?*"

The research in this chapter is design research. The methodology utilised is a mixture of the action research and design research paradigms – see among others Nunamaker and Chen (1990), Gummesson (2000) and Hevner *et al.* (2004). We follow Hevner *et al.*'s (2004) recommendations to utilise a real case as an essential part of the evaluation in the development of a design, because "*by creating new and innovative artefacts it tries to extend the boundaries of current information systems knowledge*". Davidsson *et al.* (2005) concluded that much of the published research on multi-agent systems stops with a description of the possible concept or design, and never evaluates/tests it. We perceive the evaluation of a real case to be an essential part of evaluation, because it includes implementation aspects and goes beyond plain system designs. (Van Aken, 2004). Following the work done in Chapter 1, the design and prototype are evaluated using multiple methods.

The chapter starts with a case description in section 5|2. This is followed by a section that describes the design (5|3). Evaluation comprises the largest part of the chapter. The prototyping functioned as an evaluation (section 5|4) of methods and concepts. Furthermore, we performed simulations (see section 5|5), a validation with experts (5|6), and an in-company field-test (5|7). The chapter concludes with a discussion section in section 5|8.

5|2 Case description

5|2|1 Company description

Post-Kogeko, with headquarters in Maasland, The Netherlands, was founded in 1979 under the name Post by Dirk Post. It was founded as a trucking company active in the transportation of sand and concrete. Later Post expanded into the newly emerging container transport market, where it started working for the American carrier Sealand (nowadays part of Maersk). In 1990, Post took over Kogeko, which was a transport firm active in the transport of fruits and vegetables. Investments were made in cool-containers and a new headquarters, and the name changed to Post-Kogeko. Another takeover took place in 1995, when Van Die Transport was added to the group. The Van Die brand still exists, its fleet is among others active for the Albert Heijn distribution centre in Pijnacker. In 1998, Post-Kogeko, together with Visbeen and Norfolk Line, initiated the joint-venture DailyFresh Logistics, specialising in the transport of fruits and vegetables to the UK. In 2008, Post-Kogeko acquired Zutrans from Campina, which was active in dairy transport.

Post-Kogeko is a complete LSP active in distribution, forwarding and transport. Post-Kogeko's focal areas are: (1) transport within Europe; (2) groupage to the UK and Ireland; (3) distribution; (4) conditioned storage and transhipment; and (5) sea-container transport. Furthermore, they operate a truckwash and offer financial and administrative services to their clients. Post-Kogeko currently employs ± 575 employees and operates ± 500 trucks.

The company is growing steadily, and is referenced by the Commissie Van Laarhoven (Van Laarhoven *et al.*, 2006) as one of the innovators in its industry. In April 2009, Post-Kogeko ranked 35th on the list of largest LSPs in The Netherlands (Logistiek, 2009).

5|2|2 Container transport

Container transport is one of Post-Kogeko's main activities. It is the process of picking up a container at a container terminal and transporting it to a customer-specified location. Arriving at the customer location, the container is unloaded (while remaining on the truck's trailer, with the driver present), and the empty container is brought back to the same or another terminal or empty depot. This concludes the (import) order, and the truck is then ready for its next order – see Figure 5.1 for an illustration of this process. The process is reverse for export containers. Figure 5.2 shows one of the trucks in action. Please note that one of the trajectories is always empty; there is no matching between empty import containers and customers that require a (empty) container for export.

Around 40 trucks (and drivers) are dedicated to the container unit. In the case of needed additional capacity in a specific unit, Post-Kogeko sometimes temporarily deploys drivers and trucks from other units. Furthermore, external carriers can be hired.

Post-Kogeko makes use of flexible trailers, which are capable of transporting 20ft, 40ft and 45ft containers. The containers can be either standard containers, high-cubes (which are higher than standard), or so called reefers (which is a container equipped with freezing equipment). Post-Kogeko is specialised in conditioned transport (which only makes up 9% of the container market – see Verweij (2009)); as such 95% of the containers it transports are reefers. Reefers limit the dynamic reallocation of empty containers. Reefers have to be cleaned before re-entering the circuit of empty containers.

An analysis of order data from the first three weeks of January 2005 revealed that an average of 81 container orders are handled each day (Oink, 2005). The 80-20 rule seems to hold for Post-Kogeko. The 9 largest customers (out of a total of 27 different customers in that period) turned out to be responsible for 84% of all orders – the top 4 alone were responsible for 57%. 4 out of 38 terminals are the most important, and account for 77% of the orders. These terminals are ECT Delta, RST Noord Deka, ECT Home, and APM Terminal. Almost all containers have a destination in the Netherlands; the biggest part remains in the larger Rotterdam area.

Transporting containers means interacting with terminals and customers. The interaction with the customers can be split in four different moments: (1) First, the order intake phase takes place, when Post-Kogeko receives its customers orders. Generally these orders specify the pickup of a container the next day. The orders received on Thursday specify a transport on Friday, the orders received on Friday specify a transport for the next Monday, and so on. Orders are generally received by fax or e-mail; last-minute orders sometimes come in by phone. (2) During the execution day, contact about the order-status occurs. Topics covered include if the order has been assigned and picked up already, where the truck is and when can it be expected for delivery, and so on. This type of contact generally takes place by phone, and is generally customer-initiated. (3) Then the actual delivery takes place, where the driver delivers a full or empty container, and waits for unloading/loading of the container. (4) The invoicing occurs afterwards. An invoice is generally an accumulation of several orders in one invoice, sent after a specified period.

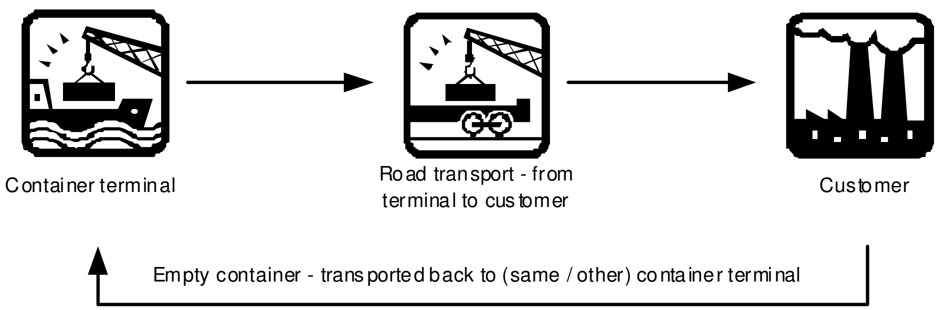


Figure 5.1 – Standard physical flow of containers in the Post-Kogeko container unit

Figure 5.2 – Post-Kogeko truck carrying a CoolBoxx container

Terminals are the locations where full or empty containers have to be picked up. Before picking up a (full) container Post-Kogeko needs to know that the container is available. It may happen that containers have not yet left the ship, or (as is more often the case,) are not yet administratively available. The latter is due to administrative procedures, either between the carrier and the customer– which generally means non-settled payments, or because it has not yet released been by customs authorities. Container release by ECT, the largest terminal operator (which includes the ECT Delta, ECT Home, and Hanno terminals), can be monitored by means of a web interface, and the PortInfolink RoadPlanning application. The APM and Uniport terminals have their own Internet status systems, next to RoadPlanning. Most of the smaller terminals still have to be contacted by phone. Container status information (such as present, departed, expected and not available) is crucial in the planning process, as we found that between 40-60% of the containers are generally not available at the start of operations on any given day. Post-Kogeko has to announce the pickup of containers electronically through the PortInfolink RoadPlanning system several hours in advance, through a so-called pre-notification message. This way, terminals can prepare for pickup, and arrange their administration to ease the pickup process. When containers are still expected to arrive by seaship or barge on a certain day, Post-Kogeko planners can check the ship's status and location through an online system (provided by the company Dirk Zwager). This is generally not needed, since most customers assure beforehand that the containers they instruct Post-Kogeko to pick-up are physically present at the terminal. In addition to the sources of information discussed above, planners utilise traffic information (for example from the ANWB website `www.anwb.nl/verkeer`) in their decision-making.

5|2|3 Planning container trucks

The planning of container trucks involves several distinct phases. First, there is the order intake process, as mentioned before. This is generally one (working-)day before required execution. It is the request from a customer to Post-Kogeko to pick-up a container at a container terminal (in case of an import container) and transport it to the customer, with delivery within a certain time window. Order intake is just an intake process; no price or timeframe negotiations are performed. Customers are generally return customers who conduct a large quantity of orders at known tariffs.

The second step is the actual planning of the execution phase, which involves the assignment process from orders to trucks. This is split in two phases. First, one day in advance, the assignment of the first trips is performed. This phase includes a decision on the amount of trucks to use. In case more trucks are required than the container unit has available, it requests additional trucks from other units within Post-Kogeko or hires external carriers. The second phase starts at execution day when the trucks finish their first orders of the day. The planner monitors truck locations, statuses and container availabilities, and assigns empty trucks to orders. See the activity diagram in Figure C.1 in Appendix B for a schematic illustration of this part of the process. The reason that this process is split in two is that trucks generally start operating very early in the morning; 4 o'clock is not an uncommon starting time. The planners start working later. The planner that prepares execution for a certain day is responsible the day after for execution control and assignment of the rest of the orders.

The third step is no longer a planning process, rather the actual process of transporting containers physically from one location to the other, and back. Figure C.2 in Appendix B illustrates this for an import container. For an export container, the process would be reverse. Generally, the PickupAddress is the same as the ReturnAddress; although this is different for around 25% of the orders. The planner keeps in contact with the driver, updates the statuses, and anticipates future orders. Sometimes, a fourth step is needed: the cleaning of the container before it is returned to the terminal or an empty depot. This depends upon the usage of the container, the customer and the carrier. The last step is financial settling: this is an invoice that triggers a payment, generally for a set of orders performed within a certain period.

The container planning team at Post-Kogeko consists of two planners, and two persons responsible for data-entry and administrative support. When planners generate assignments, it is most common to utilise one truck for one complete order. Sometimes, however, it happens that a container (on a trailer) is dropped at a DeliveryAddress by Truck A, and later picked up again (after unloading or loading has taken place) by Truck B. Very rarely it also happens that containers are picked up a couple of days in advance from a terminal, or brought back a couple of days later, sometimes residing at Post-Kogeko in Maasland. If so, it sometimes happens that the different trajectories are executed by different trucks, but with the same trailer chassis. The different entities including their attributes and methods, and the relationships between them, are shown in the class diagram depicted in Figure C.3 in Appendix B.

5.2.4 Information systems

The planners are supported by two information systems. The first is QFreight, a Transport Management System (TMS). The system contains all information about the orders for a specific day, and is utilised throughout the entire organisation. New orders are (manually) entered into the system, order assignment is (manually) done, and after completion, this system forms the basis for sending out invoices to the customer. It furthermore includes a module to generate management reports. During the different transport phases, the planner manually updates the statuses in the system. As such, QFreight provides an up-to-date view on all order-related information. QFreight is provided by Continental Software Services and utilises an MS SQL Server as its backbone, and also contains historical data.

The second system is CarrierWeb – provided by e-Freighttrac Ltd. CarrierWeb is a track&trace and two-way communication system, which basically is a combination of the onboard computers in the trucks and a web-based interface for the planner (see Figure 5.3 for an illustration). Part of the onboard computers is a GSM/GPRS module for communication with the CarrierWeb backbone. The system can be utilised to send messages from planner to driver, and vice versa. An important part of the onboard computer is its ability to connect to different sensor-systems; it can connect not only to the GPS receiver, but also to temperature sensors, and CanBus motor management information. This information is sent to the back-end system. Planners can continuously trace truck locations, and receive messages when the temperatures of cool compartments move outside set boundaries.

5.2.5 Points for improvement

Analysing the processes described in the previous sections, we recognise several “pain points”. These include pain points due to the type of (external) environment one is dealing with, and pain points related to the control and execution of tasks. Furthermore, we identify issues with respect to the role of automation.

The external environment directs the internal processes: the continuous last-minute character of orders combined with few ICT-based external linkages shapes an environment with a serious amount of manual control and relatively little space for optimisation. Customers hardly ever reveal orders on a longer than one-day-in-advance timescale. Lacking electronic interfaces between upstream and downstream partners makes information exchange relatively slow and expensive.

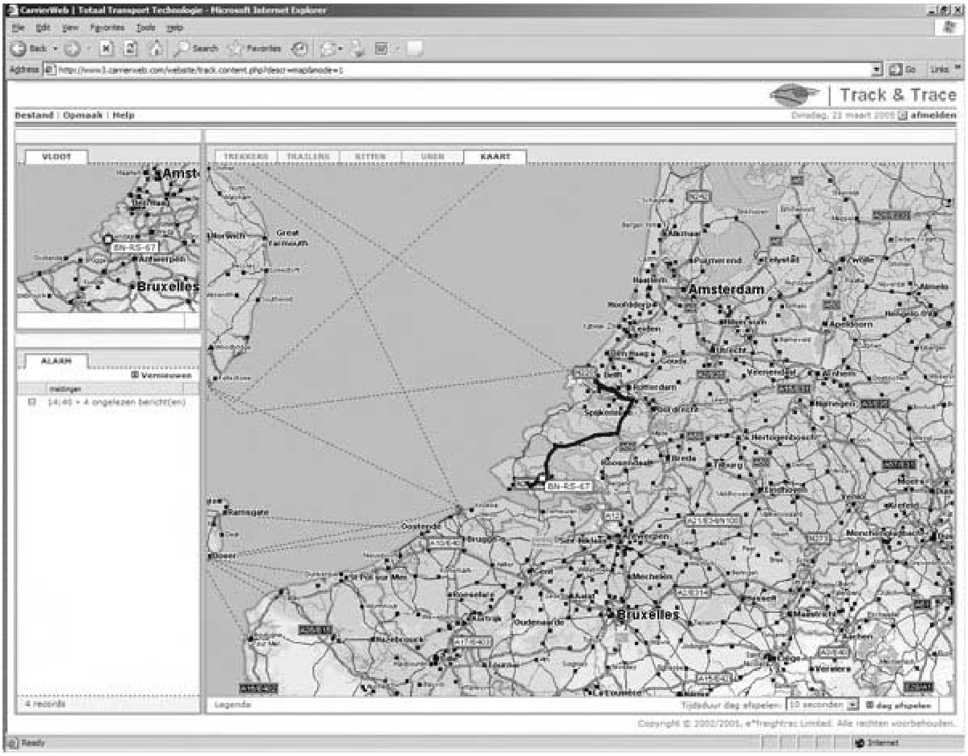


Figure 5.3 – Screenshot CarrierWeb system

What complicates matters from an optimisation perspective is that not all containers are available at the start of operations early in the morning; either they have not physically left the ship yet, or are delayed for administrative reasons – which can be due to non-settled payments or customs. Post-Kogeko can only transport released containers, which are allowed to leave the container terminal. This complicates optimisation, since not all information is known beforehand, and will only become available sometime during the day. Furthermore there is a large variety in amount of work per day: in number of orders as well as distance to travel per day. Doing the entire planning *a priori* is not possible.

Despite the fact that container transport is an activity that heavily depends upon external information, and coordination, there is little electronic process integration between parties. The planning heavily relies upon manual operations. This is surprising since many of the customers are returning customers (see the observation on the 80-20 rule). Likewise, the containers are picked up from a limited number of locations. These terminals have automated systems in place to perform availability and status checks of containers (web / PortInfolink services). Currently, this is also largely a planner-triggered manual activity.

Although supported by several information systems, the assignment process is not automated either, and permanently requires two planners. In the assignment process, the planner combines all information available and assigns a certain truck (and driver) to a specific order.

Potential for savings can be found at the following points:

- Automation of the planning processes (assignment process as well as external information processing tasks) could result in a larger span of control for a planner, making it possible for the planner to plan more trucks.
- Better assignments, resulting in fewer mistakes, less unnecessary waiting, and less empty kilometres. See the analysis by Oink (2005).
- Chain coordination – involving customers and terminals – enables a higher level of optimisation. It no longer solely reacts upon (last-minute) orders, but has the potential to anticipate orders. This might result in shorter throughput times, less unnecessary waiting, lower costs (since one can further optimise resources and routes) and is likely to lead to more satisfied customers.

5.2.6 Multi-agent system potential

Many multi-agent system projects referenced in literature have been focused on building alternative planning engines as alternatives to more traditional architectures – as we found in our literature study. Agent researchers herein claim that their technologies and designs are more flexible and can better cope with change, whereas the OR community points to the fact that central assignment mechanisms are generally more optimal.

Having the situation as described above, we however perceived the challenge differently. Constructing a multi-agent system for such an application does not solely deal with constructing a different alternative to existing technologies. It has to be based on the intrinsic need for a real-time assignment engine, which is, although enterprise-centric, inter-organisational in nature. Container planning is an inter-organisational coordination problem; the planning problem is not as complex in its planning nature as in its information coordination nature. Technology to support and enable this should be focused on coordination through communication, rather than on optimisation alone. Preferably, this should be done through processes understandable to the human planner, as Krauth *et al.* (2007) showed. Many companies do not (yet) buy fully into autonomously operating planning systems, rather prefer a decision support system that automates most of the pain, keeps the planner in final control, and in the meantime helps to increase a planner's span of control. A system that better fits the needs is more understandable (by mirroring human-styled processes), and derives solutions through communication.

This is where a multi-agent system could potentially excel; see also our observations in the previous chapters. The sections that follow report on our efforts, steps and findings.

5|3 Design

5|3|1 The process of design

The initial design phase spanned a period of almost a year, from first work floor visit to the last design workshop we did. Arthur Oink (2005) wrote his MSc thesis about this. After that period, prototype development began, followed by internal tests, and validation/evaluation. Roughly, we followed the design cycle suggested by Van Strien (1986), which we included in Figure 1.2.

All phases we went through we had feedback sessions with Post-Kogeko planners, IT and management, in order to verify our understanding, and to keep them involved throughout the process. These feedback sessions were “workshop style” meaning we discussed their processes and issues for improvements, supported by slides we prepared, graphs, and Excel sheets with analysed data. For an overview of all sessions we held, see Table D.1 in Appendix D. The table shows that we organised sessions and workshops with the planners, the management, the most important customer, the largest container terminal, and domain experts at Erasmus University.

For designing the MAS, we utilised Agent UML (also: AUML) – see also section 2|7|2. In our design, we made sure to model according to the processes we observed, to match closely the processes as they are in current practise. Although we were not yet aware of the article by Mes *et al.* (2007), see also section 2|7|2, we implicitly followed the steps they recommend in designing a MAS.

A conceptual design is only a small first step; actually building the design is a different thing. That was the next step we took in our research, in order to pilot novel concepts and technologies. We went through many iterations, different designs and versions in our development process. In the period between May 2006 and June 2008, an estimated 300+ man-days were spent by the two developers on the development of the different versions of the prototype. Throughout the different phases we frequently interacted with Post-Kogeko; for an illustration, see Figure 5.4 – showing a picture of Alberdine van Velzen and Richard Crans from Post-Kogeko.

5|3|2 Agents in the system

The design started from the idea that all entities in the system that have to do with planning should have their own agent. See Figure 5.5 for an illustration. The idea was born to develop a real-time planning system, which consists of different types of software agents monitoring their environments. This includes TruckAgents monitoring truck movements and traffic jams, and OrderAgents monitoring container availability and customer preferences. Orders are assigned in real-time to trucks, based upon a mechanism that considers the order details (minimizing lateness of orders), the movements of the fleet, reduces empty miles, and potential delays due to traffic jams. The design took the current way human planners work as a starting point, and modelled the existing planning processes in agent behaviours. The decisions it makes are relatively human-understandable.

Figure 5.4 – Continuous interaction with Post-Kogeko employees

In the first version of the system we developed, there are only two types of agents: TruckAgents, and OrderAgents; for each truck in the system a TruckAgent, and for each order an OrderAgent. As soon as a truck becomes available, its truck agent will notice and starts actively searching for the next job to execute. This is done by contacting the different order agents and then calculating a score for each order – this mechanism will be explained in the next section. As a result, the truck agent gets an (ordered) array of orders to execute with a list of scores. To avoid (very) local optimisation, the truck agent then takes the highest ranked order and asks for a bid from the other trucks in the system in that specific order. The scoring incorporates the time trucks are still busy, represented by a truck's ETA, and the place where it will become available. If there is no other truck that is better capable of executing this particular order, the truck agent claims the order, and the truck is instructed to execute this specific order.

A TruckAgent can be an initiator, which means that it is actually searching for an order, as described above, or a participant. A participant responds to other initiators, but is not currently searching for an order. The Class Diagram for the initiator role is given in Figure C.4 – AUML Class Diagram – TruckAgent (initiator role)Figure C.4 in 1.1.1.1.1.1Appendix A. The order assignment procedure is given in Figure 5.6.

Figure 5.5 – Agents in the system and their characteristics

5.3.3 Scoring mechanism

The calculations a TruckAgent performs in order to calculate a score for a specific OrderAgent exists of different elements that together construct the score. A TruckAgent builds a scorelist – see also the second element in Figure 5.6:

1. Construct a list of orders $O = \{o_1, o_2, \ldots o_m\}$. These are all (m) orders still to be executed today. Orders that have been completed and orders that are currently executed are left out.
2. Decide which score-elements S to use in evaluating alternatives, and calculate the corresponding scores. The different score-elements are, $S = \{s_1, s_2, s_3, s_4\}$:
   - S_1: Customer Time Window – The score for the fit with the time the container has to arrive at the customer.
   - S_2: Customer Importance – The score for the importance of the customer.
   - S_3: Empty Mileage – The score for the empty miles to drive to the pickup of a particular container.
   - S_4: Traffic Jam Avoidance – The score for the avoidance of expected traffic peaks.
3. Look up the weights of the criteria; not all weights are equal. In this case “empty mileage” or “customer time window” can be expected to outweigh other criteria. These are: W = {W1, W2, W3, W4}.
4. Calculate the total score outcomes Dw (μ_{i1}, μ_{i2}, μ_{i3}, μ_{i4}) where w = {1, 2, …, m}. The matrix then looks like Table 5.1.

Table 5.1 – Matrix showing order scorelist for TruckAgent

	Partial Scores				Total
	W1	W2	W3	W4	
Alternatives	S1	S2	S3	S4	
O1	μ11	M12	…	μ14	Dw1
O2	μ21	μ22	…	μ2n	Dw2
…	…	…	…	…	…
Om	μm1	μm2	…	μmn	Dwm

5|3|4 Technology choices

The process to choose a particular agent-development toolkit was a pragmatic one. First, we decided to utilise a known agent-development environment (instead of, for example, a business process engine, or engineering from scratch). Second, we looked for environments in which we could utilise our experience with either Java or Visual Basic. Third, we looked for agent environments with a large user community, preferably an open-source platform. Fourth, the existence of a support environment would be an advantage; as we like to test functionalities and system design concepts, rather than engineer a new agent environment.

JADE turned out to be among the most utilised agent toolkits currently available on the market – see also Garcia and Lucena (2008). It is often used for research purposes – see a.o. Cheeseman *et al.* (2005), Chmiel *et al.* (2005), and Boer-Sorban (2008). To illustrate the popularity of JADE and its concepts, we refer the reader to Wohlin (2007) in which the Bellifemine *et al.* (2001) paper that introduced JADE ranked as the13th most cited article in software engineering for the period of 1986-2005. It is, furthermore, a (very) popular download from the ACM Digital Library.

JADE stands for Java Agent Development Environment. JADE is a software framework fully implemented in Java. It simplifies the implementation of multi-agent systems through a middleware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases. The agent platform can be distributed across machines, and they do not even need to share the same operating system. The configuration can be controlled via a remote GUI – see for example Figure 5.7. The full FIPA communication model is implemented and integrated: interaction protocols, envelopes, ACL, content languages, encoding schemes, ontologies and transport protocols.

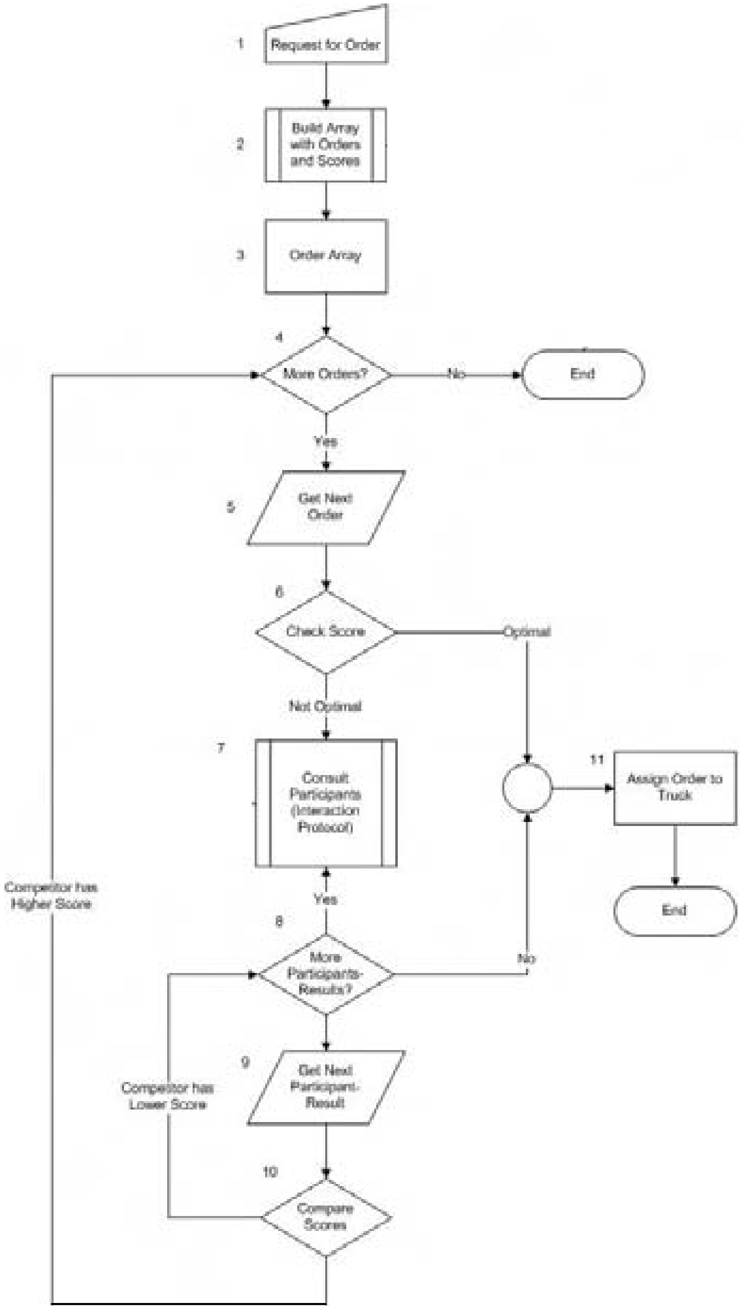


Figure 5.6 – Flowchart – Agent Head Automata of a TruckAgent (initiator role)

JADE brings features and functionalities specifically focused on agent-based system development. One of the important differences when compared with plain Java development is that agents can switch between different behaviours, and are set up to communicate with other agents via the standardised FIPA protocols. Every JADE agent is composed of a single execution thread and all its tasks are modelled and can be implemented as Behaviour objects; Behaviours can also be explicitly executed in another Java thread, but this is not the default. Behaviours are at the core of an agent. Behaviours can be of different types; an overview and short description of these types is given in Table E.1 in Appendix D.

Furthermore, an integral part of JADE is a Directory Facilitator agent (*the DF*) that keeps track of other agents in the system and their capabilities. The DF can be contacted by agents to discover other agents – this is similar to the UDDI functionality in the WebServices technology stack. See Figure 5.8, for an illustration of the DF interface. Last but not least, JADE comes with a set of graphical tools. For example the *Sniffer agent,* which can sniff and display interactions between other agents. The sniffer allows for the tracking of messages exchanged in a JADE agent platform. When the user decides to sniff an agent, or a group of agents, every message directed to or coming from that agent is tracked and displayed in the sniffer window. The user can view, save, and load, every message track for later analysis. This helps in visualising and understanding the internal processes, which is handy for debugging purposes. See Figure 5.9 for a screenshot of the *Sniffer Agent*.

Another tool included is the so-called *Introspector Agent* (see Figure 5.10), which allows one to monitor and control the life-cycle of a running agent and its exchanged messages – messages in the queue of sent and received messages. It also actively shows running behaviours, which become interesting when agents are equipped with a stack of different behaviours that they execute at the same time. Along with the other tools, this is helpful in monitoring the characteristics of the system, and in order to see whether the system works in the way it is intended.

That the JADE platform is very efficient and scalable is shown by Chmiel *et al.* (2005). They tested agent message exchange, agent creation, and migration to another machine or container. Their conclusions include the observation that JADE as a toolkit does not introduce substantial (system) overhead, and its applications are only limited by the standard limitations of Java and the Java Virtual Machine (JVM). Even on “*ancient hardware*”, they were able to run experiments with thousands of agents effectively.

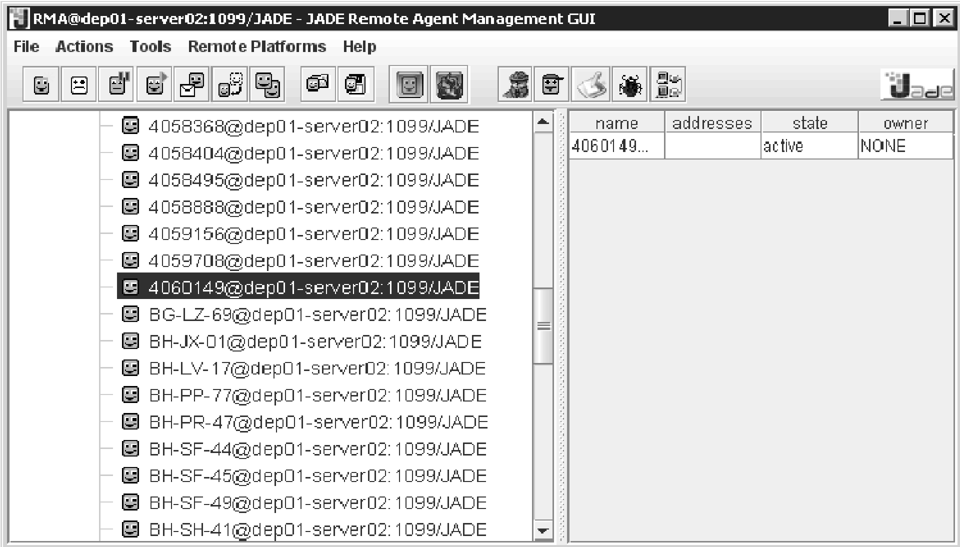


Figure 5.7 – The JADE Remote Management Agent GUI

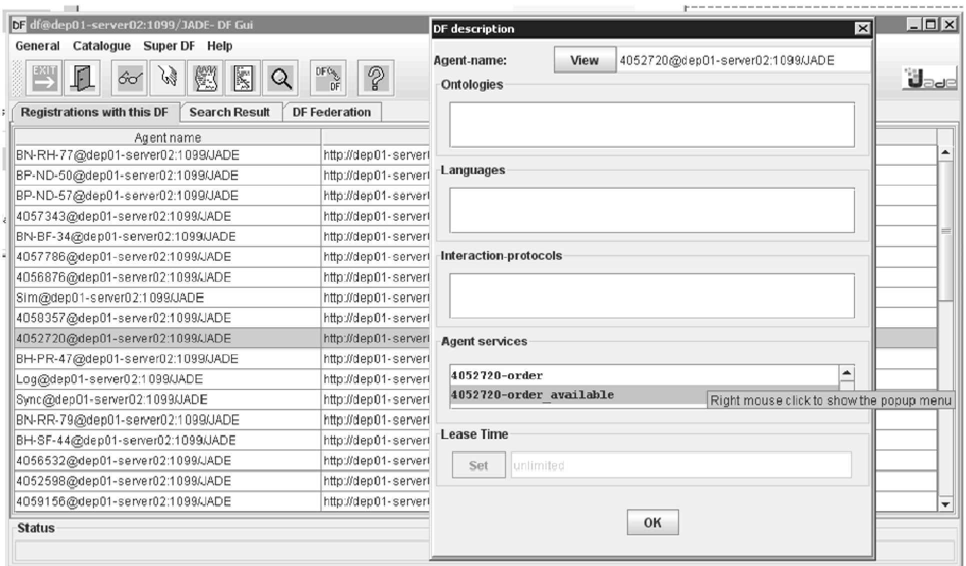


Figure 5.8 – The JADE DF GUI

For more details about JADE, both technical and functional, we refer the reader to Bellifemine *et al.* (2007). Bellifemine and his co-authors have been, and still are, at the core of the JADE development team and wrote a book that provides detailed insight in the possibilities of JADE. It also covers extensions to JADE that we did not use, including the LEAP add-on, which makes it possible to run JADE on handheld devices, such as PDAs and Mobile Phones. A special add-on has recently been released for the new Google Android (mobile) platform. The JADE platform also includes a WebServices integration gateway, which enables developers to link JADE-based agent systems with SOA applications (Sonntag, 2006).

The development of the multi-agent system was performed in Eclipse, an open-source multi-platform Java Development environment, on two Microsoft Windows XP desktop systems. The engine gets its order data from the QFreight database, and the location information either from the simulator or the CarrierWeb system database. Both databases are Microsoft SQL Servers. The SQL Servers (version 2005) contain the order information and run on Microsoft Windows 2003 Servers, and the truck simulator on a RedHat Enterprise Linux Server. The technical specifications of these physical machines can be found in Table 5.2. The AND road-network route-planning functionality was provided by our DEAL research partner Almende. The underlying map we used was Benelux only, thus allowing route calculations only within the Benelux countries. Since only a very small percentage of container trips go outside of the Benelux countries, this was not perceived to be a problem. The scoring mechanism deploys fuzzy (scoring) functions. We utilised the FuzzyJ toolkit. See `http://www.iit.nrc.ca/IR_public/fuzzy/`.

A (human) user needs a user interface. Although the JADE support tools, such as the earlier described Sniffer and Introspector, were useful for development, we needed a human-understandable, user interface for the in-company tests. The choice was made for a web-based GUI, which made it relatively easy to integrate a map. The maps used came from Google Maps, for which an open API is available – see `http://code.google.com/apis/maps/`. For the web development, we utilised the Google Web Toolkit (GWT), an open source AJAX toolkit provided by Google, which perfectly complements our development environment. GWT makes it possible to code a web-based GUI utilising a subset of Java. The GWT engine translates that into browser-executable JavaScript. GWT is perfectly suited for client-server applications. This allows a Java program to be executed at the server, which connects with the client-side in the browser (which is (sometimes browser-dependent) JavaScript) through both synchronous and asynchronous remote procedure calls. For full details, see the documentation at `http://code.google.com/webtoolkit`. On top of the GWT toolkit, we utilise the GWT-EXT (`http://gwt-ext.com`) widget library as well as the GoogleMaps-GWT API, for additional interface enhancements. A screenshot from the GUI as designed is shown in Figure 5.11.

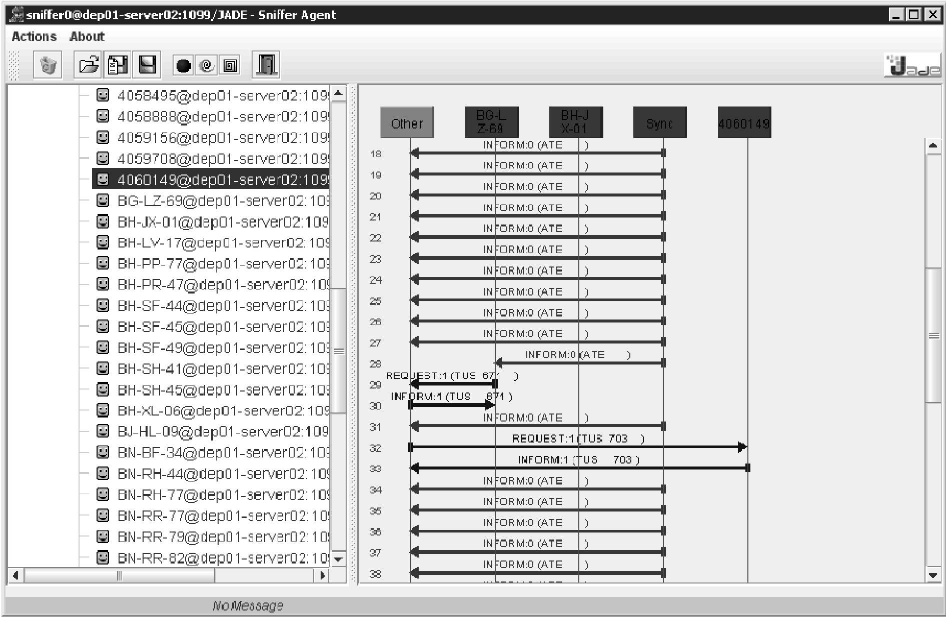


Figure 5.9 – The JADE agent sniffer

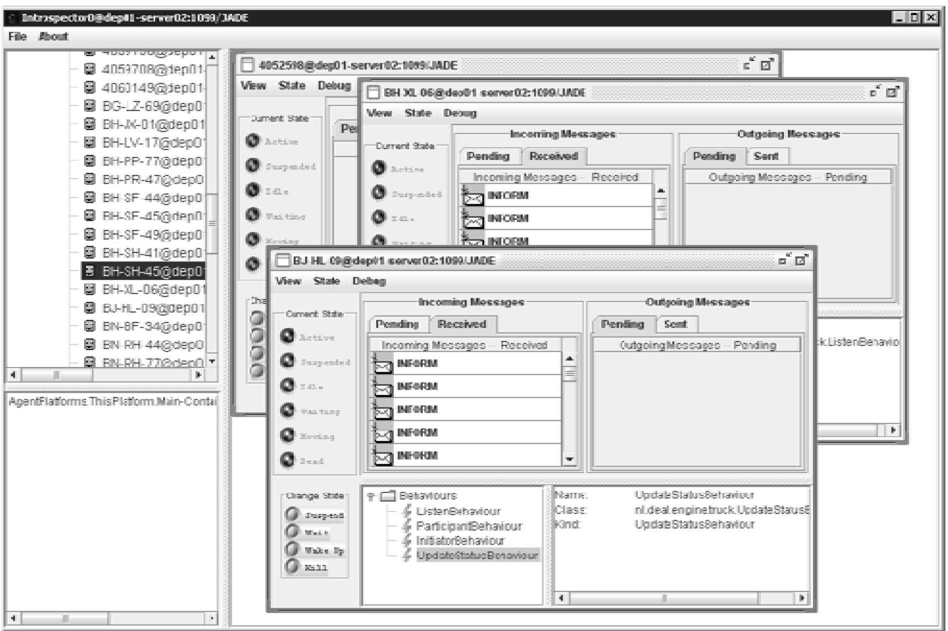


Figure 5.10 – The JADE introspector agent

Table 5.2 – Machines used in the development trajectory

Machine	Description
Development machine 01	Windows XP SP2 on a Dell Optiplex 620 (Desktop), 2,8 GHz Intel Pentium IV HT,2 GB RAM, 40 GB 7200 rpm HD
Development machine 02	Windows XP SP2 on a Dell Latitude C640 (Laptop), 2,2 GHz Intel Pentium IV, 1,0 GB RAM, 60 GB 5400 rpm HD
Server 01 (Simulator)	Redhat Enterprise Edition, via a VM on a Dell PowerEdge SC 1425 (Server), Dual–XEON 3,0 GHz, 3 GB RAM
Server 02 (SQL server, TomCat java application (web-)server)	Windows Server 2003 Enterprise Edition SP2, via a VM on a Dell PowerEdge SC 1425 (Server), Dual–XEON 3,0 GHz, 3 GB RAM

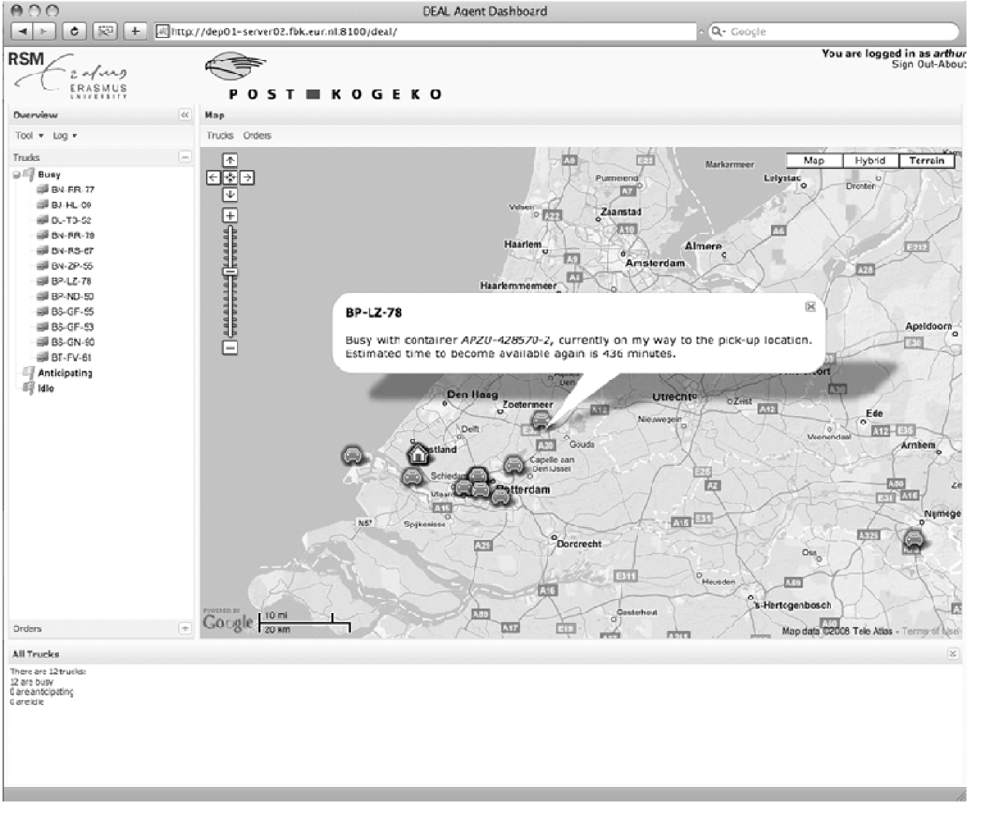


Figure 5.11 – Browser based UI, developed with GWT

5|3|5 The agents in the system

The agents in the system process the information they receive – such as truck locations, orders, order availability, customer preferences, and traffic jams. TruckAgents derive to order assignments through real-time negotiation with the other agents in the system. This way always considering the most up-to-date situation.

Three/four different versions of the prototype system have been constructed – see also Figure 5.12. We started off with the most extended (and perhaps most advanced) version, a fully automated version that runs on a simulator. Evolving from that version onwards, we first extended to a version with a UI, in order to visualise the working of the system, and to interact from a web interface with the multi-agent system and the simulator. This version runs on a web server (Apache Tomcat), and can be manually started and stopped, and set to simulate a certain day in a particular scenario. For practical reasons, we will not refer to this version as a separate version throughout the remainder of this chapter. In turn, this version was the starting point for constructing a manually operated version that could run on the (same) simulator, and eventually a version without the simulator, which connected to the Post-Kogeko backend systems. The three different versions, with their respective agents are listed in Table 5.3. For pragmatic reasons, we started with the simulator version first. First, we had to construct a "proven version" of the system converging with a version for the field-test – see section 5|7|2. Running against a simulator made it possible to experiment with different mechanisms and designs. The manual version was only developed in the last week before the field-test.

In the three versions of the system we utilise different types of agents. These agents and their behaviours are listed in Table 5.3 and Table 5.4. The most important agents are the TruckAgents and the OrderAgents. **TruckAgents** actively search and negotiate for orders with each other. To do so, they have several behaviours that communicate with other agents, the simulator, or the user interface (in the case of the manual version), or they execute specific tasks. Examples of the latter are the behaviours related to the search and selection of a new order: the InitiatorBehaviour, ParticipantBehaviour, and FindOrderBehaviour (*) in the simulator version, and the FindOrderBehaviour (#), the AssignOrderBehaviour, and the AnticipateOrderBehaviour in the manual version. TruckAgents actively communicate with OrderAgents to find out which order would be the best match.

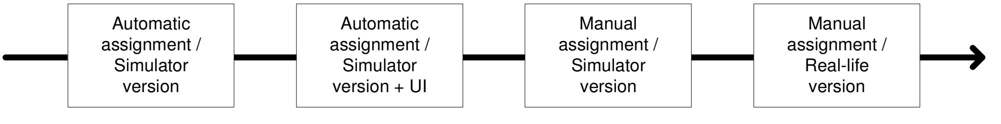


Figure 5.12 – Evolution through time of versions of the prototype

Table 5.3 – Agents in the different versions of the prototype

Agents	Automatic assign. / Simulator version	Manual assign. / Simulator version	Manual assign. / Field-test version
TruckAgent	X	X	X
OrderAgent	X	X	X
ManagementAgent	X	X	X
LogAgent	X	X	
TimeAgent	X	X	
SyncAgent	X	X	
SimAgent	X		

The FindOrder algorithm is visualised in the sequence diagram in Figure 5.13, the CalculateScore algorithm does a call to four subclasses that calculate four scores for, respectively: a CustomerImportanceScore, a CustomerTimeWindowScore (explained in detail in 0, aspect #3), an EmptyMileageScore, and a TrafficJamAvoidanceScore. The total score – for the combination of a specific truck and specific order at a certain point in time – is the sum of these four subscores times a set parameter – see Equation 5.1. TruckAgents store all information of importance to that particular truck and the calculations that need to be made by the agent. With every status update, TruckAgents calculate their expected ETA (Estimated Time of Arrival) until the moment they will be free to execute their next order. This is used in calculations when CFPs are sent out by other TruckAgents with a request for a proposal.

$$\begin{aligned}\boldsymbol{TotalScore} = {} & (\boldsymbol{customerTimeWindowWeight} * \boldsymbol{customerTimeWindowScore}) \\ & + (\boldsymbol{customerImportanceWeight} * \boldsymbol{customerImportanceScore}) \\ & + (\boldsymbol{emptyMileageWeight} * \boldsymbol{emptyMileageScore}) \\ & + (\boldsymbol{trafficJamAvoidanceWeight} * \boldsymbol{trafficJamAvoidanceScore})\end{aligned}$$

Equation 5.1 – Central formula CalculateScore (TruckAgent)

OrderAgents encapsulate all details from an order that needs to be executed on the particular date. An order stores information on its preferred customer time-window (with the preferred delivery times), the type of customer, the pickup, delivery and return locations, corresponding kilometres, and the availability of the order at the pickup terminal. An OrderAgent has behaviour to automatically check availability with the system from the terminal.

The **ManagementAgent** is a supportive agent that performs several supportive activities. The ManagementAgent is the agent that is among the first to be started in the system; it has functionality to generate the proper amount of OrderAgents (as many OrderAgents as there are orders for that day), and a sufficient amount of TruckAgents. In the field-test setting this step is bypassed. In this setting, the ManagementAgent then looks-up a database, and generates the number of trucks (with the proper number plates, used as the ID), as listed in the system for that day. Furthermore, the ManagementAgent keeps track of the amount of orders still to be executed and the amount of trucks still active in the system. It possesses the functionality (in the simulator version) to check whether a day has been completed (that is: all orders executed), in which case it communicates with the SimAgent that it can restart the system and simulate another day.

The agents and behaviours marked with a (*) are only present in the simulator versions of the system; the ones with a (#) only in the manual versions. In the simulator setting we need additional supportive agents and behaviours; more specifically, the SimAgent, SyncAgent and the TimeAgent. The **SimAgent** is the agent responsible for starting and executing simulations. In fact, it sets a particular date, starts the simulator server (running on a Linux server), and initialises the agent platform. It then starts the SyncAgent and the TimeAgent, which in turn start the ManagementAgent, which then generate the OrderAgents and the TruckAgents. It has only three behaviours: a ListenBehaviour, that listens to completion messages from the ManagementAgent; a RestartBehaviour, that is executed to restart the simulation on a brand new day; and a DelayRestartBehaviour which automatically executes after 24 (simulated) hours, or when an event occurs. The **SyncAgent** takes care of the needed communication with the simulator and syncserver, which are running at a Linux server in the network. The simulator typically simulates another date and time than the current system-time. Furthermore, it can run at a faster(or slower)-than-normal pace. The **TimeAgent** has functionalities and behaviours to handle this. The synchronisation with the simulator is handled by the SyncTimeBehaviour. Other agents contact the TimeAgent to find out the current time in the simulated world, this is handled by the ListenBehaviour of the agent.

Not yet mentioned is the role of **MyAgent**. It is not a separate agent, rather an abstract superclass of functionalities that each agent implements. MyAgent has standard functionality to log messages in an XML file, to communicate with a database engine, to register itself in a convenient way in the DF, to read properties from a `properties.xml` file, to print DEBUG messages, and to update the properties. It furthermore provides a standard LogBehaviour for logging, and a standard DoDeleteBehaviour that terminates an agent. Terminating agents can also be achieved by calling the agent. DoDelete functions from the agent, but that sometimes results in not properly ending still-active behaviours. When the DoDeleteBehaviour is called, the termination arrives as a behaviour that is only executed when currently running behaviours have ended or been actively blocked.

In order to communicate with an external application, we choose the direct O2A (Object-to-Agent) functionality, which is included in recent versions of the JADE platform. O2A can be implemented by an agent simply by `setEnabledO2ACommunication(true,0)`; it makes it possible to communicate directly with this agent from external code. Another way to communicate from an external application would be through a GatewayAgent, a concept and technology introduced in the paper by Sonntag (2006). Both approaches have their advantages, and we experimented with both. Since we however always want to communicate directly with a particular agent that we know by its Agent Identifier (AID), we see O2A as the easiest approach. Sonntag claims that a GatewayAgent offers the advantage of a single entrance point for external applications, which can be beneficial from a security point of view. GatewayAgents can furthermore implement a full webservices description, including UDDI functionality, and are thus relatively easily integrated within a larger SOA platform. The TruckAgents and OrderAgents update their status for display in the GUI in two XML files we utilised in the interface, namely `trucks.xml` and `orders.xml` (see for an example Appendix E). The GUI interprets these files, and graphically displays them on a map – see Figure 5.11, which shows the truck view.

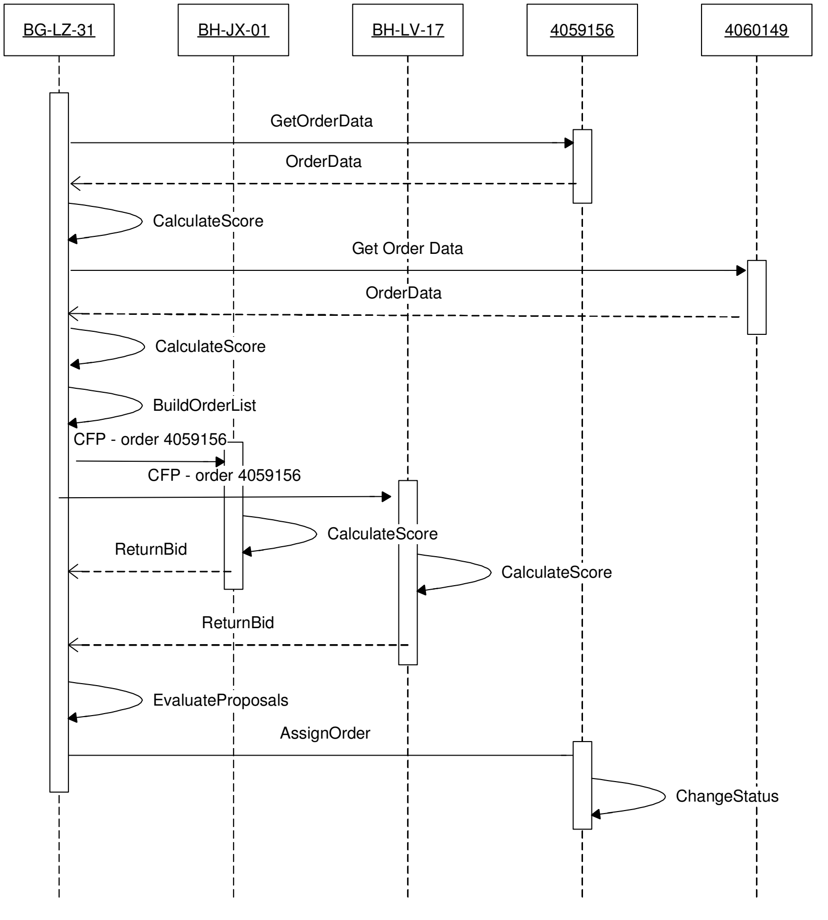


Figure 5.13 – Sequence diagram – FindOrder process

The LogBehaviour is a MyAgent behaviour and can thus be implemented by each agent in the system. We introduced this feature to let agents log their actions and its impacts, so that we could easily analyse what happens and what has happened. This was also done in an XML file, namely in `log.xml`. The GUI has a feature to display the logs, and analyse the log file. We included an analysis of Truck Hours, Truck KMs, Order Hours, and Order Availability. For a screenshot of the Truck Log information see Figure 5.14.

In the Appendix we included some of the code we wrote for this application. Shown is the code for the `TruckAgent.java` (in Appendix F) and two of its behaviours, namely the `ListenBehaviour.java` (in Appendix G) and the `InitiatorBehaviour.java` (in Appendix H). Including all code would consume too much space. These three examples, however, illustrate the inner-working of the agent system as they include communication with other agents, DF consultation, internal operations, logging, *et cetera*.

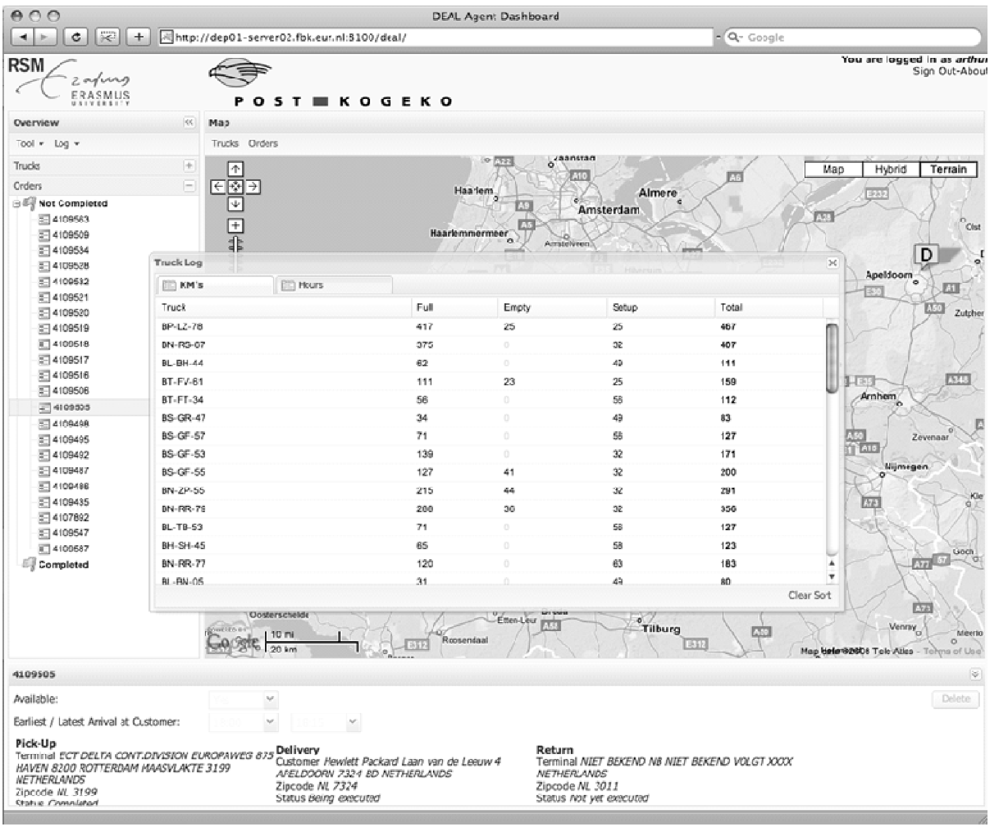


Figure 5.14 – The user interface, showing the log dialog

Table 5.4 – Overview of agents and their behaviours (* sim only; # manual only)

Agent	Agent Behaviours	Role
MyAgent (abstract class)	LogBehaviour	OneShotBehaviour fired to write a message to the log.
	DoDeleteBehaviour	OneShotBehaviour to kill an agent (including deregistration)
TruckAgent	InitialBehaviour	Initial behaviour that starts the different behaviours of the TruckAgent
	ListenBehaviour	Cyclic behaviour that listens to messages from other agents
	ObjectListenBehaviour (#)	Cyclic behaviour that listens to messages from external Java applications (through O2A communication)
	InitiatorBehaviour (*)	Cyclic behaviour that listens to messages to trigger the search for a new order (than the truck becomes the initiator). There is, at maximum, one initiator at any given time.
	ParticipantBehaviour (*)	Cyclic behaviour that listens and responds to messages from TruckAgents with an initiator role.
	FindOrderBehaviour (#)	Behaviour that searches for a new orders and contacts the other TruckAgents. Different from the initiator and participant behaviour, since this behaviour produces a result to be interpreted by a human.
	AssignOrderBehaviour (#)	Behaviour that assigns a specific order (result of manual response on FindOrderBehaviour) to a TruckAgent.
	AnticipateOrderBehaviour (#)	Behaviour that directs a TruckAgent to anticipate a certain order (result of manual response on FindOrderBehaviour).
	CWUpdateBehaviour (#)	Behaviour that updates the location of the TruckAgent from CarrierWeb.

	FindOrderBehaviour (*)	Behaviour that is needed to find a suitable order – typically started by the Initiator and Participant Behaviour.
	DelayFindOrderBehaviour (*)	Waker behaviour that only runs in case a TruckAgent is idle, still searching for a new suitable order, and not yet gone home.
	HitTheRoadBehaviour (*)	Waker behaviour that is executed to change the status from "waiting at a location (terminal or customer)" to "driving again" after a specified amount of time. Needed in the simulation scenario.
	UpdateStatusBehaviour	Cyclic behaviour that updates the status and location of the TruckAgent in the database, in reaction to messages coming from the tracking system (being the syncserver or the CarrierWeb system).
	SetETABehaviour	Behaviour that calculates the new ETA of the TruckAgent.
OrderAgent	InitialBehaviour	Initial behaviour that starts the different behaviours of the OrderAgent
	ListenBehaviour	Cyclic behaviour that listens to messages from other agents
	ObjectListenBehaviour (#)	Cyclic behaviour that listens to messages from external Java applications (through O2A communication)
	CheckAvailabilityBehaviour	Behaviour that performs an order availability check.
	DelayCheckAvailabilityBehaviour	Ticker behaviour that starts the CheckAvailabilityBehaviour after a defined time.
ManagementAgent	InitialBehaviour	Initial behaviour that starts the different behaviours of the ManagementAgent
	ListenBehaviour	Cyclic behaviour that listens to messages from other agents
	ObjectListenBehaviour (#)	Cyclic behaviour that listens to messages from external Java applications (through O2A communication)

	GenerateTrucksBehaviour	Behaviour that calculates the amount of needed trucks, and starts a corresponding amount of TruckAgents.
	GenerateOrdersBehaviour	Behaviour that extracts the orders from the database, and starts a corresponding amount of OrderAgents.
	Delay…Behaviour	Simple TickerBehaviours that start the CheckOrdersBehaviour, GenerateOrdersBehaviour, and CheckDayCompletionBehaviour, after a defined amount of time.
	CheckDayCompletionBehaviour (*)	OneShotBehaviour that checks completion of a simulated day
SyncAgent (*)	ListenBehaviour (*)	Cyclic behaviour that listens to messages from other agents and the syncserver (used for simulation).
	InformBehaviour (*)	OneShot behaviour that sends the XML message, as received from the syncserver, to the right agent in the system.
TimeAgent (*)	ListenBehaviour (*)	Cyclic behaviour that listens to messages from other agents.
	SyncTimeBehaviour (*)	Cyclic behaviour that syncs the time and simulation rate with the syncserver (used for simulation).
LogAgent	LogBehaviour	Ticker behaviour that performs different log activities (on screen, XML file, *et cetera*)
	ListenBehaviour	Cyclic behaviour that listens to messages from other agents.
SimAgent (*)	ListenBehaviour (*)	Cyclic behaviour that listens to messages from other agents.
	RestartBehaviour (*)	OneShotBehaviour that restarts the simulator on the next day to simulate; it also creates the proper simulator logfile.
	DelayRestartBehaviour (*)	Waker behaviour that fires the RestartBehaviour after a specified amount of time.

5|3|6 Five specific agent design choices

We will not document the entire development process, since that lies beyond the scope and focus of this dissertation, but would like to discuss five specific design decisions we had to make when we were constructing the system. We will discuss specifically these engineering decisions because we think these represent well the type of choices one has to make in agent-based designs. The problems are briefly described in Table 5.5, which also explains the solutions we chose. In the Appendix we included a more detailed description of the specific problems – see Appendix I.

Table 5.5 – Five specific engineering decisions in the prototype building process

#	Problem	Short Description	Solution
1	A Super Truck "eats up" all orders [see Appendix J|1]	A well-positioned (other) TruckAgent wins multiple auctions from the same TruckAgent. This is due to the way the scoring mechanism works (and the exclusion of future orders and capacities).	Supertrucks that claim all orders, but cannot execute them: this is not realistic. When another TruckAgent wins an auction, it will no longer compete in the auction of the "next best" order.
2	Which agent decides when trucks can go home? [see Appendix J|2]	A problem of autonomy. The agent decides either for itself, or a supervisory agent assists in the decision. Underlying problem is the fact that there should remain enough trucks active in order to serve the remaining orders in due time.	We tried two mechanisms: (1) ManagementAgent decides whether a TruckAgent may go home if it asks to go home; (2) TruckAgents themselves reason whether they could go home now, and communicate this with other TruckAgents. After experimentation we decided to go with the second option.
3	How to handle orders that cannot be served on time anymore? [see Appendix J|3]	The membership function for the CustomerTimeWindow in the scoring mechanism aims at delivery within the specified time window. However, if orders become impossible to be delivered by any truck on time, how to handle these? Should you give these an absolute priority?	We decided that such orders should get an absolute priority. But, only in the case that it is not possible anymore for any truck to deliver such an order in time. We refer to the latter as GloballyTooLate. This since otherwise the scoring mechanism (rating higher for too late orders)might result in delaying orders until they are "just too late".

4	How to handle idle trucks? [see Appendix J|4]	How to handle idle trucks. That means: trucks that are not busy with an order, but are not allowed to go home yet? Should you let them wait at their current location? And let the TruckAgent try again in a couple of minutes? Or should you build in anticipating behaviour that anticipates expected orders?	We decided that trucks should try to anticipate future orders when they become idle. An anticipating truck reconsiders the total order set every so many minutes. This way, trucks are earlier at the new location. An anticipating truck locks the order, to prevent that future anticipating trucks anticipate the same order.
5	What if a human has the final decision? [see Appendix J|5]	Initially we designed the system as a fully autonomous planning engine, which needed no manual assistance. In a real implementation, this scenario is not very likely; most companies want to keep the end control in the hands of an experienced planner.	This influences the system heavily. The system becomes more a Decision Support System (DSS), and many of the choices discussed above are no longer the type of aspects the system truly needs. The system operates more in a sense that it makes suggestions to the planner. For example, with new assignments, or trucks to go home. The planner makes the final decision. For the field-test we made several of these changes; these will be documented in more detail in section 5|7.

5|3|7 Design evaluation approach explained

Design research would not be serious research without proper evaluation steps. As we discussed in Chapter 1, there are many different evaluation methods possible (see Table 4.1). We reasoned that there is no single best approach. The design and prototyping process as such, for example, is already an important evaluation approach, following Nunamaker and Chen (1990). We will discuss this in 5|4; in which we review the process, and specifically pinpoint one major change we made in the design.

Following the discussion in Chapter 1, we chose three additional evaluation approaches, with different levels of evaluation, levels of control and different rigor/relevance balances – see also Table 4.1. In section 5|5, we discuss a simulational comparison against two other planning prototypes – a method with a high level of control and thus high on the rigor axis, however, relevance is relatively low (and limited by all limitations made in the simplified simulation setting). Furthermore, it only assesses the processing level of the system. A third evaluation method we utilise is reported on in 5|6: an expert evaluation forum. Note that we also presented a prototype version of the system at the

ICIS-WITS 2007 workshop – see Moonen *et al.* (2007) – however, the feedback received there was not written down structured enough for incorporation in our evaluation. Section 5|7 discusses an in-company field test/experiment. This scores very high on the relevance axis, but its rigor is low – it is difficult to repeat, is researcher dependant, and also planner dependant. One instrument to overcome the latter might be to involve multiple planners and multiple researchers/observers, although these might influence each other.

5|4 Evaluation I: development and prototyping

5|4|1 Continuous design loops

Few programmers are gifted enough to construct faultless software code from scratch. Although integrated development environments (IDE), such as Eclipse, help in writing compiler ready code, bugs tend still to be present in the code. This is either due to faults in the coding, or unforeseen processes or circumstances, but can also be caused by a different programming paradigm, as we found out through the process of prototyping a multi-agent system. The development of a multi-agent system requires a mental switch for classically schooled programmers.

Running, tracing and solving bugs, and along that line, rethinking mechanisms and discovering smarter mechanisms are therefore parts of the construction process of the prototype. The prototype reported in this chapter has been worked on by two developers: a main developer, who did most of the programming, and a functional designer (the author of this dissertation) who assisted in the programming. We dare to state that the frequent and sometimes very intense discussions between the two developers benefitted the prototype enormously, as did the contact with outsiders on it.

Throughout the development process, continuous testing of the code took place, for which we utilised a test set distilled from historical execution data from Post-Kogeko. Two rather simplified days were used, one with only 12 orders to be executed by 5 trucks, and one with 52 orders to be executed by 26 trucks. An average day in practise generally has a larger amount of orders and trucks. For testing purposes however, these simplified days made life easier – due to a reduced setup time for the creation of order and truck agents. Short setup times are important in the debugging phase when many runs are required.

5|4|2 The introduction of the order agent

The design decisions as documented in Table 5.5 are an example of some of the decisions made throughout the development process. An important decision made relatively late in the process was not mentioned before: the introduction of the OrderAgent. Although present from the beginning in our initial design, we first tried for a solution in which we only had TruckAgents and Order Objects (along with the supportive agents as documented in Table 5.4). The order information was obtained from the QFreight database each time a truck initiated a FindOrder event. The ManagementAgent performed an availability check, and updated the order data in the QFreight database.

Although this solution worked in principal, it resulted in a large amount of database calls at each FindOrder event, and it complicated the creation of smart behaviours for orders. Instead, we had the ManagementAgent checking availability of orders, and updating the order database. It would be more logical, and more in line with agent principles, to construct an OrderAgent that has its own behaviours: such as checking and updating its availability, its estimated execution time (the duration of the trip) and communicating its details to a TruckAgent when requested. It furthermore prepared a basis for future intelligent extensions, something we will discuss later. OrderAgents contacted the order database just once, only at setup to read in all required order details. In the past it sometimes happened that, due to the fact that we were dealing with literally tens of thousands of database transactions (and connections), the system sometimes failed for hard-to-trace reasons despite the static and synchronised methods we applied.

5|5 Evaluation II: simulation

5|5|1 Context

Within the context of the DEAL project, other researchers have been working on different aspects of multi-agent systems. Mahr and Srour have done, and are still doing, extensive research into multi-agent system algorithmic design, the performance of MAS, and looking into the comparison of MAS algorithms with more traditional OR techniques. See, among others, Mahr *et al.* (2008). They report on a series of simulations they performed in order to compare a MAS design with an online optimisation algorithm and an estimation for a baseline optimum. It is shown that in a context with a relative high dynamism, the MAS approach chosen performs similar to the online optimisation.

Although our prototype was never designed to solve the exact same problem they have been working on, we decided to compare our approach to theirs, nevertheless. It gives insight into how our agents perform, and delivers insights for redesign. We do not explain details of the competing prototypes here, but refer the reader to (Mahr *et al.*, 2008). The dataset used for simulation is the same one, and it is based on transaction data from Post-Kogeko.

5|5|2 Description of experimental environment

For the simulation experiments we used a dataset based on real execution data, coming from a large dataset with execution data spanning the period from January 2002 to October 2005 and January 2006 through March 2006.

We could not use the data in its raw form. For many orders we had to correct the zip code in the address information, since many addresses referred to postal boxes instead of physical terminal locations. A postal box has a zip code positioned in the city centre; generally not the location of a terminal or customer. Furthermore, we excluded all non-Benelux orders. From the adjusted dataset, we extracted a random sample of jobs in order to generate a set of 33 days with a total of 65 orders per day. The number of 65 represents relatively well the average daily job load. Each order consists of a pickup location, customer location, and return location.

To standardise the data for our experimental purposes, we specified time windows at all locations as follows: for the terminals (the pickup and return locations), the time windows span a full twelve hour work day from 06:00 to 18:00, and the time windows at the customer locations are always 2 hours. The start of each of the 65 customer time windows occurs throughout the day, and roughly follows a uniform distribution. Given the variation in customer locations, the workload per day varies similarly. On average, each job requires approximately 4,2 hours of loaded distance (including 2,5 hours at terminals and customer).

We rendered the 33 days of data into four scenarios with varying levels of order arrival uncertainty. This was done by altering the order release time from the terminal, i.e., the point-in-time from which onwards the order can be picked up from the terminal. We set this to be either at the start of operation (06:00 hours), or two hours before the start of the customer time window (i.e., four hours before the end of the customer time window, leaving slightly less than two hours on average before the latest departure time from the pickup location). We generated these points-in-time in the day by using a uniform distribution. We used a uniform distribution since the original data did not fit with other distributions. The four different scenarios are given in Table 5.6.

The amount of trucks to be used for every day equals an average of 34 trucks, which means that every truck serves a little less than two orders a day. The actual number of trucks for a certain day can be calculated using Equation 5.2. As the formula shows, the needed amount of trucks depends mainly on distances to be driven for delivery (from the pickup to the delivery, and from the delivery to the return; respectively D_{pd} and D_{dr}). The processing times at the pickup (P_p), delivery (P_d) and return (P_r) are set to a constant 1:00, 1:00 and 0:30 hours respectively. Eight working hours are taken as the standard length of a working-day for drivers, although a driver works longer, as daily setup time is considered to be on a driver's own time. These variables have been provided by Post-Kogeko. The amount of trucks used thus differs per day, but lies between 32 and 36.

$$\frac{\sum_{i=1}^{65} \mathbf{P_p} + \mathbf{D_{pd}} + \mathbf{P_d} + \mathbf{D_{dr}} + \mathbf{P_r}}{\mathbf{8}\ (\boldsymbol{hours})}$$

Equation 5.2 – Amount of trucks depends on distance and processing time

Table 5.6 – Four simulation scenarios

Scenario	Orders per day	% / [#] known at start of operations	% / [#] unknown at start of operations
A	65	100% [65]	0% [0]
B	65	50% [33]	50% [33]
C	65	10% [7]	90% [58]
D	65	0% [0]	100% [65]

The simulations have been performed at a speed six times faster than normal, which means that a twelve hour work day can be simulated in two hours. In this setting, the agents prepared a plan (in our case a plan containing solely one assignment) that was sent to the simulator and there executed. The feedback from the simulator system was fed to the agent system again. The messages exchanged between the MAS prototype and the simulator were logged in files. These files could later be analysed by an analyser, referred to as "The Judge", which is an instrument developed and used by Mahr and Srour in their experiments – see also Mahr *et al.* (2008).

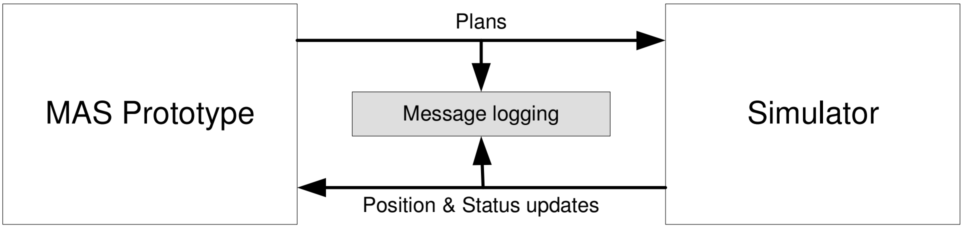


Figure 5.15 – Communication and logging in the simulator setting

5|5|3 Simulation results

In total, we simulated the four scenarios with 33 different days in four different experiments, using different parameter settings for our agent-planning engine. This means that we simulated 4 x 4 x 33 different days, which results in a total of 528 days. Since we could perform 12 simulation days per 24 hours, simulation spanned a total of 44 consecutive days. Due to the way the simulation was set up, the agents were idle most of the time, generally only updating their status and GPS position. We cannot provide numbers on the total server load, but noticed that this was low during the experiments, except for the start up of a new day when all TruckAgents were created, searched for new orders, and negotiated with each other.

The parameter settings for the different experiments are shown in Table 5.7. The table shows that the experiments only made use of two out of the four built-in scoring-factors. Since the simulator does not include any delays and traffic jams, incorporating this factor into the assignment decisions would be useless. The same holds true for the customer importance. The four different experiments differed in the weighting of the total score (made up as described in Equation 5.1), used to reflect the importance of meeting the customer time window versus the empty mileage score. The CustomerTimeWindowScore is set high, since that is the single most important factor Mahr and Srour evaluate their results upon.

The log files analysed by "The Judge" result in a long list of results. The first thing that appeared to us was the fact that the analysis revealed that many days showed an execution of more than 65 orders; namely 66, 67 or even 68 orders on that particular day. The mechanism goes on until all orders have been processed; thus, this means that for some reason, on these days the same order was executed by two TruckAgents.

An analysis of (a part of) these results is given in Table 5.8, Table 5.9, Table 5.10, and Table 5.11. Table 5.8 shows the values of the amount of idle and empty hours and the amount of trucks used for the four different scenarios in the 80-20 experiment. It can be noticed that the difference between the scenarios is not large. Table 5.9 is a correction on Table 5.8, which only shows the results for days with 65 orders. As can be noticed, only 42% of the simulated days contained just 65 executed orders. The difference between the two tables is, however, relatively small.

Table 5.10 shows the values for the deliveries for the four different scenarios for the same experiment [80-20]. What can be noticed is the high number of "too early" deliveries, the relatively high number of "too late", and the strikingly low number "on-time". In Table 5.11, the same set of seven days is compared for Scenario C. Those were all days that had 65 executed orders, not more, across the different experiments. The numbers of these days were in fact: 1, 11, 15, 19, 22, 28 and 33. Again the similarity between the numbers is striking.

Table 5.7 – Parameter settings in four different experiments

Experiment	Customer Time Window Weight	Empty Mileage Weight	Customer Importance Weight	Traffic Jam Avoidance Weight
50-50	50	50	0	0
80-20	80	20	0	0
90-10	90	10	0	0
95-05	95	05	0	0

Table 5.8 – Mean values hours and trucks [80/20 experiment]

Scenario	# of days	Hours idle	Hours empty	Trucks used
A	33	0,45	40,3	32,9
B	33	0,43	39,9	32,9
C	33	0,44	40,2	32,9
D	33	0,46	40,1	32,9

Table 5.9 – Mean values hours and trucks [corrected 80/20 experiment]

Scenario	# of days	Hours idle	Hours empty	Trucks used
A	14	0,44	39,2	32,6
B	10	0,43	39,6	33,2
C	17	0,44	39,7	32,8
D	15	0,44	39,6	32,9

Table 5.10 – Mean values deliveries [corrected 80/20 experiment]

Scenario	Orders/day	Early	Late	On time	Rejections
A	65	43,1	12,5	9,4	0
B	65	43,0	12,4	9,6	0
C	65	42,4	11,9	10,7	0
D	65	43,5	12,1	9,3	0

Table 5.11 – Comparison of results from the same seven days [scenario C]

Experiment	Idle (seconds)	Empty (seconds)	# Early	# Late	# On time	# Total
50-50	11426	996645	299	84	72	455
80-20	11066	995770	299	85	71	455
90-10	11848	995488	297	85	73	455
95-05	12524	995377	299	84	72	455

5.5.4 Reflection on the results

When comparing the results with the results reported by Mahr et al (2008), we see several important differences. Their results have very high “idle times”, though those are close-to-non-existant / minimal in our approach, but they have less “empty hours”. Another difference is in the job rejections, which is something they observe in their experiments, and we do not. The largest difference however lies in the amount of “on time” orders which is disappointingly low in our approach – see among others Table 5.10 and Table 5.11. Please note that Mahr and Srour left out several simulated days from their analysis, since they could not calculate a feasible optimal in time for these days, and only consider a subset of 26 out of the original 33 days – we decided nevertheless to simulate all 33 days.

The low number of “on-time” deliveries has to do with engine design. The TruckAgents in the system always try to find a new order, and, when doing so, they select just the highest scoring order. However, with deliveries spread over the day, at a certain point the orders that have to be delivered early have been, or are being, served by other trucks already, and the next TruckAgent selects an order which is going to be served “too early”. TruckAgents only wait when orders are not yet physically present at the terminal, but not for orders that are going to be delivered (a little bit) too early. The total “idle time” in the system is very low, as Table 5.9 shows. In fact, that number represents solely the time needed to initialise the system, and send all TruckAgents to work. When comparing with Mahr and Srour we notice that they generally have several hours of idle time per truck, whereas our idle time is less than ten seconds per truck (and, incorporating the simulator factor, this is even less of an issue).

Instead, the empty time is relatively high in our approach. On average (see Table 5.9), this is 39,5 hours per simulated day, whereas Mahr and Srour report empty times as low as 33,0 or even 31,5 hours for their approaches – with a claimed *a-priori* optimum of 27,7 hours. Empty time is the sum of all time used while not working directly on orders, thus also the time travelling empty from one (delivery) terminal location to the pickup terminal of the next order. With respect to our prototype, it also includes the time needed to find the next order; a process that only starts when the simulator signals that the previous order has been completed. The mechanism designs by Mahr and Srour plan in advance, incorporating the fact that the simulator executes the plan exactly as planned. This thus brings in another source of empty hours; since a FindOrder process, although fast, consumes time anyway.

Another factor due to the engine design is the non-existence of job rejections. The MAS design does not allow any job rejections. The last TruckAgent can never go home and quit working when all orders have not been served.

A serious issue in our design was revealed by the fact that in the largest part of our simulation runs, a double delivery of orders could be observed. A TruckAgent selects a particular order, releases the initiator lock, the OrderAgent is signalled by the TruckAgent, and accordingly updates it status – see Figure K.1 in Appendix J. However, it sometimes happened that the next initiator TruckAgent was quicker in selecting a list of potential OrderAgents than that the OrderAgent which was claimed by the previous TruckAgent had updated its status, despite the delay in the generation of TruckAgents that have been built in as a safety mechanism – see Figure K.2 in Appendix J for an illustration. This implies the need for another mechanism to synchronise activities between agents. Due to time limitations, we were not able to integrate this into our prototype and rerun the experiments. Nevertheless, we think it is an important lesson for further work.

Mahr *et al.* (2008)) has already made critical remarks concerning the dynamism of the simulation setting. The only dynamic aspect in these simulation scenarios was the percentage of order release information known at start of operations. Terminal and customer processing times are constants, driving speeds are constants and no accidents such as breakdowns occur. Simulation is always an abstraction, but this is perhaps too abstract. Especially for a prototype that has been built from the idea that the world changes every minute, and that real-time information makes a difference.

Another observation we made is that running at six times normal speed, the agents are still idle most of the time. This could be noticed observing the system performance monitor tool built into the Windows 2003 server. In fact, this means that there is a possibility to put agents to work in during the time they are idle; there is a chance for them to do more “intelligent stuff” in the time they are idling, for example preparing future decisions, or doing re-optimisation.

Thus, we can conclude that first the prototype performed not well, compared to the approaches discussed by Mahr and Srour *et al.* (2008). The plan is constructed in real-time, all time needed for calculations, communication and negotiation worsens our results. Second, the design is as such that the trucks do not stand idle. If work can be executed, it will be executed. Even if it is delivered too early. This decision was made in the design phase together with Post-Kogeko planners – they do not work with as strict time windows. Third, the way of analysind and comparing the results have been taken from the stream of work started by Mahr and Srour. In this approach “idle time” is not perceived bad, and “empty time” is. Furthermore, executions are judged on their fit with the exact time window, something our prototype had not been particularly designed for. Fourth, the simulation environment does not include any accidents or unplanned obstructions – it is not very dynamic. Driving speeds are constant (and known beforehand), and customer preferences do not change over the day. Hence, if there is nothing to react upon in real-time, why should one plan or assign in real-time?

5|6 Evaluation III: expert evaluation

5|6|1 Description of the workshops

Early 2008, we organised a one-day event in which workshops were organised with a total of 45 experts participating. Eight people were involved in the active organisation of the workshops. Table 5.12 gives an overview of the workshop participants – the organisers are not included in the numbers. We clustered the participants in different categories according to the primary industry they work in. As can be seen, most participation came directly from the logistics sector; a category which includes shippers, 3PLs, and port operators. The other categories are ICT, Consultancy, Policy Makers and Research Institutes / Universities.

The objectives of the workshops were threefold. First, the workshops enabled us to evaluate and validate two prototype MAS systems and their underlying concepts with experts from the field, and to gather their opinions and feedback. These prototypes are the prototypes developed for the Post-Kogeko container planning (this chapter), and the APPROACH prototype as discussed briefly in the next chapter and in Douma's dissertation (Douma, 2008). Second, we discussed implementation aspects of future multi-agent systems: "*How to move these prototypes into real practise, and what hurdles can be expected?*". Third, valorisation was our target: to share what we did within research, and to investigate with the industrial participants whether possibilities (and interests) for future research existed.

The event consisted of four parts. The entire group of participants first were involved in an introductory session, after which the group was split in two separate groups. These groups went to the same two parallel workshops, but followed these in a different order – see Figure 5.16. After the two workshops, all participants grouped together again for a general concluding session, the fourth part of the workshop. In this last session, we looked back at the reactions from the two workshops, and discussed success- and failure-factors for inter-organisational multi-agent systems with the group. One workshop concentrated on the application of multi-agent system technology within road planning (the prototype/design as discussed in this Chapter), the other on multi-agent application to plan barge rotations – which is the topic of the next Chapter (see specifically 6|6). Participants in group A first went to the barge planning workshop, and then to the road planning; participants in group B had a reversed schedule. In the road planning workshop three researchers from the Erasmus University discussed the application of Multi-Agent Systems & concepts within the road planning domain, whereas the barge planning workshop was hosted by two researchers from the University of Twente.

Table 5.12 – Overview of workshop participants

Industry	Number of participants
Logistics industry	20
ICT	8
Consultancy	9
Policy Makers	3
Research Institutes / Universities	5
Total	45

During the discussions, the workshop hosts took notes, which they cross-checked afterwards with the other hosts. This way they assured that all issues discussed were captured in the notes. The expert feedback forms were collected at the end of the workshops; unfortunately, however, not all forms came back completely filled in. We received a total of 27 completed forms. A form consisted of a set of questions covering the road planning case, a set covering the barge planning case, and a set of questions asked after the two workshops, the general part. In this general section of the form, questions were asked on the success- and failure-factors for inter-organisational multi-agent systems. The questions asked are shown in Table 5.13; the first two we discuss here, the third and fourth in the next chapter, and the last three, the more general questions, are discussed in Chapter 1.

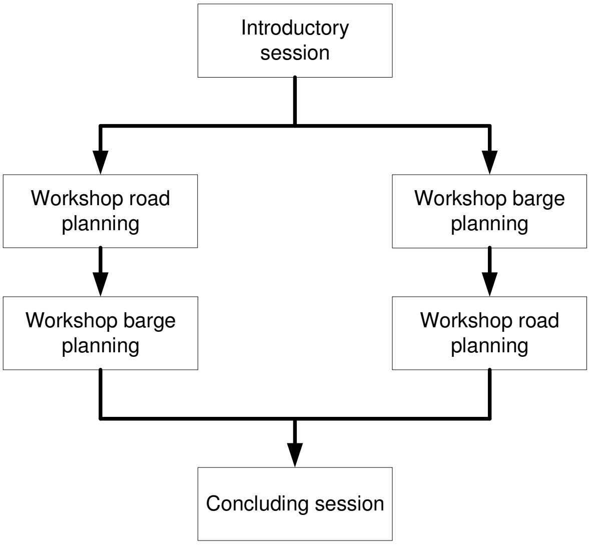


Figure 5.16 – The expert workshops

Table 5.13 – Questions asked during the workshops

#	Workshop part	Question asked
1	Roadplanning	What advantages and disadvantages of the shown multi-agent system do you see for road transport?
2	Roadplanning	How should the multi-agent system support the planner in his/her activities?
3	Bargeplanning	What advantages and disadvantages of the shown multi-agent system do you see for barge-rotation planning?
4	Bargeplanning	What advantages and disadvantages do you see in the exchange of waiting profiles?
5	General part	What do you perceive to be the success factors for multi-agent systems for transportation?
6	General part	What do you perceive to be the failure factors of multi-agent systems for transportation?
7	General part	What other application domains do you see for multi-agent systems?

5.6.2 Prototype demo

The prototype application was demonstrated in an interactive setting, which left room for questions. The application demoed was the automatic simulator version of the system, see Table 5.3. Demoed was the interface, but the inner-workings of the agent platform were also made clear through a demonstration of the JADE sniffer, DF, and agent introspector tools. As part of the demo, the design process and some ideas for future extensions were presented. After the demonstration, the participants were asked to fill out the first part of the questionnaire; specifically, question [1] "*What advantages and disadvantages of the shown multi-agent system do you see for road transport?*", and [2] "*How should the multi-agent system support the planner in his/her activities?*", see also Table 5.13. The participants had ten minutes for answering these questions on the form. After this, there was room for open group discussion in which the questions on the forms were the leitmotiv. The discussion was moderated by the researchers. Only after everyone had the chance to write down his or her own answers did the group discussion take place.

5.6.3 Results from the group discussion

The discussion in the first group in the road planning workshop, for which this was the first workshop, centred around several aspects of the prototype and especially the underlying concepts. The group, which consisted of twenty-five participants, discussed technology aspects of multi-agent systems, the potential of the shown mechanism for inter-organisational chain applications, the question of how to realise the latter, and several other issues. Interestingly enough, the second group, consisting of twenty participants, took a different perspective and discussed – triggered by the same two questions – other aspects. The position of the human planner in such a system, especially in future implementations, got a lot of attention. Another issue that got attention was the large potential of these concepts for other applications, next to the ones shown. It was not a surprise that this became topic of discussion in the second group, since this group had just gone through two workshop sessions instead of just one, as opposed to the first group. In this second group, the issue of implementation also got serious attention. One participant reported on his experiences over the past years in bringing multi-agent systems to market. A selection of the remarks from the group discussion is clustered in Table 5.14; a more detailed list can be found in Table L.12 and Table L.13 in Appendix K.

Table 5.14 – Selection of remarks from group discussion roadplanning workshops

A selection of remarks made in the group discussions
"Robustness is an advantage. If the system crashes or fails, only part of the system goes down [in a truly distributed setting]."
"Is another generation of programmers needed, or can we train current developers?"
"The largest benefits of agents are revealed in true chain applications (applications that require coordination between links in the chain)"
"This demo only shows a limited application: it is highly operational. A future real-life system should include also longer-term planning (tactical / strategic) functionality."
"A large advantage of agent application is that the solution-finding process is very human-like and thus understandable and explainable. This could be a major factor in the acceptance of such a system."
"The daily real-world practise deals not so much with optimality. It is about a good enough solution that will be acceptable in the market. This is not necessarily the same."
"Be aware that human planners always have cold-water-fear when new information systems are introduced: the planner perceives such a system as a possible replacement for himself."
"It is important to test the prototype system in real practise. That will provide really useful insights."
"Coordination, as such, is very important in logistics."
"It is definitely not the technical question. It is mainly the question how to introduce such technology, systems and processes in a complex setting."

5|6|4 Structuring the feedback received

The answers to the questions and the discussion in both the road- and barge-planning workshops can be roughly split in two parts: One, feedback was received concerning the prototypes – participants identified weaknesses, and gave suggestions for further improvement. Two, feedback was gathered on the concepts that underlie the prototypes demonstrated, and its wider possibilities for future applications. The largest part of the feedback concerned feedback in the second category.

Despite the perceptual nature of the feedback received, we think that it provides useful insights. A large and diverse group of experts participated. The experts had different backgrounds, and all had experience with either the design and implementation of logistical information systems, the inner-workings of actual operations and processes within logistics, or more managerial aspects. Furthermore, several researchers participated that have been working on similar or slightly different concepts and systems. The answers received on the forms were grouped and clustered together, and frequencies of occurrence were counted. The answers have been left out of this Chapter, but are included in Appendix K.

5.6.5 Direct feedback on the prototype

Several participants showed their appreciation for the prototype. They appreciated the underlying MAS principles of having a system composed of autonomous agents that derive a solution through communication and negotiation with other agents rather than, as one participant formulated it, "*uninformed optimisation*". Someone else mentioned that MAS is "*well connected to the manual way of working, with local decision autonomy*". A third participant formulated it this way: "*I like this very much; human planners cannot handle the dynamics and complexity of the (current) reality*".

Several critical comments were made. Someone remarked that it is solely a limited prototype, and far from a real system. A list of missing aspects in the current prototype was created. Mentioned was to integrate smarter algorithms in the system; to include opening times of terminals and customers in the decision-making process; and to find a structured way to integrate knowledge of human planners in the MAS. Someone else mentioned that the prototype has a highly operational/executional nature: "*How does this prototype connect with the strategic and tactical levels of planning?*" Similarly, a critical note was made that we should better make clear how this approach differs from other systems doing real-time optimisation. Someone else mentioned that real advantages are only to be realised when companies start collaborating; "*Despite the fact that the shown technology might be a platform for this, the ease of implementation is questionable*." This implied that it is an organisational behavioural issue rather than purely technical.

As concrete extensions for this prototype, several participants mentioned extensions within the supply chain: such a system should make it possible to connect with terminals and customers, and dynamically negotiate timeslots for pickup and delivery. Another extension mentioned was a connection to electronic marketplaces, where agents (trucks/drivers) can automatically find orders. A third possible extension was to include a driver-agent in the system: currently it is the truck-agent who decides, and the driver lacks autonomy. Fourth, the system should calculate and compare several decision scenarios, which it presents the planner, rather than making an autonomous choice. Fifth, although the system does include a multi-attribute decision-making process, human planners include many other factors, too. In response, one participant remarked that "*everyone always tells that systems cannot replace planners with 25 years of experience, but I am convinced that a good system can do at least as good*." Not everyone agreed with his statement.

Several participants pointed at the importance to test the prototype in practise. A test in practise will provide many other insights. For a complete overview of all feedback to these two questions we refer to Table L.2 and Table L.3 in Appendix K.

5.6.6 Feedback on question "How to support the planner"

On the second question, how a multi-agent system should support the planner, many comments were received. First of all, general comments that hold for any planning system, such as "*the system should ease the planning task*", and "*the planning decisions the system makes should be reliable and correct*". Several participants mentioned that it is nowadays important to have systems facilitating real-time decision-making.

Several participants remarked that MAS-based planning "*should facilitate management-by-exception. The planner should focus on problems solely (80/20 rule)*". A connected issue is that "*the planner should move up a level and work more on strategic/tactical aspects of planning; the agents can do the operational decision-making*". Someone mentioned that the "*planner should be able to overrule an agent's decision; in turn, the agent should learn from this for future decisions*". A fourth issue mentioned was that "*in the background, agents should continuously search for possibilities and opportunities*".

With respect to the user interface, several opinions were heard. One participant mentioned that agents should not present the planner alternatives, rather just replace the planner, since agents do better than humans anyway. Nevertheless, most workshop participants see a strong role for human planners in future multi-agent systems. Agent feedback in the form of multiple (ranked) alternatives to the planner, was frequently mentioned. Argumentation of these decisions, through visualisation and dashboard functionality (including meta-information, decision impact and statistics) are perceived useful and important as well. For a complete overview of feedback to this question we refer to Table L.4 in Appendix K.

5.7 Evaluation IV: field-test

5.7.1 Introduction

A fourth evaluation method we utilised was a real-life confrontation with the environment the prototype was developed for. Together with Post-Kogeko, we decided to test the prototype in the daily practice of container planning, and analyse its workings. Testing a new technology within a company is a useful idea, however, it is not evident how to do it, and what to measure. What should we measure is in turn linked to: What could we actually measure? Performance? Potential impact? Or, does it solely deal with planners' perception, planners who perhaps even perceive of a system as a potential threat to their daily jobs, and thus might be highly critical? Use of technology? The perspective of management? Serious questions, and serious concerns, without crystal clear answers.

We decided to aim for an evaluation session, in which we would connect live to the Post-Kogeko systems and let the multi-agent system run in parallel with the normal planning process. In order to do so, the simulator-based version of the system had to be rebuilt into a version that was able to run in parallel to the real-life QFreight and CarrierWeb systems. Also, the agents lose part of their autonomy, since the planner keeps control over the final decision. As such, human interaction with the agents also became an issue.

5|7|2 Technological differences, changes in the mechanism and lessons learned

Table 5.3 already revealed that the real-life version of the system consisted of fewer agents than the simulator versions – several agents perform simulator specific tasks, which are not needed in a real-life version. Constructing the field-test ready version, we had to remove the simulator agents (SimAgent, SyncAgent and TimeAgent), and connect to the real-life systems. As a first step we decided to construct an in-between version: a version of the system is operated manually, but runs against a simulator. The main reason for this was that debugging the system becomes difficult in the field-test version. See Figure 5.12

In the manually operated designs the system no longer fully autonomously responds to changes; it becomes now entirely planner operated. That means that planners press the "find order" button. In the real-life version they also (manually) change the availability status of orders, and change the truck status when a truck changes its state. The system resides passive until a planner fires an event. In case of a FindOrder event, the planner waits for a list of recommended orders before making a choice.

The first time we pressed the button, it worked, but the response times were dramatically long, especially for large instances. The waiting time between pressing the button, and getting feedback on the screen was unacceptably long, running the system from Eclipse at Development Machine 01 (see Table 5.15). . The small instance sample went pretty well, but testing the larger instance (with 52 orders and 26 trucks) the response was dramatic: 48 seconds passed between firing the button and the result on the screenAt first we thought that nothing had happened, since no single feedback was fed back through the interface, giving us a signal that something was happening. That let us decide to introduce a status bar, which showed that the button was fired and a response was being generated. Nevertheless, 48 seconds was still unacceptable, even with feedback from the status bar.

We started a search for possible fixes. One fix could be to change the calculate score mechanism, for example by leaving out the consultation round to the other TruckAgents. This was a change that was easy to make, just commenting out several lines in the FindOrder behaviour code, and resulted in faster feedback. The results from five attempts now came back within (on average) 10 seconds.

However, this was not the proper way to go, since the evaluation of alternative options, namely, the consultation of competitor trucks, was one of the core features of the design.

Hence, we had to look in more detail at the FindOrder and CalculateScore mechanisms. The amount of calls made to the RoadNetwork class turned out to be one of the main problems. Each call resulted in a route calculation in RoadNetwork. Analysing the CalculateScore mechanism we found that each call to CalculateScore resulted in 12 calls to RoadNetwork. These operations are relatively time-intensive, since they do a distance or time calculation (over the route network) or a slightly less time-consuming node conversion. Furthermore, the system utilises only one single (synchronised) RoadNetwork instance, which means that all operations have to take place sequentially. The RoadNetwork class needs to be synchronised otherwise it crashes. An increase in the amount of orders leads to an explosion in the amount of calculations. As Table 5.15 shows, the amount of RoadNetwork calls in our initial scenario (for 26 trucks / 52 orders) was (an estimated) 8424. It will surprise no one who has ever worked with an LP-based route planner that this is a time consuming activity – especially also when all calls have to be handled sequentially.

Analysing the total set of RoadNetwork calls, we found that several calls were made twice, and that many calls were not truck-order combo specific, but only specific to that particular order. The distance or time from the PickUp to the delivery and back to the Return location is, in principal, the same for each order independent of the truck that executes it. Realising this, it was an easy step to let the OrderAgent calculate these route details (distance, times and location nodes) once, and let the TruckAgent contact the OrderAgent to get the details when needed. This way, we were able to reduce the amount of RoadNetwork calls in the CalculateScore mechanism to only two calls when fired. This resulted in a factor of six fewer RoadNetwork calls.

Two other improvements we made concerned a reduction in the amount of interactions between agents. In the initial mechanism, for each CalculateScore event a TruckAgent contacted an OrderAgent three times with a request for certain order details (from the different subscore classes). Likewise, the TimeAgent was contacted twice (even for the same information!) from the separate score classes. We decided to contact each agent just once, and then pass the proper parameters to the subscore classes that need this information for further calculations.

Last but not least, we went through all code, and removed and optimised the code further by removing unnecessary parameters, parallelising execution of code, et cetera.

In Table 5.15 we give an overview of the impact these changes had on the (average) response times after a FindOrder event was fired from the UI. The amount of CalculateScore calculations can be estimated with Equation 5.3.

$$2\left(n + \sum_{i=0}^{n} \mathrm{i}\right)$$ **(where *n* is the amount of TruckAgents in the system)**

Equation 5.3 – Estimated amount of CalculateScore calls

This is an estimation, since the exact number depends upon the amount of TruckAgents and OrderAgents in the system, the amount of orders in execution already, and the relative position from the initiator TruckAgent as compared against the participant TruckAgents. The amount of OrderAgents we estimate on 2n; since the average amount of orders in the system is twice the amount of trucks. For the amount of interactions of the participants, the sum factor, we estimate twice the sum ofi from 0 to n, for the reason that participants are often no better than the initiator, and the participants try again the next round on the following bid.

Table 5.15 shows that a leap in performance, e.g., faster response times, was the result of the optimisation of the scoring mechanism. It taught us that even in times of very fast servers and desktop machines, writing efficient code that only calculates the same thing once and then passes it on, is still important. Optimisation of the mechanism brought us a response in less than twenty percent of the original time. The only thing we could not immediately improve upon was the browser-side response time: the building of the response grid on the screen. Note, however, that the times measured are the response times on the Eclipse-executed code on the development machine, not on the web server. It is known that GWT Java-JavaScript conversion is relatively time consuming in GWT hosted-mode. Hosted-mode lets your application run as Java byte code within the Java Virtual Machine (JVM), which allows debugging of Javacode (instead of hard-to-interpret JavaScript errors). The disadvantage is that all Java-JavaScript conversions consume a relatively large amount of time. Running it on the development machine however was the only mechanism to monitor – through a `System.out.println()` message – the end of the Agent calculations, and thus measure response times for the UI response. The UI delay is less an issue when deployed on a properly installed webserver, however, we are sure that improvements could be made in this code as well.

5|7|3 Practical changes made for the field-test

To get the system working in the Post-Kogeko practise, we had to make additional changes. First of all, we had to connect the system to the real-life databases and servers at Post-Kogeko. The order and truck details had to be written into the respective OrderAgents and TruckAgents at setup from the QFreight database. Second, the truck-movements could be retrieved from a live connection to the QFreight system, which in our practice became a Microsoft Access database at our own server, which was held up to date through a SOAP connection with the CarrierWeb webserver. Each TruckAgent could poll its position in the database.

Table 5.15 – Response times on FindOrder event (all orders are still available)

Simple scenario with 26 trucks / 52 orders	Initial mechanism	Only initiator score mechanism	Improved mechanism
Estimated number of CalculateScore calculations	702	52	702
Amount of RoadNetwork calls	8424	624	1404
Amount of TruckAgent – OrderAgent interactions	2106	156	702
Amount of TruckAgent – TimeAgent interactions	1404	104	702
Total duration FindOrder	48 sec.	10 sec.	9 sec.
UI delay (building the grid)	3 sec.	3 sec.	3 sec.
FindOrder mechanism duration	45 sec.	7 sec.	6 sec.

Several practical problems had to be solved in order to prepare for the field-test. To mention a few: up until now, we had worked with a relatively standard orderset: each order had precisely one pickup address, one delivery address, and one return address. However, different types of orders are also present in the daily practise (although less common): the so-called “overrijders”. These can be divided in three types: (1) Inter Terminal Transport (ITT) which is a transport from a terminal to the container scanner (from the customs authorities) and back to the terminal again. In principal these orders are not planned by the Post-Kogeko container planning, however, the orders are visible in QFreight; (2) Offhires: a long list of (empty-) containers to transport from one terminal to the other, not planned by the planning; one order number can contain up to one hundred different containers to be transported, which can be spread over a longer period of days or even weeks; (3) Offhires that became high-priority: an offhire order that has to be transported at a specific date, booked as a normal order in the system, with the difference that it only contains two locations: a pickup and return location. Only type (3) is planned by the planning and should be included in our system.

Additional issues we observed were very practical in nature. First, it turned out that not all container trucks had at that point been equipped with CarrierWeb; these could not be traced in the system, and were positioned on our map at geographical position 0”, 0” (which is west of the African coast). Second, we ran into a problem with our (AND) route planner which could only handle Benelux locations. In principal, 98% of the trips take place within the Benelux, but sometimes trips go to France, Germany or the UK. The system was not able to handle these; we made the practical choice to neglect these trips. Third, we noticed that several orders in the systems were not information-complete at start up; often the return terminal was not yet known, and an address code "*Onbekend*" was included. Here, we pragmatically chose to use the pickup location as the return location instead. Fourth, not all customers and terminals have the right address information in QFreight. One problem we observed with large frequency is that the address information concerns the streeta ddress and zip code of a P.O. box instead of the physical location. A zip code of a P.O. box always has the form of ..00. The main problem herewith is that route planners cannot handle such zip codes. We handled this in two different ways. For the most frequent customers, we manually checked the zip codes in the database, and manually updated these codes. For the less frequent and new customers, we chose to pragmatically change ..00 into ..11; a zip code that always exists. Nevertheless, this brings in an error. For example: everything within the greater Rotterdam area that has a P.O. box zip code 3000 then pinpoints to 3011; which is a zip code in the city centre – generally not the location of customers nor terminals.

5|7|4 Methodology

Sitting down with the planners, we utilised the Thinking-Aloud technique to gather evaluation data. The Think-aloud technique is pretty much what it sounds like. Someone is asked to do a task, and to think aloud about what he/she is doing. This method makes it possible to acquire insights and feedback that is difficult or even impossible to reach through other methods (Nielsen *et al.*, 2002). The classic source for the think-aloud technique is the work by Newell and Simon (1972). Today, it is the most common method in use in the Human-Computer-Interaction field – see among others Wright and Monk (1990), Nielsen *et al.* (2002), and Yang (2003). Wright and Monk (1990) have proved that the think-aloud methodology not only is effective as such, but also is a useful method to be used by the developers of a system or prototype with their intended users. Developers have turned out to be inaccurate in predicting what problems they will observe in the evaluation. As such, using this evaluation method is a useful instrument in a design cycle, even when used by researchers involved in the development.

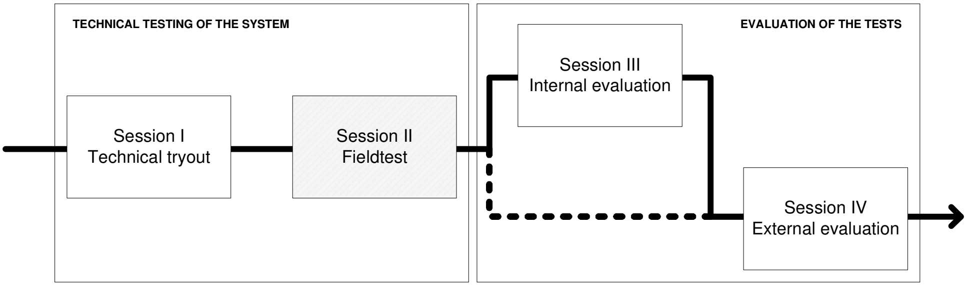


Figure 5.17 – The different sessions related to the field-test and the evaluation

Our field-test and follow-up workshops involved four sessions on different days spread over a period of two weeks' time, in the spring of 2008; this is graphically depicted in Figure 5.17. In the first session, we tested the technical functioning of our prototype. In the second session, we spent an entire day on the floor, sitting next to the planners – the system ran parallel to the planning, and advised when a plan decision had to be taken. Note that we did not start at start of operations, which is at 3:30 in the morning, but only at 8:45 when the planners arrived. The so-called "*first-work*" is always planned in advance. In the third session, we performed an internal evaluation session, in which we discussed the on-floor event (which had taken place the day before) and prepared for the fourth session, one week later. For the fourth session, we invited chain partners from Post-Kogeko (both up- and downstream) and discussed with them the results of the test, the prototype as such, and thoughts for the future.

5|7|5 The tests itself

When the test day began, we initialised the system. We double-checked the settings in the configuration file, and started the agent engine from the UI, which resulted in the generation of the ManagementAgent, which in turn first generated the OrderAgents and than the TruckAgents. These agents got their data from the QFreight and CarrierWeb databases. Since QFreight does not include as many different order-states as we do, and because often order-states change too late in QFreight, we decided to manually synchronise order-states. The same held for order availability information. Changing GPS positions from CarrierWeb were automatically fetched by the TruckAgents.

A day like this generates many results across several categories. First of all, such a test is an instrument to find bugs and identify missing features in the prototype. Second, and more important from the perspective of our research, is the question of whether the prototype and its concepts as developed hold in real practise, and whether the planners who have to work with it "buy it". Third, it is a moment to discuss future extensions to the prototype and its underlying concepts.

During the tests two researchers (also the developers of the system) were present. Notes were made by both, and shared and compared afterwards. The user interaction (of the planner) with the system, the plan experiences, and interaction about pros and cons of the system were all documented. Two planners participated in the test; one senior-planner in the container unit, with 21+ years of experience, and a planner with 8+ years of experience (of which only a couple of months had been spent in container planning).

During the tests, a couple of bugs and a longer list of missing features were collected. These are included in Appendix L. Some of the missing features were due to design or development choices we made earlier in the process. Examples of the latter are, for example, the list of off-hire orders we left on purpose out of the system, the Benelux route planner we chose to use (since that one was freely available to us), and the fact that a standard order comes with three addresses (but in practise, sometimes more or fewer addresses exist). Another example of an aspect we neglected in our design was the fact that the Coolboxx 45ft containers are longer than the standard 40ft containers and thus need a different chassis; however, these can be transported by a multifunctional chassis that most trucks are equipped with. In practise, however, it turns out that specific trucks "specialise" in the transport of 45ft containers. A last issue to mention is that we made a simplification so that an order is always executed by precisely one truck. However, it is possible that orders (with chassis) are dropped somewhere – generally at Post-Kogeko headquarters – and later on transported to a customer or return terminal by a different truck.

The feedback received on the working of the prototype was very positive. In Table 5.16 and Table 5.17, we included a selection of feedback received from the two planners involved. In previous contacts, we always had interesting and challenging discussion with senior-planner Ben van Zeijl, who was very sceptical about our attempts to construct a multi-agent planning system from the research's beginning. Having gone through the briefing and the test itself he turned out to be more positive than we expected. Some of the comments he made have been clustered in Table 5.16, the underlying theme is that looking beyond the limitations of this first prototype, he did recognise a future for such systems. A system that keeps up-to-date with internal systems, and proactively advises with proper feedback is very much welcomed. Furthermore we noticed several times Ben's response: "O*h yes, that would also be an option, I haven't thought of...*" in response to the system's suggestion.

Also, the planner Arno Pieper provided positive feedback. Arno was not familiar with the prototype, the design, or the principles of MAS before the test, but was positive about what we tested and evaluated with him. Arno has been recently appointed to a new job, in which he is responsible for better balancing the fleet of Post-Kogeko trucks, in order to avoid unnecessary empty trips. He reflected the experiences with the prototype not only on the container planning, but also on his work in this new job. He recognised a wider application of the prototype on the Post-Kogeko corporate level, over different units: "*It would be very handy to have an overview of all Post-Kogeko trucks, also from other (than container) units, and get signals from these trucks when they are empty, and on what trajectories they are on*". Arno furthermore reflected on the use of traditional OR-based planning solutions, with which he worked in his previous job at Post-Kogeko in Zoetermeer. "*In traditional planning systems, planners hardly ever press the automatic planning button due to too many hard restrictions, and the feeling of being out-of-control*". Rather, he appreciated "*a system that thinks along with the planner and gives advice would be very handy, as this prototype shows*." Some of his other feedback is clustered in Table 5.17.

The planners especially appreciate the integral view of the system (which integrates all information in one screen), and the proactive features to keep itself up-to-date. Also, the fact that the system runs in parallel to the planning and the existing systems is perceived very valuable. The truth is, of course, that planners do not really care whether a system is a multi-agent system or not. They want a system that offers a solution to their daily problems, by automating routine work, and providing assistance in the planning tasks. Nevertheless, we noticed that they appreciated the system's real-time character, the proactive behaviours, and the underlying solution-finding through negotiation and communication.

To give an example of the latter, extensions of the prototype to terminals and customers were mentioned as potentially very interesting. Arno Pieper for example stated that "*often customer requirements are not as hard as one thinks. When one contacts a customer to ask whether an order is really needed tomorrow, or at a specific time, it often turns out that it can be changed – systems generally do not include this, but the approach here shown might enable this… perhaps even through automated negotiation with customers*".

Table 5.16 – Feedback on the prototype, field-test planner I

Experiences Ben van Zeijl (Senior Planner containers, 21+ years at Post-Kogeko)
"It is clear that this prototype does have a long list of limitations and issues that still have to be solved"
"Keeping a system up-to-date and synchronised with the other systems should not consume time, otherwise it is not of use" [please note that we choose to synchronise manually to avoid unexpected errors]
"System should be overrulable"
"System should run in parallel to the planner, think along with him, and proactively come up with suggestions"
"Concerning the score feedback [after pressing Find Order button]: sorting on empty miles score is most important"
"Concerning the score feedback [after pressing Find Order button]: the other scores are less important, but very handy to have and to include in the decision-making process"
"The CustomerTimeWindowScore especially plays a role for the first set of orders, not later during the day"
"Most of the time the advice given by the agents is very usable, except for the fact that some restrictions are not yet included in the system"
We noticed that several times Ben's response was: "oh yes, that would also be an option that I haven't thought of" in response to an assignment by the system
"The baseline as tested today is interesting; of course it is not perfect, however the concepts are very promising, and I am looking forward to see such smart systems make the step from the lab to our company floor…"
"In a follow-up test, we should try to avoid the long setup and manual synchronisation times, which would make it possible to focus more on the assignment steps and process, and really run every decision in parallel"

Table 5.17 – Feedback on the prototype, field-test planner II

Experiences Arno Pieper (Planner containers/overall, 8+ years at Post-Kogeko)
"It is interesting and useful to get back several criteria that orders are ranked upon" *"Most suggestions the system makes definitely make sense"* *"A system that thinks along with the planner and gives advice would be very handy, as this prototype shows. In traditional planning systems, planners hardly ever press the automatic planning button due to too many hard restrictions, and the feeling of being out-of-control"* *"I prefer a system and a planning workflow that is as digital and electronic as possible"* *"Ideal would be a system that bundles all planning-related information in one screen"* *"Often customer requirements are not as hard as one would think. When one contacts a customer to ask whether an order is really needed tomorrow, or at a specific time, it often turns out that it can be easily changed – systems generally do not include this, but the approach that you show might enable this… perhaps even through automated negotiation with customers"*

Richard Crans made it clear how important an integrated overview and visualisation is for the planning practise. He illustrated this with the example of the CarrierWeb service, which was down for two days in the week before our test: "*The lack of overview resulted in two far less efficient days, with more inefficient trips, et cetera*."

In conclusion, we look back at a successful test. An overview of conclusions drawn about the test is given in Table 5.18. In short: the technology did work well, ignoring some minor bugs, and the system was received with much interest. Although the planners, in principal, do not care what type of technology they work with, they appreciated many of the aspects in our multi-agent system design. The assistance in the assignment process was appreciated, and worked relatively well. The UI, and interaction with the agents was clear, and especially the integration of the GoogleMaps widget was welcomed.

5|7|6 Discussing (chain) implications in two follow up workshops

The field-test was followed by two workshops. The first one was internal within Post-Kogeko. The second one included participants from some of Post-Kogeko's supply chain partners. Representatives from the two largest container terminals in the Port of Rotterdam were present in this second workshop: ECT and APM terminals. Furthermore, a representative from CoolControl, Post-Kogeko's number-one container-customer, participated. Also present was someone from PortInfolink. From Post-Kogeko itself, four people were present, and three researchers from Erasmus University and the University of Twente. For the complete list of the thirteen workshop participants, we refer the reader to Table N.1 in Appendix M. Below we give a short summary of the workshop.

Table 5.18 – Conclusions field-test at Post-Kogeko, Spring 2008

Conclusions summarized
The planners do not really care if it is a multi-agent system or not. They want a system that comes with solutions to their daily problems, by giving assistance to the daily job, automating routine work, and providing assistance in the planning tasks.
The prototype as demoed did come with some minor bugs, and it was not feature complete. The session, as such, resulted in a long list of suggestions for additional features. The planners evaluated the features that were present, independent of missing features, and tried to solely consider underlying technologies and concepts.
The concept of a smart system that runs in parallel to the manual processes, and comes autonomously with suggestions for assignments was very much welcomed.
The multi-criteria order assignment mechanism, as was implemented deliberately, resulted in reactions such as, "*Oh yes, that would also have been an interesting option*"
In daily practise, multiple decision criteria exist, however, be aware that there are, for example, no fixed narrow time windows, or pertinent rules that some customers are always served faster. Most constraints are pretty flexible.
The largest troubles currently exist at the chain level: late customer changes, little information from (and interaction with) the terminals, long queues. This prototype could be a basis for a further integration with the up- and downstream supply chains, and help in automatic checking, updating and perhaps negotiating. This would be a real relief.
From a technical level, the test was a success: all agents kept working properly, and the UI (the part the planners interfaced with) made it easy to interact with them.
From a research perspective, this test was also very interesting, however, it should not stop here. The interesting challenge remaining is to make the next step towards real implementation.

The session started with an introductory round in which the participants briefly introduced themselves and their functions. This was followed by a short presentation and demonstration of the MAS prototype and its underlying concepts by the researchers. This was followed by a short reaction on the test by one of the planners who participated, Ben van Zeijl. The roundtable was opened through an open question raised by Frans Denie, general director Post-Kogeko: "*How could the chain partners, here present, collaborate better in their daily interaction processes*". This question was inspired by the ideas for coordination-driven optimisation as present in the prototype, and the opportunities identified by the planners working with the prototype in the test. A basic chain interaction drawing, Figure 5.18, supported the question and opened up the floor for discussion.

The invitation was well received. The two hours that had been planned for the workshop were hardly enough. Most parties present in the session do normally not discuss the impact of new technologies or approaches with each other in such a concrete way, despite the fact that they are partners in the supply chain and strongly depend on each other.

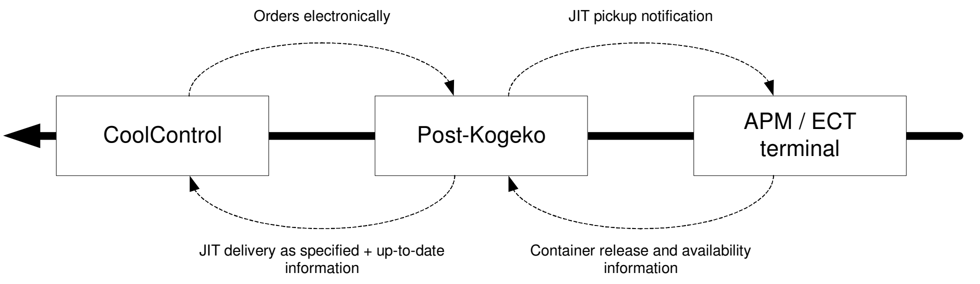


Figure 5.18 – Basic chain interaction diagram

It was identified that the terminals are further with their information technologies than most LSPs know; only a small amount of LSPs use that to the full extent. At the same time, the terminals have troubles handling trucks in time, and face large arrival peaks. In response, they have started working on initiatives such as time-slotting and remote check-in to streamline truck arrival, and to better balance truck handling. PortInfolink offers applications to support coordination in the port on multiple levels; unfortunately many parties use the existing technology wrong. But, not all processes have been supported well, as the example given by ECT on the container pre-announcements shows: "*The pre-announcement comes both too late as well as too early,*" according to Tom Niels. "*It is too late for sorting containers coming from the ship, it is too early (and unprecise) for scheduling pick-up by trucks*". Coolcontrol did realise during the roundtable that their behaviour causes continuous crisis management for their LSPs and the terminals. Coolcontrol knows already three weeks in advance what container is coming when. However, by only starting communications in the last days or even hours, there is nothing left to streamline operations with.

Many issues were identified that could be largely improved upon through chain-wide layered information exchange. A compilation of these issues, non-conclusive, is listed in Table 5.19.

One of the day's main conclusions was that the chain as a whole would thrive after information exchange and coordination at multiple planning levels. Several weeks before the actual arrival of the containership, an initial exchange of information could take place: identifying the customer for a specific container, the expected modality, and expected LSP. When the actual pickup approaches more detailed coordination – on container availability, customs handling, and eventually an announcement of a specific visit – should take place. The latter should also be in multiple steps, the last step, for example, 15 minutes before actual arrival at the loading dock, so the terminal could already start container handling. The current way of working results in many last-minute operations, leaving no space for any optimisation.

The participants agreed that this workshop should not remain a one-time event only. Everyone present found it very useful. This should result in follow-up actions, perhaps in the form of Post-Kogeko's participation in a test with APM's truck-time-windows scheduling, or a future research program following the work done within DEAL and the Transumo Diploma project, as was suggested by Professor Jos van Hillegersberg.

Concluding, we can state that this research project that began with the development of a multi-agent system prototype for a single-enterprise planning environment, eventually triggered discussion on the application of such technologies (and novel processes) in the wider supply chain. The prototype and its real-world test became an instrument to discuss future application for coordinated road planning in the port. "*To be continued…*"

Table 5.19 – List of identified issues that result from wrong information exchange

Party	List of issues
Terminals	Long queues of waiting trucks Large amount of unnecessary repositioning of containers over the terminal Unbalanced capacities with high peaks Destination and modality of containers arriving unknown (wrong stacking)
LSP	Long waiting times at terminals (on average 30-45 minutes before container is placed on a truck – terminal waits for truck to get there) Late arrival of orders Rush orders become the standard (although containers have already been aboard the seaship for many weeks)
Customer	Price for service is premium (due to inefficiencies of other parties in chain) Every order seems a rush-order, but many are not

5|8 Discussion

This chapter describes a design research project we worked on over the past few years. The design was made for a multi-agent system for the real-time planning of container trucks at Post-Kogeko, initial problem to tackle was the real-time coordination of activities. Post-Kogeko was involved in the design trajectory. The design was turned into a prototype system utilising a set of different technologies. The agent engine was constructed utilising the JADE toolkit. A user friendly UI has been built utilising GWT and GoogleMaps.

Prototyping the system provided hands-on experience in not only the construction of the system, but also the true conceptually different thinking behind multi-agent systems. We documented the most important decisions made in the design throughout this chapter, one example being the introduction of the OrderAgent. Several versions of the prototype were developed, most importantly a fully autonomous simulator version, and a human planner-controlled real-life version. Unfortunately, time did not allow us to construct a true cross-enterprise application in which we would also employ TerminalAgents and CustomerAgents.

The prototype(s) has been evaluated and validated through different methods – following the recommendations from the previous chapter (Chapter 1). First of all, the design process, as such, proved to be an important evaluation approach – something mentioned earlier by Nunamaker and Chen (1990). Second, we tested the prototype in a simulation environment, in which we compared its performance against two other prototype systems. Third, we organised an expert evaluation session, in which we discussed the system, its design, and the visions behind it with a group of industrial experts. Fourth, we performed a field-test with the prototype in the Post-Kogeko practice.

Honestly speaking, from an optimisation perspective, we did worse than the optimisation-based approaches we compared with in the simulation – see section 5|5. However, this did not surprise us, due to the setup of the simulation experiments. The test setting was too artificial for our prototype. It lacked disturbances that profit from active planning mechanisms. The "*fit with the time window*" was the most important evaluation characteristic in the experiment; whereas we focused on keeping the trucks in our system busy – often resulting in orders that arrived (a little) too early. In the Post-Kogeko field-test, we found again that in practise time windows are important only to a limited extent, efficient use of trucks and and avoidance of empty kilometres is perceived as more important.

Our evaluations of the prototype with practioners showed that industry professionals are intrigued by the idea of having information systems that find solutions by utilising real-time information, and coordination with other (internal or external) systems. End-users of planning tools, or logistical professionals, are not *per se* interested in multi-agent systems as a technology, rather in its functionality and potential. The concept of solving a logistical problem through real-time coordination and negotiation is, however, welcomed. Which exact technology to use in the system, the user seems to hardly care. Krauth (2008) found that users are influenced by the type of feedback they get on the inner-workings of a planning system, when the system performs its calculations. We cannot confirm nor falsify this, since we did not test these aspects in our research. Rather we have ground to believe that end-users, who are no technicians, are relatively quickly impressed by new technologies and concepts behind a certain system; independent whether these are agents, objects, or Web services, to name a few. A user-friendly graphical user interface with a nice look-and-feel can easily give them the impression of a state-of-the-art system.

This research project did not result in a concrete implementation. Nevertheless, we tried to research and discuss future implementation in our different evaluation stages. The list of factors identified by Bold (2005) identified chain collaboration already as the main source for further cost savings for LSPs. Both the field-test as well as the expert evaluation revealed that the true potential is expected for this type of systems in chain interaction. Utilise agent concepts to coordinate activities with up-and downstream supply chain partners. At multiple levels of planning. However, this is easier said than done.

The research in this chapter concentrated on the fifth sub-research question – "*Can multi-agent systems contribute to better performing, and easier implementable systems for transportation?*". As with any (design) case-study, it is hard to generalise our findings (Eisenhardt, 1989; Yin, 2003). Nevertheless, this chapter showed that multi-agent systems are an interesting instrument for real-time and inter-organisational coordination problems in transportation; especially the frequent changes, and need for last-minute reactions make MAS a promising approach. Real implementation is a necessary next step to investigate the claimed easier implementability and gain better insight into performance compared to other approaches.

Chapter 6 Barge rotation planning

6|1 Introduction

With the increase in container flows around the globe (Levinson, 2006), the Port of Rotterdam faces a massive increase in the number of containers shipped through its terminals every year. Nowadays, barges account for around 30% of the transport to the hinterland. In the port, barges make rotation trips: a visit along several terminals where they unload and load containers. Since barges are competing for the same (limited) capacities at terminals, and visits to terminals are depending upon visits to other terminals, planning and coordination of barge rotations is necessary. With the increase in ship movements and container flows, coordination between barges and terminals becomes more inflexible every day.

This chapter discusses this case, and a research project aimed at solving the coordination problem through multi-agent system architecture. In this research, sub-research question five was: "*Can multi-agent systems contribute to better performing, and easier implementable systems for transportation?*" An important difference with the case in the previous chapter is that this setting deals with the design of an Inter-Organisational System (IOS) for a network of companies. It lacks a central firm, and is positioned and initiated as a system for a network of competitors. This is in contrast to the IOS discussed in the previous chapter, which was initiated within one central firm.

The methodology utilised throughout this chapter is similar to the methodology followed in the previous chapter. One important difference is that our involvement in this research started with an industry workshop (see section 6|2), which aimed at understanding the problems as they exist in practise. This workshop concluded research done in a previous project, entitled APPROACH. Data analysis of the available transactional data was the next step (6|3), followed by a design (6|4) and evaluation of different scenarios in simulation (see section 6|5). Industrial parties were involved throughout the entire process for input, validation, and the provision of feedback. Towards the end of the research, an expert forum provided feedback on our thoughts – see section 6|6. The chapter is concluded with a discussion section (6|7) which addresses issues open for further research. The work in this chapter is only a subset of work done within the larger research project – for more details we refer the reader to Van Groningen (2006), Moonen *et al.* (2007), Douma *et al.* (2008), and the dissertations of Douma (2008) and (forthcoming) Lang (2010). This chapter could not have been written without the contributions of Albert Douma and Bastiaan van de Rakt.

6.2 Manual planning of barge rotations

6.2.1 Barge shipping

The port of Rotterdam is a key container transhipment hub for Europe. Increasingly, the quality and the accessibility of the port and the port's hinterland connections is becoming a decisive competitive factor. Truck transportation was (and is) the primary hinterland connection. Since the early 1980s, however, the river Rhine has increasingly been recognised as a 'natural' connection to the German hinterland. Currently commanding a 30% market share (Havenbedrijf, 2008), inland shipping has developed into a vital hinterland connection. Although barges are not a fast mode of transport, they can be operated according to regular shipping schedules. Their success can largely be attributed to the scale of operations and the ability to operate regular on-time services. Inland shipping has become an inexpensive and reliable link in the logistics chain (Melis *et al.*, 2003; Schut *et al.*, 2004).

As a result of the spectacular growth, container transhipment capacity in Rotterdam is under pressure. Barges are handled at the terminal's quayside, using the same transhipment capacity (i.e., cranes and quays) as large seagoing vessels, placing ever-greater demands on effective and reliable planning. Another complicating factor affecting transhipment capacity planning is the fact that barges in the port of Rotterdam call at eight different terminals, on average. An average rotation time is approximately 22.5 hours, of which only 7.5 hours are used for loading and unloading. The remaining time is spent sailing and waiting. The complicated nature of the rotation planning is illustrated by the fact that, in 1998, only 62% of the barges left the port of Rotterdam on time (RIL, 1998), thuschallenging the perceived high reliability. Although more recent numbers are lacking, there is reason to assume that the current practise is worse. Container traffic in general, and barge traffic in particular has grown at a faster pace than terminal capacity (measured in quays and cranes).

The pre-planning of terminal visits – a barge's rotation – is recognised (Melis *et al.*, 2003; Schut *et al.*, 2004) as one of the key sources for supply chain inefficiencies. Currently, this planning is performed manually, one day before actual execution.

6.2.2 Controlled workshop to analyze current practice

To study the inefficiencies in the manual planning of barge rotations, a game-styled workshop was organised in September 2004. Participants were representatives from container terminals and barge-operators. The barge operators had to (manually) make a rotation planning for a list of terminals to visit and communicate their plans with the terminals. The goal of the workshop was: "*Analyse, in a controlled workshop setting, the (manual) barge rotation planning process as terminal- and barge-operators currently perform it in practise*." After the manual session, the APPROACH prototype (Schut *et al.*, 2004) ran utilising the same data, to generate an alternative planning. In the second part of the workshop, the results of the manual planning and the APPROACH system were presented, compared and discussed with the group of participants. The workshop focused on a critical evaluation of the developed concept, and investigated the possibilities for implementation and further research.

The game reflected a simplified real-world situation. Only six barge-operators and eight container terminals were used in the scenario. In total, 22 rotation trips had to be planned for a period spanning 24 hours. Each barge-operator had to plan three or four rotation trips, whereas in practise it is sometimes only one trip a day per barge-operator, depending on its size. The barge-operators and container terminals, as used in the game, are shown in Table 6.1 and

Table 6.2, respectively. The latter table also shows the number of quays to be used by the terminals in parallel, and the time-window restrictions that apply.

In the game setting, a rotation trip to be planned consists of a visit to a minimum of four, and a maximum of eight terminals. The average handling-times at the terminals are in line with practice. The sailing-times are calculated based upon an average sailing-speed of 15 kilometres per hour and the geographical location of the terminals. The geographical position of the terminal locations in the game are the same as they are in the real port. The rotation trips are based upon an old Bargeplanning 1.0 database – containing data from 2001. The database delivered the data for the 22 rotation trips. It contained, for each rotation trip, information such as estimated time of arrival (ETA), estimated time of departure (ETD), number of terminals, requested terminal order, call sizes, and the inter-terminal sailing-times. The quays of ECT Delta, ECT Home and APM are partly blocked, since in the scenario these terminals handle seaships at the same quays as barges.

The barge operators planned their terminal visits on paper cards, and these were handed over to the container terminal planners. It was one-way communication only; a feedback loop from the terminals to the barge planners was, as in daily practise, non-existant. Although all planning was done by hand and written on paper, the barge operator also entered the data on a computer screen with a similar look as the PortInfolink Bargeplanning 2.1 application, as was then in use in the Port of Rotterdam. The container terminal players fed their final planning in the screen also. Although the barge operators received this feedback, they could not rearrange their planning. Figure 6.2 shows the computer screen as used by the barge operator planners, showing the Interfeerder 1. The rotation trip, and its order, can be found in the lower part of the screen, showing that in this planning the Interfeeder 1 will visit the terminals in the following order: (1) ECT Delta, (2) HTHolland, (3) Hanno, (4) ECT Home, (5) RST. The Uniport and Waalhaven terminals are not planned yet.

The container terminal operators were equipped with a large wall-mounted planboard, showing the planning for the terminal’s quays. This included information on seaships and closing-times of the terminal. See Figure 6.1 for an illustration of a terminal’s planboard. The planners placed the small plan cards from the barge operators on the planboard. The task was to assign the quay space as well as possible, based on the requests from the barge operators. After the plan was made, the planners had to enter the plan in the database – see Figure 6.3 for an illustration. As mentioned, this resulted in a feedback signal signaling the planned start time back to the barge-operators' screen; see the last column in the lower part of Figure 6.2. All plan data was captured in a Microsoft Access database. This allowed comparison with the output data from the APPROACH run.

Table 6.1 – Barge operators as used in the workshop

Barge operator	Number of rotation trips
Alcotrans	4
BBT	3
CCS	3
Interfeeder	4
Rhine Container	4
Danser	4
Total	22

Table 6.2 – Container terminal operators as used in the workshop

Terminal operator	Number of quays	Opening times
ECT Delta	2	All day
ECT Home	2	All day
APM	2	All day
Hanno	1	All day
Uniport	1	All day
RST	1	All day
Waalhaventerminal	1	06:00-22:00 hours
HT Holland Terminal	1	06:30-21:00 hours

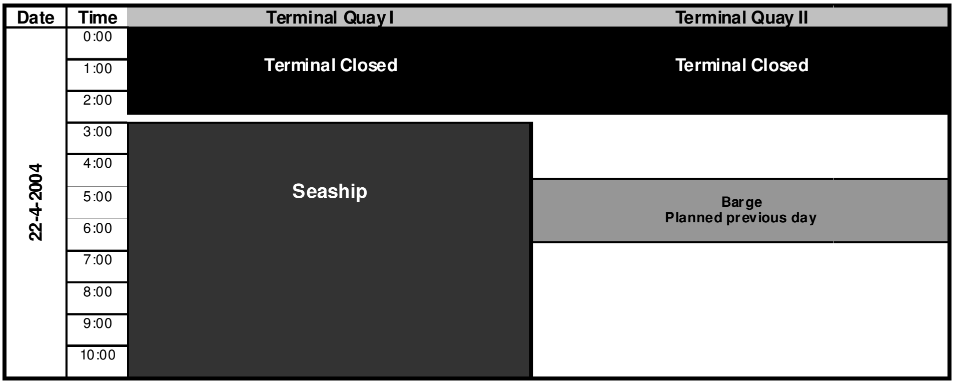


Figure 6.1 – Plan board container terminal, showing available quay space

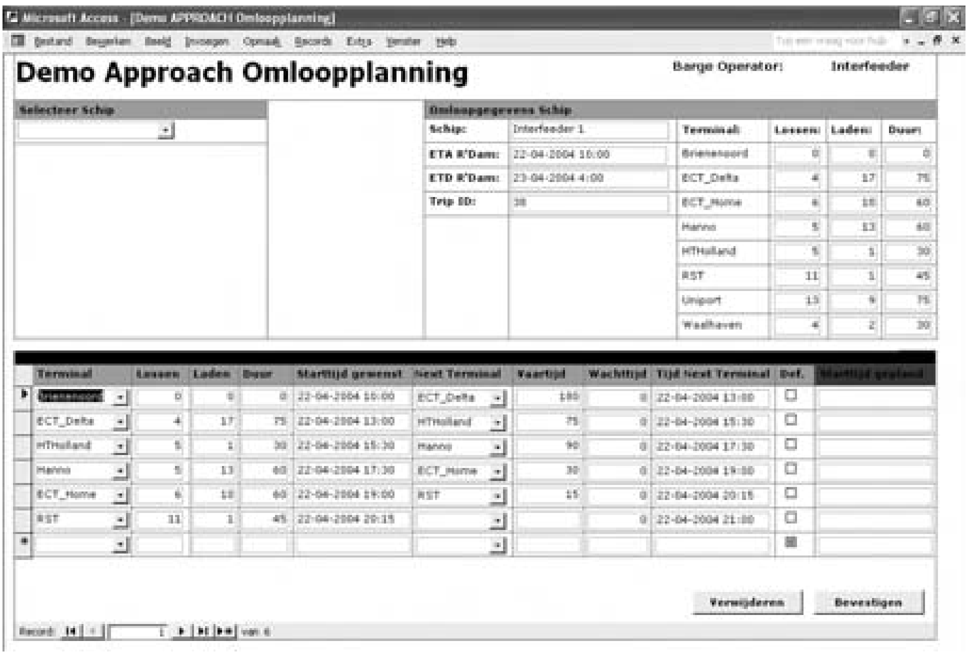


Figure 6.2 – Computer input screen barge operator planner

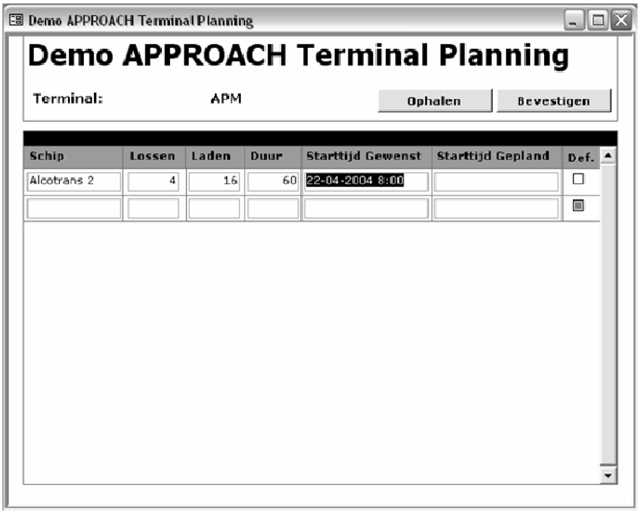


Figure 6.3 – Computer input screen container terminal operator planner

6|2|3 Workshop results

The planned rotations resulting from the manual planning process were stored in the database. Here, we discuss the example of the rotation for the Rhine I, one of the ships from barge operator Rhine Container. The Rhine I has an ETA on 22-04-2004 at 00:30, and its ETD on 22-04-2004 is 18:30. The barge therefore has a total of 18 hours for its rotation trip, and in this trip it needs to visit six terminals. Table 6.3 shows the initial plan made by Rhine Container. This plan was communicated to the different terminals, requesting time slots for visits. In Table 6.4 the confirmed final rotation plan is shown. This is the outcome of the planning by the container terminal operators. Note that there was a shift in the order of terminal visits: HTHolland is now planned before ECT_Delta. This adds an additional waiting time of 4:00 hours, since HTHolland has opening time restrictions.

After the manual planning, the APPROACH pilot system utilised the same data. Its plan consists of rotation schemes for all 22 barges. The results of the APPROACH system were also stored in a Microsoft Access database, which enabled easy comparison of results.

The manual planning often results in confirmed rotation plans that are not feasible to execute. Requested time slots are not honoured, since more barge-operators request/claim the same time slot at a terminal. The terminal can only accept one (or, in case of more quays, sometimes two), barges at the same time. The other barges are moved to an earlier or later time. As a result, barges are frequently double-booked: they are expected at multiple terminals at the same time. Terminals reserve capacity they should not have reserved: the barge can never arrive in time. A barge arriving with a delay often finds another barge along the quay, or even a queue of waiting barges. Looking at the Rhine I example (Table 6.4), several double bookings are present in the manual plan. At 08:05

AM for example, the barge is expected at two terminals at the same time: the ECT_Delta and RST terminal. The decision to be made is as follows: the Rhine I is ready at the HTHolland terminal at 07:00, however, sailing to the ECT_Delta consumes 1:15 hours. It can never be there at the planned 07:15. Nevertheless, it may consider going there, and arrive one hour late. The decision could also be to first go to the RST or Waalhaven terminal and skip ECT_Delta, and instead do ECT_Delta later on in the rotation.

Table 6.5 shows the total double-book time (per barge, per rotation trip). From the twenty-two rotation trips planned manually, nineteen trips are unfeasible due to double-bookings. Seven trips even contain a double double-booking.

The late arrival of barges at container terminals, caused by the previously mentioned double-bookings, is illustrated in Figure 6.4. Shown is the percentage of barges arriving too late at the terminal, according to the confirmed manual (pre-) planning. The figures illustrate, for example, that at the Uniport terminal, 6 out of 18 barges (= 33%) will arrive too late. There is a significant difference between the APM, ECT_Delta and the ECT_Home terminals, and the other terminals. These three terminals have (in this game setting) two quays to handle barges, whereas the others have solely one. Not surprisingly, more quays result in more flexibility in the planning.

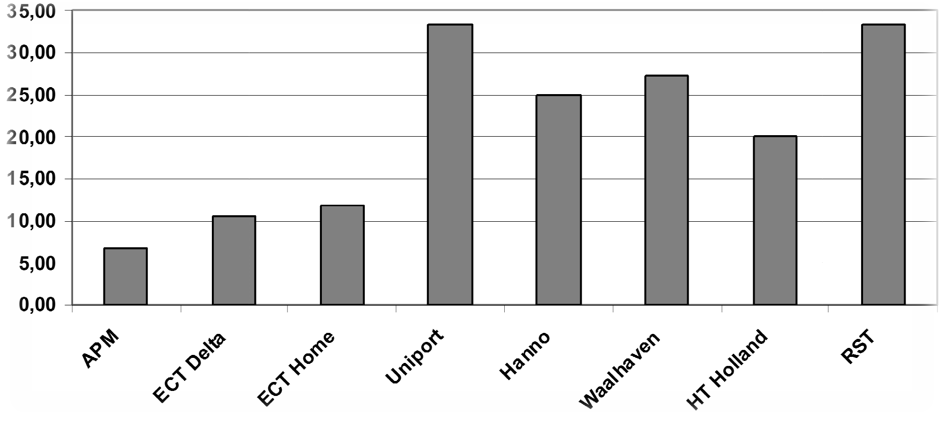


Figure 6.4 – Percentage of barges arriving too late at the terminal

Table 6.3 – Workshop output: initial manual plan for the Rhine I

Terminal	Duration	Starting time	Shipping time	Next terminal	Next arrival	Waiting time
Brienenoord	0:00	22-04-04 00:30	3:00	ECT_Delta	22-04-04 03:30	
ECT_Delta	1:45	22-04-04 03:30	1:15	HTHolland	22-04-04 06:30	
HTHolland	0:30	22-04-04 06:30	1:00	RST	22-04-04 08:00	
RST	0:45	22-04-04 08:00	0:45	Waalhaven	22-04-04 09:30	
Waalhaven	0:45	22-04-04 09:30	0:15	Hanno	22-04-04 10:30	
Hanno	1:30	22-04-04 10:30	0:15	Uniport	22-04-04 12:15	
Uniport	1:00	22-04-04 12:15	1:00	Brienenoord	22-04-04 14:15	

Table 6.4 – Workshop output: confirmed manual plan for the Rhine I

Terminal	Duration	Starting time	Shipping time	Next terminal	Next arrival	Waiting time
Brienenoord	0:00	22-04-04 00:30	2:00	HTHolland	22-04-04 06:30	4:00
HTHolland	0:30	22-04-04 06:30	1:15	ECT_Delta	22-04-04 08:15	
ECT_Delta	1:45	22-04-04 07:15	2:30	RST	22-04-04 11:30	
RST	0:45	22-04-04 08:00	0:45	Waalhaven	22-04-04 09:30	
Waalhaven	0:45	22-04-04 09:30	0:15	Hanno	22-04-04 10:30	
Hanno	1:30	22-04-04 10:30	0:15	Uniport	22-04-04 12:15	
Brienenoord	0:00	22-04-04 00:30	2:00	HTHolland	22-04-04 06:30	4:00

Table 6.5 – Total double book time for the manual planning results

Barge	Double-bookings (hours)	# of double-bookings	Barge	Double-bookings (hours)	# of double-bookings
Alcotrans 1	2:00	3	Danser 2	2:15	2
Alcotrans 2	0:45	3	Danser 3	0	0
Alcotrans 3	2:15	4	Danser 4	0:30	1
Alcotrans 4	1:30	1	Interfeeder 1	2:00	2
BTT 2	1:00	1	Interfeeder 2	0	0
BTT 3	0	0	Interfeeder 3	0:30	1
BTT 4	0	0	Interfeeder 4	0	0
CCS 2	2:30	2	Rhine 1	4:30	3
CCS 3	1:15	2	Rhine 2	0	0
CCS 4	1:15	1	Rhine 3	0:45	1
Danser 1	1:00	1	Rhine 4	1:45	1

6|2|4 Workshop outcomes

The workshop revealed that the traditional way of manual planning of barge rotations results in severe problems. It also illustrated that the first APPROACH system was capable of establishing a feasible plan with zero double-bookings.

The industry participants were surprised, but found it not entirely unexpected, to see the magnitude of problems their manual planning processes caused, and were enthusiastic about the APPROACH pilot system's results. Reactions were, among others: "*When can we have this system working? It would help us solve (part of) our business problems*" and, "*How can we contribute to further development of this system?*" The participants also vented some critical remarks about the demo. They noted, for example, that the outcome of the APPROACH planning contained illogical rotations; rotations with longer sailing distances than needed. The system sometimes scheduled rotations in an order a human planners would never allow. Several additional remarks received related to restrictions missing in the prototype. The position of human planners was a topic for extensive discussions: Will they disappear because of APPROACH, or will their tasks change? What about the knowledge and experience they possess?

The system as developed within the APPROACH project – see Schut *et al.* (2004) – was a first attempt to solve the planning problem. Looking at it critically, the system basically operated through a trial-and-error, puzzle-solving mechanism, with only a small amount of intelligence deployed (Moonen *et al.*, 2007). The system's architecture was basically a central planboard, where software agents tried to find solutions. The system kept on retrying until a satisfying solution was reached. The different parties in the system did not communicate with one another, nor synchronise activities.

As initially designed, the APPROACH system solved the problem of pre-planning, leaving out re-planning or real-time event handling.

As a follow-up to the workshop, the initiator of the project, INITI8, started the development on a software product named SYNCHRON8. In parallel, it formed a consortium of research parties to conduct further research. The following sections demonstrate some of the outcomes of the work within this new research project, which of course got a headstart due to all the knowledge present in the consortium, and the experiences gained throughout the earlier project – see, among others, Leenaarts *et al.* (2003), Melis *et al.* (2003), Schut *et al.* (2004), and Moonen *et al.* (2007).

6.3 In-depth problem description

6.3.1 Barge rotation planning in more detail

Every day, between 75 and 100 barges visit the port, and these barges visit, on average, eight terminals. There are a total of 37 terminals in the port, in 11 locations. Most terminals have closing hours and handle sea-going vessels at the same quays. Both factors limit the freedom of barges to plan rotations. Moreover, there is no contractual relationship between terminals and barges, which means that they cannot force each other to deliver at predefined service levels. Coordination is done by telephone, fax or e-mail. This is time consuming, and it is complex to handle changes and disturbances. As a matter of fact, it is often not possible to execute a rotation or quay plan as planned. The inter-organisational processes between barge- and terminals operators in the port are depicted in Figure 6.6 (adapted from Schut *et al.* (2004)). The scheme illustrates that there are three important points in time, which we refer to as 24, 4 and 0 hours before execution. The pre-planning (24H) starts a day in advance: terminals need to arrange personnel, from the shared labour pool, to operate the cranes to (un)load containers. In fact, this is not exactly 24H in advance, but between 14 and 38 hours before execution: requests for tomorrow have to be made the latest today at 10 O'clock in the morning (see Figure 6.6). For ease of readability, we refer to this as the 24H planning moment. 4H prior to processing, a barge has to announce to the terminal which containers it wants to (un)load. At 0H, the terminal faces operational decisions, such as which team processes a specific barge.

An analysis of data from 2001[1] stored in PortInfolink's BargePlanning database revealed that there are three main sources for delays in the execution phase. The first are delays due to

[1] Data from 2001 was used for the reason that that was the year BargePlanning was introduced. By then barge operators did enter a description in the system for being early or late. Although still existing, this feature is now hardly in use anymore.

additions/subtractions in number of containers (compared with the initial plan), the second are delays due to late arrivals (which are, for the largest part not further specified, but also includes late arrivals due to capacity problems at ECT), and third are administrative problems with documents, container clearing and/or loading/unloading lists.

Each individual barge rotation depends on the rotation of other barges and available terminal capacity. Barge operators are competitors and are hesitant to give competitors insight into their books, especially where it concerns competitive information. The paradoxical setting is that they need to coordinate with each other in order to smooth their operations, and avoid or reduce unnecessary waiting times. With a yearly increase in container flows and barge traffic, this can be expected to become ever more necessary.

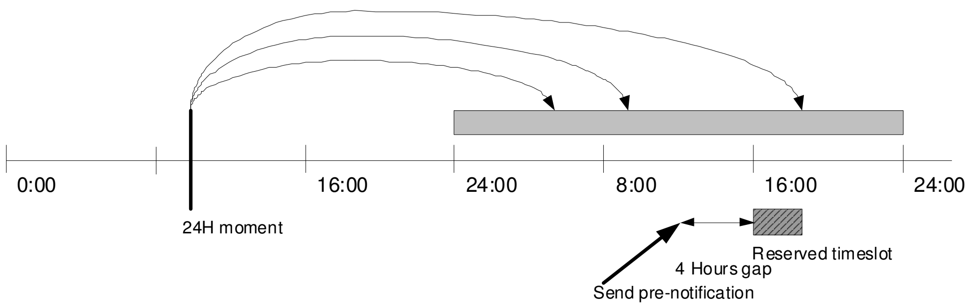


Figure 6.5 – Planning moments depicted in time

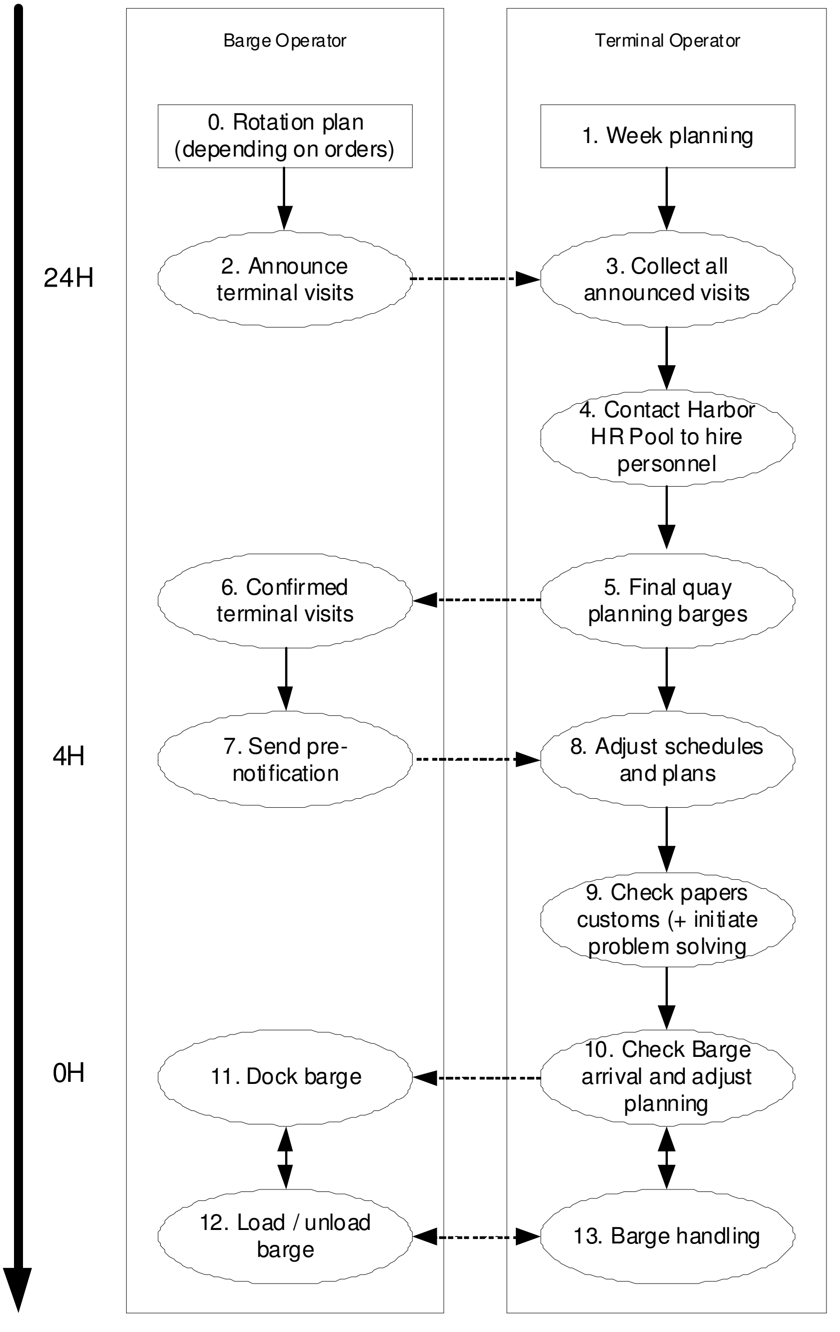


Figure 6.6 – Inter-organisational processes spanning barge- and terminal-operators

Table 6.6 – Call sizes dependent on type of barge traffic

	Number of calls	Average call size	Average load (per call)	Average unload (per call)	Number of containers
Inland	11208	23,10	12,86	10,24	258905
Rhine shipping	4728	42,56	21,25	21,30	201224
Antwerp	1631	88,36	45,16	43,20	144115
Total	17567	33,77	17,78	15,98	604244

6|3|2 Implementation environment

Important design considerations are related to the implementation environment. Here, several issues play a role:

- Many different parties with conflicting objectives. Terminal operators and barge operators have different objectives. These are conflicting, to a certain extent.
- Highly competitive environment. Barge operators, but also terminals, compete with each other and do not want to give their competitors too much insight.
- Strategic behaviour: Strategic behaviour by the barge operators worsen the way the current (manual) system works. Barge operators try to increase their chance be processed in a timely manner, by either over- or under-exaggerating the amount of containers to be processed. Data analysis of execution data from PortInfolink's BargePlanning (Moonen and Rakt, 2005) and interviews revealed that strategic behaviour takes place to a very substantial extent.
- Gain-sharing. Gains should be measured for each party individually, and for the system as a whole. If the system does not meet the expectations of a single party, this party has an incentive to quit, which is not desirable (Van Groningen, 2006).
- Yearly growth. The container market has shown yearly growth over the past decade in the double digits; the same trend is foreseen for the near future.
- It is hard to establish a trusted party that coordinates all operations. Barge and terminal operators will not quickly accept an authority coordinating everyone's actions. They want to stay autonomous and in control of their own operations.

6|4 Design directions

In the APPROACH2 vision document (Moonen and Rakt, 2005) we proposed several system architectures for a multi-agent system for barge rotation planning. These designs are given in Figure 6.7. The first column shows the manual way of working, the second the way the foreseen (first)

SYNCRON8 system would change this. This SYNCHRON8 is the result of the work done within the first APPROACH project, and only focuses on the 24H in advance pre-planning phase. Three novel architectures are shown in the figure as columns three, four and five. (1) An architecture that takes the SYNCHRON8 situation as a starting point, and adds continuous replanning and event management in the execution phase. (2) An architecture that only focuses on (near) real-time assignment of terminal visits. This approach does not include 24H pre-planning, simply since that plan will change anyway. (3) This architecture is a combination of the earlier two: rough capacity planning beforehand, along with the smart assignment in real-time based on the actual situation.

Furthermore we proposed to rethink existing processes when making a novel design. A specific example is the order-intake. Order-intake should not be considered disconnected from the pre-planning of terminal visits, which is how it is now. In practise, order intake and order acceptance continue until the very last minute: when capacity is available, orders are accepted until the moment of docking at a terminal's quay. See Figure 6.8 for an illustration. When a barge operator operates more than one barge, or operates a barge that returns every day to the Port of Rotterdam (in case of Antwerp- or domestic inland shipping), it could optimise its visits and load plans. For example, this can be done by grouping orders for a specific terminal (or destination) at barge B1, while grouping orders from another terminal at barge B2. Grouping can likely lead to fewer calls and larger call sizes. Barge operators can do this dynamically for new arriving orders in the order intake, but might consider so-called decommitment strategies (real-time optimisation of current execution plans) also: if a better idea, and still before execution, operators can reassign an order to a barge other than originally assigned. In a multi-agent system design, this can be achieved by barge operators' order intake agents, negotiating with barge agents about their bill-of-lading, perhaps even consulting terminal agents to request if sorting can be done in time.

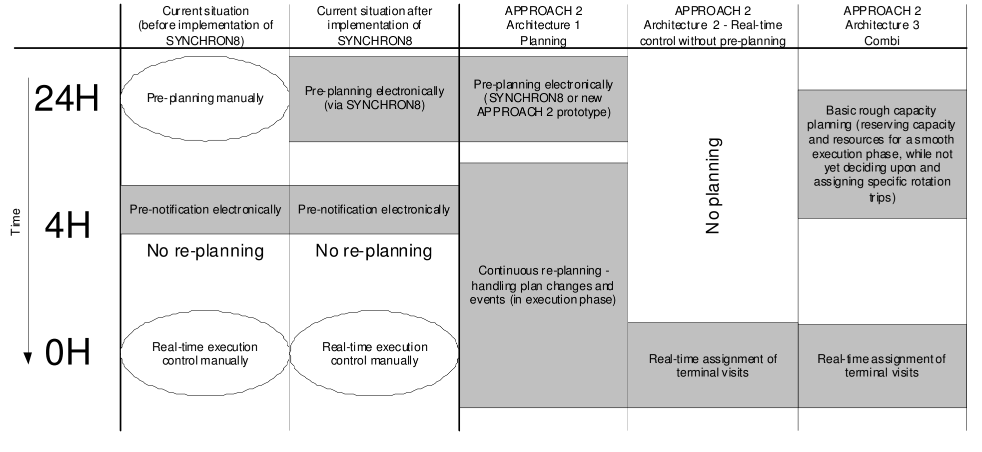


Figure 6.7 – Different architectures to be researched

Unfortunately, due to changes in the project structure and organisation, we have not been able to develop prototypes for the different architectures, and the work on re-planning and real-time planning of barge rotations has not yet lifted off. Rather, much work has gone into the construction of new approaches for the 24H planning problem – which is nicely documented in the dissertation by Douma (2008) and the book chapter by Douma *et al.* (2008).

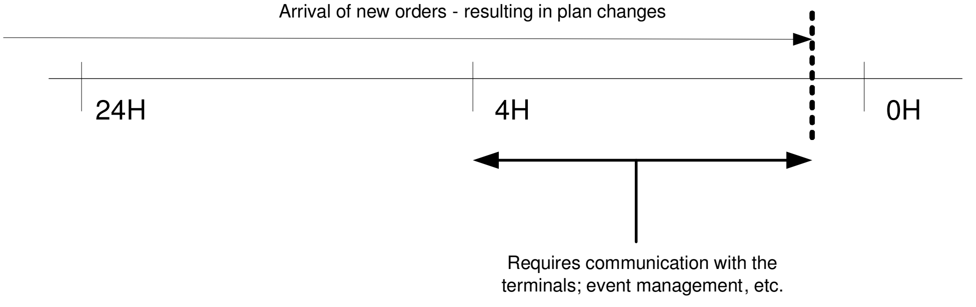


Figure 6.8 – Order acceptance and pre-planning of barge visits

6|5 Coordination through waiting profiles

6|5|1 Exchanging waiting profiles

We propose a direct communication mechanism which mirrors daily practice. Direct communication means that every barge operator contacts all terminal operators it has to visit during its rotation about convenient times for loading and unloading. Based on the response of the terminal operators, the barge operators decides when it is going to visit every terminal and in which sequence. This is actually the way the manual system works currently. Automated through agents, this might go through several rounds of communication and negotiation. An illustration of the proposed agent model is given in Figure O.1 in Appendix N, which shows a barge operator contacting terminal operators. As we have seen before, when sailing its actual rotation, it can be necessary for a barge to cancel or re-plan terminal visits if unexpected disturbances occur that would cause late arrival at terminals. This is especially important when terminals refuse to process barges that arrive too late. If a terminal refuses to process a barge, this barge has to make new appointments. This probably means a delayed rotation since it is likely that a terminal has no free capacity in the first couple of hours. A barge would thus try to make robust rotation plans which are less sensitive to disruptions.

The choice for a suitable communication mechanism is not trivial, and several mechanisms can be considered. Four alternative mechanisms are discussed in detail by Douma *et al.* (2008). For a short overview of the alternatives, see Table O.1 in Appendix N.

Let us look briefly at the fourth alternative, the waiting profiles approach. In the first stage, a barge operator agent asks a terminal agent for the expected waiting times during a day and the terminal replies to the barge with a waiting profile. This waiting profile – see Figure O.2 in Appendix N for an illustration – shows the expected waiting time during the day and can be customised for a specific barge operator agent based on its characteristics or its reputation. The expected waiting time is a maximum. This provides barges more certainty about the time processing would finish. The barge has to be at the terminal at the time promised. If it does not arrive in time, its reservation is cancelled and a new appointment has to be made. Waiting times have an additional advantage since they give the terminal the option to integrate slack in its schedule to cope with uncertainties. In fact, exchanging waiting times is more general than exchanging time slots since the latter can be derived from the former. The second stage consists of the construction of a rotation, followed by making appointments with terminals – see Figure O.3 in Appendix N. The rotation is constructed based on the waiting profiles and expected waiting times. A barge operator agent aims to find a rotation that minimises the sum of expected waiting, handling and sailing times. Once a barge agent has determined the best rotation, it announces the time it expects to arrive at the terminal and sends a confirmation. If the barge is not on time at the terminal, it has to make a new appointment. In the future, one could also incorporate the "reputation" of barges in the system, such as "always on time", "often too late", et cetera. This reputation can be used in models to adapt waiting profiles and to force barges to incorporate enough slack into their rotations. This results in self-regulation of the system, i.e., barges cannot deviate too much from their appointments although this is not contractually enforced.

6|5|2 Simulation

A field-test to test the concepts is unrealistic, since INITI8's SYNCHRON8 has not yet been implemented. As such, it would require parties to change their way of working, adapt their systems, and foremost, would require a huge investment in systems, implementation and people. Hence, that simulation is utilised as a research method – see also the discussion in Chapter 1. The same set of rotations needs to be planned as were used in the workshop reported on in section 6|2. This provides the opportunity to compare with an artificial real-life situation; namely, the human planning made then. Two different multi-agent system approaches (scenario II and scenario IV) are compared with a traditional, non-agent based algorithm which is both static and deterministic – e.g., it knows everything in advance with certainty. The static-deterministic algorithm plans all barge rotations with the objective to minimise maximum lateness, in other words, to minimise the delay of the barge that is delayed the most. The three different approaches, including the static benchmark, are described in more detail in the dissertation by Douma (2008) and the article by Douma *et al.* (Douma *et al.*, 2008).

Table O.2 in Appendix N presents the results from the simulation and comparison. It is clear that scenario IV, which is based on waiting times and travel times, in this case results in a much better performance than the approach following scenario II, which only uses travel times to plan a rotation.

The solution provided by the static benchmark is optimal in terms of the objective and was obtained within one minute of processing time. One should note that, as the results show, the two agent approaches do not perform much worse than the static benchmark. The latter is truly a baseline scenario, since it assumes all information to be statically known in advance (without any disturbances and re-planning throughout execution).

6|6 Expert evaluation

6|6|1 Expert evaluation workshop

An expert evaluation of the ideas and prototype as described in the above sections was organised in parallel with the expert evaluation session for the road planning cases – described in the previous Chapter (see 5|6). That chapter also described the structure and organisation of the workshops. Questions asked during the evaluation event are listed in Table 5.13. The two questions for this specific workshop are questions number three and four:

[3] What advantages and disadvantages of the shown multi-agent system do you see for barge-rotation planning?

[4] What advantages and disadvantages do you see in the exchange of waiting profiles?

Two categories of feedback were gathered, direct feedback on the prototype/game as such, and general comments concerning the implementation aspects of the system. The feedback of the second category is bundled with the comments on the road network prototype and the feedback to the general questions, and will be discussed in section 7|2 in the next Chapter. The direct feedback on Question III is discussed in section 6|6|3, and the feedback on Question IV in section 6|6|4. A complete overview of all feedback received can be found in Appendix K.

6|6|2 Description of the game

This evaluation workshop was constructed around a multi-player game. In the barge planning game, players had to plan a rotation along a number of terminals. To plan a rotation, players could request time slots at the terminals concerned. The terminals answered the players automatically, but with a random delay between zero and ten seconds. If a requested time slot was confirmed by a terminal it turned green on the player's screen. If the time slot was refused it turned red, indicating that the player should propose a new time slot. A screenshot of the UI is included as Figure 6.9.

Before playing the game the attendants got to perform a planning exercise on paper. For each of the eight terminals, information was given about the terminal opening times, and sailing times were provided. The game experience illustrated the dynamic setting, as the availability of a terminal was influenced by the time slot requested by other players. In the game, players could experience the value of information sharing (through the exchange of waiting profiles) and the information communication in a multi-agent system. For a more detailed description of the idea behind waiting profiles, see section 6|5|1, Douma *et al.* (2008) and Douma (2008).

The game provided the participants insights into how a multi-agent system might be of help in such a setting. Similarly to the road planning workshop, after the interactive component the participants had to fill out a questionnaire followed by group discussion. Due to a larger time consumption of the game component in the workshop, less time for discussion was available compared with the road planning workshop. For a more detailed description of the game and its use in the workshop, we refer the reader to Douma *et al.* (2008).

6|6|3 Feedback received on the prototype / concept (Question III)

Many participants appreciated the workshop's setting. They remarked that the game "*illustrates the capacity problems of container terminals well*", as well as the "*practical wishes of (future) end users*". The game illustrates that "*such a system should especially be beneficial for the barge operators / shippers*".

When developed into a real system, such future application would have the potential to "*formalise the decentralised planning that currently exists*", this way ensuring a system acceptable to all parties. Also ,it would provide support through automation, some kind of system-wide optimisation/levelling, and the possibility for event management. "*Real-time re-planning when events occur*" is mentioned as one of the advantages of a MAS approach. Several participants also mention the importance of "*management-by-exception*". Also mentioned was that MAS is an ideal type of automation in situations "*where many parties have different objectives*". A system, as demonstrated, has the potential to enlarge the capacity of port terminals, barges and thus hinterland transport flows.

Critical notes were also submitted. The fundamental question was raised as to whether this "*could not better be solved centrally*". A lively discussion emerged, which made a split visible in disciples of centralised and decentralised approaches. With respect to the specific game scenario, questions arose concerning the "*role of the shipper*", "*the flexibility of the barges in the time slots*", and the "*prioritisation of barge arrivals at terminals*". Many comments received through the questionnaires concerned future implementation of the prototype and its concepts. Technical issues were brought forward, such as: "*a radical change of the planning process is required*"; "*a standardisation of interfaces is needed*"; and "*planners should keep the decision authority (the system only advises)*". It furthermore "*requires special skills from the planners that need to be taught*".

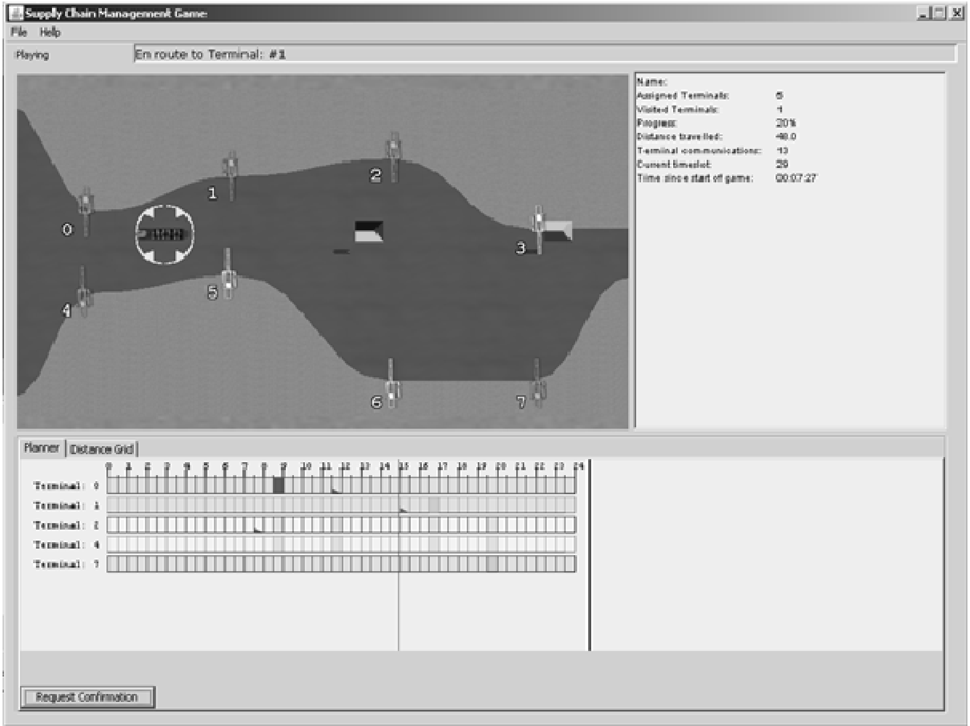


Figure 6.9 – Screenshot bargeplanning game

Concerns about the complexity of the market were mentioned, along with the question how to roll out such a system and get it adopted. Several participants remarked that it is likely that not all barge operators have the required (financial) resources. A question several participants raised is "*how to handle competitive information?*"; similarly, participants expect "*little transparency from the side of the container terminals*". Someone else even formulated it this way: "*It is a good and useful solution – however, due to an expected lack of willingness to cooperate on the side of the terminals, it is very unlikely to succeed*". Someone suggested to have a "*neutral and independent third party operating the system*" to overcome such hurdles. Someone else wondered whether some kind of currency should be introduced to enable coordination. It was mentioned that "*everyone should participate to improve the whole*". For a complete overview of all feedback to these questions we refer to Table L.5 and Table L.6 in Appendix K.

6|6|4 Feedback received on the waiting profiles (Question IV)

On Question IV, concerning the waiting profiles, a list of both advantages and disadvantages was received – see the feedback clustered in Table L.7 and Table L.8 in Appendix K. The number of issues on the list of disadvantages outnumbered the identified advantages. Perceived advantages were the issue that the information to exchange in the waiting profiles exists already, it can be easily distilled from existing systems. Furthermore, sharing is beneficial for both parties, and when shared, less communication is needed compared with a Q&A-style coordination mechanism. Someone else mentioned that insight into waiting profiles of terminals makes it possible to include coordination with up-river waterworks in the decisions.

Disadvantages were identified in three different clusters. First, entirely opposite to other experts who identified information sharing to be no problem, several (other) experts showed their concern that the "*parties involved do not want to share information such as this*", or that the "*intention to share information reduces when commercial or sensitive information is shared (a.o., with competitors)*". At the same time, for barge-operators it will be "*difficult to acquire information from multiple terminals, since this is competitive information*". Someone suggested that one way to tackle this is "*to first show all parties the benefits when information would be shared*".

Second, critiques on the use of the terminal waiting profiles was put forward. When "*many barge-operators use the same waiting profiles, this might result in requests for the same terminal capacity by several parties at the same time*". Someone else remarked that "*waiting times and utilisation rates of barges are very important too – include that explicitly in the solution*".

Third, worries were voiced concerning the question of how to respond to inappropriate / unintended use of waiting profiles. More specifically, it was stated that "*This approach runs the risk of cheating, parties that play it unfair – try to define a mechanism that minimises this*", and "*Parties can frustrate each other when changes occur and all parties start responding independently and unorchestrated to these changes*".

6|7 Discussion

6|7|1 Summary

This chapter discussed a MAS for an inter-organisational planning problem in the Port of Rotterdam. Barges connect sea terminals (in Rotterdam) with inland terminals in the hinterland. Every barge has to visit a series of terminals in Rotterdam within a limited period of time. Terminals have limited quay and handling capacities. Barges have to reserve time of arrival with the terminals. Since one terminal visit is generally part of a larger rotation trip through the port, coordination is required. In the current manual way of working, barges plan rotations 24 hours in advance, and communicate the intended arrivals with the terminals. These terminals utilise this information to decide upon the capacities (quay space and personnel) they need to reserve for the next day. They fit the barge in their schedules, and set and confirm the expected arrival time. There is no explicit feedback loop, barge operators do not re-plan their rotation based on the confirmation.

A workshop was organised to mimic a controlled and simplified version of the real-world, in which barge- and terminal-operator-planners planned a day's operations. The outcomes were shocking, and gave insight into the problems that occur. The pre-planning results in infeasibilities in plans. Examples are double bookings in a barge's rotation, and conflict situations at terminals where barges compete for the same quay space at the same time.

The workshop triggered a new round of (funded) research. Specific focus was on researching smarter algorithms and methods, real-time coordination, and implementation aspects. This chapter describes some of the design steps we went through, and compares two different agent coordination mechanisms with a traditional optimisation-based planning approach. The simulation experiments show that the multi-agent systems do not perform substantially worse than the optimisation-based approach.

Expert interaction workshops were organised to evaluate and validate the waiting profiles approach in a multiplayer game setting. The experts discussed the (dis-)advantages of a MAS to the barge-rotation planning problem, and the (dis-)advantages of the waiting profiles. This resulted in a list of identified issues.

Throughout this chapter sub-research question five was addressed, which asked "*Can multi-agent systems contribute to better performing, and easier implementable systems for transportation?*" As in the previous chapter this also concerned a design case that has not made it (yet) to real implementation. Nevertheless, synthesizing the research, we derived three main observations, with respect to this sub-research question, concerning the issues "*why agents fit*", the "*type of coordination*", and "*real implementation*". We discuss these in the sections following.

6.7.2 Observation one: Why agents fit

Throughout our research, we found in most contact with barge- and terminal operators that they have serious problems with the foresight of a centralised system for barge rotation planning. Perhaps only a matter of perception, but reasons mentioned include "*competitive reasons*" and "*existing legacy systems*" that have to be part of future systems, et cetera.

Multi-agent systems in such networks can provide a valuable perspective over (more traditional) centralised architectures. Every agent can operate autonomously, in the interest of a specific company and can encapsulate knowledge. Furthermore, a multi-agent system can mirror (to a larger extent) the way the network is currently organised in practise. In the expert evaluation workshops, the possibility to function in environments "*where many parties are having different objectives*" and the functionality for *"real-time re-planning when events occur*" were mentioned as important benefits of MAS. Nevertheless, the experts were critical, the fundamental question raised in the discussion was whether this "*could not be better solved centrally*". The discussion showed a split in disciples of centralised and decentralised approaches. The simulation results discussed in this chapter, and the dissertation by Douma (2008), show that the multi-agent system approaches do not perform substantially worse than centralised optimisation approaches.

6|7|3 Observation two: Type of coordination

A second set of important observations can be made in relation to the type of coordination. The largest shortcoming of the current manual coordination lies in the limited amount of iterations that can be made. When a schedule turns out to be infeasible, re-planning and re-coordination would simply consume too much time. The current practice is to make a rough plan with major infeasibilities in advance, and do last-minute coordination over the phone when sailing the actual rotation. The barge shipper then contacts the terminals it needs to visit and tries to arrange time slots.

The automation of inter-organisational coordination can be done in different manners, as this chapter showed. The coordination mechanisms researched are all mechanisms to do point-to-point coordination between barge operators and terminal operators, thus involving vertical coordination. Reasoning back, we think that there are three explicit choices to make in the design phase when designing the coordination mechanism: (1) the architectural choice [*centralised or decentralised, or even a hybrid form?*]; (2) the type of coordination structure [*vertical or horizontal, or cross-organisational*]; and (3) the mechanism or process of coordination as such [*the type of mechanisms, as shown in Table O.*1]. Modelled in a framework, this looks like Figure 6.10 – note that the third dimension is here in the filling of the matrix. Kumar and Van Dissel (1996) make a similar split when they identify different forms of inter-organisational interdependence – see Table 2.2. Their model incorporates the second and third decisions we discussed above, but does not make these explicit. The architectural dimension is left out of their division. In section 3|2, we introduced a framework to classify inter-organisational processes; see Figure 3.1. In this model, the coordination structure is an explicitly important element, since we divide over horizontal or vertical collaboration.

The agent scenarios discussed in this chapter only deal with coordination process approaches for the lower left quadrant of the figure: vertical coordination, utilising an agent-based decentralised planning approach. For future research, it might be interesting to compare approaches from other quadrants. The current approaches researched take the perspective of the barge operator: a barge schedules its rotation by contacting terminals. This framework gives thought for other mechanisms, since terminals might consider horizontal coordination (with order terminals) in order to better coordinate peaks and labour capacity. Furthermore, hybrid structures could be thought of, for example through agents working on a global (semi-centralised) level, that help in levelling different regions. The port consists of three main areas with terminal locations: the City, the Botlek and the Maasvlakte. Each region could have an agent that controls and protects against imbalance, for example, a situation where the City is underutilised and the Maasvlakte overutilised. A related question, with respect to the architecture, is whether one should aim for solely peer-to-peer communication and coordination, or utilise a structure with mediator agents. These kind of design questions are important for future acceptance of the system. We learned from the workshops and interviews that future users (at least when adopting the system) would like to understand what happens, and why it happens.

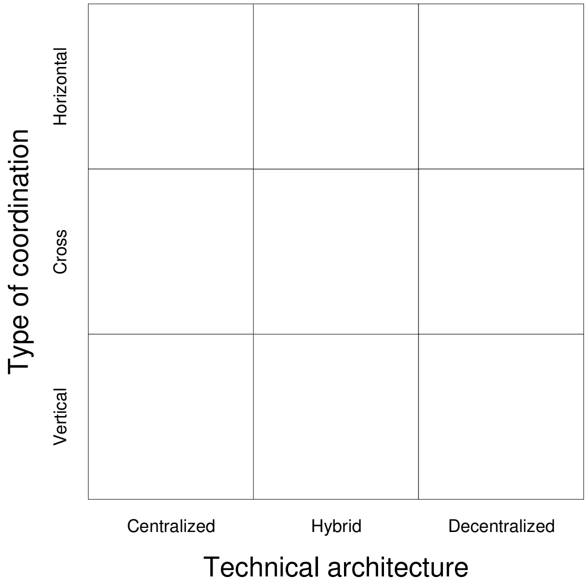


Figure 6.10 – Coordination mechanisms in IOS

6|7|4 Observation three: Real implementation

The APPROACH and SYNCHRON8 systems have not yet been implemented in the port of Rotterdam, despite years of research and development. The first project ran under the umbrella of Connekt and started back in 2000.

From our list of factors that hinder agent adoption in industry, as identified in our review of literature in Table 2.6, we can only link the factors "*cost*", "*standards*", and "*the legacy of legacy systems*" to the non-adoption of the system. The project did have a severe academic component, nevertheless the factor "*stuck in academic prototyping*" does not hold in this case. The project was initiated by a commercial firm, INITI8, which aimed at bringing a real product to market, but also the academic trajectory was performed in continuous interaction with the market.

A factor that hinders implementation is the complex inter-organisational environment. One of the aspects that makes it complex is the fact that almost all involved parties have planning systems in use currently. A future system should be able to connect with diverse backend systems, through, for example, a service-oriented architecture. Unwillingness to share competitive information is another factor. The same holds for the questions of "*who is going to pay what*" for this service, "*what the* *costs and benefits for individual parties will be*", and "*how we can be sure that parties are treated in an equal manner*".

The expert evaluation revealed the role of planners/users in such a system to be critical in implementation. It requires "*different skills*", and "*a process change*", but foremost, "*planners should keep the decision authority (advised by the system)*". The experts pointed to the crucial role of information sharing in the demoed system.

Recent developments will perhaps speed up (future) implementation. The Vereniging van Inland Terminals, the Vito, has started to handle barges at a dedicated crane at the Maasvlakte. This way, barges do not have to visit the different ECT Delta and/or APM terminals. Planning the dedicated Vito crane requires insight into barge movements throughout the port, and an understanding of arrival and handling patterns – as the MSc thesis by Van Andel (2007) demonstrated. Another development is the concept of a so-called Container-Transferium which the Port Authorities are studying. A Container-Transferium is a cross-dock terminal outside the port areas, where trucks pick-up and drop-off containers. This way truck traffic can be reduced significantly in the port. It will require a large amount of coordination and real-time control.

Note that both the dedicated Vito crane, as well as the container transferium, are examples of investments in additional physical infrastructure and extra capacity, not so much advances in planning. Investments in planning might impact handling times at terminals, reduce queues and introduce more certainty in the process.

Chapter 7 Towards implementation

7|1 Introduction

The previous two chapters reported on practical research projects in which we experimented with multi-agent systems. This chapter discusses implementation of (future) multi-agent systems. A full implementation requires something different than solely a prototype or proof-of-concept. In addition to a further engineered system, it also requires an implementation process. In section 7|2 of this chapter, we report research findings from the previously mentioned expert evaluation sessions. The expert opinions are confronted with literature, which provides a series of useful insights that are currently less-documented in literature. Furthermore, we surveyed recent literature focused on implementation aspects in relation to multi-agent systems. This is included in section 7|3.

7|2 Expert feedback on implementation issues

7|2|1 Introduction

In the same workshops as previously discussed in sections 5|6 and 6|6, we interacted with experts about implementation issues of multi-agent systems. Three questions were asked in the general part of the workshops, see questions 5 to 7 in Table 5.13 or below:

[5] What do you perceive to be the success factors for multi-agent systems for transportation?

[6] What do you perceive to be the failure factors of multi-agent systems for transportation?

[7] What other application domains do you see for multi-agent systems?

An overview of the feedback received on these questions can be found in Table L.9, Table L.10, and Table L.11 in Appendix K.

The workshop discussion (and the feedback gathered) centred around three areas that are also present in literature on multi-agent systems adoption and implementation. These three areas are: (1) (Dis-)Advantages with respect to information system design; (2) (Dis-)Advantages of MAS within the field of logistics; and (3) Factors that hinder adoption and initial implementation of multi-agent systems. We discuss these areas separately in the sections that follow, and confront the gathered feedback with literature.

In comparing the aspects identified by the experts to the literature, we find many similarities. Many aspects known from literature were, of course, not mentioned. Furthermore, we recognise several factors that the experts brought up, which are either described differently in literature, or are lacking entirely. Table 7.1 shows a model we used to cluster the feedback received, showing different categories of answers. The model has two axes: an axis differing between known/unknown in literature, and a second axis that divides the items into mentioned/not mentioned by the experts. In case an aspect is known in literatur, and mentioned by the experts, it can either be alike, or different. Not surprisingly, the category "not known from literature, and not mentioned by the experts" (category IV) is a category we could not fit too many aspects to.

Please note that, although we introduced our experts to the topic before the workshops started, we did not give them a complete lecture on the pros and cons that MAS architectures potentially offer. As such we can say that the answers they gave were mainly formed by their experiences, combined with the impressions gained throughout the day by viewing the demo and playing the simulator game. By letting them fill out the forms first, we ensured that everyone was able to write down his/her own opinions before group discussion took place (which could have otherwise been a factor of influence).

Table 7.1 – Model to cluster feedback

Aspects	Known from literature	Not known from literature
Mentioned by experts	I	III
Not mentioned by experts	II	IV

7|2|2 Advantages and disadvantages with respect to information system design

Many of the factors Lea *et al.*(2005) mention with respect to the expected benefits of a distributed MAS architecture as opposed to more traditional (ERP-styled) enterprise systems are confirmed by the experts. They confirm the aspects of "*impact of system failure*", "*update, modification and maintenance*", "*effective decision-making via real-time information sharing across different functional area*", "*technology readiness*", "*employee training*", and "*customisation capability to support current business processes*". Lea *et al.* (2005) mention that "conflicting goals" is a problem with traditional architectures as opposed to multi-agent architectures, whereas the experts particularly identify these as serious concerns in a multi-agent architecture, specifically: "*Objectives of an agent can be conflicting: handle the current order, versus jumping on a future order*". Likewise, the experts have their doubts concerning the amount of business process redesign needed for a multi-agent system implementation. Specifically, they mention the need for different processes and mechanisms of control.

One of the potential benefits of MAS architectures is the distributed nature of the application, which unleashes large computing power for the application. This factor was specifically mentioned by the forum of experts and can also be found in (Mönch *et al.*, 2006). Four out of the five basic strengths of multi-agent system architectures, as identified by Caridi and Cavalieri (2004), are confirmed by the experts; namely: "*modularity*"; "*decentralisation reduces impact of local modifications on other system modules*"; "*embedding multi-objective functions*"; and, "*designing systems can be a step-wise process*". Only the factor "*allows to effectively model time-varying physical systems*" is not mentioned. The advantages that "*problem solving in the system is closely based on problem solving in the human organisation*" and "*the responsiveness of the autonomous agents system*" as Wooldridge (2005) summarizes them, are confirmed by the experts.

An important difference mentioned by the experts, which seems to be less thoroughly studied, is the observation that a different type of programming skills is needed for developers developing multi-agent system applications, as opposed to more traditional software development skills. Our own experiences developing the prototypes show that a different mindset is indeed needed: one has to start thinking in terms of agents, behaviours and interactions. Jennings and Bussmann (2003) notice that although agent-based software development is a different method, they do not perceive the programmer's skills to be a problem. Wooldridge (2005) even states that developing multi-agent systems is easier for developers and designers than developing traditional systems. That might be true for skilled MAS developers and designers. Parunack (2000) signaled that a change in the educational system is needed to get agents implemented in practise. We can only conclude that this change has not yet taken place.

7.2.3 Advantages and disadvantages of MAS in the field of logistics

The experts we consulted considered the application of multi-agent systems in the shown logistic application domains to be appealing. The four primary reasons for MAS application in transportation, as identified by Fischer *et al.* (1996), were all confirmed, namely: it is "*an inherently distributed task*"; "*capability to handle dynamic events*"; "*an alternative solution to central optimisation, which focuses only on utilising local information*" and "*firms engage in a high-level of negotiation and cooperation in performing their daily transport tasks*". Davidsson *et al.* (2005) also concluded that the problem characteristics of logistics closely match those of an ideal multi-agent application.

An issue that, to the best of our knowledge, seems to be understudied, is the question: "*How can the (tacit) knowledge get from a planner to the system and then be translated into decision rules for the agents?*" We realise that this has been a topic in the expert system (research) domain for many years, but think that this becomes equally important when building real-world multi-agent systems. A separate domain of research is "*planner – (agent) system interaction*". Much of the feedback we got from the experts concerned practical design issues, but also fundamental issues were raised. For example: "*Should the agent learn from planner interaction, and if so, how?*"

The experts commented that the prototypes shown were both examples of operational/executional level planning systems. Questions were asked how this should fit with applications with longer planning horizons, e.g., strategic/tactical. This is an aspect currently less documented in literature, as Davidsson *et al.* (2005) have already concluded.

Several other useful insights that are not widespread in scientific literature were provided by the experts. One of the important comments made by a series of experts is that they perceive the inter-organisational domain to be the application area with the largest potential for multi-agent systems. Another important comment was the critical observation that, in practise, "optimality" means something other than what optimality among scholars means. In practise, it is less about achieving the theoretical optimal, and much more about good-enough solutions that are accepted and easy to implement. Several experts commented that this might be an additional benefit of multi-agent system approaches. A third important comment we picked up is that we should be aware that too much information exchange, especially asymmetric information exchange, can negatively influence one's competitive position.

7|2|4 Factors that hinder adoption and initial implementation

In the discussion of factors that hinder adoption and implementation of multi-agent systems in logistics, the experts confirmed many factors found in literature. These factors included: cost; accuracy and correctness of the results; acceptance by users; the role for human decision-makers; the need for professional development methods; standards; the legacy of legacy design and techniques; and the factor that many academic research projects never make the transition from the lab to practice. We listed these factors in Table 7.2 (under category I), with references to literature that made to similar observations. In Table 7.2, we list implementation aspects discussed in literature but not mentioned in the discussion or the returned questionnaires. These can be found under category II.

The experts mentioned several issues that we have not found documented in literature, these are listed in category III, Table 7.2. The common denominator for these issues is the remark that "*the real question the research community should be working on is how to get multi-agent systems implemented in real practise*". Benefits of multi-agent applications in logistics seem to be substantial at first glance, however, there are still so many open issues that real implementation could take some time.

Concrete issues mentioned by the experts to include in future systems implementations include: "*How to divide gains (and losses) within a networked system*"; "*There is a need for real business cases, other than solely good ideas*"; and "*What should be the role for stimulation from government and sector organisations?*" This has been mentioned before by Moonen *et al.* (2007) but has not been researched to a large extent.

Table 7.2 – Adoption and implementation aspects from expert feedback

Category	E	L	Aspects mentioned [E: Experts; L: Literature; Y: Yes; N: No]
I	Y	Y	Cost Accuracy of results / Guarantees of operational performance Acceptance by users Central role human decision-makers Professional development methods Need for (shared) standards The legacy of legacy design & techniques Stuck in academic prototyping Need for education and training
II	N	Y	Legal / ethical issues Scalability Security Misapplication (cannot solve all problems)
III	Y	N	Real question: How to get MAS implemented in practise? How to divide gains (and losses) within a networked system? There is a need for real business cases, other than solely good ideas What should be the role for stimulation from government and branche organisations?
IV	N	N	n.a.

It was mentioned that non-existing contractual relationships between parties, which have to collaborate with each other to smooth operations, hinder chain applications and hard-wired integrations. Participants from a large terminal suggested that a possible instrument to lead to more information sharing and collaboration throughout the chain – in situations where contractual relationships are lacking – might be to provide premium treatment to chain partners providing reliable information. In contrast, this service would not be provided to parties that do not provide such information.

7|2|5 Discussion

The prototypes discussed in the workshop are not yet production-ready systems, and as such we received several comments that the prototypes are oversimplified, and miss many important aspects that a real implementation would need. Nevertheless, the prototypes turned out to be useful artefacts for interaction with experts. Rather than dealing with abstract concepts, it helps to visualise and explain the unique characteristics and inner-workings of a system, and gives a common ground for discussion about implementation issues. The feedback we received can be split in two groups. We received very concrete feedback on the prototypes themselves, with suggestions for improvement (these can be found in sections 5|6 and 6|6, respectively). A different group of feedback concerned the underlying concepts demonstrated and visualised in the prototypes, and the wider potential for future applications. The largest part of feedback was received in the second category.

This expert evaluation does have limitations. First, its academic rigor may be questioned. The time taken with the experts was relatively limited, the experts were not carefully selected but registered themselves for the event, and the prototypes were demoed and shown, but time did not allow a full demonstration of all details. However, a large group of experts was present, with diverse backgrounds, and relevant feedback was received from the structured answer forms and the group discussions. Second, although the focus in the workshops was largely on the implementation aspects of multi-agent systems, the two demoed prototypes have not yet been implemented, nor aim, in their current designs, to be full-blown implementations. Third, despite the comparison with a large body of literature – see Lang *et al.* (2008) and Chapter 2 – when comparing with literature, one compares, by definition, with only a subset of literature.

7|3 Discussion of insights from literature

Triggered by our last remark, we decided to dive into literature again, and search for additional insights related to the implementation of multi-agent systems, or similar technologies, such as service-oriented architectures, in industry. We discuss these below.

Already over a decade ago, Vervest (1994) foresaw the need for more dynamic inter-organisational systems that would enable what he called the "*dynamic search for capabilities*" and "*event-driven planning*". He drew the parallel that the boardroom of the future would look more-and-more like the war-rooms of modern military generals. Coordination, as such, becomes very important to businesses and larger supply chains. The need for third-party coordination in supply chain governance is nicely discussed by Bitran *et al.* (2007). They draw the parallel to the Li&Fung case – for more info see also Margretta (1998). Literally, they state that "*there is an emerging need for entities that have the knowledge and skills to manage functionally diverse and geographical dispersed supply networks*." Also, Pil and Holweg (2006) describe that value creation in industries nowadays is becoming less linear. Steering demand, chain information, and extensions to other chains are identified as very important. Although the above-mentioned papers discuss important changes in business models and in the ways companies function, they do not discuss nor hint at specific technologies, as such. Let us now look at the current state-of-the-art in technology and implementation.

The current situation in the enterprise information systems field is discussed by Rettig (2007). He reasons that the promise of agility was never realised. Instead, most implementations created rigidity.

Systems and processes turned out to be, once implemented, difficult to change. Also, the promise to realise one single system with a standard way of working, something which is still seen as a solution by some – see for example Mahato *et al.* (2006) – turned out to be difficult to achieve. Reality often shows 30+ systems running in parallel at larger companies.

Over the recent years, software vendors and consulting firms have put forward SOA (Service Oriented Architectures) as an instrument to overcome such complexity, introduce agility and at the same time leverage investments in the corporate information systems infrastructure. However, Rettig (2007) very critically comments that: "*To the extent that these service-oriented architectures use subsets of code from within ERP and other enterprise systems, they do not escape the mire of complexity built over the past 15 years or so. Rather, they carry it along with them, incorporating code from existing applications into a fancy new remix. SOAs become additional layers of code superimposed on the existing layers. That means it is possible that a process will fail at some point due to some fault in the layers below, and in order to understand and fix that problem, software engineers will need to deal with the layers of enterprise applications below the modular business processes.*"

We would like to raise the question of whether services that are built heavily on top of existing legacy actually result in new flexibility, or if they reduce instead flexibility through the creation of new spaghetti code. Is it, as such, only a solution in the short run, and should we aim for totally different (green-field) architectures? That is also something SOA could play a role in, but as Rettig (2007) observes: "*Technical realists point out that many difficult technical problems must be solved before SOA can become the backbone for a new strategic architecture, including robust protocols for accessing the applications, high-quality integrated data stores and a sound methodology for managing the overall process*."

This is where multi-agent systems could play a role. In the past five years of research, we have observed ourselves that agents were, and still are, largely an academic topic. Eventually, concepts from the MAS research community will find their way to industry, as Wooldridge (2005) underscores. McBurney and Luck (2007) reflect more on this issue. They state: "*We're at the point where we can now build open and dynamic systems, which underpin nearly all views of future computing, but we haven't yet done so to any great extent. […] Once we do, the prevailing model of computing changes – even current large-scale distributed systems are not open and dynamic in the manner envisioned – so it makes sense to think and work in terms of the agent conceptualization and the associated technologies that go with it.*" Object Orientation (OO) is hardly ever considered a specific technology anymore, rather, it is seen as the current standard way of conceiving and engineering computer systems. McBurney and Luck (2007) reason that it is not illogical that, in the near future, agent concepts will go down the same path and replace the OO paradigm. The question perhaps should be *when* it will occur, rather than *if* it will occur.

The current architectural design tools to construct agent systems (primarily used in research) make it difficult to construct real-world software for professional software developers, as Garcia and Lucena (2008) state. A software architect has to include many aspects which are currently not supported through professional software development environments. Examples mentioned include the handling of agent characteristics, designing roles, structuring behaviours, adaptivity, et cetera. They plead for better design tools and technology toolkits, and mention that toolkits such as JADE (which we utilised in Chapter 1) and JACK are a start, but have a long way to go.

Pretty much in line with agent principles, Bae and Seo (2007) make the case for the necessity of splitting the decision algorithmic and the communication layer in inter-organisational supply chain applications constructed through SOA-like technologies. By separating the process logic from the decision logic, it might be easier to construct such systems. They specifically mention BPEL technology as an instrument.

The functioning of companies, and employees in these companies, is more than ever before impacted by new information (and Internet) technologies. Brynjolfsson and McAfee (2007) foresee a future in which employees will become technology-empowered knowledge workers who operate in networked enterprises. Collaborative technologies will be utilissd within and between companies, they state. These technologies are becoming so user-friendly and powerful that end-users within companies have started developing and implementing new tools and features themselves. Bughin and Manyika (2007) observed this in their survey on Web 2.0 technologies. The ease of exploring these technologies is cited as a factor to avoid typical barriers to implementation by quickly pulling together prototypes. Bughin and Manyika reason that this is an important difference from traditional enterprise system implementations: "*Instead of big bang top-down approaches, now [the implementations] start at a company's grassroots level: small groups of interested individuals can launch informal pilots to test their viability.*" Furthermore, one should note that a specific novel technology should never be the sole motivator for change in a company, instead, change has to be driven from the potential for improvement in operations. Technology can help in realising change and, as Merrifield *et al.* (2008) formulate it, to help in "*building ultraefficient and flexible [intra- and inter enterprise] operations*" that are connected through distributed software and networks. They suggest that companies should perceive SOA not as technology, but rather as an instrument to help reengineer their processes. Their paper discusses several examples.

7|4 Conclusions

"*The real question the research community should be working on is how to get multi-agent systems implemented in real practise*", was a remark we received. It shows the interest from industrial parties for concrete experimentation with multi-agent systems which was something reported on before by Luck *et al.* (2004), and Van Hillegersberg *et al.* (2004). Inter-organisational applications have the largest potential, according to the participants.

The experts confirmed many facts known from literature, but also added new insights. For example, it was mentioned that a different set of programming skills and education is needed. Not surprisingly, most new insights concerned aspects that become important when moving towards real implementations. Mentioned was that optimality in real practise is not the single most important factor. Furthermore, the comment was received that both prototypes demoed were examples of applications with a limited time-horizon, whereas in practise planning is needed at different planning levels. Also, an observation was made that unequal information exchange might be an important factor that hinders adoption of systems. The division of gains and losses in networked applications is a fourth factor.

Our literature survey revealed that industrial environments and supply chains are currently going through a major change, and available technologies are changing in a fast pace. Agents could potentially profit from the services hype, but in order to do so, much needs to change – an example being the development tools. SOA, at the same time, faces certain limitations where agents could fill in. A last observation is that end-user development has been increasingly taking place.

Chapter 8 Conclusions and discussion

8|1 Introduction

This dissertation is the result of several years of research into the application of multi-agent systems in inter-organisational systems for transportation. While working on our initial research questions, we came across new questions. Some of these have been answered; others are still open for future research. We will discuss these in the sections that follow.

Section 8|2 summarises our key findings, utilising our initial research questions. In section 8|3, we discuss the scientific and managerial implications of our work. Suggestions for future work are discussed in section 8|4.

8|2 Results

The largest contribution of this dissertation is the insight gained in the application of multi-agent systems to the SCM domain. Starting from a broad research base, we worked and interacted with practioners and scholars, in workshops and in real-life design cases, on the application of multi-agent systems within supply chains, and particularly in transportation. We followed a design research approach, in which we designed, developed and evaluated IT artefacts in and with their intended usage environments. Despite the fact that we did not make it to full implementation in the two cases, the artefacts developed made it possible to interact with users and experts about future implementation.

The research in this dissertation was performed around the main research question: "*How can multi-agent systems be successfully applied to design and implement better performing inter-organisational systems for transportation?*" Subquestion 5 "*Can multi-agent systems contribute to better performing, and easier–to-implement systems for transportation?*" drove our design and experimentation phases, reported on in Chapter 1, Chapter 6 and Chapter 1. The hands-on experience in the design research provided a different perspective on multi-agent systems, design and implementation, different than we would have obtained through literature alone. Along the way we touched upon unforeseen questions, of which the question of how to evaluate a prototype MAS system, which is covered in Chapter 1, is an example.

This dissertation results in five key findings that deal with technology, application, implementation, the research process, and/or combinations of these. The key findings are listed in Table 8.1 and discussed below.

Our research illustrates that MAS have the potential to realise systems that are both inter-organisational and operate in (near) real-time; as such MAS offer potential for supply chains and, in particular, transportation activities – see the two cases in Chapter 1 and Chapter 6, and the expert feedback received in sections 3|3|2 and 7|2|3. MAS make it possible to switch to a different planning and control paradigm, focusssed on coordination through communication and negotiation rather than an isolated (single-tier) optimisation. This can be done in (semi-) real-time, and both intra- or inter-organisational. It makes it possible to utilise real-time sensor data integral to decision-making, such as, for example, GPS or RFID information. One of the important comments received from a series of expert workshops is that "*the inter-organisational domain is [perceived to be] the application area with the largest potential for multi-agent systems*". It turned out that MAS are especially useful in logistical situations that require coordination, last-minute decision-making, and face several – sometimes partially conflicting – objectives. From an information systems design perspective, MAS are a different approach to establishing systems. MAS make it easier to include perspectives of stakeholders. The approach to centre a system on communication and coordination opens new perspectives for IOS.

Second, multi-agent systems bring to inter-organisational applications not only the feature of information sharing, but also the feature of selective information hiding. In many chain applications, parties do not want to reveal too much about their competitive position, and thus want to limit the data to be exchanged. This especially plays a role in industry-wide systems, such as the barge planning application discussed in Chapter 6, and it was also one of our conclusions from our interactions with experts – see specifically section 7|2|3. It is less an issue in MAS systems designed from a central enterprise's perspective, as the Post-Kogeko design case illustrates. One possible instrument to hide competitive information (from competitors) would be through a trusted third party (TTP). A TTP generally is an intermediary that facilitates interactions between two parties who both trust the third party; they use this trust to secure their own interactions. TTPs potentially have an important role to capture and redistribute gains and losses among participants, see section 6|3|2 describing the important design requirements for a barge rotation implementation. Setting up a TTP structure is far from trivial – see also the literature and expert observations in sections 2|7 and 7|2.

Third, the evaluation of our two cases revealed that for future MAS application in industry, hybrid architectures that integrate existing legacy, which and are not of a fully decentralised nature, or most likely. This thesis combines two essential observations. First, little implementations will be green field implementations. Companies generally have existing information systems in place, and a MAS system needs to become part of the larger information systems infrastructure within or between enterprises. In the Post-Kogeko design, several existing systems had to be integrated, see sections 5|2|4 and 5|3|4. In the case of barge planning, it is likely that the different parties have diverse back ends (see sections 6|6|3 and 6|7|4). Second, MAS designs will be either part of, or have to integrate with, multiple hierarchical control layers with different planning processes depending upon the planning horizon. A MAS which solely performs real-time assignment as we piloted at Post-Kogeko – see Chapter 1 – is of limited value. One of the experts formulated it as follows: "*This (…) only* *shows a limited application: it is highly operational. A future real-life system should include also longer-term planning (tactical / strategic) functionality.*" – see also sections 5|6|3 and 7|2. The process analysis in the barge planning case (see section 6|3|1) already hinted at the need for a multi-level planning design – see our suggestions for further work in section 8|4|2. Hybrid system design does not necessarily interfere with MAS principles. In fact, it might be the only realistic way to get such systems implemented in practise.

Fourth, we found that although many papers exist which focus on MAS in logistics and/or transportation, relatively little research published to-date took place outside of labs and is linked to the industrial practise. MAS, as a research domain, is mainly academic. In our interactions with industry we received the critical comment that "*the real question the research community should be working on is how to get multi-agent systems implemented in real practice*" – see section 7|2|4. The MAS and the (more practise-oriented) SOA domains can positively influence each other, see section 2|5|5 and 7|3. Current MAS technologies make it possible to go through relatively quick design and development iterations, allowing to test and evaluate novel ideas in practise and translate these into systems, as our own design efforts with Jade in Chapter 1 illustrate. MAS design should be made part of academic curricula, as the design of MAS requires different skills from developers. This we experienced ourselves, found through literature study (see 2|6|3) and heard back from experts (see 7|2|2). The two design cases reported on in this dissertation will be transformed into teaching cases that can be used by technical and business school students to understand the application of multi-agent system concepts for inter-organisational systems in supply chains.

Table 8.1 – Key findings from this dissertation

#	Key finding
1	Multi-agent systems have potential for supply chains, and, in particular, transportation. They can enable a different planning & control paradigm that is focussed on coordination through communication & negotiation rather than isolated optimisation.
2	Multi-agent systems do not only offer information sharing, but also selective information hiding, which is very important in inter-organisational applications.
3	For most future MAS applications, we foresee hybrid architectures that integrate existing legacy, and are not entirely of a decentralised nature.
4	The largest part of MAS research to-date takes place solely in labs and is too disconnected from reality. This is a pity.
5	Applied research into novel inter-organisational systems requires a different view on prototype evaluation than currently common within the research community.

The fifth finding is that applied research into novel inter-organisational systems requires a different view on prototype and/or software-artefact evaluation than is currently most common in research. Evaluation is an essential part of any design research. Moving towards the evaluation of our two design cases, we found that it is not crystal clear how to properly evaluate an inter-organisational agent prototype with all of its complexities – see Chapter 1. For the research in this dissertation we utilised a multi-method evaluation approach that is based upon an overview of different evaluation methods (see Table 4.1). The later sections of Chapter 1 and Chapter 6 show how we utilised the multi-method evaluation.

8|3 Discussion of implications

8|3|1 Scientific implications

The research in this dissertation has several implications for science. This thesis contributed, in particular, two detailed design cases of MAS to practical supply chain applications, an area in which relatively little work has been done. Multi-agent systems have the potential to change the construction of supply chain systems. MAS systems are primarily based on coordination (in the chain) rather than isolated (single-tier) optimisation. Our research illustrates that MAS has the potential to realise systems that are both inter-organisational and operate in (near) real-time. We found large interest from practioners for the shown concepts and prototypes.

The founding principles of supply chain management have been, up until now, difficult to realise through (traditional) information systems (Sharman, 2003). SCM might be better helped with multi-agent systems. The potential benefits of multi-agent systems to supply chains are their coordination and negotiation mechanisms, which make it possible to consider fundamentally different ways to automate supply chain management. Only a limited amount of “agents in SCM” papers, as we have seen in section 2|5, deals with chain relations, coordination, and negotiation. This dissertation might inspire future work in this domain, as we expect a revolution in planning and planning systems.

Furthermore this dissertation developed a perspective on prototype evaluation. We propose a multi-method evaluation approach for MAS and provide an initial overview of different methods (see Table 4.1). These insights can be beneficial beyond the initial focus of our work. The simulation, as part of our evaluation of the road-planning prototype (see section 5|5) illustrated the importance of a different perspective on evaluation. Otherwise, one compares apples with oranges (or pears, for the Dutch reader).

In our own research we found that iteratively confronting theoretical concepts and practical insights provides added value. The targeted end-users of our research, practioners, are often not very interested in underlying technologies or concepts – see for example the remarks received in sections 5|7|5 and 7|2|3. For them, the eventual application of technology or concepts matters. Nevertheless, their practical insights are seldom documented in literature. In our interaction with experts we found, for example, that practioners posses a lot of practical experience concerning the implementation of complex inter-organisational systems, which differs at several points from what is documented in literature, and vice versa – see section 7|2. This dissertation made a first contribution, and we recommend expert interaction as important for future research.

8|3|2 Managerial implications

The research in this dissertation has several managerial implications. First of all, it illustrates the importance of collaboration between scientists and practioners. Scientists are searching for relevance and practioners have practical problems and challenges to be helped with. Second, this dissertation distils several lessons suitable to management, which can be valuable input for organisational restructuring and system (re-)design.

The Commission Van Laarhoven came up with the slogan "*logistics = smart ways of organising*". The research in this dissertation along with its five key findings (listed in Table 8.1) provides several handles to help logistics become smart and organised. It is important to better utilise information already available in organisations and supply chains. Electronically enabled coordination with supply chain partners can (and will) become very important. As a matter of fact, coordination in supply chains is the place where large cost reductions can still be realised (Sharman, 2003; Sutherland, 2003).

The two cases discussed in this dissertation especially illustrate the potential multi-agent systems have for spanning organisational boundaries. In the road planning case, discussed in Chapter 1, we worked on a system design for internal operations, which was linked to the larger supply chain context. We identified that although internal operating costs can be saved – less planner hours, smarter assignments – the true potential lies in chain application: information exchange at different levels, resulting in cost savings for the LSP, the terminal and the customer. The automated use of real-time information in the coordination with chain partners has large potential for both cost reductions and improved service. The example of barge rotation planning, discussed in Chapter 6, taught us that there is large potential for an information system to support the coordination hassles of planning barge rotations in the port. Parties behave autonomously, but largely influence each other. Managers should start thinking beyond corporate borders. Technologies have the potential to change the way companies and chains operate, however, it is not solely a technology issue as it touches upon all aspects of enterprising (Merrifield *et al.*, 2008).

We recommend discussing the findings and lessons from this dissertation internally as well as externally with chain partners, and to seeing how processes can be improved through information exchange and coordination with chain partners or within the own organisation. As the road planning example showed, new technologies and different approaches have the potential to change inter-organisational processes – see section 5|7|6. It is important to realise that the world of computing and information is rapidly evolving into a ubiquitous environment that enables decision- making anywhere and at any time. The world becomes more digitalized every day; (sensor) data will soon be everywhere. Carefully review if and where your company's processes and/or those of your chain partners have become digitalised in the past years, and how this opens up possibilities for MAS. Real applications of multi-agents systems will likely be hybrid systems. It is important to think about the influence multi-agent concepts have on existing information systems infrastructure. Furthermore we strongly recommend, starting experimenting with MAS technologies and concepts.

8|4 Reflection and future work

8|4|1 Limitations

The research in this dissertation does have limitations. Due to the nature of the topic, the research developed into a relatively broad work. Although we chose to concentrate on multi-agent systems, we do realise that in practise many different alternatives to solve similar problems exist. We were not able to study these to the full extent in this dissertation, as we focussed on MAS. At the same time we realise that we perhaps have been too positive and optimistic about multi-agent systems at times throughout our research. We tried to avoid this natural side effect of a design/action research approach.

With respect to the design research approach chosen, we did not make it to full-blown implementations. Rather, we evaluated the developed prototypes and concepts through different evaluation methods. A real-implementation would provide further insights.

Another limitation is the algorithmic choices made in the development of our designs. We did not start with the idea to design the smartest planning engine ever, but aimed at constructing a relatively smart mechanism that utilises information on the fly, and interacts with external systems – similar to how human planners work. We made these pragmatic choices, but nevertheless perceive algorithmic choices as important for future research. We would like to here refer to the work by Kemppainen (2005) who discusses the use of dispatching rules in operational planning, and the work on agent algorithmic design by Mahr *et al.* (2008), De Weerdt (2003), and Mes (2007).

The generalisability of our research is limited. The cases have not been carefully selected, but developed from projects we were involved in. Furthermore, we did not include a true cross-case analysis, since our cases, and our research approaches within these, are too diverse. This would be something for follow-up work. Nevertheless, several of the findings are broadly applicable, which is also due to our extended review of literature across several domains, and the frequent interactions with experts. The experts were not randomly selected, but were rather invited through selected mailings to known contacts, and advertising through specialised websites such as `logistiek.nl` and transumo.nl. This resulted in a collection of experts who are open-minded towards research and eager to discuss novel approaches, concepts and techniques.

A final limitation is the trouble to grasp what really happens in industry. Whereas researchers tend to document their journeys, practioners generally do not – or only through white papers that often lack detail. We concluded that in industry hardly anyone speaks about "agents" or "multi-agent systems"; however, it might be that many concepts are utilised in practise already.

8|4|2 Discussion of further work

This dissertation answered several questions, but it also brought up many more. We will now discuss these suggestions for further work.

With respect to planning in logistics, we were intrigued by the fact that in many companies, planners who work with advanced planning software do not utilise these packages to the full extent. The automatic plan button is often neglected; planners often only utilise the software for its user-friendly graphical visualisation, or to create a sorted list. This raises questions about the role of planners in such systems, but also about usability of the software.

From an inter-organisational systems perspective, the question of how to divide wins and losses in the chain is interesting. In addition, the question of how to get parties in the chain aligned to adopt a certain system: Is a dominant chain party needed? A shared performance measurement system is suggested (Folan and Browne, 2005; Slobodow *et al.*, 2008) to contribute to the success of an IOS – an interesting topic for further research.

In a recent paper by Van der Horst and De Langen (2008), coordination challenges in port-hinterland transport chains are discussed, including a specific consideration of the Rotterdam situation. They identified five categories of coordination problems: (1) Unequal distribution of costs and benefits of coordination; (2) Lack of resources or willingness to invest on the part of at least one firm in the chain; (3) Strategic considerations ("reluctant, if also competitors benefit"); (4) Lack of a dominant firm; and (5) Risk-averse behaviour and a short-term focus. A reference to some of our work is included. Four mechanisms are proposed to enhance coordination: (1) The introduction of incentives; (2) The creation of inter-firm alliances; (3) Changing scope; and (4) The creation of collective action. Future implementation research might consider including these mechanisms and validating them in real-life settings, perhaps building upon the work in this dissertation.

From an information systems point-of-view, the question of agility and flexibility becomes intriguing. How can hybrid systems be constructed on top of existing legacy that do not create new legacy, but rather bring flexibility? Or is this a contradictio in terminus? What can MAS further contribute to developments in SOA? Also, are systems constructed from multiple agents indeed less of a black box to a user – see also Krauth (2008), and our observations in section 7|2|2? A different aspect is responsibility: Who is responsible when an autonomous system makes decisions? This is, of course, a topic of discussion for all automated trading, but the more autonomous systems become, the larger the issue.

Another important question deals with the topic of how to get multi-agent systems from the lab to real practice; is there a need for different development methodologies, tools and techniques? This in turn triggers the question whether there is still too much low-hanging fruit in practice. Is industry not yet ready for MAS concepts? Or are the concepts not yet ready to be applied in industry? Luck *et al.* (2003) reason that the "*lack of industrial take-up can also be understood through the absence of a migration path. We cannot hope to establish multi-agent systems radically and from scratch. […] Instead, we need to show how industry can migrate to agent-based solutions gradually, while protecting existing investments in hardware, software, and skills.*"

The paper by Lea *et al.* (2005) describes potential benefits multi-agent systems could bring to enterprise systems. The article is not very critical, and it is unclear where the claimed benefits are derived from. We find their ideas (see Table B.1 in Appendix A) however intriguing as such. Our field research confirms the claimed benefits of "effective decisions through real-time information sharing", "less need for BPR", "customisation capability", and to a certain extent also "employee training" and "technology readiness". However, with respect to implementation – specifically "implementation time" and "implementation cost" – we doubt whether, at least in the first years of multi-agent system implementations, their statements hold. Global consulting firms and enterprise software vendors have large groups of implementation consultants with strong experience in ERP implementations. Custom software development can be expected to take more time, and other skills. For future work it would be interesting to reconsider the list of benefits and test them in practise.

What are the design decisions we are dealing with when designing multi-agent systems? In (Douma *et al.*, 2008) we discussed a series of design decisions for multi-agent systems, related to the barge planning case – here documented in Chapter 6 – see Figure 8.1. The figure illustrates that design decisions are very diverse in nature, and range from technological choices to choices about mechanisms, agent societal structures, logistical mechanisms (as presented in, for example, Figure 6.7), intelligence to include within the agents, and the role of humans in the system. Is this a basis for a future reference model?

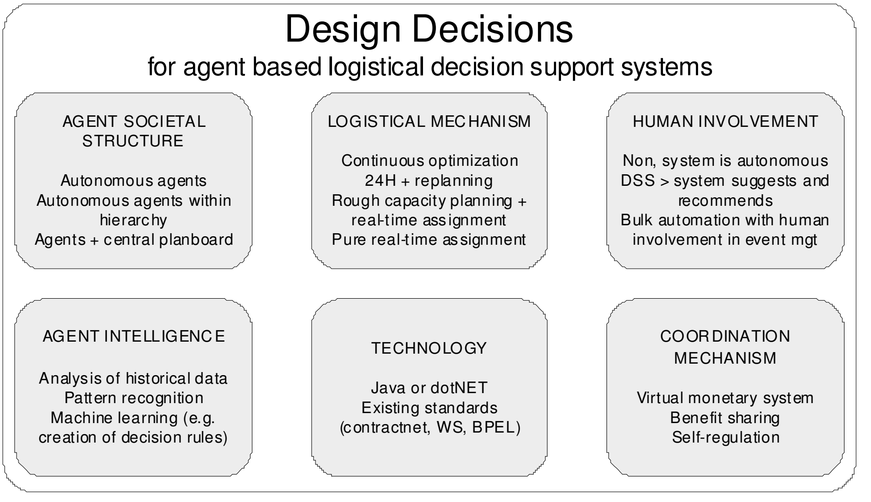


Figure 8.1 – Some design decisions in MAS system design (Douma *et al.*, 2008)

8.4.3 The "façade of false certainty"

A last issue we would like to discuss, is something we like to refer to as the "*façade of false certainty*". In the two cases, but also in other work we were involved in, we found that today's planning practises too often generate a façade of false certainty. This is caused by an implicit design choice for a reduction in the information processing need, to state it in Galbraith's (1974) terms. By fixing a certain date or quantity in advance, the planning has something to work towards. However, often the originally set date or quantity changes, which is too often never properly incorporated in the planning. This generates a lot of unnecessary last-minute fire-fighting, and has resulted in implementations of systems that assist in exactly doing that (= fire-fighting). Perhaps it would be smarter to integrate uncertainty explicitly in the planning, to keep track of changes that impact (un)certainty and update systems accordingly. Planning then becomes a continuous exchange of information with a certain probability. Negotiation and coordination of activities are the primary planning and control techniques, not single-tier optimisation.

Uncertainty is not a problem, as long as it not treated as certain. Layered information exchange through time, and managing uncertainty should be essential elements in future system designs that incorporate it in decisions. Keep on exchanging information: not just once at order intake and initial scheduling, but keep updating schemes and schedules when information becomes more or less certain – see Figure 8.2 for an illustration. Note that in practise, many of these curves (with different due dates and/or quantities) exist in parallel, which is valuable at an aggregated level. In such systems steering becomes a continuous process, with the continuous reconsideration of one's position when new information becomes available. As such, we are the first to agree that the real-time assignment approach we utilised in Chapter 1 is clearly over-simplified for most applications.

Companies in almost all industries have gone through long enterprise software implementations over the past decades. Nowadays, core in most enterprise systems is generally a propagation engine. Service-oriented architectures (SOA) and multi-agent systems (MAS) can be instruments to establish supply chain systems of the future that treat certainty in a different manner, namely, by handling frequent information updates.

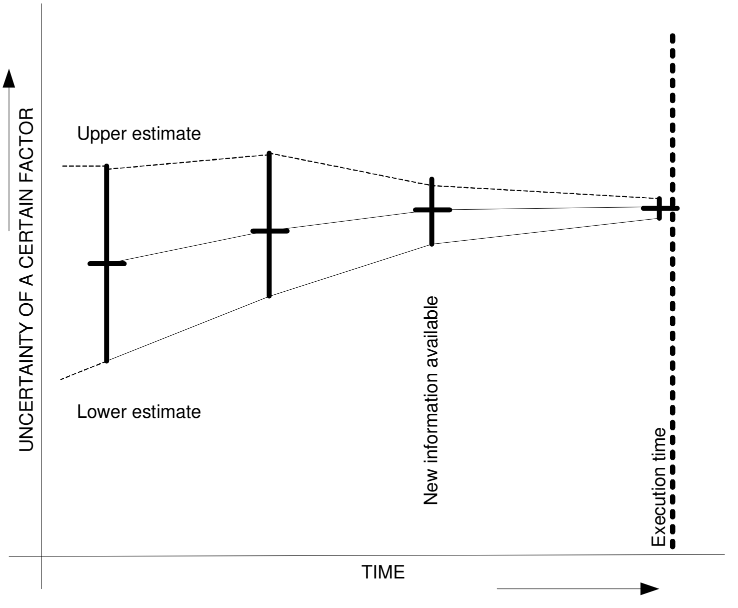


Figure 8.2 – Uncertainty in planning through time

8.5 Closing words

We stand at the beginning of a revolution in planning and organising logistics activities. The right technologies have arrived; now it is time to redesign processes in and between companies. No longer perceive planning as something to protect against uncertainty, but exploit uncertainty to the fullest extent, and utilise smart inter-organisational coordination systems to create the future in enterprising. MAS concepts have large potential to contribute.

Mobile phones have changed the way we coordinate our daily lives. We do not make precise plans anymore, but start with a rough plan, and perform last-minute fine-tuning. However, we make it very complex for ourselves if we do not complete our rough planning in time: otherwise we might end up in the wrong town, at the wrong restaurant, or be there at a wrong time.

However, if all works out fine, we nicely fulfil our appetite! And now, now it is time to digest...

APPENDIX

Appendix A Literature study

This Appendix lists the journals covered in the literature (see Table A.1), lists the academic search engines used for an additional general search (in Table A.2) and lists the most important key words used in the search (Table A.3).

Table A.1 – List of journals covered [2000-2006]

Management Information Systems Quarterly	Management Information Systems Quarterly Executive
Information Systems Research	Information Systems
Informs Journal on Computing	Communications of the ACM
Communications of the AIS Harvard Business Review	Journal of Management Information Systems
California Management Review	MIT Sloan Management Review
Administrative Science Quarterly	Interfaces
Decision Sciences	Management Science
Production Planning & Control	Operations Research
Journal of Operations Management	Transportation Science

Table A.2 – Academic search engines used

ABI/Inform	ACM Digital Library	Google Scholar
ISI Web of Science	IEEE Digital Library	Science Direct

Table A.3 – Key words used

planning	logistics	application	web service	real-time
multi agent systems	distributed system	decision making	supply chain (management)	transportation

Appendix B Potential MAS benefits

This appendix shows a list of potential benefits multi-agent systems can bring enterprise information systems, coming from (Lea *et al.*, 2005).

Table B.1 – Potential benefits MAS architecture for Enterprise IS (Lea *et al.*, 2005)

Implications	Traditional information systems	Commercial ERP suits	MAS based ERP systems
Impact of system failure	Local impact	Global impact	Local impact
Data redundancy, integrity, and accuracy	Data redundancy is likely unavoidable and will result in integrity and accuracy problems	Not a problem	?
Update, modification, and maintenance	Difficult	Easy	Easy
Effective decision making via real-time information sharing across different functional areas	Ineffective decision making due to outdated data or difficulty of obtaining cross functional data	Improved decision quality because real-time information sharing across functional areas	Improved decision quality because agent can obtain most appropriate/accurate information for used for decision
System interdependencies	Often ignored	Considered	Considered
Conflicting goals	Often unavoidable	Not a problem	Not a problem
Needs for business process redesign due to ERP integration	Not applicable	Often unavoidable	Very limited
Technology readiness	Often not a concern	Critical to implementation success	Minimum impact
Implementation cost	Existing	Often underestimated	Minimum
Implementation time	Existing	Often underestimated	Faster implementation
Employee training	Not needed	Intensive training needs	Minimum
Customization capability to support current business processes	Not applicable	Limited	Flexible

Appendix C UML diagrams Post-Kogeko

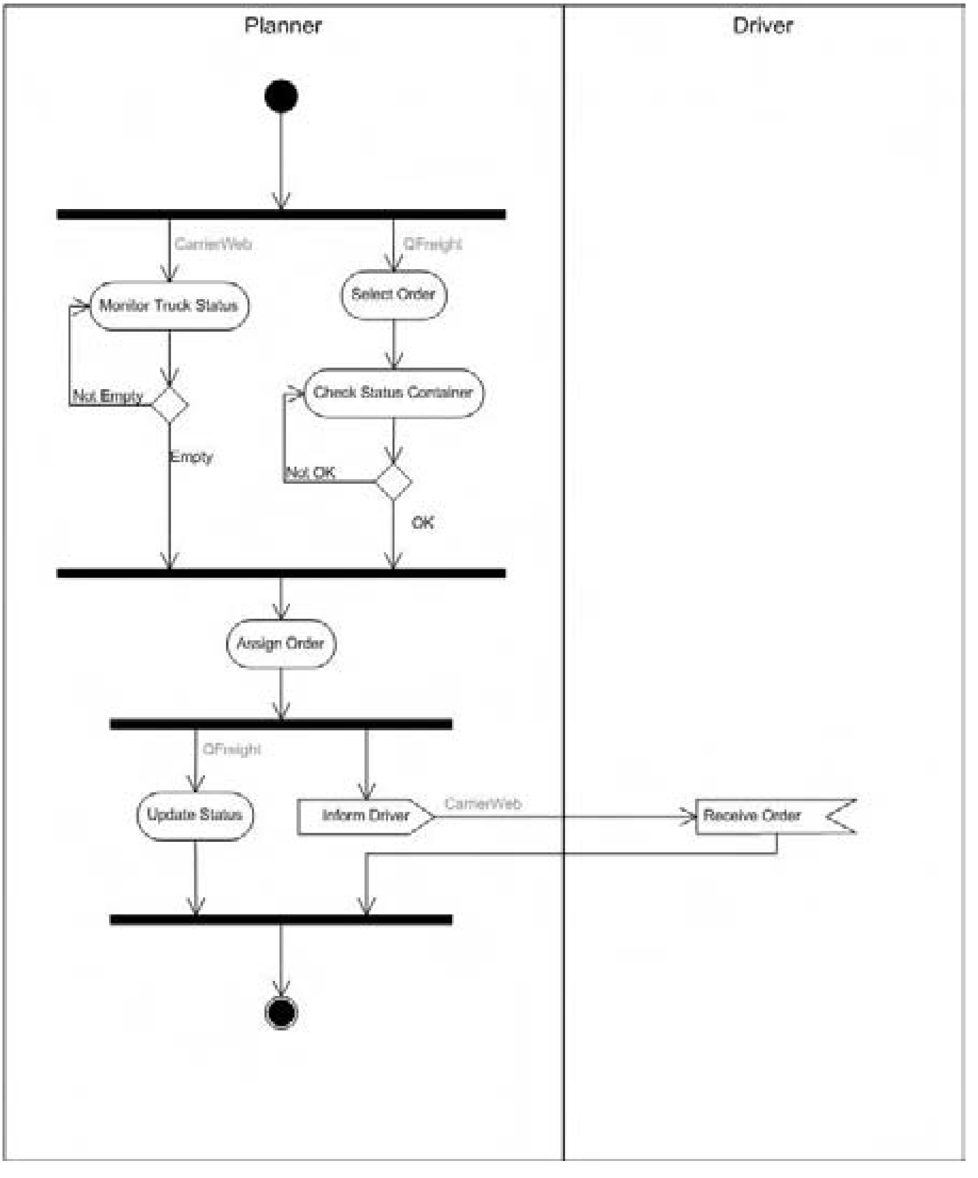


Figure C.1 – Activity diagram – Assign second (or later) order

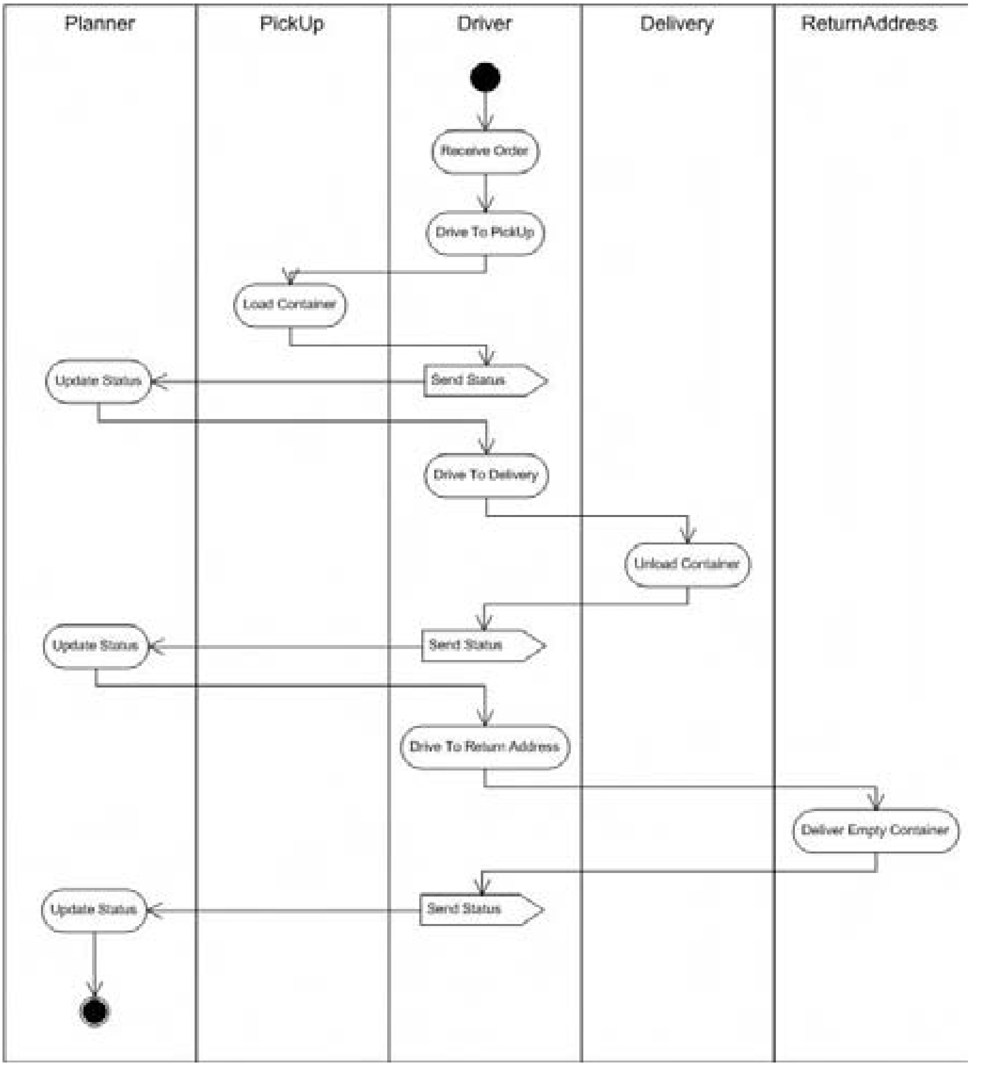


Figure C.2 – Activity-diagram – Execution of an order

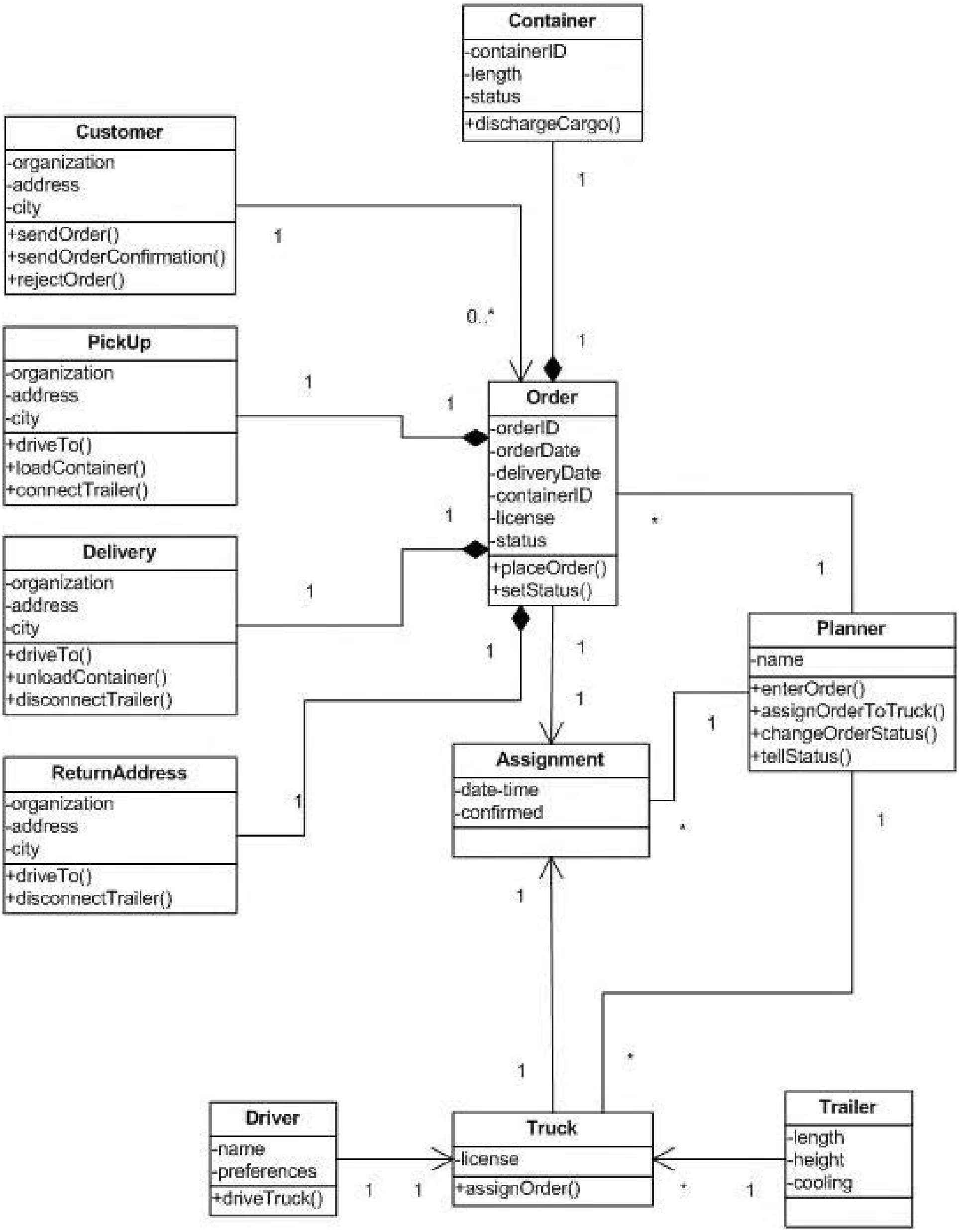


Figure C.3 – UML class diagram showing entities in the container planning

<<TruckAgent>>	
BH-SF-44 / Initiator, Participant	
Role	
Initiator	
Participant	
State-Description	
Truck is empty; trying to find an order that still needs to be executed	
Truck is busy with an order; participant is asked by an initiator agent to calculate a score for a particular order	
Actions	
<<pro-active>>	
Find best order for this truck	
<<re-active>>	
Calculate score for a particular order, and communicate with initiator truckagent	
Methods (public methods only)	
getCurrentLocation()	setCurrentOrderAddresses()
getHomeLocation()	setInitiatorInDF()
getOrderStatus()	setParticipantInDF()
getETA()	getCurrentOrderPickUp()
getCurrentOrderID()	getCurrentOrderDelivery()
getStatusChanged()	getCurrentOrderReturn()
getOrderAnticipatingUpon()	getLocationNode()
getTruckStatus()	getStartTime()
getFirstOrder()	getOrderStatus()
getRate()	
Behaviours	
InitialBehaviour	DelayFindOrderBehaviour
ListenBehaviour	HitTheRoadBehaviour
InitiatorBehaviour	UpdateStatusBehaviour
ParticipantBehaviour	SetETABehaviour
FindOrderBehaviour	
Protocol	
FIPA Interaction Protocol	

Figure C.4 – AUML Class Diagram – TruckAgent (initiator role)

Appendix D Overview feedback sessions

Table D.1 – Feedback sessions throughout the design process

Date	People present	Topic
16/02/2005-18/02/2005	Nico Kroos, Ben van Zeijl, Richard Crans, Alberdine van Velzen, Frans Denie, Hans Moonen, Arthur Oink	On-floor session in the planning department – to understand the planning practice
23/02/2005	Richard, Hans, Arthur	In-depth explanation CarrierWeb
29/03/2005	Richard, Hans, Arthur	Double checking the process descriptions
09/05/2005	Richard, Alberdine, Ben, Frans, Hans, Arthur	Workshop to evaluate the process descriptions and first ideas for improvements
23/06/2005	Ronald van Meurs (CoolControl), Alberdine, Hans, Arthur	Workshop at CoolControl to discuss about their requirements and ideas for process integration
28/07/2005	Ronald, Frans, Richard, Alberdine, Hans, Arthur	Feedback sessions CoolControl findings
26/10/2005	Alberdine, Richard, Fred van Rijn, Hans, Arthur	Discussion about electronic customer interface
21/11/2005 - 24/11/2005	Several experts Erasmus University, Arthur, Hans	In-depth discussion design – for an overview see the list and outcomes in (Oink, 2005).
25/11/2005	Ruud van der Ham (ECT), Jo van Nunen, Arthur, Hans	ECT first discussion about opening up their systems for LSPs.
12/01/2006	Alberdine, Fred, Ben, Hans, Arthur	In-depth feedback session about customer interface for QFreight
10/02/2006	Hans Klok (ECT), Tom Niels (ECT), Jos van Hillegersberg, Hans, Arthur	In-depth discussion about future pilot system and possible connectivity to ECT platforms.
22/05/2006	Alberdine, Richard, Hans, Arthur	In-depth feedback session – ECT connectivity, and prototype development
25/08/2006	Jos, Hans, Arthur	In-depth session prototype development

12/12/2006	Richard, Hans, Arthur	Feedback session prototype
08/05/2007	Alberdine, Geoffrey Robbemond, Hans, Arthur	Evaluation session planning functionality
23/07/2007	Alberdine, Richard, Frans, Hans, Arthur	Workshop prototype evaluation
03/08/2007	Sandra Waenink, Alberdine, Richard, Hans	Interview for TTM – expert validation
07/11/2007	Richard, Frans, Hans, Arthur	Prototype demonstration and discussion about field-test setup
16/01/2008	Arthur, Hans, Jos, Elfriede Krauth, and others	Expert evaluation seminar Transumo
28/04/2008	Ben, Richard, Alberdine, Frans, Hans, Arthur	Setup details field-test + workshop
19/05/2008	Richard, Hans, Arthur	Field test, scheduled but cancelled. Planner Ben sick at home. Technical working of system tested with Richard + first feedback on workings.
29/05/2008	Ben, Hans, Arthur, Richard, Arno Pieper	Field test. Mirror planning + evaluation with two planners.
30/05/2008	Ben, Frans, Alberdine, Richard, Hans, Arthur	Evaluation session internal – ideas for future, preparing for external session
03/06/2008	Ben, Frans, Alberdine, Richard, Hans, Arthur, Tom, Evert van Hoven (ECT), John Monteiro (PortInfolink), Frans van den Nobelen (APM), Jeroen de Rijcke (APM), Ronald, Jos	Evaluation session external parties and discussion future research / concrete implementation

Appendix E Types of Jade Behaviours

Table E.1 – Overview of different types of Behaviours within Jade

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

Appendix F XML files UI

In Table F.1, Table F.2 and Table F.3 examples are given of the XML files used for display in the UI (`trucks.xml` and `orders.xml`) and for logging purposes (`log.xml`).

Table F.1 – Example of log.xml file

```
<?xml version="1.0" ?>
<log>
<entry>
             <name>BH-PP-77</name>
<time>10:41:06</time>
             <message>order (4057532) assigned to
truck</message>
      </entry>
<truck name="BH-PP-77" type="truckIdle">
             <time>1217248630984</time>
             <message>2724781</message>
      </truck>
</log>
```

Table F.2 – Example of orders.xml file

```
<?xml version="1.0" ?>
<orders>
<order availability="true" id="4060048" name="MWCU-676088-1"
status="1">
<pickup lat="519594" lng="40284" name="MAERSK DELTA / APM
TERMINAL" zip code="3199" />
<delivery lat="519580" lng="42114" name="COOL CONTROL B. V."
zip code="2676" />
<return lat="519594" lng="40284" name="MAERSK DELTA / APM
TERMINAL" zip code="3199" />
      </order>
<order availability="true" id="4061541" name="KKFU-117303-5"
status="1">
<pickup lat="519594" lng="40284" name="ECT DELTA CONT.
DIVISION" zip code="3199" />
<delivery lat="522937" lng="47545" name="NIPPON EXPRESS B.
V." zip code="1119" />
<return lat="518775" lng="43037" name="MORCON" zip
code="3197" />
      </order>
</orders>
```

Table F.3 – Example of trucks.xml file


```xml
<?xml version="1.0" ?>
<trucks>
<truck name="BP-ND-76">
		<latitude>519177</latitude>
		<longitude>42514</longitude>
		<eta>14400.0</eta>
		<order>4061684</order>
	</truck>
<truck name="BH-SF-49">
		<latitude>519580</latitude>
		<longitude>42114</longitude>
		<eta>0</eta>
		<order>4059235</order>
	</truck>
</trucks>
```

Appendix G Sourcecode TruckAgent

In this Appendix we include the sourcecode from the TruckAgent as an example. The code is documented where needed. Notice that from the TruckAgent different behaviours are started. This version of the TruckAgent is the one as integrated in the simulator version of the system.

```
package nl.deal.engine.truck;

import jade.content.lang.Codec.CodecException;
import jade.content.lang.sl.SLCodec;
import jade.content.onto.OntologyException;
import jade.content.onto.UngroundedException;
import jade.content.onto.basic.Action;
import jade.core.AID;
import jade.domain.FIPANames;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

import nl.deal.engine.LogBehaviour;
import nl.deal.engine.MyAgent;
import nl.deal.util.order.OrderAddresses;
import nl.deal.util.order.OrderAddressesOntology;
import nl.deal.util.order.OrderDataOntology;

import roadNetwork.RoadNetworkException;
import roadNetwork.SynchronizedRoadNetworkInterface;

/**
 * @author <a href="mailto:aoink@rsm.nl">A.C. Oink</a> and
 *   <a href="mailto:hmoonen@rsm.nl">J.M. Moonen</a> <br>
 *   <a href="http://www.rsm.nl">Rotterdam School of
Management</a> <br>
 *   Erasmus University of Rotterdam <br>
 *   Department of Decision & Information Sciences <br>
 *   Copyright (c) 2008.
 */

public class TruckAgent extends MyAgent {

    private static final long serialVersionUID = -
7856763770847928308L;

    private String currentOrderID = "no_order"; // The order the
truck is processing.
    private long[] homeLocation = new long[2];
    private long[] startAnticipateLocation = new long[2];
```

```
    private long[] currentLocation = new long[2]; // The current
location of the truck.
    private double eta = 0; // The estimated duration till truck
available again (in seconds).
    private long statusChanged; // The last time the status has
changed (in milliseconds).
    private String currentOrderPickUp; // The PickUp-zipcode of
the current order
    private String currentOrderDelivery; // The Delivery-zipcode
of the current order
    private String currentOrderReturn; // The Return-zipcode of
the current order
    protected ServiceDescription[] sd; // The service description
(DF registration)
    private String orderAnticipatingUpon = "null";      // The
order anticipating upon
    private String truckStatus;
    private boolean firstOrder = true;
    private long locationNode;// If busy with order return node,
else current location.
    private long rate;        // The simulation rate.
    private long startTime;   // The starttime of the truck

    /* ----------------------------------------------- */

protected void setup() {

DEBUG = false;

// The start address of the truck.
String startZipcode = "2676";

// Register the codec for the SL0 language.
getContentManager().registerLanguage(new SLCodec();
FIPANames.ContentLanguage.FIPA_SL0);

// Register the ontology used by this application.
getContentManager().registerOntology(OrderDataOntology.getInstanc
e());
getContentManager().registerOntology(OrderAddressesOntology.getIn
stance());

// Register the agent in the yellow pages as an order agent.
sd = new ServiceDescription[3];
ServiceDescription tempSD = new ServiceDescription();

tempSD.setType("truck");
tempSD.setName(getLocalName() + "-truck");
sd[0] = tempSD;

// Register currently a participant; other option is:
'initiator'.
```

```
tempSD = new ServiceDescription();
tempSD.setType("participant");
tempSD.setName(getLocalName() + "-participant");
sd[1] = tempSD;

// Register truck idle; other options are: 'busy' and
'anticipating'.
tempSD = new ServiceDescription();
tempSD.setType("idle");
tempSD.setName(getLocalName() + "-idle");
sd[2] = tempSD;

register(this, sd);

truckStatus = "idle";
statusChanged = System.currentTimeMillis();

// Start behaviours after receiving rate and time.
addBehaviour(new InitialBehaviour(this));

// The truck calculates and sets its start-location, before it
starts operations
setStartLocation(startZipcode);

debug("Agent " + getAID().getLocalName() + " is up and running");
    }

    /* ----------------------------------------------- */

    protected void takeDown() {

// Printout a dismissal message
debug("Agent " + getAID().getLocalName() + " terminating.");

deregister(this);
    }

    /* ----------------------------------------------- */

    public long[] getCurrentLocation() {
return currentLocation;
    }

    /* ----------------------------------------------- */

    public long[] getHomeLocation() {
return homeLocation;
    }

    /* ----------------------------------------------- */

    public int getOrderStatus() {
```

```
if (!currentOrderID.equals("no_order")) {

// Request the status from the OrderAgent.
ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
msg.addReceiver(new AID(currentOrderID, AID.ISLOCALNAME));
msg.setConversationId("GET_STATUS");
msg.setReplyWith("status" + System.currentTimeMillis());
this.send(msg);

// Wait for a reply.
MessageTemplate mt =
MessageTemplate.and(MessageTemplate.MatchConversationId("GET_STAT
US"),
MessageTemplate.MatchInReplyTo(msg.getReplyWith()));
ACLMessage reply = this.receive(mt);

// As long as there is no reply, the thread pause (steps 10
millisecs)
while (reply == null) {
try {
          Thread.sleep(10);
    } catch (InterruptedException e) {
          e.printStackTrace();
    }
    reply = this.receive(mt);
}

// Reply received; retrieve the current time in seconds.
int status = Integer.parseInt(reply.getContent());
return status;

} else
return 0;
}

    /* ----------------------------------------------- */
    public double getETA() {
return eta;
    }

    public String getCurrentOrderID() {
return currentOrderID;
    }

    public long getStatusChanged() {
return statusChanged;
    }

    public String getOrderAnticipatingUpon() {
return orderAnticipatingUpon;
    }
```

```
    public String getTruckStatus() {
return truckStatus;
    }

    public boolean getFirstOrder() {
return firstOrder;
    }

    public long getRate() {
return rate;
    }

    /* ---------------------------------------------- */
    /**
     * Set the current order adresses. Request these from the
OrderAgent.
     */
    public void setCurrentOrderAddresses() {

// Request the data from the Order Agent.
ACLMessage msg = new ACLMessage(ACLMessage.REQUEST);
msg.addReceiver(new AID(currentOrderID, AID.ISLOCALNAME));
msg.setConversationId("GET_ADDRESSES");
msg.setReplyWith("order_addresses" + System.currentTimeMillis());
this.send(msg);

// Wait for a reply.
MessageTemplate mt =
MessageTemplate.and(MessageTemplate.MatchConversationId("GET_ADDR
ESSES"),
MessageTemplate.MatchInReplyTo(msg.getReplyWith()));
ACLMessage reply = this.receive(mt);

// As long as there is no reply, the thread will pause (in steps
of 10 milliseconds)
while (reply == null) {
try {
Thread.sleep(10);
} catch(Exception e) {
          e.printStackTrace();
}
reply = this.receive(mt);
}

reply.getContent();

// Initialize the result object
OrderAddresses result = null;

try {
```

```
Action a =
(Action)super.getContentManager().extractContent(reply);
result = (OrderAddresses)a.getAction();

currentOrderPickUp = result.getPickupAddress();
currentOrderDelivery = result.getDeliveryAddress();
currentOrderReturn = result.getReturnAddress();
locationNode = Long.parseLong(result.getReturnNode());

} catch (UngroundedException e) {
e.printStackTrace();
} catch (CodecException e) {
e.printStackTrace();
} catch (OntologyException e) {
e.printStackTrace();
}
    }

    /* ----------------------------------------------- */

    protected void setHomeLocation(long latitude, long longitude)
{

homeLocation[0] = latitude;
homeLocation[1] = longitude;
    }

    /* ----------------------------------------------- */

    protected void setStartAnticipateLocation(long latitude, long
longitude) {

startAnticipateLocation[0] = latitude;
startAnticipateLocation[1] = longitude;
    }

    /* ----------------------------------------------- */
    /**
     * Set the current location of the truck; sends a relocate
message to the simulator.
     * @param latitude latitude of the truck's current position
     * @param longitude longitude of the truck's current position
     * @throws RoadNetworkException
     */
    protected void setCurrentLocation(long latitude, long
longitude) {

// Update parameters.
currentLocation[0] = latitude;
currentLocation[1] = longitude;

// Update locationNode if truck is not busy with an order.
```

```
if (!truckStatus.equals("busy")) {
try {
locationNode =
SynchronizedRoadNetworkInterface.nodeFromLatitudeLongitude
(latitude, longitude);
} catch (RoadNetworkException e)
}
    }

    /* ----------------------------------------------- */

    /**
     * Sets the order status of the truck. This means that the
OrderAgent is updated on
 * its new status, and that the TruckAgent updates its
description with the DF.
     * @param orderStatus the OrderStatus of the truck
     */
    protected void setOrderStatus(int orderStatus) {

// Update the description of the TruckAgent in the DF registry
// First, set whether the truck is available or not
if (orderStatus < 7) {
ServiceDescription tempSD = new ServiceDescription();
tempSD.setType("busy");
tempSD.setName(getLocalName() + "-busy");
sd[2] = tempSD;
} else if (orderStatus == 7) {
ServiceDescription tempSD = new ServiceDescription();
tempSD.setType("idle");
tempSD.setName(getLocalName() + "-idle");
sd[2] = tempSD;
}

modify(this, sd);

// Send a message to the OrderAgent to update it's status.
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(new AID(currentOrderID, AID.ISLOCALNAME));
msg.setConversationId("UPDATE_STATUS");
msg.setContent(Integer.toString(orderStatus));
this.send(msg);
super.debug("Order status changed; message forwarded to the
OrderAgent");

if (orderStatus == 1) {
setCurrentOrderAddresses();
} else if (orderStatus == 7) {

// Clear variables.
currentOrderPickUp = null;
currentOrderDelivery = null;
```

```
currentOrderReturn = null;
currentOrderID = "no_order";
this.setTruckStatus("idle");

super.debug("Order completed.");

// Find new order.
msg = new ACLMessage(ACLMessage.REQUEST);
msg.addReceiver(this.getAID());
msg.setConversationId("FIND_ORDER");
this.send(msg);
}
    }

    /* ----------------------------------------------- */
    protected void setETA(double eta) {

this.eta = eta;
    }

    /* ----------------------------------------------- */
    protected void setCurrentOrderID(String currentOrderID) {

this.currentOrderID = currentOrderID;
    }

    /* ----------------------------------------------- */
    protected void anticipateOrder(String orderID) {

this.currentOrderID = null;
this.eta = 0;

//new TruckUtil(this).addOrder(currentOrder.getID());

// Send a message to inform the OrderAgent that it has a truck
anticipating
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(new AID(orderID, AID.ISLOCALNAME));
msg.setConversationId("UPDATE_ANTICIPATE");
msg.setContent("true");
msg.setReplyWith("update_anticipate" +
System.currentTimeMillis());
super.send(msg);
super.debug("Truck started to anticipate! Message forwarded to
OrderAgent");

// Wait for a reply.
MessageTemplate mt = MessageTemplate.and(MessageTemplate.
MatchConversationId("UPDATE_ANTICIPATE"),
MessageTemplate.MatchInReplyTo(msg.getReplyWith()));
ACLMessage reply = this.receive(mt);
```

```
// As long as there is no reply, the thread will pause (in steps
of 10 milliseconds)
while (reply == null) {
try {
Thread.sleep(10);
} catch(Exception e) {
          e.printStackTrace();
}
reply = this.receive(mt);
}

String answer = reply.getContent();

// Only if DONE returns, than do:
if (answer.equalsIgnoreCase("done")){
// Update the description in the DF registery
ServiceDescription tempSD = new ServiceDescription();
tempSD.setType("anticipating");
tempSD.setName(getLocalName() + "-anticipating");
sd[3] = tempSD;
modify(this, sd);

this.orderAnticipatingUpon = orderID;

// The message "not_allowed" returned; which means that the order
is anticipated
// upon by another truck already; therefore, restart the search
} else {
this.orderAnticipatingUpon = "null";

// Find new order.
msg = new ACLMessage(ACLMessage.REQUEST);
msg.addReceiver(this.getAID());
msg.setConversationId("FIND_ORDER");
this.send(msg);

super.debug("Order anticipation went wrong... find next
order...!");
}
    }

    /* ----------------------------------------------- */
    protected void clearAnticipateOrder() {

// Send a message to inform the OrderAgent that the truck stopped
anticipating
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(new AID(this.orderAnticipatingUpon,
AID.ISLOCALNAME));
msg.setConversationId("UPDATE_ANTICIPATE");
msg.setContent("false");
super.send(msg);
```

```
super.debug("Truck informed order that it stopped anticipation on
an order!");

// Update the agentdescription in the DF registery
ServiceDescription tempSD = new ServiceDescription();
tempSD.setType("idle");
tempSD.setName(getLocalName() + "-idle");
sd[2] = tempSD;
modify(this, sd);

// Clear the variable in the agent
this.orderAnticipatingUpon = "null";

    }

    /* ---------------------------------------------- */
    protected void setFirstOrder(boolean firstOrder) {
this.firstOrder = firstOrder;
    }

    protected void setOrderAnticipatingUpon(String
orderAnticipatingUpon) {
this.orderAnticipatingUpon = orderAnticipatingUpon;
    }

    protected void setRate(long rate) {
this.rate = rate;
    }

    public void setStartTime(long startTime) {
this.startTime = startTime;
    }

    /* ---------------------------------------------- */
    protected void setTruckStatus(String newTruckStatus) {

long currentTime = System.currentTimeMillis();
String timeActivity = Long.toString(currentTime - statusChanged);
String activity = null;

if (this.truckStatus.equals("idle"))
activity = "truckIdle";
else if (this.truckStatus.equals("busy"))
activity = "truckBusy";
else if (this.truckStatus.equals("anticipating")) {
addBehaviour(new LogBehaviour(this, "truck", "kmSetup",
Long.toString(calculateAnticipateKM())));
activity = "truckAnticipating";
}

addBehaviour(new LogBehaviour(this, "truck", activity,
timeActivity));
```

```
this.truckStatus = newTruckStatus;
statusChanged = currentTime;
    }

    /* --------------------------------------------- */
    /**
     * Set a start location.
     * @param startZipcode the starting zipcode of the truck
     */
    private void setStartLocation(String startZipcode) {

try {
locationNode =
SynchronizedRoadNetworkInterface.nodeFromName(startZipcode);
long latitude =
SynchronizedRoadNetworkInterface.latitudeOfNode(locationNode);
long longitude =
SynchronizedRoadNetworkInterface.longitudeOfNode(locationNode);

setCurrentLocation(latitude, longitude);
setHomeLocation(latitude, longitude);

// Create and relocate the truck in the simulator.
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(new AID("Sync", AID.ISLOCALNAME));
msg.setConversationId("RELOCATE");
msg.setContent("<relocate truck=\"" + this.getLocalName() + "\">"
+
"<longitude>" + currentLocation[1] + "</longitude>" +
"<latitude>" + currentLocation[0] + "</latitude>" +
"</relocate>");
super.debug("XML-message created: " + msg.getContent());

// Send the simulator message to the SyncAgent
send(msg);

super.debug(this.getLocalName() + "'s start location is: " +
startZipcode + " (" + latitude + ", " + longitude + ")");

} catch (RoadNetworkException e) {
e.printStackTrace();
}
    }

    /* --------------------------------------------- */
    public void setInitiatorInDF() {

// Register that the Truck has become the initiator
ServiceDescription tempSD = new ServiceDescription();
tempSD.setType("initiator");
tempSD.setName(getLocalName() + "-initiator");
```

```
sd[1] = tempSD;

modify(this, sd);
    }

    /* ----------------------------------------------- */
    public void setParticipantInDF() {

// Register that the Truck has become the initiator
ServiceDescription tempSD = new ServiceDescription();
tempSD.setType("participant");
tempSD.setName(getLocalName() + "-participant");
sd[1] = tempSD;

modify(this, sd);
    }

    /* ----------------------------------------------- */
    public String getCurrentOrderPickUp() {
return currentOrderPickUp;
    }

    public String getCurrentOrderDelivery() {
return currentOrderDelivery;
    }

    public String getCurrentOrderReturn() {
return currentOrderReturn;
    }

    public long getLocationNode() {
return locationNode;
    }

    public long getStartTime() {
return startTime;
    }

    /* ----------------------------------------------- */
    /**
     * Translates a truckstatus number to human readable string.
     * @param s the number
     * @return the string
     */
    public static String getOrderStatus(int s) {

String status = null;

switch (s) {
case 0:
status = "no order assigned.";
break;
```

```
case 1:
status = "on my way to the pick-up location.";
break;
case 2:
status = "at the pick-up location.";
break;
case 3:
status = "on my way to the delivery location.";
break;
case 4:
status = "at the delivery location.";
break;
case 5:
status = "on my way to the return location.";
break;
case 6:
status = "at the return location.";
break;
case 7:
status = "order completed.";
break;
}
return status;
    }

    /* ------------------------------------------------ */
    private long calculateAnticipateKM() {

long result = 0;

try {
long nodeStart = SynchronizedRoadNetworkInterface.
nodeFromLatitudeLongitude(startAnticipateLocation[0],
startAnticipateLocation[1]);
long nodeEnd = SynchronizedRoadNetworkInterface.
nodeFromLatitudeLongitude(currentLocation[0],
currentLocation[1]);
result = SynchronizedRoadNetworkInterface.
distance(nodeStart, nodeEnd) / 10;

} catch (RoadNetworkException e) {
e.printStackTrace();
}
return result;
    }
}
```

Appendix H Sourcecode ListenBehaviour

In this Appendix we include the sourcecode from the ListenBehaviour from the TruckAgent as an example; this behaviour listens for messages from other agents, and reacts upon.

```
package nl.deal.engine.truck;

import jade.core.behaviours.CyclicBehaviour;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

/**
 * @author <a href="mailto:aoink@rsm.nl">A.C. Oink</a> and
 *    <a href="mailto:hmoonen@rsm.nl">J.M. Moonen</a> <br>
 *          <a href="http://www.rsm.nl">Rotterdam School of
Management</a> <br>
 *          Erasmus University of Rotterdam <br>
 *          Department of Decision & Information Sciences <br>
 *          Copyright (c) 2008.
 */

public class ListenBehaviour extends CyclicBehaviour {

    private static final long serialVersionUID =
6167165414188594331L;
    private TruckAgent myAgent;

    /**
     * Constructor.
     * @param myAgent The agent this behaviour belongs to.
     */
    public ListenBehaviour(TruckAgent myAgent) {
super(myAgent);
this.myAgent = myAgent;
    }

    /* ----------------------------------------------- */

    /**
     * This behaviour listens for messages from other agents.
     */
    public void action() {

MessageTemplate m1 = MessageTemplate.MatchConversationId("ETA");
MessageTemplate m2 =
MessageTemplate.MatchConversationId("DELETE");
MessageTemplate m1orm2 = MessageTemplate.or(m1, m2);

ACLMessage msg = myAgent.receive(m1orm2);
```

```
if (msg != null) {
if (msg.getConversationId().equals("ETA")) {
// The TruckAgent received a request for it's ETA.
ACLMessage reply = new ACLMessage(ACLMessage.INFORM);
reply.addReceiver(msg.getSender());
reply.setConversationId("ETA");
reply.setContent(Double.toString(myAgent.getETA()));
reply.setInReplyTo(msg.getReplyWith());
myAgent.send(reply);

} else if (msg.getConversationId().equals("DELETE")) {
// The TruckAgent received an instruction to delete itself.
try {
myAgent.doDelete();
} catch(Exception e) {
e.printStackTrace();
}
}

} else {
block();
}}
```

Appendix I Sourcecode InitiatorBehaviour

In this Appendix we include the sourcecode from the InitiatorBehaviour from the TruckAgent as an example. The code is documented where needed. The InitiatorBehaviour runs when a TruckAgent takes up the initiator-role, and searches actively for a new order.

```
package nl.deal.engine.truck;

import java.util.Random;

import nl.deal.engine.MyAgent;

import jade.core.behaviours.CyclicBehaviour;
import jade.domain.DFService;
import jade.domain.FIPAException;
import jade.domain.FIPAAgentManagement.DFAgentDescription;
import jade.domain.FIPAAgentManagement.ServiceDescription;
import jade.lang.acl.ACLMessage;
import jade.lang.acl.MessageTemplate;

/**
 * @author <a href="mailto:aoink@rsm.nl">A.C. Oink</a> and
 *   <a href="mailto:hmoonen@rsm.nl">J.M. Moonen</a> <br>
 *          <a href="http://www.rsm.nl">Rotterdam School of
Management</a> <br>
 *          Erasmus University of Rotterdam <br>
 *          Department of Decision & Information Sciences <br>
 *          Copyright (c) 2008.
 */

public class InitiatorBehaviour extends CyclicBehaviour {

private static final long serialVersionUID = -
2662106840521483150L;

private TruckAgent myAgent;

/**
 * Constructor.
 * @param a      the agent this behaviour belongs to
 */
public InitiatorBehaviour(TruckAgent myAgent) {

super(myAgent);
this.myAgent = myAgent;

// Find the first order.
findOrder();
```

```
}

/* ----------------------------------------------- */
/**
 * This behaviour is activated when a REQUEST-message is send to
the agent.
 * This is an action initiated from the GUI.
 * Or, in automatic mode, when an agent finishes the previous
order.
 */

public void action() {

MessageTemplate mt = MessageTemplate.and(
MessageTemplate.MatchPerformative(ACLMessage.REQUEST),
MessageTemplate.MatchConversationId("FIND_ORDER"));
ACLMessage msg = myAgent.receive(mt);

if (msg != null) {
// INFORM-message received; process it...
myAgent.debug("Received a REQUEST-message from "
+ msg.getSender().getLocalName() + "; now acting as initiator.");

// ... find a new order.
findOrder();
} else {
block();
}
}

/* ----------------------------------------------- */
/**
 *  FindOrder. First check whether another truckagent is
initiator (by consulting
 * the DF) IF yes, check again after a couple of seconds. IF not,
claim the
 * initiator status (update the DF description) and start the
process
 */

private void findOrder() {

int numberOfInitiators = 0;

//  Double check, by consulting the DF
try {
DFAgentDescription dfd  = new DFAgentDescription();
ServiceDescription sd = new ServiceDescription();
sd.setType("initiator");
dfd.addServices(sd);

DFAgentDescription[] result = DFService.search(myAgent, dfd );
```

```
numberOfInitiators = result.length;

} catch (FIPAException fe) {
fe.printStackTrace();
}


// In case there is another initiator active, add a new
DelayFindOrder behaviour
if (numberOfInitiators > 0) {
Random generator = new Random();
int timeToWait = 1000 + generator.nextInt(19000);

// Another agent holds the initiator-lock, try again later.
int period = (int) (timeToWait / myAgent.getRate());

if (period < 1000)
period = 1000;

myAgent.addBehaviour(new DelayFindOrderBehaviour(myAgent,
period));

// No other initiator, claim the initiator position, and start
search for new order.
} else {
// Update the DF
myAgent.setInitiatorInDF();

// Start looking for a new order.
myAgent.addBehaviour(new FindOrderBehaviour(myAgent));
}
}

}
```

Appendix J Design problems

In this Appendix we discuss the problems listed in Table 5.5 in more detail. The five engineering decisions are discussed in different sections below.

J|1 Supertruck eats up orders

Due to the way we calculate scores, and consult competitive (participant) TruckAgents, we ran into the problem that a well positioned TruckAgent “eats up” all orders from a less well positioned (initiator) TruckAgent. Well positioned does not only involve the physical location, but also the other elements from the scoring mechanism (such as the fit with the customer time window).

This is a natural result of the fact that we decided to not construct any plans longer than the current order, and the fact that no single agent oversees the entire set of orders, capacities, and possible combinations.

In principal the mechanism works as initially designed, since we do not want to assign orders to a TruckAgent that is less well positioned than a competitor (who is no initiator yet, but will become this again in due time (since that has been included in the scoring)). However, it would be very strange if a certain participant wins 7 out of 12 auctions. This agent might be better positioned for these orders than the current initiator agent, considering each order individual; however, it will never be able to execute all these orders – as it also has to go through a round of scoring and auctioning with competitor agents.

As a matter of fact we decided for the pragmatic solution to remove winning participants from any further (“next best”) order auctions. Note that this does not result in any assignments for those participants; it only reduces complexity in the mechanism. TruckAgents that become empty, still have to go through the mechanism, and might well end up with another order than the one earlier won in the previous auctions. This engineering choice thus prevents “supertruck” behaviour, and accelerates the assignment process of a truck (as the set of participants reduces with each auction won by a participant, resulting in less interactions and calculations to be made).

J|2 TruckAgents go home

As the amount of (available) orders to execute reduces the situation can arise that initiator TruckAgents cannot find a match with an order anymore on which they outperform the participant agents. In this particular case the question arises whether the TruckAgent should go home, and leave the work to the other TruckAgents, or remain active, and for example anticipate on an order that is still unavailable, or just wait at the current location for future work. The issue is that on a system (or corporate) level, there should remain enough trucks active in the system in order to serve the remaining orders in due time. In fact, we experimented with two mechanisms. The first mechanism was to let the ManagementAgent decide whether a TruckAgent may go home, if it asks the ManagementAgent to go home. The ManagementAgent kept track of the amount of trucks and orders in the system, and thus could estimate whether enough capacity was available. In principal this worked, however, it was more logical, and better in line with agent based design, to let the TruckAgent make the decision itself; by contacting the DF to consult the amount of work still to be performed, and the amount of trucks still active. The second mechanism which is now integrated became TruckAgents that reason themselves whether they should go home. In case they go home, they update the DF.

J|3 Orders that cannot be served on time anymore

The membership function for the CustomerTimeWindow in the scoring mechanism aims at delivery within the specified timewindow, through a trapezoid styled-function – see Figure J.1. When comparing between two TruckAgents this is a useful mechanism to choose for a truck that delivers the order on-time. However, should the function truly be trapezoid styled? Should not orders that are late be delivered as soon as possible? Thus, should the function look as Figure J.2? Yes and no. The scoring should be implemented as such that a late order gets a higher priority. However, if we implement it simply as in the Figure, we get the behaviour that TruckAgents score better for late delivery, since the priority scores rank higher than. In fact, when two identically positioned TruckAgents bid on the same order, one on time, one too late, the latter will be scoring higher. That is unwanted behaviour.

As such we decided that indeed orders should get an absolute priority. But only, in case no truck is possible anymore to deliver such an order in time. We refer to the latter in our code as GloballyTooLate. This since otherwise the scoring mechanism (rating higher for too late orders) might result in delaying orders until they are "just too late". As a matter of fact we utilize a combination of two membership functions for our scoring mechanism, namely the earlier discussed trapezoid function for "on-time" orders (see Figure J.1), and an exponentially descending membership function as depicted in Figure J.3 for orders that cannot be delivered in time anymore by any truck. As such, all the agents bidding use either one of the two curves, but all utilize the same curves in the same bidding round.

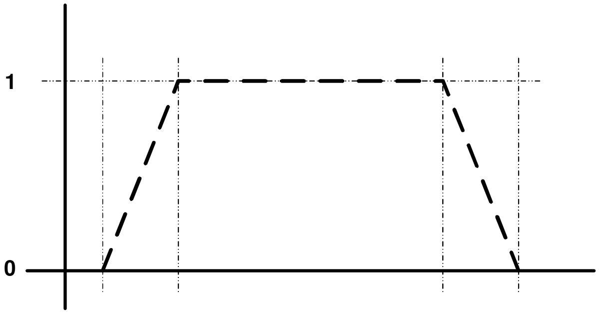


Figure J.1 – Trapezoid function

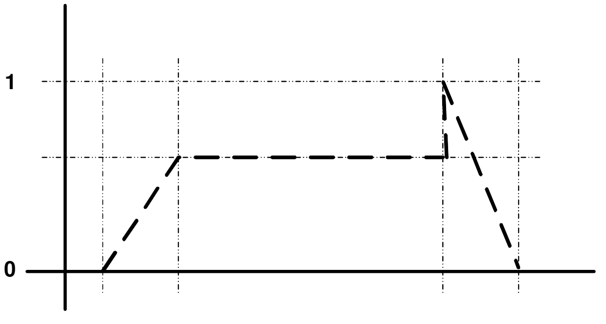


Figure J.2 – Customer time window scoring function

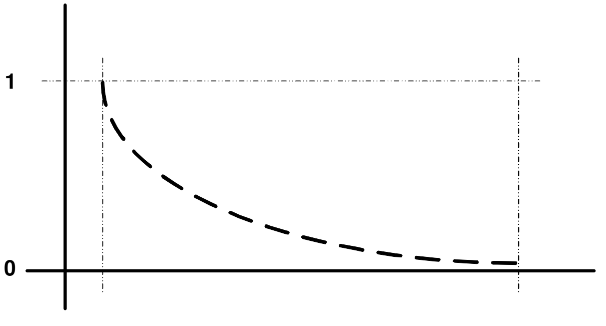


Figure J.3 – Too late, but now prioritized function (SFuzzySet)

J|4 Idle trucks

How to handle idle trucks; that means: trucks that are not busy (anymore) with an order, but are not allowed to go home yet? Should these trucks reside and wait at their current location? And let the TruckAgent try to find an order again in a couple of minutes? Or should anticipation behaviour be included that anticipates on expected orders? Anticipation as such means driving to the pickup address of a not yet available order.

We decided that trucks should try to anticipate on future orders in case they become idle. An anticipating truck reconsiders the possible orders every so many minutes (by taking the initiator role up again). This way trucks can be expected to arrive earlier at the new location, and utilize their idle time better.

Please note that in practice such waiting times are often spend for different purposes, such as having lunch, filling gas, or for administrative purposes. For the simulation it has however been included in the code.

J|5 Human decision making

Initially we designed the system as a fully autonomous planning engine, which needed no manual assistance. In a real implementation this scenario is not very likely; most companies do want to keep the end control in the hands of an experienced planner.

This influences the system and its design heavily. The system has to become more like a Decision Support System (DSS), and many of the choices discussed above are no longer the type of aspects the system truly needs. The system would operate more in a sense that it makes suggestions to the planner. For example: new assignments, or trucks to go home. The planner makes the final decision. Also a visual interface, and reporting functionality become very important. For the field test we made several of these changes; these are documented in more detail in section 5|7.

Appendix K Agent communication

The figures in this appendix show how the sequence of interactions between TruckAgents, OrderAgents and the DF should be (see Figure K.1), and sometimes turned out to be in the simulations (see Figure K.2).

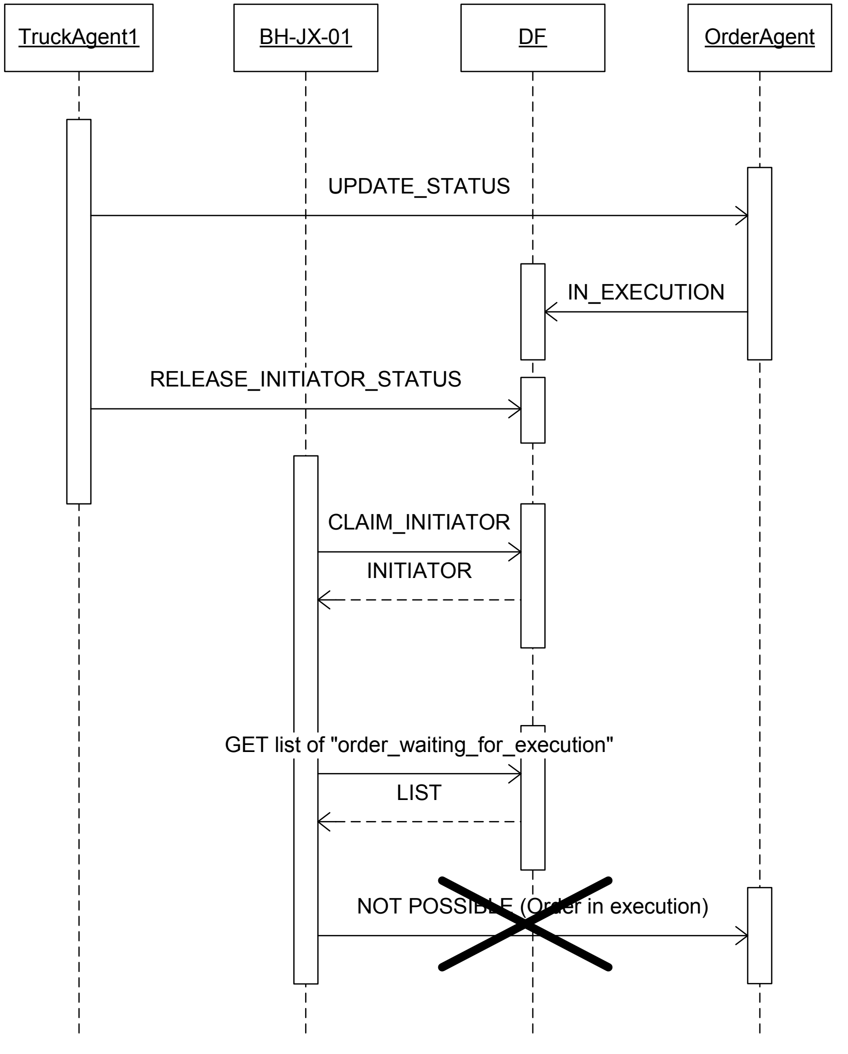


Figure K.1 – Should Be, TruckAgent-DF-OrderAgent communication sequence

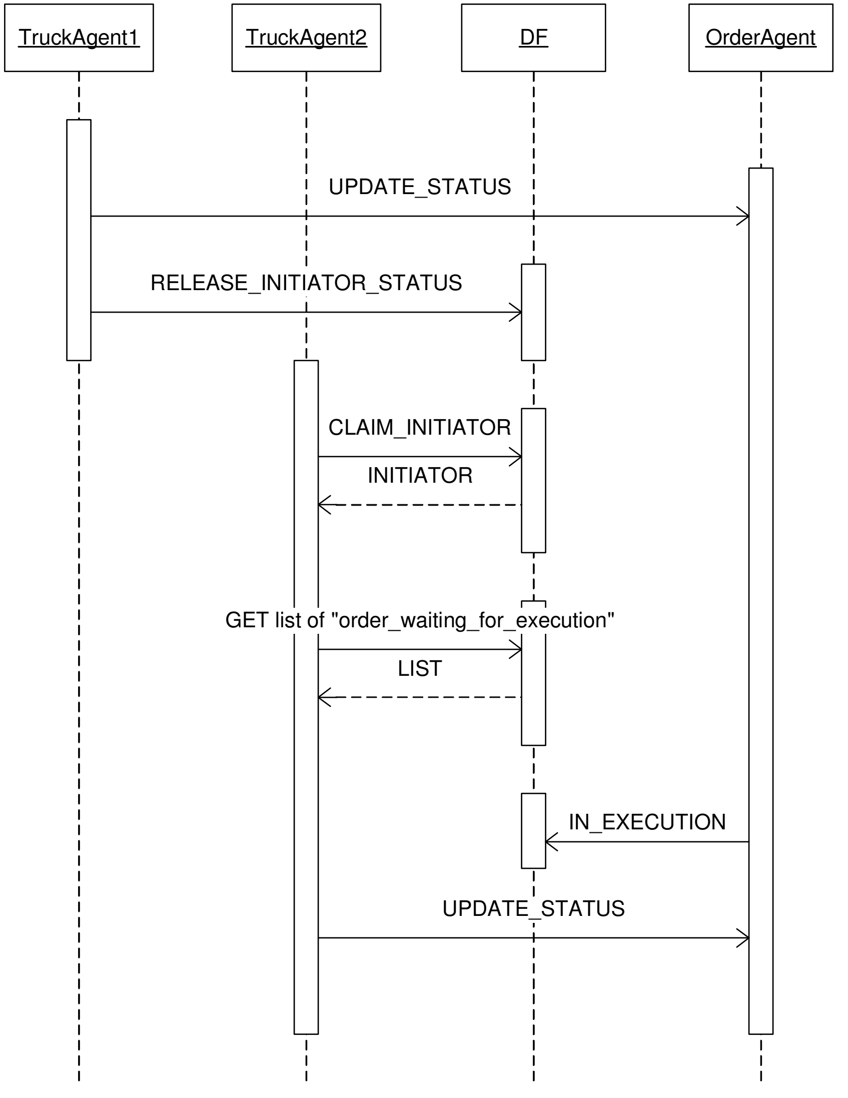


Figure K.2 – Wrong TruckAgent-DF-OrderAgent communication sequence

Appendix L Expert feedback

In this Appendix we include the expert feedback we received from the workshops as documented in 5|6 (on page 93), 6|6 (on page 129) and 7|2 (on page 137). Table L.1 can serve as an index to the other tables in this appendix. Please note that the "From" column contains either an F or a D. The F stands for Feedback Form, the D for Discussion in Group. The workshop questions the last column refers to are listed in Table 5.13.

Table L.1 – Index of tables with expert feedback

Where	What	From	Which workshop question
Table L.2	Advantages road transport prototype	F	Question I
Table L.3	Disadvantages road transport prototype	F	Question I
Table L.4	Feedback on how to support the planner	F	Question II
Table L.5	Advantages barge planning prototype	F	Question III
Table L.6	Disadvantages barge planning prototype	F	Question III
Table L.7	Advantages of the exchange of waiting profiles for barges	F	Question IV
Table L.8	Disadvantages of the exchange of waiting profiles for barges	F	Question IV
Table L.9	Success factors that drive adoption	F	Question V
Table L.10	Factures that explain adoption failure	F	Question VI
Table L.11	Potential other applications	F	Question VII
Table L.12	Results from the discussion in the first group in the road transport workshop	D	Questions I & II
Table L.13	Results from the discussion in the second group in the road transport workshop	D	Questions I & II

Table L.2– Advantages road transport prototype

Cluster	Frequency	Feedback	Comment
Every good planning system	2		Less empty miles for trucks
	2		Increase efficiency, reduce losses
Specific advantages agent concept			Enables incident management approach
	4		Fast & real-time information processing and decision making
			Current systems difficult to change
	2		Robustness due to decentralized implementation (continuity in case of system failures)
			Possibility to acquire external orders from electronic marketplaces (due to autonomy of the truck/driver combination)
	2		Chain application
			Online planning makes it possible to improve the performance of the chain
			Learning characteristic of agents
Specific IOS application			Information coordination and negotiation of such a solution is a large advantage of the shown solution
			A negotiation driven approach makes it possible to better balance the peaks at container terminals
	3		The possibility to improve terminal processing and waiting times due to better coordination
			Well connected to the current manual way of working, with local decision autonomy
			Environmental advantages due to better coordination and less "waste"
	3		Better levelled use of the road network
General comments			The concept looks good and convincing - imho applicable in real practice
			I like this very much: people cannot handle the dynamics and complexity of the current reality

Table L.3 – Disadvantages road transport prototype

Cluster	Frequency	Comment
Aspects of the concept		Objectives of an agent can be conflicting: handle the current order, versus jumping on a future order
		New generation software developers needed
	3	Who is in control when several agents compete for the same order? The coordination of decisions can be difficult.
		Can/should you use standard business rules?
		Many communication lines and interfaces

	Stock-exchange effect: a small disturbance somewhere can result in an explosion somewhere else The decision process is less tangible
Aspects of the system (suggestions for improvements)	Opening-times of parties throughout the chain (specifically terminals & customers) is an important factor to include in the decision process Exchanging too much information can influence someone's competitive position Coordination of trips (loads & customer) needs to be very solid in order to work Process to handle exceptions Translate knowledge of planner to the agent system Build in smarter algorithms (an example: more centralized mechanisms) This is only the operational level; what about the strategic and tactical levels; when someone has more time to react and optimize. How do such systems integrate and link? Should be capable of calculating and compare several (possible future) scenarios Integration needed with higher-level planning tools How does this compare with other systems that do real-time optimization? Inflexible aspect is the lacking autonomy of a driver (it is the truck-agent who decides); shouldn't there be a driver-agent as well? The trucking business has many one-man-companies: those are unlikely to soon connect to such a system, meanwhile they can disturb operations The contractual relationship between the terminal and LSP is not existing: this makes it difficult for a terminal (such as ECT) to push forward strong requirements
Criticism on the current prototype	Real advantages are only realized when companies start collaborating together - the ease of implementation is thus questionable It is not yet a real system (solely a limited prototype) A true business case for implementation is needed: it has to be more than solely a good idea
General feedback 2	Uniformity of systems Like every system implementation: requires a new planning process The planner likely perceives the system as a threat The given demo was unclear

Table L.4 – Feedback on how to support the planner

Cluster	Frequency	Comment
General comments for a planning system		The planning decisions the system makes should be reliable and correct
	2	Should facilitate real-time decision making
		Should ease the planning task
Agent specific comments	2	Planner should focus on the problems solely: 80/20 rule: management by exception; as such the planner can concentrate on a higher level of steering
		Continuous search for possibilities/opportunities by the agents
		Planner should be able to overrule an agent's decision; in turn, the agent should learn from this [this way it becomes possible to let the agents do more-and-more work]
		Agents should assist in making pre-planning
		Planner should be replaced by agents wherever possible
	3	The planner should move up a level and work more on strategic/tactical aspects of planning; agents can do the operational decision making
Interface comments		Show meta-information to the planner
	3	Agents should come up with advises (which include the impact of certain decisions, and visualize these)
	3	Give multiple alternatives
		Make it a real dashboard: with several statistics (like for example: the risk for delays)
		Agents should make clear how they get to a certain decision (the why) to win the thrust of the planner
		Don't give any alternatives; agents do better than humans anyway: just let the agents do it...

Table L.5 – Advantages barge planning prototype

Cluster	Frequency	Comment
The setting of this particular case		Game illustrates the capacity problems of container terminals
		Game illustrates the practical wishes of end users
		It is especially beneficial for the shippers
Advantages of this specific technology		Planning could largely take place in an automated manner (80/20 rule): the planner solely focuses on management-by-exception
	2	Formalizes decentralized planning [which factually is the case right now as well; as such it shows well the benefits and requirements of the different parties]

		Much calculation power, reduction of errors
		Rational and factual
		Non-stop
		Possibility for real-time re-planning when events occur
		Real-time planning can be well supported through a MAS
		Possible to use for offline planning in a situation with multiple parties with different objectives
Advantage of this approach to the port	2	System-wide optimization overcomes local sub-optimization
	4	Enlarges the capacity of port and barges
		Enlarges the capacity for hinterland transport
		Better predictable

Table L.6 – Disadvantages barge planning prototype

Cluster	Frequency	Comment
Implementation aspects case		Need for an independent organization that puts the system in the market, and functions as an independent authority
		How to enrol this throughout the chain
	2	Competitive information
		Requires special skills of the planners that need to be taught
		Transparency from the side of the terminals is unlikely
		Threshold to share information is large
	2	Everyone should participate to improve the whole
		Sub optimization of the separate terminals
	2	Do barge operators have the financial resources to invest in this?
		Adoption is difficult, due to the complexity of the market
		Requires a (radical) change of the planning process to achieve real large benefits
		Good & useful solution - however, due to an expected lack of willingness to cooperate from the terminals, very unlikely to succeed
Implementation aspects technology		Proof-of-concept -> is not yet a working and implemented system
		Standardization needed
		System should be purely advising (towards planners)
		Requires standardization of interfaces
Further development of the concepts		Should you introduce some kind of currency to enable coordination
		Wouldn't it be possible to solve this centrally?
		Flexibility of barges in timeslots should be better
		What is the role of the shipper?
		Prioritizing barge arrivals at terminals should be improved upon

Table L.7 – Advantages of the exchange of waiting profiles for barges

Cluster	Comment
Information is present	Less need for barge-terminal communication (the terminal does provide a complete overview to a barge) More up-to-date information available It is beneficial for both parties to share information (in turn that enables chain optimization) Contains little critical information, and can "to little costs" improve the process
Side-effects	Integral processing (??) Actors can change their behaviour Application that can be an extension to this: optimization of up-river waterworks arrivals, such as sluices

Table L.8 – Disadvantages of the exchange of waiting profiles for barges

Cluster	Frequency	Comment
Information exchange is a problem	3	Parties do not want to share information such as this - NEVER! Intention to share information reduces when commercial or sensitive information is shared (a.o. with competitors) Do parties (specifically terminals) actually want to share such information? [suggestion for improvement/acceptance: first show them the benefits] It is difficult to acquire information from multiple terminals, since this is competitive information
Suggestions for improvements to the concept		Many barge-operators use the same waiting profiles; this might result in requests for the same terminal capacity by several parties at the same time Waiting times and utilization rates of barges are very important too - include that explicitly in the solution
How to respond to inappropriate use	2	This approach runs the risk of cheating, parties that play it unfair - try to define a mechanism that minimizes this Parties can frustrate each other when changes occur and all parties start independently, and unorchestrated responding to these changes

Table L.9 – Success factors that drive adoption

Cluster	Frequency	Comment
The motivation for adoption such a chain system comes from		The government and branch-organizations should have a stimulating role Request should be chain driven Collaboration
Specific aspects that lead to success		Just-in-time principal should by applied throughout the entire chain (including terminals and customers) Thrust is needed for all parties in the chain (hence that transparency is important) Clarity with respect to the use of information
How to let agents succeed	 2	Agents make it possible to stay close to the current way of (manual) working [people dislike changes] Show the added value of such a solution Clear business case needed Reliability of systems large Concerns overcoming complexity

Table L.10 – Factures that explain adoption failure

Cluster	Comment
Aspects of chain management	Mechanism to divide "chain gains" among members Too little participation of companies Mechanism to maintain that parties obey the rules (for example: penalties) Spreading Too many parties, too much coordination, too many objectives
Individual organization aspects	Planners have different perceptions than management (a good system needs fewer planners...) New technology is scary
Specific agent aspects	When many variables are around perhaps the agents will not deliver; the optimum is than hard to set Show the differences, and show where the added value is

Table L.11 – Potential other applications

Dock planning
No answer for commercial reasons. But I do see several application domains.
Utilizing real-time information of the fleet (trucks and barges) to dictate maintenance planning, and the coordination with service technicians [make the fleet intelligent]
Complex dynamic environments where many parties need to coordinate activities

Table L.12 – Results from discussion in the first group road transport workshop

Technology aspects agents	Robustness is an advantage. If the system crashes or fails, only part of the system goes down [in a truly distributed setting]. Is another generation of programmers needed, or can we train current developers? [question raised by CapGemini] Watch out for sub-optimality due to a lacking global view - in response someone asks whether this actually is a problem
Potential for inter-organisational chain applications	There is no relationship between the LSP and the container terminals. However, terminals could reward LSPs that provide reliable information (concerning arrival times, and container data) with a premium treatment [this is suggested by ECT] Coordination with both customers and terminals is an important, interesting and convincing extension of the current prototype The largest benefits of agents are revealed in true chain applications (that require coordination between links in the chain) A large and important player in the chain should have the lead in the realization of a chain system
The question "How to realize chain applications?" (starting off from the prototype)	A trusted third party could play a role in actively monitoring the market (KPIs and so), and make clear which parties do benefit or have damage from particular chain decisions How to create mass [that is where the true potential is...]
Other remarks	This demo only shows a limited application: it is highly operational. A future real-life system should include also longer-term (tactical / strategic) views. A large advantage of agent application is that the solution finding process is very human-a-like and thus understandable and explainable. This could be a major factor in the acceptance of such a system. In the daily real-world practice it is not so much about optimality. It is about an acceptable (good) solution which will be acceptable in the market. This is not necessarily the same.

Table L.13 – Results from discussion in the second group road transport workshop

The factor called the "human planner"	How to get the knowledge from human planners in the system? Preferably in a flexible (non-hardcoded) manner. The shown system does included a multi-attribute decision making process. Nevertheless, be aware that human planners include many different other factors. The system should come up with proposals for the planner (in a 80/20 setting) The human planner has the final control; the agent system should learn from the feedback it gets from the planner on its proposals

	Human planners always have cold-water-fear when new information systems are introduced: the planner perceives such a system as a possible replacement for himself.
Discussion issue	In a chain environment: who (or which party) is going to make the real decisions?
Potential for application	Important to test the prototype system in real practice. That will provide real useful insights. A suggestion is to apply these concepts in the taxi branch. Mathijs de Weerdt (TU Delft) explains that his research (several years ago) showed that humans were capable of doing it efficiently by hand until a fleet size of 25 cabs. Above that an agent based system does better. There is real potential. A suggestion is to apply the prototype in a larger multi-enterprise environment, with for example 15 to 20 companies and 1500 trucks. That creates mass. Jo van Nunen (EUR) suggests a link with the 4C initiative of the Commissie Van Laarhoven The largest potential for agents is in true chain applications. Coordination as such is very important in logistics.
Implementation aspects	It is definitely not the technical question It is mainly the question how to introduce such technology, systems and processes in a complex setting It is important to show the added value for different parties; and to construct a mechanism that assures that parties cannot cheat or play unfair. The complexity and dynamics in such applications are that large that planners cannot oversee all aspects [he adds: everyone always tells the opposite, that systems cannot replace planners with 25 years of experience, but he is convinced that a good system can do at least as good]. This kind of systems has potential for huge cost savings: fewer planners needed, but additional cost savings through new processes (many of these will be in the IOS domain). The real intriguing question for the research community should be: How can we get to a phase that these systems are going to be used and adopted?

Appendix M Feedback fieldtest

This appendix includes an overview of the feedback received during the prototype field test held at Post-Kogeko with two Post-Kogeko planners. A list is given in Table M.1. Here we make a split in bugs, missing features (MF), and design choices (DC).

Table M.1 – Overview of feedback received

Bug	Timewindow for a customer in the interface is restricted between 6:00-20:00 hours. In principal orders can have wider timewindows. Found through non-clickable orders which had a window outside this region.	Easy
Bug	List of orders and trucks in the interface is not sortable. It is currently not in alphabetical (Trucks) nor numerical (Orders) order.	Easy
MF	The list of offhire orders and trucks is not included. This has been a design choice, since these are not planned on an order basis. The prototype gives thought that it should be very handy to have the offhires visible nevertheless, to exchange trucks and orders with the "normal fleet". When QFreight was implemented this was overlooked.	Relatively easy, however, notice that offhire orders are different
DC	Coolboxx containers are 45 ft long; which means these cannot be transported on a normal chassis. In fact, there is a dedicated fleet of Coolboxx trucks with a corresponding chassis.	Easy, choice was made to not include too many restrictions
MF	Orders are not always completed entirely on one day. This means that sometimes an order for tomorrow is picked-up today, or a container from yesterday is still delivered or returned today. These orders were not automatically fetched by the system, and had to be generated manually.	Average, requires a view on orders throughout the rest of the week
MF	Drivers that go home with their truck are only allowed to do so if the chassis does not carry a container, or an empty container.	Easy
DC	Order execution in the prototype is always performed by exactly one truck; in practice often: Truck A does the pickup, than the chassis is dropped somewhere (often Post-Kogeko HQ), and Truck B finishes the job.	Average, requires rethinking the mechanism

DC	Containers for the UK are brought to the Ferry at Hoek van Holland, and dropped there. They are picked up in Hull again by another truck (at a specific time).	Hard, requires also monitoring ferry movements, and control over the trucks in the UK.
DC	Possibility to enter the next job when the current job is still busy. Currently each truck is assigned to exactly one order, and it can only be assigned to a different order after the first order is finished.	Average, requires rethinking the mechanism, and a queue of future orders
MF	Information on container vessel delays is lacking (linked to containers that are delivered by that vessel). Linked to this: start a sense-and-response event when a vessel is delayed, and proactively solve troubles that arise.	Hard, requires interfacing with other systems.
DC	Currently the FindOrder mechanism for the next trip only starts when the past order is entirely completed. However, this should be done beforehand [concretely: this way queuing twice at ECT can be avoided]	Easy
MF	Score calculations. It is not only distance that matters (reducing empty miles); perhaps as important is to get a container from exactly the same return terminal – this should be an additional criterion.	Easy
MF	Lunch breaks have been left out of the calculations. Breaks for the drivers, but especially also the breaks at the terminals.	Average. Also to be included in scoring mechanism.
MF	Include the fact that some orders still need to be “declared” (need to go through an “inklaar process”). This takes often an additional stop, at a non-terminal location, and thus results in a longer trip duration.	Average. Also to be included in scoring mechanism.
MF	Orderview only shows the containernumber, whereas the orderlist only shows ordernumbers.	Easy. Information exists in the agent.
MF	Truckview does not include the name of the driver nor the driver’s code.	Easy. Information exists in DB.

DC	Routecalculation engine always calculates with maximum speeds.	Average. Option to integrate a different routeplanner.
DC	A standard order comes with three addresses. However, design choice was made that in case not all three addresses are known in advance we set the unknown addresses to something we estimate. However, in that case OrderAgent should update itself throughout the day when the orderdetails change (and do get known).	Relatively easy, requires a new check and update behaviour.
DC	Locations and trucks outside the Benelux are neglected [due to restrictions in our routeplanner]. This should not be the practice in a real-life and implemented version.	Average. Option to integrate a different routeplanner.
DC	Multi-criteria decision process (with multiple scoring points) is interesting, but criteria do change over the day. In morning customer timewindows important (for the first trip), throughout the remainder of the day hardly an issue anymore.	Average. Requires major change.
MF	The list of orders should be not only divided in “Completed” / “Not Completed”, but the latter category should be split also in “available” / “not available”.	Easy
Bug	The log tables could not be made visible (neither the truck log, nor the order log). Perhaps due to the fact that many orders did not go through all steps, but skipped steps due to the manual synchronization.	Average. Requires testing and bug finding.

Appendix N Chain workshop participants

This appendix includes an overview of the participants that were present in the chain workshop at Post-Kogeko, which was held after the prototype field test. The list is given in Table N.1. The participants marked with a (*) were also present in the internal workshop, which was organized before the external workshop.

Table N.1 – Overview of participants in the chain implication session

Alberdine van Velzen (*)	Post-Kogeko	Head of Planning
Ben van Zeijl (*)	Post-Kogeko	Senior Planner container unit
Richard Crans (*)	Post-Kogeko	Head of IT
Frans Denie (*)	Post-Kogeko	General Director
Ronald van Meurs	CoolControl	Senior Planner
Tom Niels	ECT	Operations Manager
Evert van Hoven	ECT	Consultant Business Development
Frans van den Nobelen	APM Terminals	Sr. Project Manager Development
Jeroen de Rijcke	APM Terminals	Supervisor Gate
John Monteiro	PortInfolink	Product Manager roadplanning
Jos van Hillegersberg	Universiteit Twente	Full Professor
Arthur Oink (*)	Erasmus Universiteit	Consultant ICT (+ dev. prototype)
Hans Moonen (*)	Erasmus Universiteit	PhD Candidate (+ dev. prototype)

Appendix O Barge rotation planning

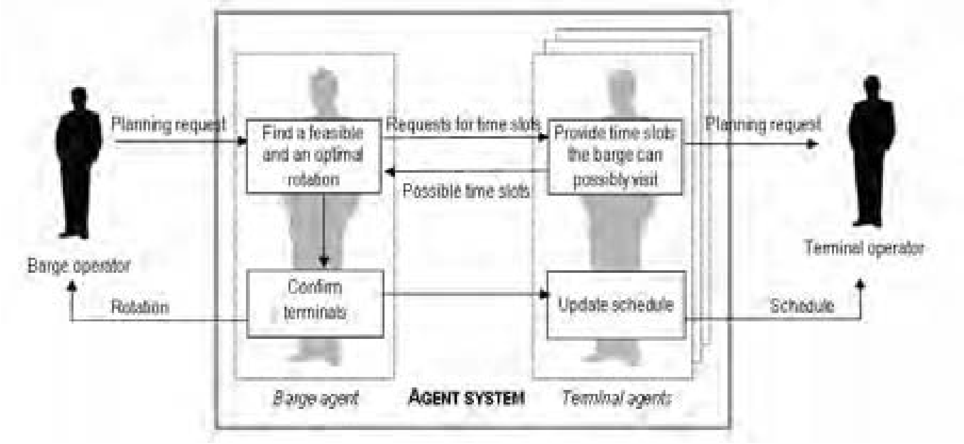


Figure O.1 – Illustration of the working of the agent model (Douma *et al.*, 2008)

Table O.1 – Four different coordination mechanisms for pre-planning

#	Short description	Advantages	Disadvantages
I	Automating current practice: prepare possible rotation, check if possible	Simple	Much communication needed Little space for optimization
II	Prepare a basic rotation, than ask the terminals for the first available timeslot that suits the rotation	Simple Not much communication	Leaves little space for any optimization Dependent upon quality of information terminals
II I	As (II), but with a multiple timeslot response from the terminals	More options for optimization	Timeslot and time needed for visit are not necessarily the same Dependent upon quality and completeness of information
I V	Ask the terminals for an expected waiting profile over the day, and use this in rotation preparation	A more global view Ample of possibilities for smart optimization	Barges have to stick to their plans

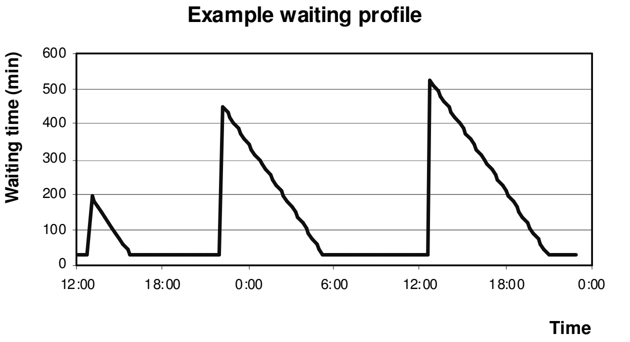


Figure O.2 – Example of a waiting profile sent to a BOA by a TOA

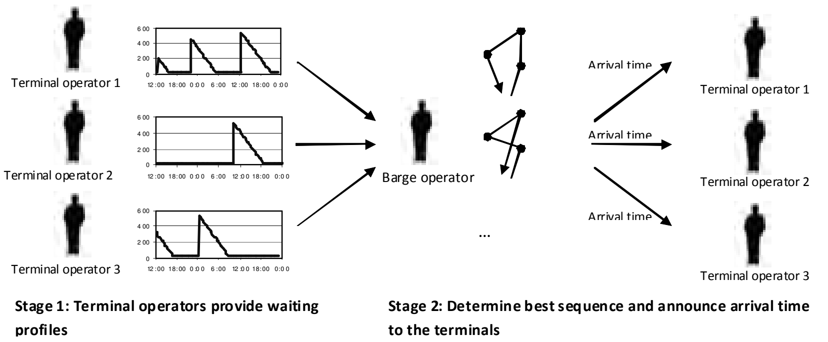


Figure O.3 – Inner mechanism of the MAS utilizing waiting profiles

Table O.2 – Results from the three different approaches

Approach	Maximum lateness	Average lateness	Number of barges delayed
Static benchmark	-24 min.	-480 min.	0
Scenario II	696 min.	-278 min.	8
Scenario IV			
Min waiting time equal to… 0 min.	140 min.	-473 min.	2
30 min.	35 min.	-420 min.	1
60 min.	140 min.	-362 min.	3

List of abbreviations

3PL	Third Party Logistics Provider
4PL	Fourth Party Logistics Provider
ABC	Activity Based Costing
ACM	Association for Computing Machinery
AI	Artificial Intelligence
AID	Agent Identifier (see also JADE)
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
APS	Advanced Planning and Scheduling
AUML	Agent UML (see also UML)
B2B	Business-to-Business
B2C	Business-to-Consumer
BI	Business Intelligence
BOA	Barge Operator Agent (APPROACH specific)
BPEL	Business Process Execution Language
BPML	Business Process Modelling Language
BPR	Business Process Redesign
CAN-bus	Controller-area network bus
CFP	Call For Proposals
CRM	Customer Relationship Management
CS	Computer Science
CTM	Collaborative Transportation Management
DC	Distribution Center
DEAL	Distributed Engine for Advanced Logistics (project name)
DF	Directory Facilitator (see also JADE)
DSS	Decision Support System
EDI	Electronic Data Interchange
EET	Estimated Execution Time
EIS	Enterprise Information Systems
ERP	Enterprise Resource Planning
ETA	Estimated Time of Arrival
ETD	Estimated Time of Departure
FIFO	First In, First Out
FIPA	Foundation for Intelligent Physical Agents
FTL	Full Truck Loads

GPRS	General Packet Radio Service (see also GSM)
GPS	Global Positioning System
GSM	Global System for Mobile communications
GUI	Graphical User Interface
GWT	Google Web Toolkit (see also AJAX)
ICT	Information & Communication Technologies (see also IT)
ID	Identification
IDE	Integrated Development Environment
IOS	Inter Organisational Systems
IS	Information Systems
IT	Information Technologies (in The Netherlands is often ICT used instead)
ITT	Inter Terminal Transport
JADE	Java Agent Development Environment
JVM	Java Virtual Machine
KPI	Key Performance Indicator
LEAP	Lightweight Extensible Agent Platform (see also JADE)
LP	Linear Programming
LSP	Logistics Service Provider
LTL	Less Than Truck Loads
MAS	Multi-Agent System
MRP-I	Material Requirements Planning (often also only: MRP)
MRP-II	Manufacturing Resource Planning
O2A	Object-to-Agent communication (see also JADE)
OEM	Original Equipment Manufacturer
OM	Operations Management
OO	Object Orientation
OR	Operations Research
PDA	Personal Digital Assistant
PhD	Philosophiæ Doctor (Latin; meaning: Doctor of Philosophy)
Q&A	Questions & Answers
RFID	Radio Frequency Identification
RFIT	Radio Frequency Information Technology (advanced version of RFID)
RUP	Rational Unified Process
SCC	Supply-Chain Council
SCEM	Supply Chain Event Management (software)
SCM	Supply Chain Management
SCOR	Supply Chain Operations Reference Model
SME	Small and Medium Enterprises

SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
TAC	Trading Agent Competition
TEU	Twenty feet Equivalent Unit
TMS	Transport Management System
TOA	Terminal Operator Agent (APPROACH specific)
TTP	Trusted Third Party
UDDI	Universal Description, Discovery and Integration
UI	User Interface (see also GUI)
UML	Unified Modelling Language
UTAUT	Unified Theory of Acceptance and Use of Technology
VMI	Vendor Managed Inventory
WADE	Workflows and Agents Development Environment (see also JADE)
WFM	Workflow Management
WSDL	Web Services Description Language
WWII	World War Two
XML	eXtendable Markup Language

Bibliography

Adali, S. and L. Pigaty (2003). "The DARPA advanced logistics project." *Annals of Mathematics and Artificial Intelligence* 37(4): 409-452.

Adler, J. L. and V. J. Blue (2002). "A cooperative multi-agent transportation management and route guidance system." *Transportation Research Part C* 10(5-6): 433-454.

Akkermans, H. and K. van Helden (2002). "Vicious and virtuous cycles in ERP implementation: a case study of interrelations between critical success factors." *European Journal of Information Systems* 11(1): 35-46.

Akkermans, H. A., P. Bogerd, *et al.* (2003). "The impact of ERP on supply chain management: Exploratory findings from a European Delphi study." *European Journal of Operational Research* 146(2): 284-301.

Alavi, M. (1984). "An Assessment of the Prototyping Approach to Information Systems Development." *Communications of the ACM* 27(6): 556-563.

Alt, R. and S. Klein (1998). *Learning from failure: The myths and magic of electronic transportation markets*. HICSS 1998, Hawai, IEEE press.

Anderson-Lehman, R., H. J. Watson, *et al.* (2004). "Continental Airlines Flies High With Real-Time Business Intelligence." *MIS Quarterly Executive* 3(4): 163-176.

Anthes, G. H. (2003). "Agents of Change." *Computerworld*(January 27, 2003).

Anumba, C. J., O. O. Ugwu, *et al.* (2001). "A multi-agent system for distributed collaborative design." *Logistics Information Management* 14(5/6): 255-266.

Anussornnitisarn, P. and S. Y. Nof (2003). "e-Work: the challenge of the next generation ERP systems." *Production Planning & Control* 14(8): 753-765.

Arnott, D., G. Pervan, *et al.* (2005). "Who pays for decision support systems research? Review, directions and issues." *Communications of the Association for Information Systems* 16: 356-380.

Bae, H. and Y. Seo (2007). "BPM-based integration of supply chain process modeling, executing and monitoring." *International journal of production research* 45(11): 2545-2566.

Bakos, J. Y. (1991). " Information Links and Electronic Marketplaces: The Role of Interorganizational Information Systems in Vertical Markets." *Journal of Management Information Systems*.

Bakos, J. Y. (1991). "A Strategic Analysis of Electronic Marketplaces." *MIS Quarterly* 15(3): 295.

Bakos, J. Y. (1997). "Reducing buyer search costs: Implications for electronic marketplaces." *Management Science* 43(12): 1676-1692.

Baskerville, R. (2003). "The LEO principle: perspectives on 50 years of business computing." *Journal of Strategic Information Systems* 12(4): 255-263.

Baskerville, R. and M. D. Myers (2004). "Special Issue on Action Research in Information Systems: Making IS Research Relevant to Practice - Foreword." *MIS Quarterly* 28(3): 329-335.

Baskerville, R. L. (1999). "Investigating Information Systems with Action Research." *Communications of the Association for Information Systems* 2(19).

Bauer, B. and J. Müller (2004). Methodologies and Modeling Languages. *Agent-Based Software Development*. M. Luck, R. Ashri and M. D'Inverno. London, Artech House: 77-131.

Becker, H. S. (1996). The Epistemology of Qualitative Research. *Ethnography and Human Development: Context and Meaning in Social Inquiry*. R. Jessor, A. Colby and R. A. Schweder. Chicago, University of Chicago Press: 53-71.

Becker, J., P. De Wit, *et al.* (2003). Intelligent Agents in supply chains - an explorative research. Delft, Connekt.

Belecheanu, R. A., S. Munroe, *et al.* (2006). *Commercial applications of agents: lessons, experiences and challenges*. AAMAS 2006, Hakodate, Japan ACM Press

Bellifemine, F., A. Poggi, *et al.* (2001). "Developing multi-agent systems with a FIPA-compliant agent framework." *Software-Practice and Experience* 31(2): 103-128.

Bellifemine, F. L., G. Caire, *et al.* (2007). *Developing Multi-agent Systems with JADE*, Wiley.

Benbasat, I., D. K. Goldstein, *et al.* (1987). "The Case Research Strategy in Studies of Information Systems." *MIS Quarterly* 11(3): 369-386.

Benbasat, I. and R. W. Zmud (1999). "Empirical Research in Information Systems: The Practice of Relevance." *MIS Quarterly* 23(1): 3-16.

Berglund, M., P. v. Laarhoven, *et al.* (1999). "Third-party logistics: Is there a future?" *International Journal of Logistics Management* 10(1): 59.

Bharati, P. and A. Chaudhury (2006). "Studying the current status of technology adoption." *Communications of the Acm* 49(10): 88-93.

Biehl, M. (2007). "Success factors for implementing global information systems." *Communications of the ACM* 50(1): 52-58

Bitran, G. R., S. Gurumurthi, *et al.* (2007). "The Need for Third-Party Coordination in Supply Chain Governance." *MIT Sloan Management Review* 48(3): 30-37.

Blake, M. B. (2002). *B2B electronic commerce: Where do agents fit in?* Proceedings of the AAAI-2002 Workshop on Agent Technologies for B2B E-Commerce, Edmonton, Alberta, Canada.

Blumberg, B., D. R. Cooper, *et al.* (2005). *Business Research Methods*, McGraw-Hill Education.

Boer-Sorban, K. (2008). *Agent-Based Simulation of Financial Markets: A Modular, Continuous-time Approach* Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam.

Bold, A. and J. Olsson (2005). What it costs to Manage Collaborative Logistics. EyeForTransport.

Bonabeau, E. and C. Meyer (2001). "Swarm Intelligence. A whole new way to think about business." *Harvard Business Review*(May 2001): 107-114.

Booth, W. C., G. G. Colomb, *et al.* (1995). *The craft of research*. Chicago, The University of Chicago Press.

Bose, I. and R. Pal (2005). "Auto-ID: Managing anything, anywhere, anytime in the supply chain." *Communications of the ACM* 48(8): 100-106.

Bowersox, D. J., D. J. Closs, *et al.* (2003). "How to Master Cross-Enterprise Collaboration." *Supply Chain Management Review* 7(4): 18-27.

Bretzke, W.-R. (2003). Supply Chain Event Management - Bridging the Gap Between Planning and Realisation. Business Briefing: Global Purchasing & Supply Strategies. 2003.

Brickman, C. and D. Ungerman (2008). "Climate change and supply chain management." *McKinsey Quarterly* 2008(July).

Bromley, P. (2001). "A Measure of Logistics Success." *Logistics Quarterly* 7(3).

Brown, C. and I. Vessey (2003). "Managing the Next Wave of Enterprise Systems: Leveraging Lessons from ERP." *MIS Quarterly Executive* 2(1): 65-77.

Brynjolfsson, E. and L. Hitt (1996). "Paradox lost? Firm-level evidence on the returns to information systems spending." *Management Science* 42(4): 541-558.

Brynjolfsson, E. and A. P. McAfee (2007). "The Future of the Web: Beyond Enterprise 2.0." *Sloan management review* 48(3): 49-55.

Bughin, J. and J. Manyika (2007). "How businesses are using Web 2.0: A McKinsey Global Survey." *McKinsey Quarterly* 27.

Caridi, M. and S. Cavalieri (2004). "Multi-agent systems in production planning and control: an overview." *Production Planning & Control* 15(2): 106-118.

Carr, N. G. (2003). "IT doesn't matter." *Harvard Business Review* 81(5): 41-+.

Castelfranchi, C. (1995). *Guarentees for autonomy in coginitive agent architecture*. Decentralized AI 3 - proceedings of the third European workshop on modelling autonomous agents and multi-agent worlds (MAAMAW-91), Elsevier Science Publishers B.V, Amsterdam.

Cervenka, R., I. Trencansky, *et al.* (2005). "AML: Agent Modeling Language Toward Industry-Grade Agent-Based Modeling." *Lecture notes in computer science*

Chapman, R. L., C. Soosay, *et al.* (2003). "Innovation in logistics services and the new business model." *International Journal of Physical Distribution and Logistics Management* 33(7): 630-650.

Cheeseman, M. J., P. Swann, *et al.* (2005). "Adaptive manufacturing scheduling: a flexible and configurable agent-based prototype." *Production Planning and Control* 16(5): 479-487.

Chen, F. G. (1999). "Decentralized supply chains subject to information delays." *Management Science* 45(8): 1076-1090.

Chen, H., M. Z. Frank, *et al.* (2005). "What Actually Happened to the Inventories of American Companies Between 1981 and 2000?" *Management Science* 51(7): 1015-1031.

Chen, I. J. and A. Paulraj (2004). "Towards a theory of supply chain management: the constructs and measurements." *Journal of Operations Management* 22(2): 119-150.

Chmiel, K., M. Gawinecki, *et al.* (2005). "Efficiency of JADE agent platform." *Scientific Programming* 13(2): 159-172.

Christopher, M. (1999). "Logistics and Supply Chain Management: Strategies for Reducing Cost and Improving Service." *International Journal of Logistics Research and Applications* 2(1): 103-103.

Clemons, E. K. and M. C. Row (1991). "Sustaining It Advantage - the Role of Structural Differences." *Mis Quarterly* 15(3): 275-292.

Coase, R. H. (1937). "The Nature of the Firm." *Economica* 4(16): 386-405.

Collins, J., W. Ketter, *et al.* (2009). Flexible Decision Support in a Dynamic Business Network. *The network experience*. P. Vervest, D. W. Van Liere and L. Zheng. Berlin, Springer Verlag: 233-248.

Coltman, T., T. M. Devinney, *et al.* (2001). "E-business: Revolution, evolution, or hype?" *California Management Review* 44(1): 57-+.

Cotteleer, M. J. and E. Bendoly (2006). "Order lead-time improvement following enterprise information technology implementation: an empirical study." *MIS Quarterly* 30(3): 643-660.

Cross, N. (2007). "Forty years of design research." *Design Studies* 28(1): 1-4.

Cruijssen, F., W. Dullaert, *et al.* (2007). "Horizontal cooperation in transport and logistics: A literature review." *Transportation Journal* 46(3): 22-39.

Currie, W. and R. Galliers (1999). *Rethinking Management Information Systems: An Interdisciplinary Perspective*. Oxford, Oxford University Press.

Da Silva, V. T. and C. J. P. De Lucena (2007). "Modeling Multi-Agent Systems." *Communications of the ACM* 50(5): 103-108.

Daganzo, C. F. (2005). *Logistics Systems Analysis*, Springer Verlag.

Davenport, T., H. , J. Harris, G. , *et al.* (2004). "Enterprise systems and ongoing process change." *Business Process Management Journal* 10(1): 16.

Davenport, T. H. (1998). "Putting the enterprise into the enterprise system." *Harvard Business Review* 76(4): 121-+.

Davenport, T. H. and J. D. Brooks (2004). "Enterprise systems and the supply chain." *Journal of Enterprise Information Management* 17(1): 8.

Davidsson, P., L. Henesey, *et al.* (2005). "An analysis of agent-based approaches to transport logistics." *Transportation Research Part C-Emerging Technologies* 13(4): 255-271.

Davies, N. J., D. Fensel, *et al.* (2004). "The future of Web Services." *BT Technology Journal* 22(1): 118-130.

De Ridder, E. (2003). Op weg naar een succesvolle introductie van een revolutionair logistiek planningsplatform. Rijswijk, Technische Hogeschool: 61.

De Toni, A., G. Nassimbeni, *et al.* (1994). "New Trends in the Supply Environment." *Logistics Information Management* 7(4): 41-50.

De Weerdt, M., A. Ter Mors, *et al.* (2006). "Multi-agent Planning: An introduction to planning and coordination." *Handouts of the European Agent Summer School*: 1-32.

De Weerdt, M. M. (2003). Plan Merging in Multi-Agent Systems. Delft. PhD.

DeLone, W. H. and E. R. McLean (1992). "Information Systems Success: The Quest for the Dependent Variable." *Information Systems Research* 3(1): 60-95.

Denning, P. J. (2005). "Is computer science science?" *Communications of the ACM* 48(4): 27-31.

Denzin, N. K. and Y. S. Lincoln (2005). *The Sage Handbook of Qualitative Research*. Thousand Oaks, CA, Sage Publications Inc.

desJardins, M. E., E. H. Durfee, *et al.* (1999). "A survey of research in distributed, continual planning." *Ai Magazine* 20(4): 13-22.

Devaraj, S. and R. Kohli (2003). "Performance impacts of information technology: Is actual usage the missing link?" *Management Science* 49(3): 273-289.

Doll, W. J. and G. Torkzadeh (1998). "Developing a multidimensional measure of system-use in an organizational context." *Information & Management* 33(4): 171-185.

Donselaar, K. v., K. Kokke, *et al.* (1998). "Performance measurement in the transportation and distribution sector." *International Journal of Physical Distribution & Logistics Management* 28(6): 434-450.

Douma, A. M. (2008). Aligning the Operations of Barges and Terminals through Distributed Planning. BETA Research School for Operations Management and Logistics. Enschede, University of Twente. PhD.

Douma, A. M., H. M. Moonen, *et al.* (2008). Designing an Agent-Based Inter-Organizational Coordination System for Planning and Control of Container Barges in the Port of Rotterdam. *Smart Business Networks - a new business paradigm*. P. Vervest, E. Van Heck and K. Preiss. Rotterdam, Smart Business Networks Initiative: 382-407.

Douma, A. M., J. Van Hillegersberg, *et al.* (2008). Using a management game to exemplify a multi-agent approach for the barge rotation and quay scheduling problem in the port of Rotterdam.

Management in Logistics Networks and Nodes. T. Blecker, W. Kersten and C. Gertz. Berlin, Erich Schmidt Verlag. 8: 227-244.

Durfee, E. H., V. R. Lesser, *et al.* (1989). "Trends in cooperative distributed problem solving." *Knowledge and Data Engineering, IEEE Transactions on* 1(1): 63-83.

Eisenhardt, K. M. (1989). "Building Theories from Case Study Research." *The Academy of Management Review* 14(4): 532-550.

Endriss, U., N. Maudet, *et al.* (2003). "Protocol conformance for logic-based agents." *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*: 679–684.

Farrell, D. (2003). "The real new economy." *Harvard Business Review* 81(10): 104-+.

Fingar, P. (2006). *Extreme Competition: Innovation And the Great 21st Century Business Reformation*, Meghan-Kiffer Press.

Fischer, K., J. P. Muller, *et al.* (1996). "Cooperative transportation scheduling: An application domain for DAI." *Applied Artificial Intelligence* 10(1): 1-33.

Fitzgerald, G. (1998). "Evaluating information systems projects: a multidimensional approach." *Journal of Information Technology* 13(1): 15-27.

Folan, P. and J. Browne (2005). "Development of an extended enterprise performance measurement system." *Production Planning & Control* 16(6): 531.

Fonseca, S., M. Griss, *et al.* (2001). "Evaluation of the ZEUS MAS Framework." *Second International Workshop in Software Agents and Workflows for Systems Interoperability*.

Fowkes, A. S., P. E. Firmin, *et al.* (2004). "How highly does the freight transport industry value journey time reliability--and for what reasons?" *International Journal of Logistics* 7(1): 33-43.

Fox, M. S., M. Barbuceanu, *et al.* (2000). "Agent-Oriented Supply-Chain Management." *International Journal of Flexible Manufacturing Systems* 12(2-3): 165-188.

Franklin, S. and A. Graesser (1996). "Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents." *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*: 21–36.

Frey, D., P. O. Woelk, *et al.* (2003). "Integrated multi-agent-based supply chain management." *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on*: 24-29.

Friedman, T. L. (2005). *The world is flat*. New York, Farrar, Straus and Giroux.

Frohlich, M. T. (2002). "e-integration in the supply chain: Barriers and performance." *Decision Sciences* 33(4): 537-556.

Galbraith, J. R. (1974). "Organization design: an information processing view." *Interfaces* 4(3): 28-36.

Gambardella, L. M., A. E. Rizzoli, *et al.* (2003). "Ant Colony Optimization for Vehicle Routing in Advanced Logistics Systems." *Proceedings of the International Workshop on Modelling and Applied Simulation*: 3–9.

Garcia, A. and C. Lucena (2008). "Taming Heterogeneous Agent Architectures." *Communications of the ACM* 51(5): 75-81.

Garfield, M. J. (2005). "Acceptance of Ubiquitous Computing." *Information Systems Management* 22(4): 24-31.

Gates, B. (1995). *The road ahead*. New York, Viking.

Gattiker, T. F. and D. L. Goodhue (2005). "What happens after ERP implementation: understanding the impact of interdependence and differentiation on plant-level outcomes." *MIS Quarterly* 29(3): 559-585.

Genesereth, M. R. and S. P. Ketchpel (1994). "Software agents." *Communications of the ACM* 37(7): 48-53.

Geoffrion, A. M. and R. Krishnan (2001). "Prospects for Operations Research in the E-Business Era." *Interfaces* 31(2): 6-36.

Glass, R. L. (2005). "The first business application: A significant milestone in software history - Celebrating the pivotal role of the first business applications software." *Communications of the ACM* 48(3): 25-26.

Goldratt, E. M. (2000). *Necessary but not sufficient*, North River Press Great Barrington, MA.

Gosain, S., A. Malhotra, *et al.* (2004). "Coordinating for flexibility in e-business supply chains." *Journal of Management Information Systems* 21(3): 7-45.

Graham, T. S., P. J. Dougherty, *et al.* (1994). "The long term strategic impact of purchasing partnerships." *International Journal of Purchasing and Materials Management* 30(4): 13-18.

Green, K. W. J., D. Whitten, *et al.* (2007). "The impact of timely information on organisational performance in a supply chain." *Production Planning & Control* 18(4): 274-282.

Gummesson, D. E. (2000). *Qualitative Methods in Management Research*. Thousand Oaks, Sage Publications Inc.

Guttman, R. H., A. G. Moukas, *et al.* (2001). "Agent-mediated electronic commerce: a survey." *The Knowledge Engineering Review* 13(02): 147-159.

Hagel III, J. and J. S. Brown (2001). "Your next IT strategy." *Harvard Business Review* 79(9): 105.

Haigh, T. (2001). "Inventing information systems: The systems men and the computer, 1950-1968." *Business History Review* 75(1): 15-61.

Haley, K. B. (2002). "War and peace: The first 25 years of OR Great Britain." *Operations Research* 50(1): 82-88.

Hammer, M. (2001). "The superefficient company." *Harvard Business Review* 79(8): 82-+.

Havenbedrijf (2008). Modal split containers 2007-2003.

Henderson, J. C. and N. Venkatraman (1993). "Strategic alignment: leveraging information technology for transforming organizations." *IBM Systems Journal* 32(1): 4-16.

Heng, M. S. H. (2003). "Understanding Electronic Commerce from a Historical Perspective." *Communications of the Association for Information Systems* 12: 104-118.

Hertz, S. and M. Alfredsson (2003). "Strategic development of third party logistics providers." *Industrial Marketing Management* 32(2): 139-149.

Hess, T. J., L. P. Rees, *et al.* (2000). "Using autonomous software agents to create the next generation of decision support systems." *Decision Sciences* 31(1): 1-31.

Hevner, A. R., S. T. March, *et al.* (2004). "Design science in Information Systems research." *MIS Quarterly* 28(1): 75-105.

Himoff, J., G. Rzevski, *et al.* (2006). "Magenta technology multi-agent logistics i-Scheduler for road transportation." *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems (AAMAS'06)*: 1514-1521.

Holmstrom, J., K. Framling, *et al.* (2002). "Collaborative planning forecasting and replenishment: new solutions needed for mass collaboration." *Supply Chain Management: An International Journal* 7(3): 136-145.

Hong, I. B. and C. Kim (1998). *Toward A New Framework for Interorganizational Systems: A Network Configuration Perspective*. Thirty-First Annual Hawaii International Conference on System Sciences (Proceedings of HICSS 1998), Hawaii, IEEE Press.

Hong, Y. B. (2002). "A new framework for interorganizational systems based on the linkage of participants' roles." *Information & Management* 39(4): 261-270.

Hoogeweegen, M. R., D. W. van Liere, *et al.* (2006). "Strategizing for mass customization by playing the business networking game." *Decision Support Systems* 42(3): 1402-1412.

Huhns, M. N., L. M. Stephens, *et al.* (2002). "Automating supply-chain management." *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*: 1017-1024.

Iastrebova, K. V. (2006). *Managers'Information Overload the impact of coping strategies on decision-making performance*, RSM Erasmus University, Erasmus Research Institute of Management (ERIM).

Ibrahim, M. (2006). Trust, dependence, and interorganizational systems. Tilburg, Tilburg University.

Iivari, J. (2007). "A paradigmatic analysis of Information Systems as a design science." *Scandinavian Journal of Information Systems* 19(1).

Jennings, N. R. (2000). "On agent-based software engineering." *Artificial Intelligence* 117(2): 277-296.

Jennings, N. R. (2001). "An agent-based approach for building complex software systems - Why agent-oriented approaches are well suited for developing complex, distributed systems." *Communications of the ACM* 44(4): 35-41.

Jennings, N. R. and S. Bussmann (2003). "Agent-based control systems: Why are they suited to engineering complex systems?" *Control Systems Magazine, IEEE* 23(3): 61-73.

Jennings, N. R., K. Sycara, *et al.* (1998). "A Roadmap of Agent Research and Development." *Autonomous Agents and Multi-Agent Systems* 1(1): 7-38.

Jones, M. C., M. Cline, *et al.* (2006). "Exploring knowledge sharing in ERP implementation: an organizational culture framework." *Decision Support Systems* 41(2): 411-434.

Kaplan, B. and D. Duchon (1988). "Combining Qualitative and Quantitative Methods in Information Systems Research: A Case Study." *MIS Quarterly* 12(4): 571-586.

Kemppainen, K. (2005). Priority scheduling revisited-dominant rules, open protocols and integrated order management. Helsinki, Publications of Helsinki School of Economics. PhD.

Kemppainen, K. and A. P. J. Vepsäläinen (2003). "Trends in industrial supply chains and networks." *International Journal of Physical Distribution and Logistics Management* 33(8): 701-719.

Khoo, L. P., S. B. Tor, *et al.* (1998). "The potential of intelligent software agents in the world wide web in automating part procurement." *International Journal of Purchasing and Materials Management* 1998(January): 46-52.

Kimbrough, S. O., D. J. Wu, *et al.* (2002). "Computers play the beer game: can artificial agents manage supply chains?" *Decision Support Systems* 33: 323-333.

Klapwijk, H. P. (2004). *Economics Reconfigured ... A Vision of tomorrow's Value Chains ...* Breukelen, Universiteit Nyenrode.

Klein, E. E. and P. J. Herskovitz (2007). "Philosophy of science underpinnings of prototype validation: Popper vs. Quine." *Information Systems Journal* 17(1): 111-132.

Klusch, M. (2001). "Information agent technology for the Internet: A survey." *Data & Knowledge Engineering* 36(3): 337-372.

Kopczak, L. R. and M. E. Johnson (2003). "The supply-chain management effect." *Mit Sloan Management Review* 44(3): 27-34.

Krauth, E. (2008). Real-time Planning Support - A Task-Technology Fit Perspective. ERIM PhD Series Research in Management. Rotterdam Erasmus University Rotterdam. PhD.

Krauth, E., H. Moonen, *et al.* (2005). "Agent-based modeling of third-party logistics performance indicators." *Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*: 262-265.

Krauth, E., H. Moonen, *et al.* (2005). "Performance Measurement and Control in Logistics Service Providing." *The ICFAIAN Journal of Management Research* 4(7).

Krauth, E. I., J. v. Hillegersberg, *et al.* (2007). *Agent-based Human-computer-interaction for Real-time Monitoring Systems in the Trucking Industry*. HICCS.

Kulmala, H. I., A. Vahteristo, *et al.* (2005). "Interorganizational operations in value chains - Experiences from networked software firms." *Production Planning & Control* 16(4): 378-387.

Kumar, K., H. G. v. Dissel, *et al.* (1998). "The merchant of Prato-revisited: Toward a third rationality of information systems." *MIS Quarterly* 22(2): 199-226.

Kumar, K. and J. v. Hillegersberg (2000). "ERP Experiences and Evolution." *Communications of the ACM* 43(4): 23-26.

Kumar, K. and H. G. Van Dissel (1996). "Sustainable collaboration: Managing conflict and cooperation in interorganizational systems." *MIS Quarterly* 20(3): 279-300.

Kurnia, S. and R. B. Johnston (2000). "The need for a processual view of inter-organizational systems adoption." *Journal of Strategic Information Systems* 9(4): 295-319.

Kushniruk, A. (2002). "Evaluation in the design of health information systems: application of approaches emerging from usability engineering." *Computers in Biology and Medicine* 32(3): 141-149.

Lai, K. H., E. W. T. Ngai, *et al.* (2004). "An empirical study of supply chain performance in transport logistics." *International Journal of Production Economics* 87(3): 321-331.

Lambert, D. M. and M. C. Cooper (2000). "Issues in Supply Chain Management." *Industrial Marketing Management* 29(1): 65-83.

Lang, N. (2010). Dissertation - forthcoming, expected 2010, Erasmus University Rotterdam. PhD.

Lang, N., H. Moonen, *et al.* (2008). Multi Agent Systems in Logistics: A Literature and State-of-the-art Review Rotterdam, Erasmus Reserach Institute of Management.

Langley Jr, C. J. and G. R. Allen (2006). *Third-Party Logistics Study Results and Findings of the 2006 Eleventh Annual Study*, Capgemini US LCC and FedEx Corp.

Lea, B. R., M. C. Gupta, *et al.* (2005). "A prototype multi-agent ERP system: an integrated architecture and a conceptual framework." *Technovation* 25(4): 433-441.

Lee, A. S. (1989). "A Scientific Methodology For MIS Case Studies." *MIS Quarterly* 13(1): 33.

Lee, G. G., H. F. Lin, *et al.* (2005). "Influence of environmental and organizational factors on the success of internet-based interorganizationaI systems planning." *Internet Research* 15(5): 527-543.

Lee, H., A. F. Farhoomand, *et al.* (2004). "Innovation Through Supply Chain Reconfiguration " *MIS Quarterly Executive* 3(3): 131-142.

Lee, H. L. (2004). "The Triple-A Supply Chain." *Harvard Business Review*(October 2004): 102-112.

Lee, H. L., V. Padmanabhan, *et al.* (1997). "Information distortion in a supply chain: The bullwhip effect." *Management Science* 43(4): 546.

Lee, J. C. and M. D. Myers (2004). "Dominant actors, political agendas, and strategic shifts over time: a critical ethnography of an enterprise systems implementation." *Journal of Strategic Information Systems* 13(4): 355-374.

Leenaarts, M., J. Treur, *et al.* (2003). "Eindrapport approach." *Technicalreport, Connekt, Delft*.

Lemoine, W. and L. Dagnaes (2003). "Globalisation strategies and business organisation of a network of logistics service providers." *International Journal of Physical Distribution and Logistics Management* 33(3): 209-228.

Lesser, V. R. and D. D. Corkill (1987). "Distributed Problem Solving." *Encyclopedia of Artificial Intelligence* 1: 245-251.

Levi, M. D. and F. G. Conrad (1996). "A heuristic evaluation of a World Wide Web prototype." *interactions* 3(4): 50-61.

Levinson, M. (2006). *The Box - How the shipping container made the world smaller and the world economy bigger*. Princeton, New Jersey, Princeton University Press.

Levinson, P. (2004). *Cellphone: The Story of the World's Most Mobile Medium and How It Has Transformed Everything!*, Palgrave Macmillan.

Levy, M. and P. Powell (1998). "SME flexibility and the role of information systems." *Small Business Economics* 11(2): 183-196.

Lewis, I. and A. Talalayevsky (2000). "Third-Party Logistics: Leveraging Information Technology." *Journal of Business Logistics* 21(2): 173-185.

Lewis, I. and A. Talalayevsky (2004). "Improving the interorganizational supply chain through optimization of information flows." *Journal of Enterprise Information Management* 17(3): 229-37.

Li, F. and H. Williams (1999). "Interfirm collaboration through interfirm networks." *Information Systems Journal* 9(2): 103-115.

Li, J., R. Sikora, *et al.* (2006). "A strategic analysis of inter organizational information sharing." *Decision Support Systems* 42(1): 251-266.

Lima, R. M., R. M. Sousa, *et al.* (2006). "Distributed production planning and control agent-based system." *International Journal of Production Research* 44(18-19): 3693-3709.

Lindgren, R., O. Henfridsson, *et al.* (2004). "Design principles for competence management systems: A synthesis of an action research study." *MIS Quarterly* 28(3): 435-472.

Logistiek (2009). Top 50 Logistiek Dienstverleners 2008 Logistiek.

Lu, X. H., L. H. Huang, *et al.* (2006). "Critical success factors of inter-organizational information systems - A case study of Cisco and Xiao Tong in China." *Information & Management* 43(3): 395-408.

Luck, M. and M. d'Inverno (2001). "A Conceptual Framework for Agent Definition and Development." *The Computer Journal* 44(1): 1.

Luck, M., P. McBurney, *et al.* (2003). "Agent Technology: Enabling Next Generation Computing." *Agent link community*.

Luck, M., P. McBurney, *et al.* (2004). "A Manifesto for Agent Technology: Towards Next Generation Computing." *Autonomous Agents and Multi-agent Systems* 9: 203-252.

Luhn, H. P. (1958). "A business intelligence system." *IBM journal of Research & Development* 2(4): 314-319.

Ma, K. J. (2005). "Web Services: What's Real and What's Not?" *IT Professional* 7(2): 14-21.

Maes, P. (1994). "Agents that reduce work and information overload." *Communications of the ACM* 37(7): 30.

Maes, P., R. H. Guttman, *et al.* (1999). "Agents that buy and sell." *Communications of the ACM* 42(3): 81.

Mahato, S., A. Jain, *et al.* (2006). "Enterprise Systems Consolidation." *Information Systems Management* 23(4): 7-19.

Mahoney, M. S. (1988). "The History of Computing in the History of Technology." *Annals of the History of Computing* 10(2): 113.

Mahr, T., J. Srour, *et al.* (2008). *Agent Performance in Vehicle Routing when the Only Thing Certain is Uncertainty*. Proceedings of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal.

Makukha, K. and R. Gray (2004). "Logistics partnerships between shippers and logistics service providers: the relevance of strategy." *International Journal of Logistics - Research and Applications* 00007(00004): 361-378.

Malone, T. W., J. Yates, *et al.* (1987). "Electronic markets and electronic hierarchies." *Communications of the ACM* 30(6): 484-497.

March, S., A. Hevner, *et al.* (2000). "Research Commentary: An Agenda for Information Technology Research in Heterogeneous and Distributed Environments." *Information Systems Research* 11(4): 327-341.

March, S. T. and G. F. Smith (1995). "Design and natural science research on information technology." *Decision Support Systems* 15(4): 251.

Margretta, J. (1998). "Fast, Global, and Entrepreneurial: Supply Chain Management, Hong-Kong style." *Harvard Business Review*(September-October 1998): 102-114.

Marik, V. and D. McFarlane (2005). "Industrial adoption of agent-based technologies." *IEEE Intelligent Systems* 20(1): 27-35.

Marquis, H. A. (2006). "Finishing off IT." *MIT Sloan Management Review* 47(4): 12-15.

McAfee, A. P. and E. Brynjolfsson (2008). "Investing in the IT that makes a difference." *Harvard Business Review*(July-August 2008): 98-107.

McBurney, P. and M. Luck (2007). "The Agents Are All Busy Doing Stuff!" *Intelligent Systems* 22(4): 6-7.

McFarlane, D. and Y. Sheffi (2003). "The impact of automatic identification on supply chain operations." *The International Journal of Logistics Management* 14(1): 1-17.

Meehl, P. E. (1978). "Theoretical risks and tabular asteriks: Sir Karl, Sir Ronald and the slow progress of soft." *Journal of Consulting and Clinical Psychology* 46(4): 806-834.

Meijer, S., G. J. Hofstede, *et al.* (2006). "Trust and Tracing game: learning about transactions and embeddedness in a trade network." *Production Planning & Control* 17(6): 569-583.

Melis, M., I. Miller, *et al.* (2003). *B. Eck van, M. Leenarts, MC Schut, and J. Treur. Distributed rotation planning for container barges in the port of rotterdam*. Delft, Eburon.

Menon, M. K., M. A. McGinnis, *et al.* (1998). "Selection Criteria for Providers of Third-Party Logistics Services: an Exploratory Study." *Journal of Business Logistics* 19(1): 121-137.

Merrifield, R., J. Calhoun, *et al.* (2008). "The next revolution in productivity." *Harvard Business Review*(June 2008): 72-80.

Mes, M., M. v. d. Heijden, *et al.* (2007). "Design choices for agent-based control of agvs in the dough making process." *Beta Working Paper*(206): 1-30.

Mes, M., M. van der Heijden, *et al.* (2007). "Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems." *European Journal of Operational Research* 181(1): 59-75.

Milojicic, D. (2004). "Microsoft's Jim Gray on Computing's Breakthoughs, Lessons, and Future." *IEEE Distributed Systems Online* 5(1).

Mönch, L., M. Stehli, *et al.* (2006). "The FABMAS multi-agent-system prototype for production control of wafer fabs: design, implementation and performance assessment." *Production Planning and Control* 17(7): 701-716.

Moonen, H., A. M. Douma, *et al.* (2008). Obstacles to Multi-Agent Systems Implementation - Expert Opinions Confronted with Literature. *Management in Logistics Networks and Nodes*. T. Blecker, W. Kersten and C. Gertz. Berlin, Erich Schmidt Verlag. 8: 269-290.

Moonen, H., N. Lang, *et al.* (2006). Creating Virtual Capacity - Utilizing real-time information to boost the performance and reliability of container supply chains. Transumo Seminar Infrastructure Reliability. Delft.

Moonen, H., A. Oink, *et al.* (2007). *A Multi-Agent system for Real-time planning of container trucks*. WITS'07 - Seventeenth Annual Workshop on Information Technologies and Systems, Montreal, Canada.

Moonen, H. and B. v. d. Rakt (2005). Approach 2 - Building the Vision. Diploma Project. Transumo. Rotterdam: 44.

Moonen, H. M. (2003). Third Parties' Influence on Sales in the Enterprise Software Market, RSM Erasmus University.

Moonen, H. M., A. C. Oink, *et al.* (2005). *Dealing With Post-Kogeko - The Design Process of a WebServices based Inter-Organizational Transport Planning System*. International Conference on Electronic Business, Hong Kong.

Moonen, H. M., A. J. R. Zwegers, *et al.* (2003). *A framework for collaborative applications*. Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Applications (CE2003), 26-30 July 2003, Madeira , Portugal, A.A. Balkema Publishers: Lisse.

Moonen, J. M., B. Rakt, *et al.* (2007). Agent Technology supports Inter-Organizational Planning in the Port. *Managing Supply Chains - Challenges and Opportunities*. R. B. M. d. Koster and W. Delfmann. Copenhagen, Copenhagen Business School Press. 1.

Moore, G. E. (1965). "Cramming more components onto integrated circuits." *Electronics* 38(8): 114-117.

Moore, P. D., K. B. Manrodt, *et al.* (2006). The Power of O3 - Optimized Strategy, Planning and Execution. Report on Trends and Issues in Logistics and Transportation.

Moyaux, T., B. Chaib-draa, *et al.* (2006). Supply Chain Management and Multiagent Systems: An Overview. *MultiAgent-Based Supply Chain Management*. B. Chaib-draa and J. P. Müller. Berlin, Springer.

Moyaux, T., B. Chaib-draa, *et al.* (2005). "Information sharing as a coordination mechanism for reducing the bullwhip effect in a supply chain." *IEEE Trans. on Systems, Man, and Cybernetics*.

Mukhopadhyay, T. and S. Kekre (2002). "Strategic and operational benefits of electronic integration in B2B procurement processes." *Management Science* 48(10): 1301-1313.

Nahm, A. Y., M. A. Vonderembse, *et al.* (2003). "The impact of organizational structure on time-based manufacturing and plant performance." *Journal of Operations Management* 21(3): 281-306.

Narayanan, V. G. and A. Raman (2000). "Aligning Incentives for Supply Chain Efficiency." *Harvard Business School teaching note*(9-600-110).

Newell, A. and H. A. Simon (1972). *Human problem solving*, Englewood Cliffs, NJ: Prentice-Hall.

Ngai, E. W. T. and A. Gunasekaran (2004). "Implementation of EDI in Hong Kong: an empirical analysis." *Industrial Management & Data Systems* 104(1-2): 88-100.

Nielsen, J., T. Clemmensen, *et al.* (2002). *Getting access to what goes on in people's heads?: reflections on the think-aloud technique*. Proceedings of the second Nordic conference on Human-Computer Interaction, ACM Press New York, NY, USA.

Nissen, M. E. (2001). "Beyond electronic disintermediation through multi-agent systems." *Logistics Information Management* 14(4): 256-275.

Nissen, M. E. and K. Sengupta (2006). "Incorporating Software Agents into Supply Chains: Experimental Investigation with a Procurement Task." *MIS Quarterly* 30(1): 145.

Nunamaker, J. F. and M. Chen (1990). "Systems development in information systems research." *System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii International Conference on* 3.

Nwana, H. S., L. Lee, *et al.* (1996). "Coordination in Software Agent Systems." *BT Technology Journal* 14(4): 79-88.

Nwana, H. S. and D. T. Ndumu (1999). "A perspective on software agents research." *The Knowledge Engineering Review* 14(02): 125-142.

Odell, J. J., H. V. Parunak, *et al.* (2002). Modeling agents and their environment. *Agent-Oriented Software Engineering Iii*. Berlin, SPRINGER-VERLAG BERLIN. 2585: 16-31.

Oink, A. (2005). Agent Technology in het Containervervoer. School of Economics. Rotterdam, Erasmus University Rotterdam. MSc.

Ordanini, A. (2005). "The effects of participation on B2B exchanges: A resource-based view." *California Management Review* 47(2): 97-+.

Palvia, P., V. Midha, *et al.* (2006). "Research models in information systems." *Communications of the AIS* 17.

Papazoglou, M. P. (2001). "Agent-Oriented technology in support of e-business." *Communications of the ACM* 44(4): 71-77.

Papazoglou, M. P. and W.-J. v. d. Heuvel (2006). "Service-Oriented Design and Development Methodology." *International Journal of Web Engineering and Technology* 2(4): 412 - 442.

Parunak, H. V. D. (1996). *Applications of distributed artificial intelligence in industry*, John Wiley

Parunak, H. V. D. (1999). Industrial and practical applications of DAI. *Multiagent systems: a modern approach to distributed artificial intelligence* MIT Press Cambridge, MA, USA: 377-421.

Parunak, H. V. D. (2000). "A Practitioners' Review of Industrial Agent Applications." *Autonomous Agents and Multi-Agent Systems* 3: 389-407.

Peña, J., M. G. Hinchey, *et al.* (2006). "Multi-agent system product lines: challenges and benefits
" *Communications of the ACM* 49(12): 82-84.

Petersen, S. A., M. Divitini, *et al.* (2001). "An agent-based approach to modelling virtual enterprises." *Production Planning & Control* 12(3): 224-233.

Peterson, D. M. (2003). "Power to the BPEL: A technology for Web services." *Business Communications Review* 33(6): 54.

Petrie, C. and C. Bussler (2003). "Service agents and virtual enterprises: A survey." *IEEE Internet Computing* 7(4): 68-78.

Pil, F. K. and M. Holweg (2006). "Evolving from Value Chain to Value Grid." *MIT Sloan Management Review* 4(47): 4.

Plant, S. (2002). *On the Mobile - The effect of mobile phones on social and individual life*, Motorola.

Pokahr, A., L. Braubach, *et al.* (2008). Simulation and Implementation of Logistics Systems Based on Agent Technology. *Management in Logistics Networks and Nodes*. T. Blecker, W. Kersten and C. Gertz. Berlin, Erich Schmidt Verlag. 8: 291-308.

Porter, M. E. (2001). "Strategy and the Internet." *Harvard Business Review* 79(3): 62-+.

Premkumar, G., K. Ramamurthy, *et al.* (2005). "Information Processing View of Organizations: An Exploratory Examination of Fit in the Context of Interorganizational Relationships." *Journal of Management Information Systems* 22(1): 257-294.

Ragowsky, A. (2002). "Enterprise Resource Planning." *Journal of Management Information Systems* 19(1): 11-15.

Rai, A., R. Patnayakuni, *et al.* (2006). "Firm performance impacts of digitally enabled supply chain integration capabilities." *Mis Quarterly* 30(2): 225-246.

Raman, A. (1995). "Coordination: An overview." *Harvard Business School teaching note*(9-696-001).

Rettig, C. (2007). "The Trouble With Enterprise Software." *MIT Sloan Management Review* 49(1): 21-27.

RIL (1998). Afhandeling Containerbinnenvaart Rotterdam. Werkdocument, Stichting RIL.

Robey, D., J. W. Ross, *et al.* (2002). "Learning to implement enterprise systems: An exploratory study of the dialectics of change." *Journal of Management Information Systems* 19(1): 17-46.

Romme, A. G. L. (2003). "Making a difference: Organization as design." *Organization Science* 14(5): 558-573.

Romme, A. G. L. (2004). "Action research, emancipation and design thinking." *Journal of Community & Applied Social Psychology* 14(6): 495-499.

Ross, A. (2002). "A multi-dimensional empirical exploration of technology investment, coordination and firm performance." *International Journal of Physical Distribution and Logistics Management* 32(7): 591-609.

Roth, V. (2004). *Obstacles to the adoption of mobile agents*. 2004 IEEE International Conference on Mobile Data Management.

Rudowsky, I. S. (2004). "Intelligent Agents." *Communications of the Association for Information Systems* 14: 275-290.

Samuelson, D. A. and C. M. Macal (2006). "Agent-Based Simulation Comes of Age." *ORMS Today* 33(4): 34–38.

Sandholm, T. (1999). Distributed Rational Decision Making. *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*. G. Weiss. Cambridge, MA, USA, MIT press: 201-258.

Saracevic, T. (1995). "Evaluation of evaluation in information retrieval." *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*: 138-146.

SCC (2003). *Supply-Chain Operations Reference Model - SCOR Version 6.0.*

Schleiffer, R. (2005). "An intelligent agent model." *European Journal of Operational Research* 166: 666-693.

Schneeweiss, C. A. (2003). *Distributed Decision Making*, Springer.

Scholz-Reiter, B. and H. Höhns (2002). *Agent-based collaborative supply net management*. Formal Proceedings of the APMS.

Schut, M., M. Kentrop, *et al.* (2004). "APPROACH: Decentralised Rotation Planning for Container Barges." *Proceedings of the Sixteenth European Conference on Artificial Intelligence*: 755-759.

Shang, S. and P. B. Seddon (2002). "Assessing and managing the benefits of enterprise systems: the business manager's perspective." *Information Systems Journal* 12(4): 271-299.

Sharman, G. (2003). "e-Supply Chain Management - Venturing beyond e-Commerce." *Business Briefing: Global Purchasing and supply chain strategies*(2003).

Shehory, O. and A. Sturm (2001). "Evaluation of modeling techniques for agent-based systems." *Proceedings of the fifth international conference on Autonomous agents*: 624-631.

Shen, W. and D. Norrie (1999). "Agent-based Systems for Intelligent Manufacturing: A state-of-the-art survey." *Knowledge and Information Systems* 1(2): 129-156.

Sheombar, H. S. (1997). "Logistics coordination in Dyads: Some theoretical foundations for EDI-induced redesign." *Journal of Organizational Computing and Electronic Commerce* 7(2-3): 153-184.

Shim, J. P., M. Warkentin, *et al.* (2002). "Past, present, and future of decision support technology." *Decision Support Systems* 33(2): 111-126.

Siau, K. (2003). "Interorganizational systems and competitive advantages - Lessons from history." *Journal of Computer Information Systems* 44(1): 33-39.

Sierra, C. (2004). "Agent-Mediated Electronic Commerce." *Autonomous Agents and Multi-agent Systems*(9): 285-301.

Simon, H. A. (1981). *The Science of the Artificial*, MIT Press.

Singh, N., K.-h. Lai, *et al.* (2007). "Intra-organizational perspectives on IT-enabled supply chains." *Communications of the ACM* 50(1): 59-65

Sink, H. L., C. J. Langley Jr, *et al.* (1996). "Buyer observations of the US third-party logistics market." *International Journal of Physical Distribution & Logistics Management* 26(3): 38-46.

Slobodow, B., O. Abdullah, *et al.* (2008). "When Supplier Partnerships Aren't." *MIT Sloan Management Review* 49(2): 77-83.

Sonntag, M. (2006). "Agents as web service providers: single agents or MAS?" *Applied Artificial Intelligence* 20(2): 203-227.

Sousa, P., C. Ramos, *et al.* (2004). "The Fabricare system: a multi-agent-based scheduling prototype." *Production Planning & Control* 15(2): 156-165.

Spieck, S., M. Weigelt, *et al.* (1995). "Decentralized problem solving in logistics and production with partly intelligent agents and comparison with alternative approaches." *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*.

Sridharan, U. V., W. R. Caines, *et al.* (2005). "Implementation of supply chain management and its impact on the value of firms." *Supply Chain Management-an International Journal* 10(3-4): 313-318.

Sriram, R., V. Arunachalam, *et al.* (2000). "EDI adoption and implementation: an examination of perceived operational and strategic benefits and controls." *Journal of Information Systems* 14(1): 37-52.

Srour, J., T. Mahr, *et al.* (2008). Performance evaluation within a networked enterprise: balancing local objectives and network relations. *Smart Business Networks - a new business paradigm*. P. Vervest, E. Van Heck and K. Preiss. Rotterdam, Smart Business Networks Initiative: 279-304.

Stefansson, G. (2002). "Business-to-business data sharing: A source for integration of supply chains." *International Journal of Production Economics* 75(1-2): 135-146.

Stewart, G. (1995). "Supply chain performance benchmarking study reveals keys to supply chain excellence." *Logistics Information Management* 8(2): 38-44.

Straub, D. W. (1989). "Validating Instruments in MIS Research." *MIS Quarterly* 13(2): 147-169.

Sutherland, J. (2003). "Collaborative Transportation Management – Creating Value Through Increased Transpor tation Efficiencies." *Business Briefing: Pharmagenerics 2003*: 4.

Thomas Jr, C. R. and F. Seibel (1999). "Adaptive cargo at Southwest Airlines." *The 4th Annual Colloquium on the Application of Complex Adaptive Systems to Business (Ernst & Young), July*: 25-27.

Tolido, R. (2006). Service Oriented Enterprise - Het beste perspectief is van buitenaf. Capgemini Whitepaper.

Tran, Q.-N. N., G. Low, *et al.* (2005). Comparison of Ten Agent-Oriented Methodologies. *Agent-oriented Methodologies*. B. Henderson-Sellers and P. Giorgini, Idea Group Pub.: 341-367.

Trappey, A. J. C., C. V. Trappey, *et al.* (2004). "Mobile agent technology and application for online global logistic services." *Industrial Management & Data Systems* 104(1-2): 169-183.

Treacy, M. and F. Wiersema (1995). *The Discipline of Market Leaders*. Boston, HarperCollins Publishers.

Tucker, D. and L. Jones (2000). "Leveraging the power of the Internet for optimal supplier sourcing." *International Journal of Physical Distribution & Logistics Management* 30(3/4): 255-267.

Vaidyanathan, G. (2005). "A framework for evaluating third-party logistics." *Communications of the ACM* 48(1): 89-94.

Van Aken, J. E. (2004). "Management Research Based on the Paradigm of the Design Sciences: The Quest for Field-Tested and Grounded Technological Rules." *The Journal of Management Studies* 41(2): 219.

Van Andel, M. (2007). Measures for improving VITO container barge handling at ECT Delta, Port of Rotterdam. Delft, Technical University of Delft. MSc.

Van Baalen, P., R. Zuidwijk, *et al.* (2009 (expected)). *Port Inter-Organizational Information Systems*.

Van Baalen, P. J., M. Van Oosterhout, *et al.* (2000). *Dynamics in Setting up an EDI Community*. Delft, Eburon Publishers.

Van Busschbach, E., B. Pieterse, *et al.* (2002). *Support of Virtual Enterprises by an Integration Infrastructure*. Proceedings of PRO-VE '02, 1-3 May 2002, Sesimbra, Portugal, Kluwer Academic Publishers, Boston.

Van Den Bogaard, A., H. Lintsen, *et al.* (2008). *De eeuw van de computer*. Deventer, Kluwer BV.

Van Der Heijden, H., R. Wagenaar, *et al.* (1995). "Redesigning process control mechanisms using EDI. An agency-theoretic perspective." *Proceedings of the 28th Hawaii International Conference on System Sciences*.

Van Der Horst, M. R. and P. W. De Langen (2008). "Coordination in Hinterland Transport Chains: A Major Challenge for the Seaport Community." *Maritime Economics & Logistics* 10(1-2): 108-129.

Van Der Vlist, P. (2007). *Synchronizing the Retail Supply Chain*. Rotterdam, Dissertation RSM Erasmus University.

Van Dijke, B. (2008). Multi-agent Systems vs. traditional Operations Research methods: a comparison for Supply Chain Management. Department of Decision and Information Sciences. Rotterdam, RSM Erasmus University. MSc.: 115.

Van Dijkum, C., D. DeTombe, *et al.* (1999). *Validation of Simulation Models*. Amsterdam, SISWO Publication.

Van Groningen, L. (2006). Performance indicators for barge planning in the Port of Rotterdam. RSM Erasmus University. Rotterdam, Erasmus University Rotterdam. MSc.

Van Heck, E. and P. Vervest (2007). "Smart business networks: how the network wins." *Communications of the ACM* 50(6): 29-37.

Van Hillegersberg, J. (1997). *Metamodelling-based Integration of Object-Oriented Systems Development*. Rotterdam.

Van Hillegersberg, J. (2006). *Information Systems for a Networked World*. Enschede.

Van Hillegersberg, J., H. Moonen, *et al.* (2004). "Agent technology in supply chains and networks an exploration of high potential future applications." *Intelligent Agent Technology, 2004.(IAT 2004). Proceedings. IEEE/WIC/ACM International Conference on Intelligent Agent Technology*: 267-272.

Van Hillegersberg, J., J. C. Tseng, *et al.* (2006). "Hub to Higher Performance? An Internet Hub for the Vos Logistics Supply Chain." *International Journal of Advanced Manufacturing Systems* 9(1).

Van Laarhoven, P., M. Damen, *et al.* (2006). Naar een vitalere supply chain door krachtige innovatie (Eindrapportage Commissie Van Laarhoven). Hoofddorp: 44.

Van Rijswijck, H. and I. Davydenko (2007). Intelligent Agents in Transportation Planning. Delft, TNO Built Environment and Geosciences: 49.

Van Strien, P. J. (1986). "Praktijk als wetenschap." *Methodologie van het Sociaal-Wetenschappelijk Handelen. Van Gorcum*.

Venkatesh, V., M. G. Morris, *et al.* (2003). "User acceptance of information technology: Toward a unified view." *Mis Quarterly* 27(3): 425-478.

Verschuren, P. and R. Hartog (2005). "Evaluation in design-oriented research." *Quality & Quantity* 39(6): 733-762.

Vervest, P. H. M. (1994). *Communication, Not Information: An Ad Hoc Organisation of the Value Chain*. Rotterdam, Erasmus Universiteit Rotterdam, Faculteit Bedrijfskunde, Rotterdam School of Management.

Verwaal, R. (2005). Logistics. An agile sector? RSM Erasmus University. Master Thesis

Verweij, K. (2009). Containers: de kurk waarop de wereldhandel drijft. Logistiek.nl. 02/03/2009.

Vessey, I., V. Ramesh, *et al.* (2002). "Research in Information Systems: An Empirical Study of Diversity in the Discipline and Its Journals." *Journal of Management Information Systems* 19(2): 129-174.

Vokurka, R. J., J. Choobineh, *et al.* (1996). "A prototype expert system for the evaluation and selection of potential suppliers." *International Journal of Operations & Production Management* 16(12): 106-27.

Wagner, T., L. Gasser, *et al.* (2005). *Impact for Agents*. AAMAS 2005, Utrecht, The Netherlands, ACM.

Wang, E. T. G. and J. C. F. Tai (2003). "Factors affecting information systems planning effectiveness: organizational contexts and planning systems dimensions." *Information & Management* 40(4): 287-303.

Wang, Y. C. W., M. S. H. Heng, *et al.* (2005). "Business-to-business integration—the mediating effects of network structure and network atmosphere." *Production Planning & Control* 16(6): 575-585.

Wareham, J., J. G. Zheng, *et al.* (2005). "Critical themes in electronic commerce research: a meta-analysis." *Journal of Information Technology* 20(1): 1-19.

Weber, M. M. (2002). "Measuring supply chain agility in the virtual organization." *International Journal of Physical Distribution and Logistics Management* 32(7): 557-590.

Webster, J. and R. T. Watson (2002). "Analyzing the past to prepare for the future: Writing a literature review." *Mis Quarterly* 26(2): XIII-XXIII.

Welmers, A. C. (2005). Op zoek naar helderheid, een aangepast UTAUT model voor digitale radio. Enschede, University of Twente. MSc.: 103.

Willmott, S., U. Cortés, *et al.* (2004). Standards for Agent Development. *Agent-Based Software Development*. M. Luck, R. Ashri and M. D'Inverno. London, Artech House: 133-166.

Wohlin, C. (2007). "An analysis of the most cited articles in software engineering journals." *Information and Software Technology* 49(1): 2-11.

Wooldridge, M. (1999). Intelligent Agents. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. G. Weiss. Cambridge, MA, USA, MIT Press: 27-78.

Wooldridge, M. (2001). Methodologies (Chapter 10). *Introduction to Multiagent Systems*. M. Wooldridge, John Wiley & Sons, Inc. New York, NY, USA: 225-244.

Wooldridge, M. (2005). Responding to Real-World Complexity: Introduction to Multi-Agent Technology, A Magenta Technology Whitepaper.

Wooldridge, M. and N. R. Jennings (1995). "Intelligent agents: theory and practice." *Knowledge management review* 1995(January): 1-62.

Wortmann, H. and N. Szirbik (2001). "ICT issues among collaborative enterprises: from rigid to adaptive agent-based technologies." *Production Planning & Control* 12(5): 452-465.

Wright, P. C. and A. F. Monk (1990). "The use of think-aloud evaluation methods in design." *SIGCHI Bulletin* 23(1): 55-57.

Yang, S. C. (2003). "Reconceptualizing think-aloud methodology: refining the encoding and categorizing techniques via contextualized perspectives." *Computers in Human Behavior* 19(1): 95-115.

Yin, R. K. (2003). *Case Study Research: design and methods*. Thousand Oaks, Sage Publications Inc.

Zani, W. M. (1970). "Real-time information systems: A comparative economic analysis." *Management Science* 16(6): 350-355.

Zhu, K., K. L. Kraemer, *et al.* (2006). "Migration to open-standard interorganizational systems: Network effects, switching costs, and path dependency." *Mis Quarterly* 30: 515-539.

Zimmermann, R., S. Winkler, *et al.* (2006). *Agent-based Supply Chain Event Management – Concept and Assessment*. 39th Hawaii International Conference on System Sciences, IEEE.

Summary

This dissertation deals with the application of multi-agent systems (MAS) in transportation. Multi-agent systems are systems assembled from autonomously interacting agents; small software programs, which have some type of intelligence and individual behaviour. Communication and coordination (between agents) are the essential elements in such systems. Specifically the transportation domain is a potential candidate for multi-agent systems application (Fischer *et al.*, 1996; Luck *et al.*, 2004; Davidsson *et al.*, 2005; Moyaux *et al.*, 2006). Indeed, a heavy interdependence on chain partners troubles the implementation and utilisation of centralised systems. Fischer *et al.* (1996) illustrate that MAS has the potential to perform similarly to traditional Operations Research approaches. MAS could provides some fundamental advantages including an increased flexibility and a real-time character. A Logistics Service Provider's activities at the interplay with customers and suppliers are areas where large improvements can still be made (Bold and Olsson, 2005). Up until now, however, few examples of MAS applications within transportation are known (Caridi and Cavalieri, 2004). The "practical" examples Moyaux *et al.* (2006) discuss, concern only academic experiments, no concrete implementations. Chmiel *et al.* (2005) conclude that most MAS research is far from realistic.

Triggered by all this, we formulated a central explorative research question:

How can multi-agent systems be successfully applied to design and implement better performing inter-organisational systems for transportation?

This was the "*leitmotiv*" for our work over the past years. We studied literature, performed various sorts of fieldwork, made designs, constructed systems, performed simulations, and interacted with experts from science and practise.

In Chapter 2, we performed an extended literature review, which resulted in an overview of the state-of-the-art. Specifically we studied the domain challenges in SCM and transportation, enterprise information systems, and inter-organisational systems. Furthermore, we looked at MAS in general, and, more specifically, MAS in logistics. Implementation and adoption of complex enterprise systems was discussed separately. We looked at design factors of MAS. The literature study resulted in a list of paradigm shifts (see Table 2.7).

Chapter 1 reports on our first attempt to identify opportunities for MAS to support supply chains, though a framework we developed and utilised in workshops with practioners. The framework based on the SCOR model was a useful way to present and discuss a new technology and concepts in practical process terms that every professional could recognise from his/her daily job.

The observations from literature and the feedback from practioners motivated further research on multi-agent systems application in practise. We chose to do this through design research, in which validation and evaluation are essential elements.

Chapter 1 discusses different research methodological perspectives on prototype evaluation. We synthesized a number of articles on prototype evaluation, some also covering the evaluation of multi-agent systems. Many of these evaluations go in-depth utilising one evaluation mechanism and thereby neglecting certain aspects of the prototype. Following the findings from literature, we signal that one should never evaluate a system on solely one factor, nor utilise one single method. The application of agent systems in practise does come with many different aspects, and asks for an evaluation of different factors. We conclude that the evaluation of novel software prototypes in a complex (social) environment asks for a multi-method validation and evaluation approach. The table given in Table 4.1 can act as an instrument to utilise when evaluating prototypes.

In Chapter 1, we discuss a multi-agent system for real-time planning of container trucks at Post-Kogeko, an LSP. The design turned into a prototype system making use of a set of diverse technologies. The agent engine was constructed utilising the JADE toolkit, and a user-friendly UI has been built utilising GWT and GoogleMaps. The system has been constructed from the vision that to tackle the true problem, one must deal with real-time coordination of activities. The prototype has been evaluated and validated through multiple methods;using the evaluation approach motivated in Chapter 1. First of all, the design process itself proved to be an evaluation approach itself, though all iterative design steps made. Second, we tested the prototype in a simulation environment in which we compared it to two other prototype systems. Third, we organised an expert evaluation session in which we discussed the system, its design, and the concepts behind it with a group of (mainly) industrial experts. Fourth, we performed a field test of the prototype in Post-Kogeko. The evaluations revealed that the prototype is not yet production ready, however its underlying concepts are promising, its principles endorsed by practioners, and that this way of constructing systems has potential for further future work.

Chapter 6 discusses the case of APPROACH, which is an example of a real-life inter-organisational planning problem in the Port of Rotterdam that deals with planning the rotation of barges to visit container terminals. The situation is complex in nature, which became clear in a workshop organised to mimic (a controlled simplified version of) the real-world, in which barge- and terminal-operator-planners performed one day of planning operations. We describe the design steps we went through, and compare two different agent coordination mechanisms with a more traditional optimisation approach. The simulation experiments show that the multi-agent system does not perform substantially worse than optimisation does. Also here, expert interaction workshops were organised to evaluate and validate a prototype system and its underlying concepts with experts from the field. This resulted in a list of identified issues open for further research and application; something also useful for INITI8, the company that initiated this particular case.

Implementation of multi-agent systems is discussed in Chapter 1. We report research findings from the earlier-mentioned expert evaluation sessions, and confront these with literature. Furthermore we review some additional literature focussed on implementation aspects in relation to multi-agent systems.

The dissertation concludes with a conclusions and discussion chapter. In this final chapter, we critically discuss our work, link back to our initial research questions, discuss five key findings (see Table 8.1), reflect on literature, and identify several possibilities for follow-up work.

The largest contribution of this dissertation is the insight gained in the application of multi-agent systems to the transportation domain. In two design cases, we looked at a series of aspects that are important in applying MAS to transportation. Both cases resulted in a concrete prototype, which provided us hands-on experience and additional insights, but not in real implementations. The prototypes were utilised as artefacts to discuss eventual implementation with future users and experts. We found that MAS have potential for supply chains, and, in particular, transportation.

Perhaps the most important observation we made in this dissertation is that planning, as a function within supply chains, is about to go through a fundamental change. Like the mobile phone changed coordination in daily life, the concepts discussed in this dissertation have the potential to fundamentally change coordination in supply chains. They can enable a different planning & control paradigm that is focussed on coordination through communication & negotiation rather than isolated optimisation. Dealing with different perspectives on certainty and uncertainty are essential elements in this, we expect.

Samenvatting (summary in Dutch)

Deze dissertatie behandelt de toepassing van Multi-Agent Systemen (MAS) in transport. Multi-agent systemen zijn systemen geconstrueerd uit een verzameling autonoom interacterende agenten, kleine software programma's die een bepaalde mate van intelligent en individueel beslis gedrag bezitten. Communicatie en coördinatie tussen agents zijn de essentiële elementen in MAS systemen. De transport sector lijkt een interessante kandidaat voor de toepassing van multi-agent systemen (Fischer *et al.*, 1996; Luck *et al.*, 2004; Davidsson *et al.*, 2005; Moyaux *et al.*, 2006). Immers, een grote afhankelijkheid van keten partners (in het transport) compliceert de implementatie en toepassing van (traditionele) gecentraliseerde systemen. Voorts hoefen MAS in een dynamische omgeving niet onder te doen voor traditionele (OR) toepassingen, zoals Fischer *et al.* (1996) lieten zien. Daarnaast brengt MAS een aantal fundamentele voordelen met zich mee; zoals een grotere flexibiliteit, en de mogelijkheid voor online (real-time) beslissingsgedrag. Speciaal de activiteiten op het raakvlak tussen de logistiek dienstverlener en haar klanten en toeleveranciers zijn gebieden waar grote verbeteringsslagen (qua kosten, tijd, en service) te maken zijn (Bold and Olsson, 2005). MAS kunnen hier een rol in spelen. Tot nu toe zijn er weinig voorbeelden bekend van MAS toepassingen in de logistiek (Caridi and Cavalieri, 2004). De "praktijk" voorbeelden die Moyaux (2006) bespreekt betreffen zonder uitzondering wetenschappelijke experimenten; geen concrete implementaties. Chmiel *et al.* (2005) concludeerd dat het merendeel van het huidige multi-agent systemen onderzoek verre van realistisch is.

Door dit alles getriggered, formuleerden we de volgende centrale onderzoeksvraag:

Hoe kunnen multi-agent systemen succesvol toegepast worden in het ontwerp en de implementatie van beter presenterende inter-organizationele systemen voor transport?

Deze vraag werd het "*leitmotiv*" voor ons onderzoek de afgelopen jaren. Chapter 2 beschrijft de resultaten van een uitgebreide literatuur studie. Specifiek beschrijven we de uitdagingen qua Supply Chain Management (SCM), enterprise informatie systemen, en inter-organisationale systemen. Tevens beschouwen we multi-agent systemen in het algemeen, en meer specifiek MAS gericht op transport. Implementatie en adoptie van complexe bedrijfsinformatie systemen worden apart besproken. Uiteindelijk bespreken we ook factoren die bij het ontwerp van MAS een rol spelen. Deze literatuurreview resulteert in de identificatie van een lijst van paradigma veranderingen (zie Table 2.7).

In Chapter 1 beschrijven we onze eerste verkenning naar mogelijkheden van MAS om supply chains te ondersteunen. In workshops met mensen uit de industrie zijn veelbelovende gebieden geïdentificeerd, aan de hand van een door ons ontwikkeld framework. Het framework maakt het mogelijk te praten over toepassing van MAS zonder diep in te gaan op specifieke technologieën of verschillende theoretische stromingen.

De bevindingen uit de literatuur en feedback uit de praktijk motiveren verder onderzoek naar praktijk toepassing van multi-agent systemen. Dit in de vorm van ontwerponderzoek waarin prototyping en evaluatie essentiele elementen zijn.

In Chapter 1 beschouwen we de verschillende onderzoek methodologische perspectieven op de evaluatie van inter-organizationele multi-agent prototypen. We bespreken diverse artikelen waarin (MAS) prototypes worden geëvalueerd. Het merendeel van deze evaluaties beschouwt welgeteld slechts één enkel aspect van het prototype, waarbij vele andere aspecten genegeerd worden. Daar de toepassing van MAS in de praktijk met heel veel verschillende aspecten te maken krijgt, moet naar ons idee een evaluatie ook meerdere aspecten beschouwen. Onze conclusie is dat de evaluatie van nieuwe software prototypen in een complexe (sociale) omgeving vraagt om een multi-methode validatie en evaluatie. De tabel zoals gegeven in Table 4.1 kan dienen als een belangrijk instrument in de evaluatie van prototypen.

In Chapter 1 bespreken we een ontwerp voor een MAS voor de real-time planning van container trucks bij Post-Kogeko, een logistiek dienstverlener. In ons onderzoek is een initieel ontwerp stapsgewijs uitgewerkt tot een prototype systeem dat verschillende technologieën gebruikt. De agent engine maakt gebruik van de JADE toolkit. De gebruiksvriendelijke user interface gebruikt een combinatie van GWT en GoogleMaps. Het systeem is ontworpen vanuit de gedachte dat het daadwerkelijke probleem ligt op het vlak van real-time coördinatie – zowel binnen de organisatie als in de logistieke keten. Het prototype is geëvalueerd gebruikmakende van meerdere methoden; zie onze motivatie in Chapter 1. Om te beginnen bleek het ontwerp proces zelf een belangrijke evaluatie methode. Daarnaast hebben we met het prototype gesimuleerd, waar we ook een vergelijking maken met twee andere systemen. Een derde methodiek is een expert evaluatie sessie; daarin zijn het systeem zelf, het ontwerp, en de achterliggende visies besproken met een groep experts uit de industrie. De vierde evaluatie methode is een evaluatie in de Post-Kogeko praktijk, waarbij we schaduw gedraaid hebben aan de planning. Uit al deze evaluaties komt naar voren dat het prototype weliswaar nog geen productieklaar en uitontwikkeld systeem is, maar dat de concepten zoals vormgegeven resulteren in interessante resultaten en veel interesse uit de industriële praktijk.

De APPROACH casus is onderwerp van discussie in Chapter 6. APPROACH is een systeem voor het container binnenvaart rotatieplanning probleem in de haven van Rotterdam. De complexiteit van de situatie werd duidelijk in een workshop waarin binnenvaart- en terminal-operator-planners in een gecontroleerde gesimplificeerde setting een dagplanning maakten. Het hoofdstuk beschrijft de verschillende design stadia, en vergelijkt twee agent coördinatie mechanismen met een traditionele centrale optimalisatie aanpak. De simulatie experimenten tonen aan dat het multi-agent systeem niet substantieel slechter presteert dan een optimalisatie aanpak. Ook hier gebruiken we een expert interactie workshop ter evaluatie – vergelijkbaar met de workshop zoals beschreven in het voorgaande hoofdstuk. Dit resulteert in een lijst van punten interessant voor vervolg onderzoek en daadwerkelijke toepassing – een mooie aanzet ook voor het bedrijf INITI8 dat betrokken is in het gehele proces.

Implementatie aspecten van multi-agent systemen wordt besproken in Chapter 1. Additionele onderzoeksresultaten uit de voornoemde expert evaluatie sessies worden besproken, en geconfronteerd met literatuur. Tevens beschouwen we additionele literatuur betreffende implementatie aspecten in relatie tot multi-agent systemen.

Deze dissertatie besluit met een conclusie en discussie hoofdstuk (Chapter 1). Hier bespreken we kritisch ons werk, kijken terug op de initiële onderzoeksvragen, bespreken vijf key findings (zie Table 8.1), reflecteren op de literatuur, en identificeren diverse mogelijkheden voor vervolg onderzoek en toepassing.

Resumerend kunnen we concluderen dat de belangrijkste bijdrage van dit proefschrift de verkregen inzichten in de toepassing van multi-agent systemen in het SCM domein zijn. In twee ontwerp casussen, is gekeken naar diverse aspecten welke bij het toepassen van een MAS van belang zijn. In beide gevallen kwam het welliswaar tot een prototype, wat veel hands-on ervaring en additionele inzichten opleverde, maar nog niet tot een daadwerkelijke implementatie. Daarover is geinteracteerd met toekomstige gebruikers en experts.

Wat dit proefschrift voor het voetlicht brengt is het de belangrijke observatie dat planning in logistiek naar alle waarschijnlijkheid aan het begin van een fundamentele verandering staat. Zoals de mobiele telefoon een verandering in de alledaagse coördinatie, bijvoorbeeld bij het maken van een eetafspraak, te weeg heeft gebracht, zouden concepten zoals besproken in dit proefschrift ook de coördinatie in logistieke ketens fundamenteel kunnen veranderen. Het op een andere manier omgaan met zekerheid en onzekerheid zullen daarbij essentieel zijn naar verwachting.

About the Author

Hans (Johannes Maria) Moonen was born in Nijmegen, the Netherlands on 9 April 1978. He attended the athenaeum at Elshof College (later renamed Kandinsky College) in Nijmegen, where he obtained his diploma in 1996. From 1996 to 2002, Hans studied Industrial Engineering & Management Science (Technische Bedrijfskunde) at Eindhoven University of Technology, where he obtained his Masters of Science in Engineering. The last two-and-a-half years of his studies were largely spent in different positions with the enterprise software vendor Baan. At Baan, Hans did an internship (in the Quebec City, Canada office), was employed part-time in the Netherlands as a Jr. Product Consultant and eventually wrote a master's thesis that constructed a roadmap for collaborative commerce software in the electronics industry under supervision of Prof. Dr. Ir. Piet van der Vlist and Prof. Graham Sharman, MSc, MBA.

Hans started as a PhD Candidate in the Department of Decision and Information Sciences at the Rotterdam School of Management, Erasmus University in February 2003, in the group of Prof. Dr. Ir. Jo van Nunen. His PhD Research, which was supervised from the beginning by Prof. Dr. Jos van Hillegersberg and Prof. Dr. Steef van de Velde, was supported by the Erasmus Research Institute of Management (ERIM) as well as externally through participation in three Dutch government funded research projects. Hans performed his research largely within the [EET/Senter-Novem] DEAL project (2003-2007), but also worked on the [Connekt] Intelligent Agents project (2003), and the [Transumo] Diploma project (2006-2008).

Hans has (co-)authored a series of papers, and presented his research at several international conferences including: the ECIS 2004, in Turku (Finland); the IEEE IAT 2004 in Beijing (China); the ICEIS 2005 in Miami Beach (US); the ICEB 2005 in Hong-Kong (China): the 2004 and 2006 sessions of the Trail Conference in Rotterdam; the 2006 Transumo conference in Delft; the IWDL 2006 workshop in Brescia (Italy); the ICIS 2007 WITS workshop in Montreal (Canada); and the HICL 2008 in Hamburg (Germany). Hans also went to the first four consecutive sessions (2004-2007) of the CEMS SCM conference in Riezlern (Austria), which was a nice combination of theory and practise.

Next to his work as a researcher, Hans has been involved in teaching courses and giving guest lectures, and also has supervised several MSc theses. He has reviewed papers for international conferences and journals. Also, Hans was employed for one day a week for the Commissie Van Laarhoven for a period spanning more than a year. The Commissie Van Laarhoven was a national committee working on innovation in supply chains. Next to that, Hans has worked on several occasions for the European Committee's IST program as a project evaluator and a project reviewer.

In January 2009, Hans made a career switch to Logica Management Consulting where he is now part of the ITS (intelligent transport systems) consulting practise and among others active in the field of "smart logistics". In his work at Logica, Hans strives to apply the concepts presented in this dissertation in real practise. Next to his work as a consultant, Hans also has stayed affiliated with acadaemia: one day a week he works as an assistant professor at the University of Twente in the Information Systems & Change Management group headed by Prof. Dr. Jos van Hillegersberg. In this way, Hans is truly able to combine and inspire theory with practise, and practise with theory.

ERIM PhD Series in Management

Agatz, N.A.H., Demand Management in E-Fulfillment, Promotor: Prof.dr.ir. J.A.E.E. van Nunen, EPS-2009-163-LIS, ISBN: 978-90-5892-200-7, http://hdl.handle.net/1765/1

Althuizen, N.A.P., Analogical Reasoning as a Decision Support Principle for Weakly Structured Marketing Problems, Promotor: Prof.dr.ir. B. Wierenga, EPS-2006-095-MKT, ISBN: 90-5892-129-8, http://hdl.handle.net/1765/8190

Alvarez, H.L., Distributed Collaborative Learning Communities Enabled by Information Communication Technology, Promotor: Prof.dr. K. Kumar, EPS-2006-080-LIS, ISBN: 90-5892-112-3, http://hdl.handle.net/1765/7830

Appelman, J.H., Governance of Global Interorganizational Tourism Networks: Changing Forms of Co-ordination between the Travel Agency and Aviation Sector, Promotors: Prof.dr. F.M. Go & Prof.dr. B. Nooteboom, EPS-2004-036-MKT, ISBN: 90-5892-060-7, http://hdl.handle.net/1765/1199

Assem, M.J. van den, Deal or No Deal? Decision Making under Risk in a Large-Stake TV Game Show and Related Experiments, Promotor: Prof.dr. J. Spronk, EPS-2008-138-F&A, ISBN: 978-90-5892-173-4, http://hdl.handle.net/1765/13566

Baquero, G, On Hedge Fund Performance, Capital Flows and Investor Psychology, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2006-094-F&A, ISBN: 90-5892-131-X, http://hdl.handle.net/1765/8192

Berens, G., Corporate Branding: The Development of Corporate Associations and their Influence on Stakeholder Reactions, Promotor: Prof.dr. C.B.M. van Riel, EPS-2004-039-ORG, ISBN: 90-5892-065-8, http://hdl.handle.net/1765/1273

Berghe, D.A.F. van den, Working Across Borders: Multinational Enterprises and the Internationalization of Employment, Promotors: Prof.dr. R.J.M. van Tulder & Prof.dr. E.J.J. Schenk, EPS-2003-029-ORG, ISBN: 90-5892-05-34, http://hdl.handle.net/1765/1041

Berghman, L.A., Strategic Innovation Capacity: A Mixed Method Study on Deliberate Strategic Learning Mechanisms, Promotor: Prof.dr. P. Mattyssens, EPS-2006-087-MKT, ISBN: 90-5892-120-4, http://hdl.handle.net/1765/7991

Bijman, W.J.J., Essays on Agricultural Co-operatives: Governance Structure in Fruit and Vegetable Chains, Promotor: Prof.dr. G.W.J. Hendrikse, EPS-2002-015-ORG, ISBN: 90-5892-024-0, http://hdl.handle.net/1765/867

Bispo, A., Labour Market Segmentation: An investigation into the Dutch hospitality industry, Promotors: Prof.dr. G.H.M. Evers & Prof.dr. A.R. Thurik, EPS-2007-108-ORG, ISBN: 90-5892-136-9, http://hdl.handle.net/1765/10283

Blindenbach-Driessen, F., Innovation Management in Project-Based Firms, Promotor: Prof.dr. S.L. van de Velde, EPS-2006-082-LIS, ISBN: 90-5892-110-7, http://hdl.handle.net/1765/7828

Boer, C.A., Distributed Simulation in Industry, Promotors: Prof.dr. A. de Bruin & Prof.dr.ir. A. Verbraeck, EPS-2005-065-LIS, ISBN: 90-5892-093-3, http://hdl.handle.net/1765/6925

Boer, N.I., Knowledge Sharing within Organizations: A situated and Relational Perspective, Promotor: Prof.dr. K. Kumar, EPS-2005-060-LIS, ISBN: 90-5892-086-0, http://hdl.handle.net/1765/6770

Boer-Sorbán, K., Agent-Based Simulation of Financial Markets: A modular, Continuous-Time Approach, Promotor: Prof.dr. A. de Bruin, EPS-2008-119-LIS, ISBN: 90-5892-155-0, http://hdl.handle.net/1765/10870

Boon, C.T., HRM and Fit: Survival of the Fittest!?, Promotors: Prof.dr. J. Paauwe & Prof.dr. D.N. den Hartog, EPS-2008-129-ORG, ISBN: 978-90-5892-162-8, http://hdl.handle.net/1765/12606

Braun, E., City Marketing: Towards an Integrated Approach, Promotor: Prof.dr. L. van den Berg, EPS-2008-142-MKT, ISBN: 978-90-5892-180-2, http://hdl.handle.net/1765/13694

Brito, M.P. de, Managing Reverse Logistics or Reversing Logistics Management? Promotors: Prof.dr.ir. R. Dekker & Prof.dr. M. B. M. de Koster, EPS-2004-035-LIS, ISBN: 90-5892-058-5, http://hdl.handle.net/1765/1132

Brohm, R., Polycentric Order in Organizations: A Dialogue between Michael Polanyi and IT-Consultants on Knowledge, Morality, and Organization, Promotors: Prof.dr. G. W. J. Hendrikse & Prof.dr. H. K. Letiche, EPS-2005-063-ORG, ISBN: 90-5892-095-X, http://hdl.handle.net/1765/6911

Brumme, W.-H., Manufacturing Capability Switching in the High-Tech Electronics Technology Life Cycle, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. L.N. Van Wassenhove, EPS-2008-126-LIS, ISBN: 978-90-5892-150-5, http://hdl.handle.net/1765/12103

Burgers, J.H., Managing Corporate Venturing: Multilevel Studies on Project Autonomy, Integration, Knowledge Relatedness, and Phases in the New Business Development Process, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2008-136-STR, ISBN: 978-90-5892-174-1, http://hdl.handle.net/1765/13484

Campbell, R.A.J., Rethinking Risk in International Financial Markets, Promotor: Prof.dr. C.G. Koedijk, EPS-2001-005-F&A, ISBN: 90-5892-008-9, http://hdl.handle.net/1765/306

Chen, H., Individual Mobile Communication Services and Tariffs, Promotor: Prof.dr. L.F.J.M. Pau, EPS-2008-123-LIS, ISBN: 90-5892-158-1, http://hdl.handle.net/1765/11141

Chen, Y., Labour Flexibility in China's Companies: An Empirical Study, Promotors: Prof.dr. A. Buitendam & Prof.dr. B. Krug, EPS-2001-006-ORG, ISBN: 90-5892-012-7, http://hdl.handle.net/1765/307

Damen, F.J.A., Taking the Lead: The Role of Affect in Leadership Effectiveness, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2007-107-ORG, http://hdl.handle.net/1765/10282

Daniševská, P., Empirical Studies on Financial Intermediation and Corporate Policies, Promotor: Prof.dr. C.G. Koedijk, EPS-2004-044-F&A, ISBN: 90-5892-070-4, http://hdl.handle.net/1765/1518

Delporte-Vermeiren, D.J.E., Improving the Flexibility and Profitability of ICT-enabled Business Networks: An Assessment Method and Tool, Promotors: Prof. mr. dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2003-020-LIS, ISBN: 90-5892-040-2, http://hdl.handle.net/1765/359

Derwall, J.M.M., The Economic Virtues of SRI and CSR, Promotor: Prof.dr. C.G. Koedijk, EPS-2007-101-F&A, ISBN: 90-5892-132-8, http://hdl.handle.net/1765/8986

Diepen, M. van, Dynamics and Competition in Charitable Giving, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2009-159-MKT, ISBN: 978-90-5892-188-8, http://hdl.handle.net/1765/1

Dijksterhuis, M., Organizational Dynamics of Cognition and Action in the Changing Dutch and US Banking Industries, Promotors: Prof.dr.ir. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2003-026-STR, ISBN: 90-5892-048-8, http://hdl.handle.net/1765/1037

Eijk, A.R. van der, Behind Networks: Knowledge Transfer, Favor Exchange and Performance, Promotors: Prof.dr. S.L. van de Velde & Prof.dr.drs. W.A. Dolfsma, EPS-2009-161-LIS, ISBN: 978-90-5892-190-1, http://hdl.handle.net/1765/1

Elstak, M.N., Flipping the Identity Coin: The Comparative Effect of Perceived, Projected and Desired Organizational Identity on Organizational Identification and Desired Behavior, Promotor: Prof.dr. C.B.M. van Riel, EPS-2008-117-ORG, ISBN: 90-5892-148-2, http://hdl.handle.net/1765/10723

Erken, H.P.G., Productivity, R&D and Entrepreneurship, Promotor: Prof.dr. A.R. Thurik, EPS-2008-147-ORG, ISBN: 978-90-5892-179-6, http://hdl.handle.net/1765/14004

Fenema, P.C. van, Coordination and Control of Globally Distributed Software Projects, Promotor: Prof.dr. K. Kumar, EPS-2002-019-LIS, ISBN: 90-5892-030-5, http://hdl.handle.net/1765/360

Fleischmann, M., Quantitative Models for Reverse Logistics, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. R. Dekker, EPS-2000-002-LIS, ISBN: 35-4041-711-7, http://hdl.handle.net/1765/1044

Flier, B., Strategic Renewal of European Financial Incumbents: Coevolution of Environmental Selection, Institutional Effects, and Managerial Intentionality, Promotors: Prof.dr.ir. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2003-033-STR, ISBN: 90-5892-055-0, http://hdl.handle.net/1765/1071

Fok, D., Advanced Econometric Marketing Models, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2003-027-MKT, ISBN: 90-5892-049-6, http://hdl.handle.net/1765/1035

Ganzaroli, A., Creating Trust between Local and Global Systems, Promotors: Prof.dr. K. Kumar & Prof.dr. R.M. Lee, EPS-2002-018-LIS, ISBN: 90-5892-031-3, http://hdl.handle.net/1765/361

Gilsing, V.A., Exploration, Exploitation and Co-evolution in Innovation Networks, Promotors: Prof.dr. B. Nooteboom & Prof.dr. J.P.M. Groenewegen, EPS-2003-032-ORG, ISBN: 90-5892-054-2, http://hdl.handle.net/1765/1040

Gijsbers, G.W., Agricultural Innovation in Asia: Drivers, Paradigms and Performance, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2009-156-ORG, ISBN: 978-90-5892-191-8, http://hdl.handle.net/1765/1

Govers, R., Virtual Tourism Destination Image: Glocal Identities Constructed, Perceived and Experienced, Promotors: Prof.dr. F.M. Go & Prof.dr. K. Kumar, EPS-2005-069-MKT, ISBN: 90-5892-107-7, http://hdl.handle.net/1765/6981

Graaf, G. de, Tractable Morality: Customer Discourses of Bankers, Veterinarians and Charity Workers, Promotors: Prof.dr. F. Leijnse & Prof.dr. T. van Willigenburg, EPS-2003-031-ORG, ISBN: 90-5892-051-8, http://hdl.handle.net/1765/1038

Greeven, M.J., Innovation in an Uncertain Institutional Environment: Private Software Entrepreneurs in Hangzhou, China, Promotor: Prof.dr. B. Krug, EPS-2009-164-ORG, ISBN: 978-90-5892-202-1, http://hdl.handle.net/1765/1

Groot, E.A. de, Essays on Economic Cycles, Promotors: Prof.dr. Ph.H.B.F. Franses & Prof.dr. H.R. Commandeur, EPS-2006-091-MKT, ISBN: 90-5892-123-9, http://hdl.handle.net/1765/8216

Gutkowska, A.B., Essays on the Dynamic Portfolio Choice, Promotor: Prof.dr. A.C.F. Vorst, EPS-2006-085-F&A, ISBN: 90-5892-118-2, http://hdl.handle.net/1765/7994

Hagemeijer, R.E., The Unmasking of the Other, Promotors: Prof.dr. S.J. Magala & Prof.dr. H.K. Letiche, EPS-2005-068-ORG, ISBN: 90-5892-097-6, http://hdl.handle.net/1765/6963

Halderen, M.D. van, Organizational Identity Expressiveness and Perception Management: Principles for Expressing the Organizational Identity in Order to Manage the Perceptions and Behavioral Reactions of External Stakeholders, Promotor: Prof.dr. S.B.M. van Riel, EPS-2008-122-ORG, ISBN: 90-5892-153-6, http://hdl.handle.net/1765/10872

Hartigh, E. den, Increasing Returns and Firm Performance: An Empirical Study, Promotor: Prof.dr. H.R. Commandeur, EPS-2005-067-STR, ISBN: 90-5892-098-4, http://hdl.handle.net/1765/6939

Hermans. J.M., ICT in Information Services; Use and Deployment of the Dutch Securities Trade, 1860-1970, Promotor: Prof.dr. drs. F.H.A. Janszen, EPS-2004-046-ORG, ISBN 90-5892-072-0, http://hdl.handle.net/1765/1793

Hessels, S.J.A., International Entrepreneurship: Value Creation Across National Borders, Promotor: Prof.dr. A.R. Thurik, EPS-2008-144-ORG, ISBN: 978-90-5892-181-9, http://hdl.handle.net/1765/13942

Heugens, P.P.M.A.R., Strategic Issues Management: Implications for Corporate Performance, Promotors: Prof.dr.ir. F.A.J. van den Bosch & Prof.dr. C.B.M. van Riel, EPS-2001-007-STR, ISBN: 90-5892-009-9, http://hdl.handle.net/1765/358

Heuvel, W. van den, The Economic Lot-Sizing Problem: New Results and Extensions, Promotor: Prof.dr. A.P.L. Wagelmans, EPS-2006-093-LIS, ISBN: 90-5892-124-7, http://hdl.handle.net/1765/1805

Hoedemaekers, C.M.W., Performance, Pinned down: A Lacanian Analysis of Subjectivity at Work, Promotors: Prof.dr. S. Magala & Prof.dr. D.H. den Hartog, EPS-2008-121-ORG, ISBN: 90-5892-156-7, http://hdl.handle.net/1765/10871

Hooghiemstra, R., The Construction of Reality: Cultural Differences in Self-serving Behaviour in Accounting Narratives, Promotors: Prof.dr. L.G. van der Tas RA & Prof.dr. A.Th.H. Pruyn, EPS-2003-025-F&A, ISBN: 90-5892-047-X, http://hdl.handle.net/1765/871

Hu, Y., Essays on the Governance of Agricultural Products: Cooperatives and Contract Farming, Promotors: Prof.dr. G.W.J. Hendrkse & Prof.dr. B. Krug, EPS-2007-113-ORG, ISBN: 90-5892-145-1, http://hdl.handle.net/1765/10535

Huij, J.J., New Insights into Mutual Funds: Performance and Family Strategies, Promotor: Prof.dr. M.C.J.M. Verbeek, EPS-2007-099-F&A, ISBN: 90-5892-134-4, http://hdl.handle.net/1765/9398

Huurman, C.I., Dealing with Electricity Prices, Promotor: Prof.dr. C.D. Koedijk, EPS-2007-098-F&A, ISBN: 90-5892-130-1, http://hdl.handle.net/1765/9399

Iastrebova, K, Manager's Information Overload: The Impact of Coping Strategies on Decision-Making Performance, Promotor: Prof.dr. H.G. van Dissel, EPS-2006-077-LIS, ISBN: 90-5892-111-5, http://hdl.handle.net/1765/7329

Iwaarden, J.D. van, Changing Quality Controls: The Effects of Increasing Product Variety and Shortening Product Life Cycles, Promotors: Prof.dr. B.G. Dale & Prof.dr. A.R.T. Williams, EPS-2006-084-ORG, ISBN: 90-5892-117-4, http://hdl.handle.net/1765/7992

Jansen, J.J.P., Ambidextrous Organizations, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2005-055-STR, ISBN: 90-5892-081-X, http://hdl.handle.net/1765/6774

Jaspers, F.P.H., Organizing Systemic Innovation, Promotor: Prof.dr.ir. J.C.M. van den Ende, EPS-2009-160-ORG, ISBN: 978-90-5892-197-), http://hdl.handle.net/1765/1

Jennen, M.G.J., Empirical Essays on Office Market Dynamics, Promotors: Prof.dr. C.G. Koedijk & Prof.dr. D. Brounen, EPS-2008-140-F&A, ISBN: 978-90-5892-176-5, http://hdl.handle.net/1765/13692

Jiang, T., Capital Structure Determinants and Governance Structure Variety in Franchising, Promotors: Prof.dr. G. Hendrikse & Prof.dr. A. de Jong, EPS-2009-158-F&A, ISBN: 978-90-5892-199-4, http://hdl.handle.net/1765/1

Jong, C. de, Dealing with Derivatives: Studies on the Role, Informational Content and Pricing of Financial Derivatives, Promotor: Prof.dr. C.G. Koedijk, EPS-2003-023-F&A, ISBN: 90-5892-043-7, http://hdl.handle.net/1765/1043

Kaa, G. van, Standard Battles for Complex Systems: Empirical Research on the Home Network, Promotors: Prof.dr.ir. J. van den Ende & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-166-ORG, ISBN: 978-90-5892-205-2, http://hdl.handle.net/1765/1

Keizer, A.B., The Changing Logic of Japanese Employment Practices: A Firm-Level Analysis of Four Industries, Promotors: Prof.dr. J.A. Stam & Prof.dr. J.P.M. Groenewegen, EPS-2005-057-ORG, ISBN: 90-5892-087-9, http://hdl.handle.net/1765/6667

Kijkuit, R.C., Social Networks in the Front End: The Organizational Life of an Idea, Promotor: Prof.dr. B. Nooteboom, EPS-2007-104-ORG, ISBN: 90-5892-137-6, http://hdl.handle.net/1765/10074

Kippers, J., Empirical Studies on Cash Payments, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2004-043-F&A, ISBN: 90-5892-069-0, http://hdl.handle.net/1765/1520

Klein, M.H., Poverty Alleviation through Sustainable Strategic Business Models: Essays on Poverty Alleviation as a Business Strategy, Promotor: Prof.dr. H.R. Commandeur, EPS-2008-135-STR, ISBN: 978-90-5892-168-0, http://hdl.handle.net/1765/13482

Knapp, S., The Econometrics of Maritime Safety: Recommendations to Enhance Safety at Sea, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2007-096-ORG, ISBN: 90-5892-127-1, http://hdl.handle.net/1765/7913

Kole, E., On Crises, Crashes and Comovements, Promotors: Prof.dr. C.G. Koedijk & Prof.dr. M.J.C.M. Verbeek, EPS-2006-083-F&A, ISBN: 90-5892-114-X, http://hdl.handle.net/1765/7829

Kooij-de Bode, J.M., Distributed Information and Group Decision-Making: Effects of Diversity and Affect, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2007-115-ORG, http://hdl.handle.net/1765/10722

Koppius, O.R., Information Architecture and Electronic Market Performance, Promotors: Prof.dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2002-013-LIS, ISBN: 90-5892-023-2, http://hdl.handle.net/1765/921

Kotlarsky, J., Management of Globally Distributed Component-Based Software Development Projects, Promotor: Prof.dr. K. Kumar, EPS-2005-059-LIS, ISBN: 90-5892-088-7, http://hdl.handle.net/1765/6772

Krauth, E.I., Real-Time Planning Support: A Task-Technology Fit Perspective, Promotors: Prof.dr. S.L. van de Velde & Prof.dr. J. van Hillegersberg, EPS-2008-155-LIS, ISBN: 978-90-5892-193-2, http://hdl.handle.net/1765/1

Kuilman, J., The Re-Emergence of Foreign Banks in Shanghai: An Ecological Analysis, Promotor: Prof.dr. B. Krug, EPS-2005-066-ORG, ISBN: 90-5892-096-8, http://hdl.handle.net/1765/6926

Langen, P.W. de, The Performance of Seaport Clusters: A Framework to Analyze Cluster Performance and an Application to the Seaport Clusters of Durban, Rotterdam and the Lower Mississippi, Promotors: Prof.dr. B. Nooteboom & Prof. drs. H.W.H. Welters, EPS-2004-034-LIS, ISBN: 90-5892-056-9, http://hdl.handle.net/1765/1133

Le Anh, T., Intelligent Control of Vehicle-Based Internal Transport Systems, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. R. Dekker, EPS-2005-051-LIS, ISBN: 90-5892-079-8, http://hdl.handle.net/1765/6554

Le-Duc, T., Design and Control of Efficient Order Picking Processes, Promotor: Prof.dr. M.B.M. de Koster, EPS-2005-064-LIS, ISBN: 90-5892-094-1, http://hdl.handle.net/1765/6910

Leeuwen, E.P. van, Recovered-Resource Dependent Industries and the Strategic Renewal of Incumbent Firm: A Multi-Level Study of Recovered Resource Dependence Management and Strategic Renewal in the European Paper and Board Industry, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2007-109-STR, ISBN: 90-5892-140-6, http://hdl.handle.net/1765/10183

Lentink, R.M., Algorithmic Decision Support for Shunt Planning, Promotors: Prof.dr. L.G. Kroon & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2006-073-LIS, ISBN: 90-5892-104-2, http://hdl.handle.net/1765/7328

Li, T., Informedness and Customer-Centric Revenue Management, Promotors: Prof.dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-146-LIS, ISBN: 978-90-5892-195-6, http://hdl.handle.net/1765/1

Liang, G., New Competition: Foreign Direct Investment and Industrial Development in China, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-047-ORG, ISBN: 90-5892-073-9, http://hdl.handle.net/1765/1795

Liere, D.W. van, Network Horizon and the Dynamics of Network Positions: A Multi-Method Multi-Level Longitudinal Study of Interfirm Networks, Promotor: Prof.dr. P.H.M. Vervest, EPS-2007-105-LIS, ISBN: 90-5892-139-0, http://hdl.handle.net/1765/10181

Loef, J., Incongruity between Ads and Consumer Expectations of Advertising, Promotors: Prof.dr. W.F. van Raaij & Prof.dr. G. Antonides, EPS-2002-017-MKT, ISBN: 90-5892-028-3, http://hdl.handle.net/1765/869

Maeseneire, W., de, Essays on Firm Valuation and Value Appropriation, Promotor: Prof.dr. J.T.J. Smit, EPS-2005-053-F&A, ISBN: 90-5892-082-8, http://hdl.handle.net/1765/6768

Londoño, M. del Pilar, Institutional Arrangements that Affect Free Trade Agreements: Economic Rationality Versus Interest Groups, Promotors: Prof.dr. H.E. Haralambides & Prof.dr. J.F. Francois, EPS-2006-078-LIS, ISBN: 90-5892-108-5, http://hdl.handle.net/1765/7578

Maas, A.A., van der, Strategy Implementation in a Small Island Context: An Integrative Framework, Promotor: Prof.dr. H.G. van Dissel, EPS-2008-127-LIS, ISBN: 978-90-5892-160-4, http://hdl.handle.net/1765/12278

Maeseneire, W., de, Essays on Firm Valuation and Value Appropriation, Promotor: Prof.dr. J.T.J. Smit, EPS-2005-053-F&A, ISBN: 90-5892-082-8, http://hdl.handle.net/1765/6768

Mandele, L.M., van der, Leadership and the Inflection Point: A Longitudinal Perspective, Promotors: Prof.dr. H.W. Volberda & Prof.dr. H.R. Commandeur, EPS-2004-042-STR, ISBN: 90-5892-067-4, http://hdl.handle.net/1765/1302

Meer, J.R. van der, Operational Control of Internal Transport, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. R. Dekker, EPS-2000-001-LIS, ISBN: 90-5892-004-6, http://hdl.handle.net/1765/859

Mentink, A., Essays on Corporate Bonds, Promotor: Prof.dr. A.C.F. Vorst, EPS-2005-070-F&A, ISBN: 90-5892-100-X, http://hdl.handle.net/1765/7121

Meyer, R.J.H., Mapping the Mind of the Strategist: A Quantitative Methodology for Measuring the Strategic Beliefs of Executives, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2007-106-ORG, ISBN: 978-90-5892-141-3, http://hdl.handle.net/1765/10182

Miltenburg, P.R., Effects of Modular Sourcing on Manufacturing Flexibility in the Automotive Industry: A Study among German OEMs, Promotors: Prof.dr. J. Paauwe & Prof.dr. H.R. Commandeur, EPS-2003-030-ORG, ISBN: 90-5892-052-6, http://hdl.handle.net/1765/1039

Moerman, G.A., Empirical Studies on Asset Pricing and Banking in the Euro Area, Promotor: Prof.dr. C.G. Koedijk, EPS-2005-058-F&A, ISBN: 90-5892-090-9, http://hdl.handle.net/1765/6666

Moitra, D., Globalization of R&D: Leveraging Offshoring for Innovative Capability and Organizational Flexibility, Promotor: Prof.dr. K. Kumar, EPS-2008-150-LIS, ISBN: 978-90-5892-184-0, http://hdl.handle.net/1765/14081

Mol, M.M., Outsourcing, Supplier-relations and Internationalisation: Global Source Strategy as a Chinese Puzzle, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2001-010-ORG, ISBN: 90-5892-014-3, http://hdl.handle.net/1765/355

Mom, T.J.M., Managers' Exploration and Exploitation Activities: The Influence of Organizational Factors and Knowledge Inflows, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2006-079-STR, ISBN: 90-5892-116-6, http://hdl.handle.net/1765

Mulder, A., Government Dilemmas in the Private Provision of Public Goods, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-045-ORG, ISBN: 90-5892-071-2, http://hdl.handle.net/1765/1790

Muller, A.R., The Rise of Regionalism: Core Company Strategies Under The Second Wave of Integration, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-038-ORG, ISBN: 90-5892-062-3, http://hdl.handle.net/1765/1272

Nalbantov G.I., Essays on Some Recent Penalization Methods with Applications in Finance and Marketing, Promotor: Prof. dr P.J.F. Groenen, EPS-2008-132-F&A, ISBN: 978-90-5892-166-6, http://hdl.handle.net/1765/13319

Nederveen Pieterse, A., Goal Orientation in Teams: The Role of Diversity, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2009-162-ORG, http://hdl.handle.net/1765/1

Nguyen, T.T., Capital Structure, Strategic Competition, and Governance, Promotor: Prof.dr. A. de Jong, EPS-2008-148-F&A, ISBN: 90-5892-178-9, http://hdl.handle.net/1765/14005

Nieuwenboer, N.A. den, Seeing the Shadow of the Self, Promotor: Prof.dr. S.P. Kaptein, EPS-2008-151-ORG, ISBN: 978-90-5892-182-6, http://hdl.handle.net/1765/14223

Ning, H., Hierarchical Portfolio Management: Theory and Applications, Promotor: Prof.dr. J. Spronk, EPS-2007-118-F&A, ISBN: 90-5892-152-9, http://hdl.handle.net/1765/10868

Noeverman, J., Management Control Systems, Evaluative Style, and Behaviour: Exploring the Concept and Behavioural Consequences of Evaluative Style, Promotors: Prof.dr. E.G.J. Vosselman & Prof.dr. A.R.T. Williams, EPS-2007-120-F&A, ISBN: 90-5892-151-2, http://hdl.handle.net/1765/10869

Oosterhout, J., van, The Quest for Legitimacy: On Authority and Responsibility in Governance, Promotors: Prof.dr. T. van Willigenburg & Prof.mr. H.R. van Gunsteren, EPS-2002-012-ORG, ISBN: 90-5892-022-4, http://hdl.handle.net/1765/362

Paape, L., Corporate Governance: The Impact on the Role, Position, and Scope of Services of the Internal Audit Function, Promotors: Prof.dr. G.J. van der Pijl & Prof.dr. H. Commandeur, EPS-2007-111-MKT, ISBN: 90-5892-143-7, http://hdl.handle.net/1765/10417

Pak, K., Revenue Management: New Features and Models, Promotor: Prof.dr.ir. R. Dekker, EPS-2005-061-LIS, ISBN: 90-5892-092-5, http://hdl.handle.net/1765/362/6771

Pattikawa, L.H, Innovation in the Pharmaceutical Industry: Evidence from Drug Introduction in the U.S., Promotors: Prof.dr. H.R.Commandeur, EPS-2007-102-MKT, ISBN: 90-5892-135-2, http://hdl.handle.net/1765/9626

Peeters, L.W.P., Cyclic Railway Timetable Optimization, Promotors: Prof.dr. L.G. Kroon & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2003-022-LIS, ISBN: 90-5892-042-9, http://hdl.handle.net/1765/429

Pietersz, R., Pricing Models for Bermudan-style Interest Rate Derivatives, Promotors: Prof.dr. A.A.J. Pelsser & Prof.dr. A.C.F. Vorst, EPS-2005-071-F&A, ISBN: 90-5892-099-2, http://hdl.handle.net/1765/7122

Poel, A.M. van der, Empirical Essays in Corporate Finance and Financial Reporting, Promotors: Prof.dr. A. de Jong & Prof.dr. G.M.H. Mertens, EPS-2007-133-F&A, ISBN: 978-90-5892-165-9, http://hdl.handle.net/1765/13320

Popova, V., Knowledge Discovery and Monotonicity, Promotor: Prof.dr. A. de Bruin, EPS-2004-037-LIS, ISBN: 90-5892-061-5, http://hdl.handle.net/1765/1201

Pouchkarev, I., Performance Evaluation of Constrained Portfolios, Promotors: Prof.dr. J. Spronk & Dr. W.G.P.M. Hallerbach, EPS-2005-052-F&A, ISBN: 90-5892-083-6, http://hdl.handle.net/1765/6731

Prins, R., Modeling Consumer Adoption and Usage of Value-Added Mobile Services, Promotors: Prof.dr. Ph.H.B.F. Franses & Prof.dr. P.C. Verhoef, EPS-2008-128-MKT, ISBN: 978/90-5892-161-1, http://hdl.handle.net/1765/12461

Puvanasvari Ratnasingam, P., Interorganizational Trust in Business to Business E-Commerce, Promotors: Prof.dr. K. Kumar & Prof.dr. H.G. van Dissel, EPS-2001-009-LIS, ISBN: 90-5892-017-8, http://hdl.handle.net/1765/356

Quak, H.J., Sustainability of Urban Freight Transport: Retail Distribution and Local Regulation in Cities, Promotor: Prof.dr. M.B.M. de Koster, EPS-2008-124-LIS, ISBN: 978-90-5892-154-3, http://hdl.handle.net/1765/11990

Quariguasi Frota Neto, J., Eco-efficient Supply Chains for Electrical and Electronic Products, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. H.W.G.M. van Heck, EPS-2008-152-LIS, ISBN: 978-90-5892-192-5, http://hdl.handle.net/1765/1

Radkevitch, U.L, Online Reverse Auction for Procurement of Services, Promotor: Prof.dr.ir. H.W.G.M. van Heck, EPS-2008-137-LIS, ISBN: 978-90-5892-171-0, http://hdl.handle.net/1765/13497

Rinsum, M. van, Performance Measurement and Managerial Time Orientation, Promotor: Prof.dr. F.G.H. Hartmann, EPS-2006-088-F&A, ISBN: 90-5892-121-2, http://hdl.handle.net/1765/7993

Romero Morales, D., Optimization Problems in Supply Chain Management, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Dr. H.E. Romeijn, EPS-2000-003-LIS, ISBN: 90-9014078-6, http://hdl.handle.net/1765/865

Roodbergen, K.J., Layout and Routing Methods for Warehouses, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2001-004-LIS, ISBN: 90-5892-005-4, http://hdl.handle.net/1765/861

Rook, L., Imitation in Creative Task Performance, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2008-125-ORG, http://hdl.handle.net/1765/11555

Rosmalen, J. van, Segmentation and Dimension Reduction: Exploratory and Model-Based Approaches, Promotor: Prof.dr. P.J.F. Groenen, EPS-2009-165-MKT, ISBN: 978-90-5892-201-4, http://hdl.handle.net/1765/1

Samii, R., Leveraging Logistics Partnerships: Lessons from Humanitarian Organizations, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. L.N. Van Wassenhove, EPS-2008-153-LIS, ISBN: 978-90-5892-186-4, http://hdl.handle.net/1765/1

Schaik, D. van, M&A in Japan: An Analysis of Merger Waves and Hostile Takeovers, Promotors: Prof.dr. J. Spronk & Prof.dr. J.P.M. Groenewegen, EPS-2008-141-F&A, ISBN: 978-90-5892-169-7, http://hdl.handle.net/1765/13693

Schauten, M.B.J., Valuation, Capital Structure Decisions and the Cost of Capital, Promotors: Prof.dr. J. Spronk & Prof.dr. D. van Dijk, EPS-2008-134-F&A, ISBN: 978-90-5892-172-7, http://hdl.handle.net/1765/13480

Schramade, W.L.J., Corporate Bonds Issuers, Promotor: Prof.dr. A. De Jong, EPS-2006-092-F&A, ISBN: 90-5892-125-5, http://hdl.handle.net/1765/8191

Schweizer, T.S., An Individual Psychology of Novelty-Seeking, Creativity and Innovation, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-048-ORG, ISBN: 90-5892-077-1, http://hdl.handle.net/1765/1818

Six, F.E., Trust and Trouble: Building Interpersonal Trust Within Organizations, Promotors: Prof.dr. B. Nooteboom & Prof.dr. A.M. Sorge, EPS-2004-040-ORG, ISBN: 90-5892-064-X, http://hdl.handle.net/1765/1271

Slager, A.M.H., Banking across Borders, Promotors: Prof.dr. R.J.M. van Tulder & Prof.dr. D.M.N. van Wensveen, EPS-2004-041-ORG, ISBN: 90-5892-066–6, http://hdl.handle.net/1765/1301

Sloot, L., Understanding Consumer Reactions to Assortment Unavailability, Promotors: Prof.dr. H.R. Commandeur, Prof.dr. E. Peelen & Prof.dr. P.C. Verhoef, EPS-2006-074-MKT, ISBN: 90-5892-102-6, http://hdl.handle.net/1765/7438

Smit, W., Market Information Sharing in Channel Relationships: Its Nature, Antecedents and Consequences, Promotors: Prof.dr.ir. G.H. van Bruggen & Prof.dr.ir. B. Wierenga, EPS-2006-076-MKT, ISBN: 90-5892-106-9, http://hdl.handle.net/1765/7327

Sonnenberg, M., The Signalling Effect of HRM on Psychological Contracts of Employees: A Multi-level Perspective, Promotor: Prof.dr. J. Paauwe, EPS-2006-086-ORG, ISBN: 90-5892-119-0, http://hdl.handle.net/1765/7995

Speklé, R.F., Beyond Generics: A closer Look at Hybrid and Hierarchical Governance, Promotor: Prof.dr. M.A. van Hoepen RA, EPS-2001-008-F&A, ISBN: 90-5892-011-9, http://hdl.handle.net/1765/357

Stam, D.A., Managing Dreams and Ambitions: A Psychological Analysis of Vision Communication, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2008-149-ORG, http://hdl.handle.net/1765/14080

Stienstra, M., Strategic Renewal in Regulatory Environments: How Inter- and Intra-organisational Institutional Forces Influence European Energy Incumbent Firms, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda,, EPS-2008-145-STR, ISBN: 978-90-5892-184-0, http://hdl.handle.net/1765/13943

Sweldens, S.T.L.R., Evaluative Conditioning 2.0: Direct versus Associative Transfer of Affect to Brands, Promotor: Prof.dr. S.M.J. van Osselaer, EPS-2009-167-MKT, ISBN: 978-90-5892-204-5, http://hdl.handle.net/1765/1

Szkudlarek, B.A., Spinning the Web of Reentry: [Re]connecting reentry training theory and practice, Promotor: Prof.dr. S.J. Magala, EPS-2008-143-ORG, ISBN: 978-90-5892-177-2, http://hdl.handle.net/1765/13695

Teunter, L.H., Analysis of Sales Promotion Effects on Household Purchase Behavior, Promotors: Prof.dr.ir. B. Wierenga & Prof.dr. T. Kloek, EPS-2002-016-MKT, ISBN: 90-5892-029-1, http://hdl.handle.net/1765/868

Tims, B., Empirical Studies on Exchange Rate Puzzles and Volatility, Promotor: Prof.dr. C.G. Koedijk, EPS-2006-089-F&A, ISBN: 90-5892-113-1, http://hdl.handle.net/1765/8066

Tuk, M.A., Is Friendship Silent When Money Talks? How Consumers Respond to Word-of-Mouth Marketing, Promotors: Prof.dr.ir. A. Smidts & Prof.dr. D.H.J. Wigboldus, EPS-2008-130-MKT, ISBN: 978-90-5892-164-2, http://hdl.handle.net/1765/12702

Valck, K. de, Virtual Communities of Consumption: Networks of Consumer Knowledge and Companionship, Promotors: Prof.dr.ir. G.H. van Bruggen & Prof.dr.ir. B. Wierenga, EPS-2005-050-MKT, ISBN: 90-5892-078-X, http://hdl.handle.net/1765/6663

Valk, W. van der, Buyer-Seller Interaction Patterns During Ongoing Service Exchange, Promotors: Prof.dr. J.Y.F. Wynstra & Prof.dr.ir. B. Axelsson, EPS-2007-116-MKT, ISBN: 90-5892-146-8, http://hdl.handle.net/1765/10856

Verheul, I., Is There a (Fe)male Approach? Understanding Gender Differences in Entrepreneurship, Prof.dr. A.R. Thurik, EPS-2005-054-ORG, ISBN: 90-5892-080-1, http://hdl.handle.net/1765/2005

Verwijmeren, P., Empirical Essays on Debt, Equity, and Convertible Securities, Promotors: Prof.dr. A. de Jong & Prof.dr. M.J.C.M. Verbeek, EPS-2009-154-F&A, ISBN: 978-90-5892-187-1, http://hdl.handle.net/1765/14312

Vis, I.F.A., Planning and Control Concepts for Material Handling Systems, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. R. Dekker, EPS-2002-014-LIS, ISBN: 90-5892-021-6, http://hdl.handle.net/1765/866

Vlaar, P.W.L., Making Sense of Formalization in Interorganizational Relationships: Beyond Coordination and Control, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2006-075-STR, ISBN 90-5892-103-4, http://hdl.handle.net/1765/7326

Vliet, P. van, Downside Risk and Empirical Asset Pricing, Promotor: Prof.dr. G.T. Post, EPS-2004-049-F&A, ISBN: 90-5892-07-55, http://hdl.handle.net/1765/1819

Vlist, P. van der, Synchronizing the Retail Supply Chain, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr. A.G. de Kok, EPS-2007-110-LIS, ISBN: 90-5892-142-0, http://hdl.handle.net/1765/10418

Vries-van Ketel E. de, How Assortment Variety Affects Assortment Attractiveness: A Consumer Perspective, Promotors: Prof.dr. G.H. van Bruggen & Prof.dr.ir. A. Smidts, EPS-2006-072-MKT, ISBN: 90-5892-101-8, http://hdl.handle.net/1765/7193

Vromans, M.J.C.M., Reliability of Railway Systems, Promotors: Prof.dr. L.G. Kroon, Prof.dr.ir. R. Dekker & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2005-062-LIS, ISBN: 90-5892-089-5, http://hdl.handle.net/1765/6773

Vroomen, B.L.K., The Effects of the Internet, Recommendation Quality and Decision Strategies on Consumer Choice, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2006-090-MKT, ISBN: 90-5892-122-0, http://hdl.handle.net/1765/8067

Waal, T. de, Processing of Erroneous and Unsafe Data, Promotor: Prof.dr.ir. R. Dekker, EPS-2003-024-LIS, ISBN: 90-5892-045-3, http://hdl.handle.net/1765/870

Wall, R.S., Netscape: Cities and Global Corporate Networks, Promotor: Prof.dr. G.A. van der Knaap, EPS-2009-169-ORG, ISBN: 978-90-5892-207-6, http://hdl.handle.net/1765/1

Watkins Fassler, K., Macroeconomic Crisis and Firm Performance, Promotors: Prof.dr. J. Spronk & Prof.dr. D.J. van Dijk, EPS-2007-103-F&A, ISBN: 90-5892-138-3, http://hdl.handle.net/1765/10065

Wennekers, A.R.M., Entrepreneurship at Country Level: Economic and Non-Economic Determinants, Promotor: Prof.dr. A.R. Thurik, EPS-2006-81-ORG, ISBN: 90-5892-115-8, http://hdl.handle.net/1765/7982

Wielemaker, M.W., Managing Initiatives: A Synthesis of the Conditioning and Knowledge-Creating View, Promotors: Prof.dr. H.W. Volberda & Prof.dr. C.W.F. Baden-Fuller, EPS-2003-28-STR, ISBN: 90-5892-050-X, http://hdl.handle.net/1765/1042

Wijk, R.A.J.L. van, Organizing Knowledge in Internal Networks: A Multilevel Study, Promotor: Prof.dr.ir. F.A.J. van den Bosch, EPS-2003-021-STR, ISBN: 90-5892-039-9, http://hdl.handle.net/1765/347

Wolters, M.J.J., The Business of Modularity and the Modularity of Business, Promotors: Prof. mr. dr. P.H.M. Vervest & Prof. dr. ir. H.W.G.M. van Heck, EPS-2002-011-LIS, ISBN: 90-5892-020-8, http://hdl.handle.net/1765/920

Yang, J., Towards the Restructuring and Co-ordination Mechanisms for the Architecture of Chinese Transport Logistics, Promotor: Prof.dr. H.E. Harlambides, EPS-2009-157-LIS, ISBN: 978-90-5892-198-7, http://hdl.handle.net/1765/1

Yu, M, Enhancing Warehouse Perfromance by Efficient Order Picking, Promotor: Prof.dr. M.B.M. de Koster, EPS-2008-139-LIS, ISBN: 978-90-5892-167-3, http://hdl.handle.net/1765/13691

Zhang, X., Strategizing of Foreign Firms in China: An Institution-based Perspective, Promotor: Prof.dr. B. Krug, EPS-2007-114-ORG, ISBN: 90-5892-147-5, http://hdl.handle.net/1765/10721

Zhu, Z., Essays on China's Tax System, Promotors: Prof.dr. B. Krug & Prof.dr. G.W.J. Hendrikse, EPS-2007-112-ORG, ISBN: 90-5892-144-4, http://hdl.handle.net/1765/10502

Zwart, G.J. de, Empirical Studies on Financial Markets: Private Equity, Corporate Bonds and Emerging Markets, Promotors: Prof.dr. M.J.C.M. Verbeek & Prof.dr. D.J.C. van Dijk, EPS-2008-131-F&A, ISBN: 978-90-5892-163-5, http://hdl.handle.net/1765/12703

Notes

Erasmus Research Institute of Management - ERIM

MULTI-AGENT SYSTEMS FOR TRANSPORTATION PLANNING AND COORDINATION

Many transportation problems are in fact coordination problems: problems that require communication, coordination and negotiation to be optimally solved. However, most software systems targeted at transportation have never approached it this way, and have instead concentrated on centralised optimisation.

Multi-agent systems (MAS) are a different approach to building software systems. Such systems are assembled from autonomously interacting agents; agents are small software programs, which have some type of intelligence and individual behaviour. Communication and coordination (between agents) are the essential elements in the construction of MAS. The transportation domain is often referred to as a potential candidate for the application of MAS.

In this dissertation, we discuss two MAS design cases related to the transport of containers. Both cases resulted in concrete prototypes, which let us evaluate a series of aspects important in applying MAS in transportation. We demonstrate the importance of a multi-method validation and evaluation approach. The prototypes were furthermore utilised as artefacts to discuss eventual implementation with future users and experts.

One of our most important observations is that planning, as a function within supply chains, is about to go through a fundamental change. Like the mobile phone changed the way people coordinate in daily life, the concepts discussed in this dissertation have the potential to fundamentally change coordination in supply chains. As part of this fundamental change, a different perspective on certainty and uncertainty is essential.

Design & layout: B&T Ontwerp en advies (www.b-en-t.nl) Print: Haveka (www.haveka.nl)

ERIM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are Rotterdam School of Management (RSM), and the Erasmus School of Economics (ESE). ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focused on the management of the firm in its environment, its intra- and interfirm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced doctoral programme in Research in Management. Within ERIM, over three hundred senior researchers and PhD candidates are active in the different research programmes. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.

ERIM PhD Series
Research in Management

Erasmus Research Institute of Management - **ERIM**
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
P.O. Box 1738, 3000 DR Rotterdam
The Netherlands

Tel. +31 10 408 11 82
Fax +31 10 408 96 40
E-mail info@erim.eur.nl
Internet www.erim.eur.nl