Agent models to control complexity in multi modal transport

Stijn Bernaer[1], Patrick De Causmaecker[2],
Greet Vanden Berghe[1], Tijs Vermeulen[1]

[1]KaHo St.-Lieven, research group Information Technology
[2]Katholieke Universiteit Leuven Campus Kortrijk,
Computer Science and Information Technology

Abstract

We present a Multi-Agent System for communication in the transport sector. It is part of an 'Intelligent Communication Platform for Multi Modal Transport' that is called MamMoeT. The motivation for a communication platform stems from the observed need for support in maintaining evolving partnerships as well as for the establishment of new partnerships. We observed a lack of commonly accepted tools to obtain quantifiable and live information. Examples are position of freight and vehicles, expected arrival and waiting times as well as alternative routes and transport modes. Multi-Agent Systems provide appropriate models for the communication platform. In the transport sector, which is distributed by nature, models with autonomous agents representing each of the logistic players match better to the requirements than one central information system would. The presented model explicitly addresses the transparency and privacy issues in order to increase the level of acceptance of this type of software platform. We introduce three models: an agent model, a privacy model and a communication model and elaborate upon negotiation, communication and cooperation skills of the agents.
Keywords: *agent models, privacy, transport, logistics*

1 Introduction

The transport sector is under pressure. Overloaded roads lead to both economical and ecological problems. The region of Flanders (Belgium) is an attractive area due to its dense transport infrastructure and its central position in Europe. However, a system supporting the transport players could relief the current pressure and conserve the competitiveness of the region.

Recently more and more attention is paid to the 'modal shift'. It promotes transport modes other than road transport and any combination of different transport modes. The MamMoeT project ('Intelligent Communication Platform for Multi Modal Transport') [1] [2] is designed to support the modal shift.

The rationale for an agent based intelligent communication platform is threefold:

1. On the one hand, there is a need of support to easily *find partners* in the logistic chain. Logistic companies want to avoid empty transport and manufacturers require information about alternative transport solutions. Often e.g. a loader (an organiser of transport flow) has connections with his well known partners. However, during busy times the loader lacks a transparent view on other possibilities. Moreover a clear view on alternative transport modes and a combination of them can help in reducing time and costs.

2. In addition, a more *efficient information flow* is needed. Loaders e.g. currently lack information about the position of freight, the expected arrival times. Although useful information is spread across the different transport players, it is often not in the right place, neither at the right time nor in the right format.

3. The transport domain, which is distributed by nature, requires a corresponding approach (from a developer's point of view).

We designed and implemented three agent models to provide an easy toolbox that offers a solution for the transport domain.

This paper is structured as follows. First we describe related work applying MAS in the transport domain. Then we present the multi-agent model. In this multi-agent architecture we address three core issues in three models: (1) we model individual agents as BDI *decision* makers as described in [8] (2) *communication* is subject to privacy restrictions described in the privacy model (3) finally *cooperation* between agents follows a communication model. Next we describe *agent types* and outline a real world scenario. Finally, we discuss observed success factors including graphical user interfaces and integration.

2 Related work

Managing complexity by using models is a wide-spread methodology [12]. Multi-Agent Systems have proven their surplus value in the domains of logistics, transport and manufacturing [9]. The MAGENTA's commercial multi-agent systems technology [7] e.g. offers support in scheduling for 'very large crude carriers' using tools to model the domain (Ontology Management Toolkit) and to design, debug agent related features (Virtual Market Engine).

Concerning several transport modes, Chiu et al [4] present a model for passenger transport supporting tourists and moving workforces in a metropolis using agents to build a flexible and scalable solution.

Privacy related terms such as trust and confidence recently gained interest in projects such as PISA (Privacy Incorporated Software Agent [3]). One possibility is to provide 'Closed User Groups' (CUG) in which end users can gradually expose their needs to potential partners. So privileges are given in the sense that one first addresses partners with an excellent reputation, if no solution is found, a group of partners with less reputation will be contacted and so on.

Systems (especially in road transport) that optimise the handling of freight and transport already exist within one or more related companies. Since data is internally used, privacy surveillance is not necessarily implemented. Such single modal solutions usually do not fit for integration with other systems or for supporting multi modal transport.

3 Multi-Agent Architecture

3.1 Agent model

An agent model has to deal with particular concerns within the MamMoeT project. In Fig. 1 we present an agent model inspired by the BDI-model [8].

The lower layer in the figure represents the agent's *beliefs*. The combination of interaction protocols (IP's) and actions makes up the *intentions*. Finally the agent type itself and the 'matching/decision making layer' define the *desires* (the design goals of the agent). Depending on the agent type, a set of actions (in BDI-terms: plans, activities) are available. With each of these actions a couple of IP's is associated. The IP's regulate the exchange of messages. Concerning this message traffic we designed a privacy model (section 3.2) through which every message should pass (in-coming as well as out-going).

Any incoming message first passes through the privacy 'filter'. Afterwards the 'decision making layer' uses the type of the message to determine which action (plan) to execute and thus which IP's are applicable to fulfill the goal/desire (also determined by the agent type). If needed the beliefs of the agent can be updated with information obtained through the exchange of

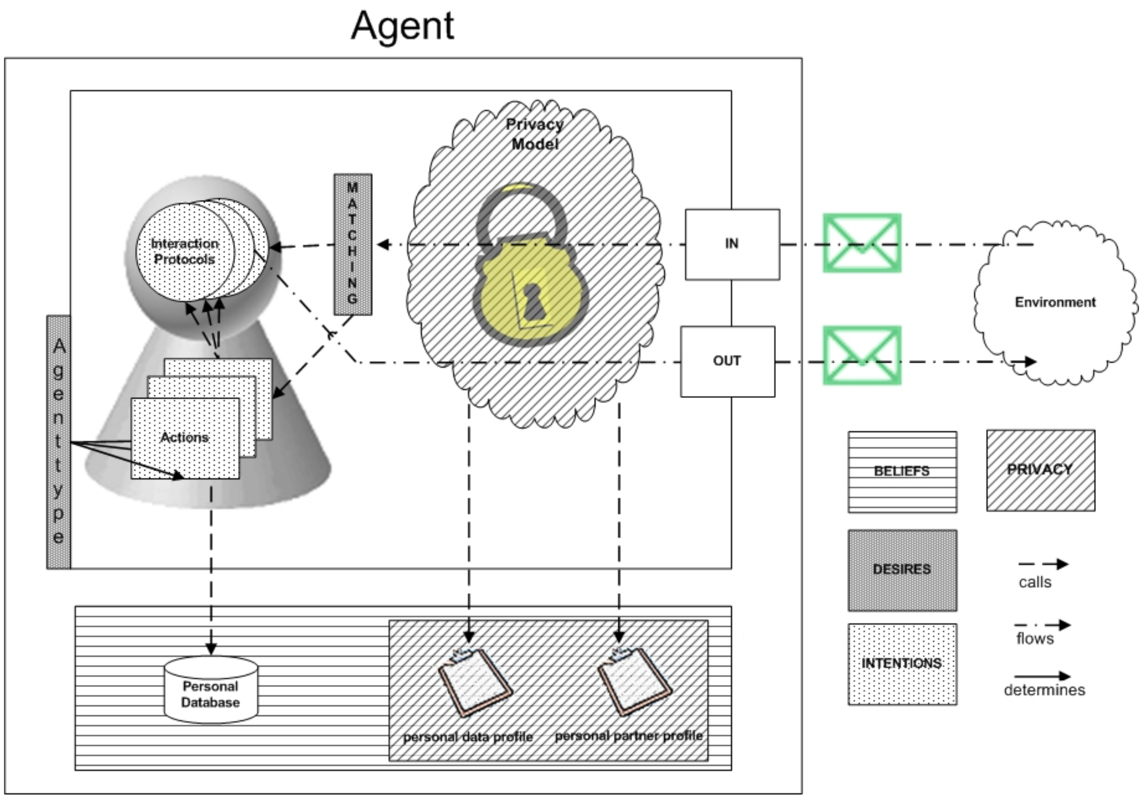

Figure 1: Agent Model

messages. The strategy of the agent is determined by the decision making part of the plans which one can visualise as a mapping of a message type to a plan. Characteristics such as ontology, sender, interaction protocol make up the message type. This variety allows to flexibly select a plan corresponding to a specific domain functionality.

With this agent model inspired by the BDI[1]-model, we want to implement the agents efficiently. This is made possible by disentangling the complex information streams of the real world and by projecting these to protocols (as a part of the intention layer). As a result, the code for exchanging messages, code for performing actions, for preserving privacy is reusable and easily adaptable, anticipating the future whimsicality characteristic of the ever changing transport domain.

[1]Believe, Desire, Intention

3.2 Privacy Model

Tim Berners-Lee stated that “No-one will take part in the new web-like way of working if they do not feel certain that private information will stay private”. This utterance perfectly summarises our point of view. On one hand a more transparent view on the transport sector is needed but this can not be accomplished if a system can not guarantee privacy. In [2] a straightforward model for preserving privacy is introduced.

3.3 Agent Community Model

Actors are often related by similar activities. Using agent communities [5] is a very natural way of projecting such a real world situation in which partners meet to inform each other and to discuss business. In [2] we elaborate on this methodology.

4 Agent types

We distinguish four types of agent. (1) Negotiation agents correspond to real world negotiators and act as delegates of those entities in our model. (2) Information agents relate to data sources and either represent real world data suppliers or serve as a wrap around an external data system. (3) Observer agents support the negotiation agents by offering a ‘publish and subscribe’ service. (4) Finally, the reasoner agents relieve the negotiation agents by isolating their reasoner capabilities into a new agent.

4.1 Negotiation agents

When attempting to match demand and supply, negotiation is indispensable. That is the case in the communication platform just as in the real world. In both cases agents depend on each other’s helpfulness [13]. In MamMoeT , we have not explicitly modelled helpfulness . An agent can be considered helpful to another agent if (1) their demand/supply request matches the supply/demand of the other agent (2) the agent has earned a reputation of the other agent. For example, a loader who offers a load of sand will not be connected to container transporters since containers are not suitable for carrying sand. The loader will be connected to bulk transporters with which he already has good relations. Once the contract between is established, the agents can start negotiating about constraints regarding the load, trajectory, pick up and delivery time, etc. We deliberately keep away from cost issues in the model since it is not the intention of the MamMoeT platform to affect the market.

The agents will use ‘calls for proposal’ to enter into agreements. We do not adopt a Blackboard protocol as-is mainly for privacy reasons [10] [6]. We do not want **all** information to be public for **every** party. Previous projects that aimed at improving the communication failed. One of the main reasons for the failure seems to be distrust in the Blackboard which announced everybody’s requests.

4.2 Information agents

Information agents are *broker* agents as they will seek and gather information on other agents and/or on existing (legacy) systems. An agent that is on the look-out for specific information only has to communicate with one information agent through well defined messages. Examples include the ‘Transport Checker Agent’ and the ‘Freight Checker Agent’. One can address one of these agents to get some (static) information about respectively transport suppliers or freight suppliers. Behind the scenes the information agents will comb for the requested data resulting in a uniform outcome (named event). If we want MamMoeT to easily integrate with existing systems, ‘wrapper’ agents can be used. An extra layer will be wrapped around the existing system in order to act like an agent within the MAS. ‘Web Services’ can act as the glue between these wrapper agents and the legacy systems [11].

4.3 Observer agents

In order to spread events fastly -without overwhelming users with trivial ones- and react on events occurring somewhere in the platform (see first rationale in the introduction) we foresee observer agents. All transport parties can subscribe to their services and be notified if required. An observer agent questions some information agents in order to find out which events are interesting for an agent to subscribe to. An agent can proclaim his interest (subscribe) in particular events so that he will be alerted when such event happens. Alternatively, information agents can spread events (publisher).

4.4 Reasoner agents

Reasoner agents receive and process messages (containing some semantic language e.g. RDF content) from other agents. Combining all these messages with some user-defined rules (constraints of the end-users, whether they are

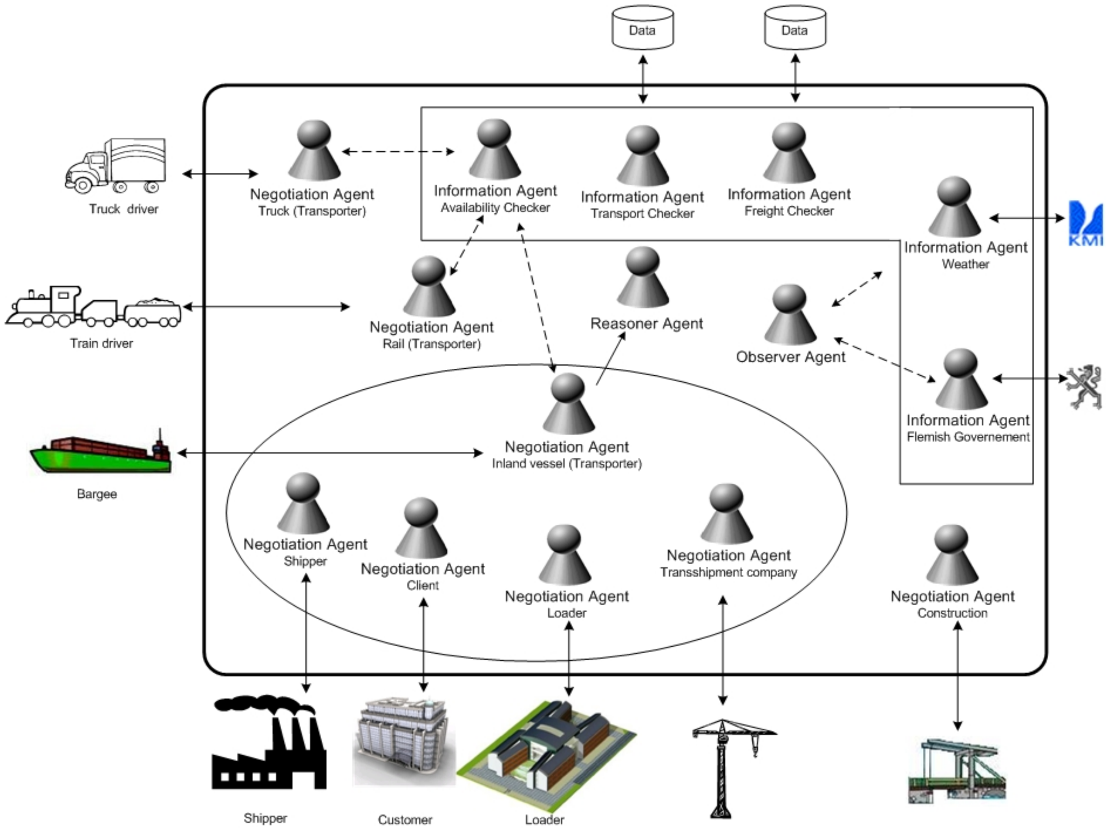

Figure 2: Interrelation between agent types

related to privacy or business) can lead to the derivation of additional data. The reasoning process can be a resource and time consuming task. For example an agent looking for a possible route for going from A to B, can start a 'reasoner agent' being responsible for making enquiries with other agents and processing recovered information to make an intelligent decision on which route to follow.

An attractive characteristic of agents is mobility. Especially reasoner agents profit from this. The negotiation agents can start reasoner agents (as slave agents) on computers with extra resources (memory, cpu) so that the computer on which the negotiation agent operates is not too heavily loaded. Consider a case where an inland bargee uses his mobile phone to connect to the MamMoeT platform. The mobile device is not capable of accommodating heavily loaded agents.

4.5 Interrelations between agent types

Fig. 2 presents the different agent types and their role in the platform. The figure also depicts the interactions with the real world. Information agents gather information in the real world outside the MamMoeT platform. They communicate with data sources of the parties on the platform, with governmental information sources on roads and waterways, with information instances on weather and weather forecast, etc. All that information will be gathered, processed and, if requested for, passed on to negotiation or observer agents.

When their own resources are scarce negotiation agents will call in reasoner agents. The majority of communication is established between the negotiation agents themselves and with the outside world. The eclipse in the figure shows a formed group of transport players making up one 'logistic chain'.

5 Scenario

In this paragraph we present a real-world scenario that considers a holistic view (see also Fig. 2) on the sequence within the settlement of a freight transport. In a first phase, partners need to find each other and make agreements. In a second phase, an efficient support during the transport of the freight is needed where relevant information is brought to the proper actors. Consider in this scenario a skipper specialised in producing fertilisers, established at location A and a customer at location B who ordered 280 tons of fertilisers. For transport handling, the customer appeals to a loader and gives some time constraints.

5.1 Defects in the current situation

Currently it is quite common that the loader (or transport company in general) makes agreements by telephone. This results in avoidable costs (personnel, time, infrastructure). Due to the lack of standards concerning formats on (electronic) documents, employees often produce the same information multiple times e.g. they type in text that was sent by fax.

Moreover, when the known contacts do not allow to satisfy the accepted tasks, the loader has to cancel some orders and may on the long term, lose some good relations.

Tedious situations (e.g. calamities) during the transport of the freight can be avoided by efficiently communicating events that happen somewhere in the system. Especially when actors belonging to different 'logistic chains'

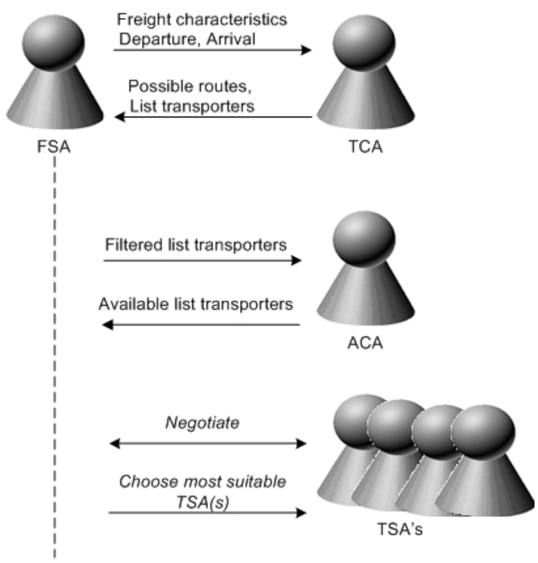

Figure 3: Match of supply and demand: a sample interaction sequence

want to communicate, they often face large administrative burdens. There is an obvious need for a streamlined communication where parties with the right privileges can hook onto.

5.2 Opportunities with MAS

Fig. 3 outlines the interaction sequence for establishing an agreement. The loader serving the client application prompts the business data (the freight and place characteristics) into his '*freight supplier agent*' (FSA, negotiation agent). That agent subsequently gets in touch with the *transport checker agent* (TCA, information agent) which works out which transporters physically match the given constraints using existing databases. Starting from the obtained sublist another information agent, the *availability checker agent* (ACA) will be addressed. The ACA verifies which transporters are available, in which transporter's schedule the job fits (e.g. without inducing empty transport when travelling to the start place). For performance reasons the ACA can cache availability parameters for some time. Next, the FSA negotiates with the '*transporter agents*' (satisfying physical as well as available constraints) of the resulting sublist of the ACA. Finally the FSA selects, based on e.g. earlier experiences, the required transporter supplier agents (TSA) with which he composes a complete transport route. At that moment one 'agent community group' is formed containing the loader, the depicted transporters, the skipper and eventually other agents which are in some way connected to the transport chain (e.g. the customer, transhipment company). After the transport handling is finished, the FSA evaluates his experience with this party, resulting in an adjustment the agent's reputation.

Note that the number of potential players is reduced gradually by contacting the information agents ACA and TCA. This already prevents unnecessary exchanges of privacy related messages. Moreover, the end user can use the CUG principle, starting to negotiate with partners with the best reputation and as long as no solution is obtained contacting parties with a lower reputation.

An advantage over negotiating in a Blackboard environment is that the privacy settings can be maintained locally (per agent).

Regularly, unforeseen occurrences take place that impact upon different players somewhere in the system. Thanks to observer agents such events can be handled efficiently. A company can make a submission for specific events (e.g. sudden changes in the water level of waterways) with an observer agent. As soon as a government publishes this event on a specific trajectory, the registered companies for which this is relevant will receive a notification.

A slightly different situation arises when companies belonging to the same transport chain set up communication. Consider a contract between a skipper, a loader and a customer belonging to the same 'agent community model'. As soon as the loader perceives an abnormal delay on the trajectory of the assigned transporters, a report will be sent to the other parties. That is very important as the customer can thus reschedule as there are extra constraints in the supply chain. An even more complex situation arises when communities are entangled. Suppose that the transporter in the scenario has a consecutive transport with a different loader and customer. In that situation, information has to flow efficiently from one community to another.

The MAS approach thus brings in some advantages; it

1. diminishes administrative burden (avoid prompting duplicate information) by constantly being online via the personal negotiation agent
2. makes a transparent view of the market while preserving privacy
3. couples the different modal transport sub-domains
4. supports efficient communication between related partners.

6 Practical success factors

So far we focussed on projecting a real world situation to MAS which results in an important decision support system for users and eases modelling and implementation for developers. On the other hand it requires an effort to increase the acceptance rate of the system for a broad public. That includes several sub aspects: (1) taking away fear by paying attention to privacy concerns (2) providing a confident looking graphical user interface (GUI) (3) making the platform easy to integrate with existing systems.

The last two aspects will be briefly discussed.

The statement of Tim Berners-Lee also has impact on the graphical aspects. During conversations we perceived that the end users attach a great value to the GUI. In fact, for end users the GUI determines how they perceive the system. We resolved to pay sufficient attention to the GUI. The confidential appearance of a GUI can support trust in a software system. Moreover it is a good means for discussing features and software requirements with non-technical people.

In the history of computer science we notice an evolution originating from a mechanistic world vision (the machine is identical to the user interface) via assembly, procedural and object-oriented languages towards agent oriented programming. Every step introduces hereby an extra level of abstraction. It makes users feel that agents are really extension of themselves.

The user interface becomes again identical to the machine, i.e. the agent model which is a projection of the real world.

Wooldridge outlined in his book "An Introduction to MultiAgent Systems" [14] the human-oriented evolution through computer science history. By making a confident GUI we want to bridge the gap between Multi-Agent Systems and the end-users.

In our implementation we provide each negotiation agent with an own personal database. The database holds settings (among which privacy) and business data. By loosely coupling the database and the agent system, integration burden is diminished.

7 Conclusion and further work

The transport and logistics domain is very complex and automatisation on decision support has not yet found its way to all the players in the sector. In this paper, we have presented an architecture through which, by making the market partly more transparent, transport players can more easily meet the orders of the day and construct long term relationships. When facing the diversity of problems in the sector, a Multi-Agent System seemed very applicable to tackle negotiation, communication and cooperation problems.

By accurately developing a Multi-Agent Architecture (consisting of three models) and different agent types, we managed to control the complex domain characterised by a big tangle of relations between different actors. The agent models enable information flow more efficient in and out the tangle. This control over the complexity is made possible by the agent paradigm offering a next step in abstraction. To increase acceptance rate a confidential looking GUI is needed (especially for the non-technical people), that way the gap between software agents and the end users become smaller. Having demonstrated the usability of software agents in a complex domain, further investigation in optional plug-ins for the platform is desirable. For example several planning tools can be integrated in the platform: production planning, lock planning, ...

References

[1] Mammoet: Intelligent communication platform for multi-modal transport. MamMoeT IWT/TETRA 40191, `http://ingenieur.kahosl.be/projecten/mammoet/ (2005)`.

[2] S. Bernaer, P. D. Causmaecker, M. Meganck, and G. V. Berghe. Designing trust with software agents: A case study. To appear in *ICES, Journal of Information, Communication and Ethics in Society*, 2006.

[3] J. J. Borking. Privacy incorporated software agent (pisa): proposal for building a privacy guardian for the electronic age. In *International workshop on Designing privacy enhancing technologies*, pages 130–140, New York, NY, USA, 2001. Springer-Verlag New York, Inc.

[4] D. K. W. Chiu, O. K. F. Lee, H. fung Leung, E. W. K. Au, and M. C. W. Wong. Multi-modal agent based mobile route advisory system for public transport network. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, 2005.

[5] D. Cristo. Community building and multi-agent systems: Creating an agent as a virtual community member. Master's thesis, Intelligent Interactive Distributed Systems Group, Vrij Universiteit Amsterdam, 2001. Supervisors: dr. N.J.E. Wijngaards and prof.dr. F.M.T. Brazier Referee: dr. B.J. Overeinder.

[6] S. Duri, J. Elliott, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J.-M. Tang. Data protection and data sharing in telematics. *Mob. Netw. Appl.*, 9(6):693–701, 2004.

[7] J. Himoff, P. Skobelev, and M. Wooldridge. Magenta technology: multi-agent systems for industrial logistics. In *AAMAS '05: Proceedings of*

the fourth international joint conference on Autonomous agents and multiagent systems, pages 60–66, New York, NY, USA, 2005. ACM Press.

[8] C.-H. Jo, G. Chen, and J. Choi. A new approach to the bdi agent-based modeling. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 1541–1545, New York, NY, USA, 2004. ACM Press.

[9] D. Ouelhadj, S. Petrovic, P. I. Cowling, and A. Meisels. Inter-agent cooperation and communication for agent-based robust dynamic scheduling in steel production. *Advanced Engineering Informatics*, 18(3):161–172, 2004.

[10] S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *Knowl. Eng. Rev.*, 19(1):1–25, 2004.

[11] L. R. Rheinheimer, J. M. Martins, and S. C. C. S. Pinto. Wsagent: an agent based on web services to promote interoperability between heterogeneous systems in the health domain. In *OOPSLA '04: Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, pages 214–215, New York, NY, USA, 2004. ACM Press.

[12] V. Robu, D. J. A. Somefun, and J. A. L. Poutré. Modeling complex multi-issue negotiations using utility graphs. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 280–287, New York, NY, USA, 2005. ACM Press.

[13] S. Talman, M. Hadad, Y. Gal, and S. Kraus. Adapting to agents' personalities in negotiation. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 383–389, New York, NY, USA, 2005. ACM Press.

[14] M. Woolridge and M. J. Wooldridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2001.