A MAS-based Infrastructure for Negotiation and its Application to a Water-Right Market

Bexy Alfonso, Vicente Botti, Antonio Garrido, and Adriana Giret

Departamento de Sistemas Informáticos y Computación Universitat Politècnica de València {balfonso,vbotti,agarridot,agiret}@dsic.upv.es

Abstract

This paper presents a MAS-based infrastructure for the specification of a negotiation framework that handles multiple negotiation protocols in a coherent and flexible way. Although it may be used to implement one single type of agreement mechanism, it has been designed in such a way that multiple mechanisms may be available at any given time, to be activated and tailored on demand (on-line) by participating agents. The framework is also generic enough so that new protocols may be easily added. This infrastructure has been successfully used in a case study to implement a simulation tool as a component of a larger framework based on an electronic market of water rights.

Keywords: negotiation, multi-agent systems, water-right market

1 Introduction

Last decades have witnessed an increasing interest in the design and application of computational infrastructures and tools, based on intelligent agents, to virtual architectures and organizations that give support to multiple ways of negotiation. Negotiation usually involves a dynamic collection of semi-independent autonomous entities (representing heterogenous software agents or humans, departments, enterprises, information resources and other organizations) each of which has a range of problem solving capabilities and resources at their disposal. These entities exhibit complex behaviors; they usually co-exist, collaborate and agree on some computational activity, but sometimes they compete with one another in a ubiquitous virtual scenario that is a sort of 'looking-glass reflection' of the real world.

Automated negotiation is essential to undertake complex behavior and architectures, including conflict identification, its management and resolution, search for consensus, assessment of agreement stability and equilibrium analysis in situations where two or more parties have opposing preferences [15]. This line of research has addressed developments for group decision support systems and meeting support systems, which can be extrapolated to automated negotiation [8, 10]. Therefore, negotiation is interesting from an application point of view but also to provide artifacts that facilitate the design, experimentation and simulation of involving agreements. In this paper we intend to profit from that experience and look at one of such artifacts: a generic negotiation MAS-based framework in which different negotiation protocols may become available. The contributions of this general framework are multiple. i) As it is defined for the Magentix2 [2] platform for open MASs, it embodies easy communication and interaction protocols among agents, roles and organizations. It also uses Jason [5] as a high-level language for programming agents, providing them with high reasoning skills. ii) Interactions among agents aim at achieving individual and global goals, and are structured via collaboration, argumentation, negotiation and, eventually, via agreements and contracts [19]. iii) It is composed of flexible negotiation mechanisms and their supporting preparatory and ending activities. iv) As a by-product, it creates standardized negotiation modules to be grafted into larger scenarios or as plug-ins in peer to peer interactions. v) It has been used as a proof of concept in mWater[6, 13], a water-right market where negotiation is essential, also embedded in a decision support system where water usage is subject to conflicts whose solution may involve different types of negotiation. vi) It provides new areas of opportunities for an agreement computing solution [19], including agility, heterogeneity, reconfigurability, cooperation, argumentation, reputation and trust issues under a MAS perspective.

2 Technological Background

There are various technologies involved in the implementation of our MAS infrastructure. First, the MAS platform in itself, which manages agents and their interactions, allowing the information exchange among them and also with the environment. Second, a language to define the agents behavior —in this case Jason, which follows the agents' BDI model. Third, to support the human-software agents' interactions it is necessary to design a Graphical User Interface (GUI) and an artifact to orchestrate the communication between the GUI and the MAS.

2.1 MAS Platform

We use Magentix2 [2] as our MAS platform because: i) it provides powerful techniques to facilitate agents' communication; ii) it supports interactions protocols between agents organizations/societies through conversations management; iii) it allows the use of high-level reasoning structures when programming the agents; and iv) it includes security issues for distributed systems, so it offers a dynamic and flexible model for complex systems. In short, Magentix2 gives us support at the three levels stated in [16]: organization level, interactions level and agent level.

Conversations Factory: an Artifact for Communication

A Conversations Factory [11] is mainly a Magentix2 mechanism to support FIPA interaction protocols [12]. Each conversations factory allows us to keep a complete interaction among two or more agents having an *initiator* (the one who starts the

conversation) and one or more *participants* (the other agents in the conversation). The two main structures supporting conversations are *CProcessor* and *CFactory*. The former manages the sent/received messages in each step of the conversation, performing the corresponding actions, and determines the next step in the conversation. The latter creates the conversations and the *CProcessors* that correspond to a specific protocol. If the agent is playing the role of *initiator*, the conversation can start without needing an external event. On the other hand, if the agent is a *participant* an event is required for it to be part of the conversation.

2.2 Programming language

Magentix2 allows us to use a high-level language for programming agents, in this case is Jason [5], which is an extension of the AgentSpeak language. AgentSpeak allows us to define agents in terms of beliefs, goals and plans. Beliefs represent the vision of each agent of the current state of the world in which such an agent is situated. Beliefs change frequently due to a 'perception' of the agent over its environment, because some information has been sent to it through a message, or because it explicitly modifies those beliefs as a consequence of some previous reasoning. Agents' goals represent the agents' intention to reach a state where they believe the goals are true, what is called 'achievement goal'. Another kind of goal is satisfied when the agents retrieve updated information from their belief base 'test goals'. Finally, plans are just a sequence of steps that allow agents to reach some goals. The fact of adding a goal acts as a triggering event for executing the corresponding planned sequence of actions. There are other actions that act as triggering events for plan executions as it is the case of the deletion of achievement goals, adding and deletion of beliefs, and adding or deletion of test goals. If this sequence of actions does not fail, the goal is successfully reached.

Jason provides a kind of action called 'internal action'. It is a structure that allows us the execution of legacy code (Java in this case). Thanks to this, the agent has access to the structures provided by the platform [3] in order to make use of the conversations factory in a more simplified way. By using some of the Magentix2 predefined internal actions, each agent can customize what it does in those steps of the conversations on which it needs to perform some 'reasoning', delegating details such as synchronization, timeouts, errors management, etc. in the platform. Magentix2 is also responsible for updating the state of each agent (by updating its beliefs) for it to make decisions, which behaves as an indirect communication.

3 Our Generic Negotiation Model

The infrastructure for a generic negotiation model can be seen as a set of entities and roles regulated by mechanisms of social order, and created in order to negotiate with some good, service or resource.



Figure 1: Generic Negotiation workflow structure. Roles: g - guest; p - participant; b - black; w - white; m - mediator; ntm - negotiation table manager; la - legal authority.

3.1 Main Structure

Our negotiation model follows a MAS specification based on conversations, and regulation on (structural) norms. It is defined as a generic organization for negotiation (see Figure 1)¹, where any participating agent may become involved in a negotiation process.

After admission is granted, each negotiation involves first a preliminary process of invitation and filtering of parties, then the negotiation process itself and, finally, some form of settlement process through which the agreements among participants are made explicit and, if appropriate, communicated to the organization.

3.2 Users and Roles

There are seven roles that interact in the model, as depicted in Figure 1. A guest role (g) is a user that wants to enter the negotiation. The guest may be specialized into a real participant (p), and furthermore as black (b) and white (w) to differentiate the parties that are acting in a given negotiation. Finally, there are four types of staff roles. The mediator role (m) represents a negotiation facilitator agent who runs standard activities, such as managing the specific parameters of the negotiation protocols. The negotiation table manager role (ntm) represents an agent who executes activities that are specific of a given negotiation protocol, for example accept valid negotiators, tune negotiation parameters of the table, mediate in the negotiation or conflict resolution process, expel negotiators, etc. The legal

 $^{^{1}}$ At a glance, each interaction/conversation represents an atomic process and/or dialog among agents; a workflow represents complex interaction models and procedural prescriptions. The dynamic execution is modeled through arcs and transitions, by which the different participating roles of the organization may navigate.

authority role (la) represents an agent who is in charge of activities for agreement enactments that are executed as a result of a successful negotiation process.

Note that, unlike other approaches, our definition introduces an explicit intelligent management into the negotiation model in the form of the mediator and negotiation table manager. These two roles have demonstrated to be very helpful to improve and facilitate the internal behavior of the organization. On the one hand, the mediator must be aware of the organizational conventions, the rules of the market and the negotiation structure. On the other hand, the negotiation table manager must obey the particular rules of the protocol to be used within the negotiation, and this is usually domain-dependant —different protocols require the application of different sequences of steps.

3.3 Workflow

The workflow activities in the generic negotiation model of Figure 1 are specified through a main structure which includes two other workflows: the *NegotiationHall* and *NegotiationTables*, plus two supporting interactions, *Admission* and *AgreementEnactment*.

Admission. It allows Guest agents to register to become a Party, and to 'jump start' a negotiation process. Once negotiation is open, this interaction allows Party agents to enter and negotiate by registering individual data for management and enforcement purposes (these data are domain-dependent and can be used, for example, for enforcing particular conventions and managing activities).

NegotiationHall. Actual negotiation starts here (see Figure 2), where Party agents become aware of any activity and/or initiate concurrent activities for negotiation. There are three interactions that provide virtual scenarios for the: i) creation of, and invitation to, negotiation tables (NTC); ii) exchange of information about active agreements and ongoing negotiation tables (IE); and finally, iii) execution of specific activities in case of an anomalous/critical situation (CS).

Negotiation Tables are created in two ways: i) by the organization itself, for example periodic negotiation tables about a set of issues, or ii) initiated on-demand by a participating agent. The negotiation tables are created in the NTC interaction, which responds to the *FIPA request* standard protocol [12]. Figure 3 and 4 show the steps of the protocol from the Party's perspective (*initiator*) and from the Mediator's perspective (*participant*), respectively. It issues the following illocution:

 $request(p_x, m, open, protocol(params), \delta, pt, at, C)$, where the semantic is as follows. Party agent p_x requests (see Figure 3) to the Mediator, m, to open a negotiation table with a given negotiation protocol. This protocol is instantiated with the set of values for the parameters params. The table is created to negotiate about a deal δ . The requesting party, p_x , will participate as pt that can take one of these values: p, that is an observer Party; a Black party b; or a White party w. at is the access type that can be *Public*, any body can be invited; or *Private*, only Party agents that fulfill the set of constraints C can participate in the negotiation table.

When the Mediator, m, receives a request to open a negotiation table (see Figure 4), it instantiates a new Negotiation Table scenario with the requested negoti-



Figure 2: Negotiation Hall workflow structure.



Figure 3: Party's behavior for requesting a New Negotiation Table.

ation protocol, for example a standard double auction, a face-to-face negotiation, a blind double auction, etc., and the given parameters. Moreover, a Negotiation Table Manager, ntm, is created to manage the execution of the negotiation table. Next, m issues an information illocution to the p_x agent who requested the table.

 $inform(m, p_x, table_{ID}, error)$, where $table_{ID}$ is the ID of the new table if it was successfully created, or a *null* value when the table can not be created due to *error* conditions.

In order to complete the negotiation table creation the Mediator needs to invite other Party agents to the new negotiation table. When the created negotiation table has a *Public* type of access, the m broadcasts an invitation message to all the participants:

inviteAll $(m, table_{ID}, protocol, \delta, C)$; in other words, the invitation message states the table_{ID} of the negotiation table that is receiving players; the negotiation pro-



Figure 4: Mediator's behavior during the conversation for Opening a New Table.

tocol protocol used in that table; the set of issues, δ , that is being negotiated; and the set of constraints, C, to participate in are also made public.

On the other hand, if the created negotiation table has a *Private* type of access, the *m* has to select first the set of possible candidates to invite, say $P_{table_{ID}}$, and then send an invitation message to every such candidate:

 $invite(m, p_y, table_{ID}, protocol, \delta, C)$, where each candidate $p_y \in P_{table_{ID}}$.

Negotiation Table. It is organized in a flexible and scalable fashion in order to easily include new negotiation protocols. Each instance of a *Negotiation Table* interaction is managed by a *Negotiation Table Manager*, *ntm*, who knows the structure, specific data and management protocol of the given negotiation protocol. The framework provides pre-defined protocols such as face-to-face, Dutch auction, English auction, standard double auction, closed bid envelope, blind double auction with mediator, among others. Nevertheless, new negotiation protocols may be easily added provided that the new definition complies with the generic structure.

Every generic negotiation table is defined as a three interaction structure (see Figure 5). The first interaction is *Registration*, in which the *ntm* applies a filtering process to assure that only valid agents can enter a given negotiation table (recall situations when a private negotiation table is executing or only a sub-group of Party agents that fulfill a set of constraints may participate in the table). The specific filtering process will depend on the given negotiation protocol and possibly on domain specific features. The second interaction is the negotiation protocol, in which the set of steps of the given protocol are specified (see bellow for a sample negotiation protocol specification). Finally, in the last interaction, *Validation*, a set of closing activities are executed, for example registering the final deals, stating the following steps for the agreement settlement, verifying that the leaving party satisfies the leaving norms of the negotiation table, etc. The set of activities to be executed in this interaction is domain specific and will also depend on the given negotiation protocol.

AgreementEnactment. Once an agreement has been successfully reached, it is settled here according to the given conventions. This may be a rather elaborate



Figure 5: Negotiation Table workflow structure.

process. First of all, the Mediator checks whether or not the agreement satisfies some formal conditions. If the agreement complies with these, a transfer contract is agreed upon and signed by the Party agents involved, and then the agreement becomes active. Once an agreement is active it may be executed and, consequently, other Party agents may initiate a grievance procedure that may overturn or modify the agreement. Even if there are no grievances that modify a contract, parties may not fulfill the contract properly and there might be some contract reparation actions. If things proceed smoothly, the agreement subsists until maturity.

4 Case Study: *mWater*, a Water-Right Market

4.1 *mWater* Overall Description

Water scarcity is a major concern in most countries. It has been sufficiently argued that more efficient uses of water may be achieved within an institutional framework where water rights may be negotiated under different market conditions [20]. In hydrological terms, a water market can be defined as an institutional, decentralized framework where users with water rights are allowed to voluntarily trade them, always fulfilling some pre-established norms, to other users in exchange of some compensation [14, 20]. Because of water's unique characteristics, such markets do not work everywhere, they are not homogenous, nor do they solve all water-related issues [20]. Also, even subtle changes in the market design (allowed participants, legislation, protocols, etc.) are very costly and difficult to evaluate.

mWater is a particular instance of the MAS infrastructure for negotiation pre-

sented above, and it is used as a simulation tool for What-If Analysis of water-right markets policies [6, 13]. More specifically, mWater assists in designing appropriate water laws and regulate, either privately or publicly, the users' actions, interactions and their eventual trade.

4.2 mWater as a Simulation Tool

mWater builds on a MAS infrastructure, simulates a flexible water-right market, and includes its own ontology for dealing with water issues and both the trading and grievance processes. We have focused our model on humans' actions: agents are the crucial component in these models and our interest relies on the social aspect of the market, which is usually missing in other markets in the literature. This simulator includes heterogeneous and autonomous intelligent agents representing the different independent entities in the market. We focus on demands and, in particular, on the type of regulatory (in terms of norms selection and agents behavior), and market mechanisms that foster an efficient use of water while also trying to prevent conflicts among parties. In this scenario, this system plays a vital role as it allows us to define different norms, agents behavior and roles, and assess their impact without jeopardizing the real-world market, thus enhancing the quality and applicability of its results as a decision support tool.

The user can configure simulation parameters such as: the group of water-users that will participate in the market², the norms and regulations that define the policies in the market, the seasons in which the water-right transfer will take place, etc. The simulation tool executes with a given configuration and the user can assess the market's behavior by means of indicators such as: number of water-right transfer agreements, volume of water transferred, amount of money, overall social satisfaction of the water-users that participated in the market, number of conflicts generated, etc.

4.3 mWater in Action

Figure 6 shows a snapshot of the mWater simulator in action. This interface allows the user, i.e. the water policy maker, to choose different input values that involve simulation dates, participants, norms (in the form of protocols used during the trading negotiation) and some decision points that can affect the behavior of the participants³.

To implement human-agents interactions, in order to have a tool for studying different behaviors and situations, it was necessary to create some GUIs with the required options for the human to make changes in the system and pass information to the rest of agents at execution time. For this we implemented a Web page with PHP as scripting language and an interface application to 'pass' all the requests

 $^{^{2}}$ It is important to point out that the simulation we have developed is a mixed-initiative simulation in which there are software agents that are completely autonomous/automated and other software agents that are simple interfaces for human users. In this way, it is very easy to include complex social behaviors that are hard to implement or highly time consuming.

³In our current implementation, these additional decision points rely on a random basis, but we want to extend them to include other issues such as short-term planning, trust, argumentation and ethical values.



Figure 6: Snapshot of the mWater simulator.

from the Web page to the MAS, and all the results from the MAS to the Web page. This makes possible to count on a MAS composed by a mixture of automated agents and humans, and even a system completely based on automated agents. Figure 7 shows how a user can participate in a Japanese Auction of a water right, by interacting with other human or automated agents.

This simulation tool allows users to analyze: i) how the conventions, norms and negotiation protocols of the market change over time; ii) how participants in these markets (re)act to these changes; and ii) how to extrapolate the empirical outcomes of the market, in terms of economic and environmental impact, to deal with the social (welfare) aspect of the market. Our preliminary experiments shed light on the benefits that a collaborative AI perspective may bring to the policy makers, general public and public administrators. Also, from the experts' evaluation we can conclude that a tool like this provides an advantageous tool to help build a more efficient water market with more dynamic norms.

5 Further Uses for the Generic Negotiation Model

The infrastructure for generic negotiation that we have presented here has several application uses, from both the academia and industry point of view. From the academia standpoint, it can be used as a testbed for other developments within the agreement technologies paradigm (http://www.agreement-technologies.org). In particular, there are several challenging questions on:

- Organization and roles. How beneficial is the inclusion of collective, heterogeneous roles, their collaboration (and trust theories) and how the policies for either flat or hierarchical group formation affect the system behavior? To answer this we need to capture all those roles currently recognized by legislation that have any impact on negotiation and agreement management, specially in grievances and conflict resolution.
- Collective decision-making, reconfigurability, cooperation, social issues and

MWATER X	2	ন ব
GTTI IA Grupo de Tecnología Info Inteligencia Artificial	ormática UNIVERSITAT POLITECNICA DE VALÈNCIA	DSIC
MWater User: VBotti	 Requesting table state Data received Table state 	
Water market: 1014	Table ID: Protocol: Role: Traded Water Rigth:	1 japanese_auction buyer 1117
	Current Bid	Remaining participants "EMDura","VB
	Time to Accept With bid!	thdraw
Trading Hall Table description		

Figure 7: Snapshot of the human-agents GUI. The user can participate in a Japanese auction with other humans and/or automated agents.

coordination. What is the impact of argumentation, judgement aggregation, reputation, prestige and multi-party negotiation in the system performance? The answer to this question is not straightforward and requires simulation tools for performance assessment, as seen in section 4.

• Institutional limitations. What type of enforcement mechanisms are necessary and how they change *w.r.t* the evolution of regulation? This is highly related to the definition, adoption and compliance of (emerging) norms and, more particularly, how to model and reason on them? To solve this, we need to face the problem of expressiveness: the type of norms we have dealt with so far has a formal representation, but other types of representation may be more complex to handle. Finally, ensuring norm compliance is not always possible (or desired), so norm violation and later detection via grievances usually makes the environment more open, dynamic and realistic for taking decisions.

From the industry standpoint, there exist further applications in the form of simple tools that can be embedded within our MAS framework:

• A decision-support tool for policy simulation. Policy-making is a hard task. Designing and taking legal decisions involves a complex balance among different factors, such as economic, social, administrative or environmental aspects. Consequently, a decision-support tool that allows policy-makers to easily predict, analyze and measure the suitability and accuracy of modified regulations for the overall system, before using other operational tools for the real floor, shows very important. Our experiments with mWater shed light on the benefits that a collaborative AI perspective for a water-right market may bring to the policy-makers, general public and public administrators.

- A GUI tool for human negotiation that facilitates the human interaction with software agents. Particularly, our GUI provides a simple, though effective way to set up parameters and dynamic changes, which affect the performance of the system, during the negotiation process (and also while simulating this process). Moreover, it intuitively provides the results generated after such an interaction process, which can be used as an analysis tool to evaluate protocols.
- A general tool open to other negotiation processes, such as other electronic markets; the workflow structure, roles and negotiation interaction remains the same. Our experiences show that our negotiation framework is general enough and can be valid for other markets. Particularly, we are applying these ideas to a by-product exchange market to boost the re-use of waste, thus being part of our current work.

6 Conclusions through Related Work

Computing has become an inherently social activity rather than a solitary one, leading to new forms of conceiving computational systems which require both interaction and negotiation. Some proposals have been effectively developed in literature to implement a negotiation framework. That is the case of the Jade platform [1, 4], which is a FIPA compliant platform that provides Java classes to handle all the FIPA interaction protocols. In this sense, the agents' interactions must be also programmed in Java by using the constructions provided by the platform. Another multi-agent platform with support for interaction protocols is Jadex [7, 17]. Jadex follows a typical BDI model and can be executed alone or under other communication platforms using *adapters*. A Jadex agent is defined through an XML file and the Java classes that implement it. Jadex also owns the 'interaction protocols' capability, offering built-in support for most of the FIPA interaction protocols. However, both Jade and Jadex use Java classes for implementing FIPA interaction protocols, so the programmer can not use other specialized programming languages, such as AgentSpeak, more expressive to model and describe agents. This does not prevent us from addressing the problem using the Java approach; in fact, so far it has been broadly used. However, in MASs, it is desirable to use tools and languages that better fit with the autonomous and proactive agents' nature. In this sense, Magentix2 [2] supports a high-level language for programming conversational agents (i.e. agents whose interactions respond to interaction protocols) and the rest of the capabilities offered by similar platforms. It also owns a conversations manager that stores and automatically adds the information required in the creation of messages during the conversation. Moreover, with Magentix2 it is possible to dynamically modify the sequence of steps in the interaction protocol in order to create more open and flexible conversations (new states and transitions between the conversation steps can be created at execution time). These features have been partially included in other platforms, whereas all of them are included in Magentix2, which makes it become an ideal infrastructure for a negotiation architecture.

From our point of view the common denominator in most of the current real, social systems is, interestingly, a negotiation process. Although some works have proposed the construction of formal conceptual models with some negotiation [9, 18], they do not always report significant advances from a collaborative AI perspective. In this paper we have established the infrastructure foundations for the specification of a multi-agent-based negotiation framework as the basis for modeling virtual scenarios, and put it into practice within a water-right market, where negotiation plays a vital role. The work presented in this paper is based on the lessons learned in [6, 13]. But now, the generic negotiation framework has been implemented in Magentix2 to offer a flexible and easy way to adapt to applications in which autonomous features in regulated environments are required. Thus, the technical contributions of this work are:

- Design a generic MAS infrastructure that captures the main steps that happen in an agent-based scenario, including mechanisms for exchanging information, negotiating and dealing with the critical situations that may appear thereafter.
- Introduce the users and intelligent roles that are necessary within an agentbased setting. Differently to existing approaches, we introduce the roles of intelligent mediators, which are very valuable for the process.
- Provide multiple negotiation strategies that are managed in a three-step unified way: registering, negotiating and validating the reached agreement. This also allows us to include different protocols in a flexible fashion.
- In order to test the applicability of this generic framework, we have put these ideas into practice with *mWater*. This water market is very illustrative and has allowed us to explore the influence that the repetitive interaction of participants exerts on the evolution of the market. Also, it has given us enough evidence that the generic framework for negotiation provides a solid foundation for complex markets.

Acknowledgments

This paper was partially funded by the Consolider AT project CSD2007-0022 IN-GENIO 2010 of the Spanish Ministry of Science and Innovation; the MICINN projects TIN2011-27652-C03-01 and TIN2009-13839-C03-01; and the Valencian Prometeo project 2008/051.

References

- [1] Jade. http://jade.tilab.com.
- [2] J. M. Alberola, J. M. Such, A. Espinosa, V. Botti, and A. García-Fornes. Magentix: a Multiagent Platform Integrated in Linux. In *EUMAS*, pages 1–10, 2008.

- [3] B. Alfonso, E. Vivancos, V. Botti, and A. García-Fornes. Integrating jason in a multi-agent platform with support for interaction protocols. In *Proceedings of the* compilation of the co-located workshops on AGERE!'11, SPLASH '11 Workshop, pages 221–226, New York, NY, USA, 2011. ACM.
- [4] F. Bellifemine, G. Caire, and D. Greenwood. Developing Multi-Agent Systems with JADE. John Wiley and Sons, 2007.
- [5] R. H. Bordini, J. F. Hübner, and M. Wooldridge. Programming Multi-agent Systems in Agent Speak Usign Jason. John Wiley & Sons, 2007.
- [6] V. Botti, A. Garrido, J. A. Gimeno, A. Giret, and P. Noriega. The role of MAS as a decision support tool in a water-rights market. In AAMAS 2011 Workshops, LNAI 7068, pages 35–49. Springer, 2011.
- [7] L. Braubach, A. Pokahr, and W. Lamersdorf. Jadex: a BDI agent system combining middleware and reasoning. In M. C. M. K. R. Unland, editor, *Software Agent-Based Applications, Platforms and Development Kits*, pages 143–168. Birkhäuser-Verlag, 9 2005.
- [8] G. B. DeSanctis and B. Gallupe. A foundation for the study of group decision support systems. *Knowledge based systems*, 33(5):589–609, 1987.
- P. Eckersley. Virtual markets for virtual goods, 2003. Available at http://www.ipria.com/publications/wp/2003/IPRIAWP02.2003.pdf (accessed May 2012).
- [10] J. Fjermestad and S. Hiltz. Group support systems: a descriptive evaluation of case and field studies. *Journal of Management Information Systems*, 17(3):115–161, 2001.
- [11] R. L. Fogués, J. M. Alberola, J. M. Such, A. Espinosa, and A. García-Fornes. Towards Dynamic Agent Interaction Support in Open Multiagent Systems. In Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence, volume 220, pages 89–98. IOS Press, 2010.
- [12] Foundation for Intelligent Physical Agents. FIPA XC00025E: FIPA Interaction Protocol Library Specification.
- [13] A. Giret, A. Garrido, J. A. Gimeno, V. Botti, and P. Noriega. A MAS decision support tool for water-right markets. In *Proceedings of the Tenth International Conference on Autonomous Agents and Multiagent Systems (Demonstrations@AAMAS)*, pages 1305–1306, 2011.
- [14] J. Gomez-Limon and Y. Martinez. Multi-criteria modelling of irrigation water market at basin level: a Spanish case study. *European Journal of Operational Research*, 173:313–336, 2006.
- [15] G. Kersten and H. Lai. Satisfiability and completeness of protocols for electronic negotiations. *European Journal of Operational Research*, 180(2):922–937, 2007.
- [16] M. Luck and AgentLink. Agent technology: computing as interaction: a roadmap for agent-based computing. Compiled, written and edited by Michael Luck et al. AgentLink, Southampton, UK, 2005.
- [17] A. Pokahr, L. Braubach, A. Walczak, and W. Lamersdorf. *Developing Multi-Agent Systems with JADE*, chapter Jadex-Engineering Goal-Oriented Agents, pages 254–258. Wiley and Sons, 2007.
- [18] C. Ramos, M. Cordeiro, I. Praça, and Z. Vale. Intelligent agents for negotiation and game-based decision support in electricity markets. *Engineering intelligent systems* for electrical engineering and communications, 13(2):147–154, 2005.
- [19] C. Sierra, V. Botti, and S. Ossowski. Agreement computing. KI Künstliche Intelligenz, 25(1):57–61, 2011.
- [20] M. Thobani. Formal water markets: Why, when and how to introduce tradable water rights. The World Bank Research Observer, 12(2):161–179, 1997.