Intelligent and Collaborative Multi-Agent System to Generate Automated Negotiation for Sustainable Enterprise Interoperability

Manuella Kadar and Maria Muntean

Computer Science Department, "1 Decembrie 1918" University, Alba Iulia, Romania
{mkadar,mmuntean}@uab.ro

Abstract. The survival of traditional enterprises within the global economy relies on their ability to embrace new ideas and new organizational forms and to imagine new approaches to collaborating in dynamic networked environments. This paper proposes a system for promoting sustainable interoperability between enterprises involved in complex networked environments through multi-level negotiation, communication and information sharing. The proposed solution is based on a multi-agent system architecture that applies rule-based negotiation at various organizational levels such as: business, ICT, workflows, data systems and people. This architecture has been tested in the case of a collaborative networked environment in which several entities, namely funding authority, beneficiary of funds, contractors and sub-contractors have to be contractually engaged through multi-level negotiations.

Keywords: multi-agent systems, sustainable interoperability, rule-based negotiation.

1 Introduction

The recent global financial and economic crisis has demonstrated that new ways of doing business and new inspirations are required. The survival of traditional enterprises within the global economy relies on their ability to embrace new ideas and new organizational forms and to imagine new ways of delivering value to customers, new approaches to collaborating in a dynamic networked environment.

Organizations can only reach the full collaboration potential if partners develop enhanced capabilities to seamless communicate, coordinate, cooperate, collaborate, and most importantly, interoperate in spite of different organizational structures, technologies or processes [1]. A broad definition of interoperability is referring to the ability of two or more systems to exchange information and use it accurately. Consequently, the lack of interoperability disturbs the creation of new markets, networks, and diminish innovation and competitiveness of business groups [2]. Apart from being only a technical issue, interoperability challenges the enterprise at organizational and semantic level, underlying the need for patterns and solutions that support the seamless cooperation among ICT systems, information and knowledge, organizational structures and people [3].

G. Jezic et al. (eds.), *Agent and Multi-Agent Systems: Technologies and Applications*, Advances in Intelligent Systems and Computing 296,
DOI: 10.1007/978-3-319-07650-8_22, © Springer International Publishing Switzerland 2014

The lack of interoperability as identified in several industrial sectors and in complex collaborative environments has a major cost, blocking the achievement of the time-to-market demanded by today's competitive environment [4].

This paper proposes a knowledge based adaptive dynamic system, namely k-MAPE (knowledge based cycle of monitoring, analyzing, planning and executing). The final goal of such system is the achievement of sustainable interoperability between enterprises involved in complex networked environments through multi-level negotiation, communication and information sharing.

The reminder of the paper is structured as follows: Section 2 describes the background for sustainability of interoperable solutions. Section 3 discloses the Multi-Agent based Negotiation System within the collaborative networked environment, the multi-layered negotiation process and communication in case of the braking down of interoperability and the requirement of re-negotiation. Section 4 provides a case study of a rule based negotiation process with detailed explanations on: i) agentification of the negotiation process and offer request design workflow; ii) communication protocols and agent behaviors; iii) rule based negotiation of an online auction and iv) implementation of the rule-based offer request design by intelligent and collaborative agents. Full implementation in the Rule Engine for the Java Platform (JESS) and Java Agent Development Framework (JADE) is provided and discussed. In Section 5 follows discussions and conclusions by pointing to future works.

2 Background for Sustainability of Interoperable Solutions

Competitive markets are becoming increasingly complex and dynamic, with companies not surviving and prospering solely through their own individual efforts [5]. Each one's success depends on the activities and performance of others to whom they do business with, and hence on the nature and quality of the direct and indirect relations [6]. Such relationships involve a mix of cooperative and competitive elements, that to cope with them, organizations need to focus on their core competencies by improving their relationships with customers, streamlining their supply chains, and by collaborating with partners to create valued networks between buyers, vendors and suppliers [7]. The collaborative process may be described as sum of coordinated and synchronous activities characterized by reciprocal interactions at high frequency that normally require the transfer of information among several organizations, i.e. knowledge sharing [8]. An emergent research challenge in seamless interoperability is rising. It focus on the sustainability within collaborative business networks, addressed by a wide complexity of interactions and a high probability of changing requirements, in the view that enterprises are complex, and adaptive systems (CAS), with factors that are making interoperability difficult to sustain over time [9]. The research challenges in the area of sustainable interoperability are several and include the study of: enterprise interoperability itself; system monitoring, behavior and adaptability; the "system" aspects of interoperability, from software component design to organizational structure to the communication, collaboration and coordination facilities; decision support to minimize the impact of changing requirements and information models; interoperability of digital ecosystems as complex systems of systems.

This approach considers that one important means to produce sustainable interoperability is designing of adapted and intelligent software components that may support effective communication, coordination and collaboration at all levels within the enterprise and the networked environment, as well. Complex tasks such as the evaluation of the maturity level of an enterprise or the design of any offer request can be successfully achieved by intelligent agents embedded with elements of artificial intelligence, e.g. expert systems and data mining classifiers. Two main aspects concerning the lack of enterprise interoperability in the digital networked environment are hereby discussed, namely: i). weak decision support at business level and ii). behavior monitoring and collaboration interruptions within the various structures of the enterprise. As concerns the decision support we proposed an enterprise interoperability maturity level assessment system carried out by intelligent agents . For instance, it is possible to create agents with Expert Systems as presented in [10]. In this paper the discussion is focused on Artificial Intelligence (AI) and algorithms that mirror the supervised learning process, that is achieved primarily through Restricted Bolzmann Machines (RBM) [11], [12] employed as a latent factor analysis application [13]. Maturity level of enterpise interoperability can be modeled by a RBM that uses a layer of binary hidden units (latent feature detectors) to model the higher-order correlations between various features of enterprises. There are no direct interactions between the hidden units that represent the latent features and no direct interactions between the visible units that represent the selected enterpresises. A simple and efficient method called „contrastive divergence" is employed to learn a good set of feature detectors from a set of training enterprises [10]. An agent endowed with a RBM is able to assess the maturity level of any enterprise before involving into a negotiation process and to inform the decision body, e.g. the Chief Negotiator on the credentials of the potential enterprise willing to get into a certain negotiation process. However, such technique provides flexible ways of making inferences, it has not been widely exploited in Multi-Agent Systems to make Negotiation Process more flexible. Consequently, modelling and implementing a MAS that employs co-ordinately learning and decision making capabilities, becomes a great challenge to provide flexible mechanisms to cope with changing market conditions, and accordingly accommodate a negotiation process with multiple offers. Behavior monitoring at various structural levels can be successfully achieved by collaborative agents and also each phase of the negotiation process and building up of the offer requests are tasks to be successfully achieved by collaborative agents endowed with suitable expert systems. The main contribution we present in this paper is the design and implementation of a collaborative Multi-Agent System to improve the Automated Negotiation Process in a dynamic networked environment.

This approach do not promote the entire substitution of human decision, it rather propose the agent technology acting in the *backstage*, helping to communicate results among different software systems. Agent technology with the inclusion of algorithms emulates intelligence and provides flexible software systems. In this sense, rule-based systems are the main AI technique that has been added to software agents. This paper pioneers on how to incorporate supervised learning in software agents to deal with complex decisions in a negotiation environment. This paper also illustrates a realistic

MAS, in which two different AI techniques are employed coordinately to accomplish a negotiation process. The design and implementation of the Multi-Agent System serve to break boundaries within a negotiation environment represented by contracting authority, contractors and subcontractors. The proposed solution is presented in the following chapters.

3 Multi-agent Based Negotiation System

Agents can be regarded as computer systems situated in some environment, being capable of autonomous action in this environment, in order to meet their design objectives [11], i.e., an agent federation that is built on some of the main attributes of software agents, namely: autonomy, social ability, reactivity and pro-activeness. Agent federations share the common characteristic of a group of agents which have ceded some amount of autonomy to a single delegate which represents the group [12]. The delegate is a distinguished agent member of the group (Chief Negotiator). Simple group members interact with an intermediary agent (BN1, CN2, SC1, etc.), which acts as an intermediary between the group and the Chief Negotiator, who further interact with the outside world, as shown in Figure 1.

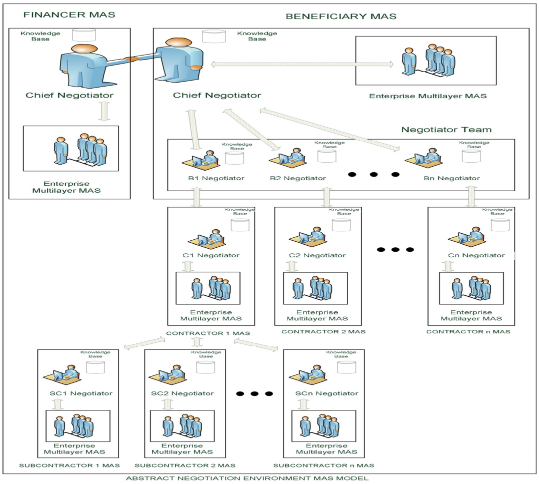

Fig. 1. MAS for abstract negotiation environment

Typically, the intermediary agent, namely, B1 Negotiator, C1 Negotiator, SC1 Negotiator, etc. accepts skill and requirements descriptions from the local agents of the enterprise multilayered MAS. The information is transmitted to the Negotiator Team who uses such information to match with requests from intermediaries representing other groups. In this way the group is provided with a single, consistent interface. The intermediary agent receives messages from its group members. These may include skill descriptions, task requirements, status information and application-level data. Such information will be communicated using some general, declarative communication language which the Negotiation Team of agents and further the Chief Negotiator understand. Outside of the group, the intermediary agent sends and receives information with the intermediaries of other groups. This could include task requests, capability notifications and application-level data routed as part of a previously created commitment. Implicit in this arrangement is that, while the intermediary must be able to interact with both its local federation members (enterprise multi-layered MAS) and with other intermediaries, individual normal agents do not require a common language as they never directly interact. This makes this arrangement particularly useful for integrating legacy or an otherwise heterogeneous group of agents.

However, a fundamental mechanism for managing inter-agent dependencies at run-time is negotiation, that is the process by which a group of agents come to a mutually acceptable agreement on some matter. Negotiation underpins attempts to cooperate and coordinate (both between artificial and human agents) and is required both when the agents are self interested and when they are cooperative. The need for a multi-agent negotiation approach is illustrated in Figure 1 in which negotiator agents represent the interests of different organizational entities (e.g., financer, beneficiary, contractors, sub-contractors).

4 Implementation of the Rule-Based Automate Negotiation with MAS

4.1 Agentification of the Negotiation Process

The environment is a contracting authority that has to accomplish a complex task, the restoration of an ancient Fortress. Besides several works, there is a road construction for which he has to negotiate with several contractors and subcontractors. The type of negotiation is an online auction. The contracting authority set forth the technical and financial characteristic of the offer request and also the potential contractor's profile requirements. The maximum admitted total price of the road construction and the technical features such as: time of achievement, length of the road, raw materials, equipment employed, archaeological rescue plans, work warranty are designed by the Beneficiary's departments. The potential contractor's profile requirements are set up by the Beneficiary's Accountant and Juridical Departments and refers to the turnover, profit, number of employees, previous similar works achieved, flexibility and capacity of working in special settings, IT infrastructure, and equipment. The offer request is launched by the Beneficiary's Chief Negotiator to one of his Negotiators responsible for negotiating Construction works, namely the B1 Negotiator Agent. B1Negotiator Agent further launches the offer request to Constructors. The Constructor's Negotiator Agent is responsible for providing data to it's own agents in order to elaborate the offer request and to further send the offer to B1Negotiator Agent. The Designer Agent as well as the Raw Material Agent as part of the Constructor's MAS face the complex problem of elaborating the best suitable offer according to the requirements. To handle such complexity a rule based expert system has been attached to the MAS.

The design of the negotiation process as a workflow is presented in Figure 2.

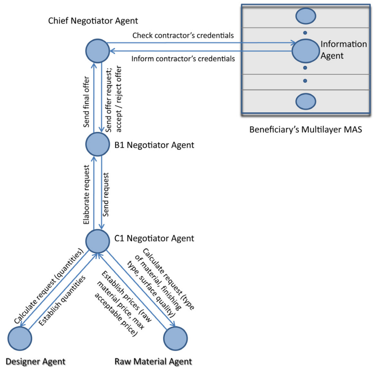

Fig. 2. Design of the negotiation process as a workflow between agents

4.2 Communication Protocols and Agents Behaviors

The MAS is globally coordinated by means of a communication protocol. In Figure 3 we present a simplified version of the proposed model. The left side shows the protocol between the Chief Negotiator and the B1Negotiator agents. On the right side, a similar model is used between the C1Negotiator and the Designer and Raw Material agents. According to the AUML notation , the solid arrows at the end of each message represent synchronous exchanges. The communication between the Chief Negotiator and the B1Negotiator is not initiated until the C1Negotiator agent sends the offer request. The entire communication process ends as soon as the Chief Negotiator updates the value of the offer in the EIS. When this occurs, the negotiation process is ended. The actual message content that is uttered by the ChiefNegotiator, B1Negotiator, C1Negotiator, Designer, RawMaterial agents comes from the emulation of intelligent behaviour.

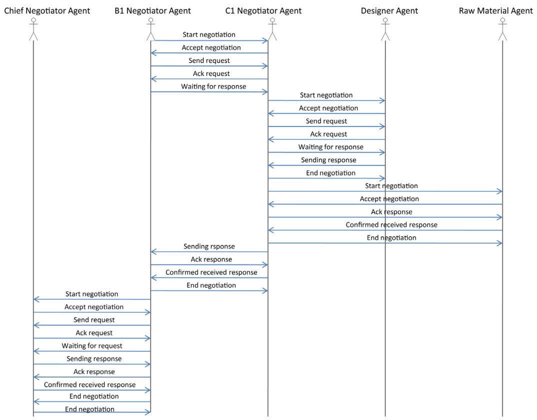

Fig. 3. Communication protocol

JADE uses the Behavior abstraction to model the tasks that an agent is able to perform and the agents instantiate their behaviours according to the needs and capabilities.

In order to develop an agent, the Agent class should be extended and the agent-specific tasks should be implemented through one or more Behaviour classes. Finally, these classes will be instantiated and added them to the agent. The Agent class, which is a common superclass, allows to inherit a basic hidden behaviour (that deals with all agent platform tasks, such as registration, configuration, remote management, and so on), and a basic set of methods that can be called to implement the application tasks of the agent (e.g. send/receive messages, use interaction protocols, query a knowledge base). The behaviors are described in Table 1.

Table 1. Behaviours

Crt. no.	Behaviour
1.	StartNegotiation = "100 StartNegotiation|"
2.	AcceptNegotiation = "101 AcceptNegotiation|"
3.	DenyNegotiation = "102 DenyNegotiation|"
4.	SendRequest = "200 SendRequest|"
5.	AckRequest = "201 AckRequest|"
6.	WaitingForResponse = "203 WaitingForResponse|"
7.	SendingResponse = "300 SendingResponse|"
8.	AckResponse = "301 AckResponse|"
9.	ConfirmedReceivedResponse = "302 ConfirmedReceivedResponse|"
10.	EndNegotiation = "500 EndNegotiation|"
11.	Error = "9999 Error|"
12.	ConversationIdB1NegotiatorC1Negotiator = "MessageB1NegotiatorC1Negotiator"
13.	ConversationIdC1NegotiatorDesigner = "MesssageC1NegotiatorDesigner"
14.	ConversationIdC1NegotiatorRawMaterial = "MesssageC1NegotiatorRawMaterial"
15.	ConversationIdB1NegotiatorChiefNegotiator = "MesssageB1NegotiatorChiefNegotiator"
16.	JaRBM = "jarbm"

4.3 Rule Based Negotiation for an Online Auction

The experiment describes an online auction for a contract of road construction and uses FIPA (Foundation for Intelligent Physical Agents) standards organization. Jess (the Rule Engine for the Java TM Platform) [14] is a rule-based system shell entirely written in Java. For representing the “accept” or “reject” proposals JESS templates have been used. The considered facts were the negotiation price, the construction time (in months) and the surface of the construction (in square meters).

The next rule fires if the negotiation price between the Contractor and one of its Sub-contractors is greater than 100000, and if the construction time is greater than five months and if the construction surface is less or equal to 500 square meters.

```
(defrule negotiation-protocol-reject
    "This is the negotiation protocol between contractor and subcontractor1, the
case: reject proposal"
    (negotiation {price > 100000})
    (negotiation {construction_time_months > 5})
    (negotiation {surface_square_meters <= 500})
    =>
    (printout t "Reject proposal" crlf))
```

If the price is less than 100000 and the construction time is less or equal to 5 months and the construction surface is greater than 500 square meters, than the "accept proposal" rule will be fired.

```
(defrule negotiation-protocol-accept
    "This is the negotiation protocol between contractor and subcontractor1, the
case: accept proposal"
    (negotiation {price < 100000})
    (negotiation {construction_time_months <= 5})
    (negotiation {surface_square_meters > 500})
    =>
    (printout t "Accept proposal" crlf))
```

4.4 Experimental Results and Discussions

The negotiation environment was tested with agents that were designed with specific rule templates, where rules assert information in their private fact base. The agents respond to this information by sending messages according to the Send - Receive Protocol defined in the system. Once the system agent receives the confirmation of starting a negotiation, an agent will negotiate with his counterpart and automatically make decisions based on his negotiation parameters. The system also provides a functionality of viewing the communications between agents and the systems via a sniffer agent. Various types of rules and queries were used for representing valid, posted and active proposals, depending on the phase of the negotiation process.

For instance, in the C1Negotiator-Designer negotiation, the Designer agent has to evaluate if the offer proposed by C1Negotiator agent exists in his fact base. In order to do this, he runs the following query:

```
(defquery search-by-name
    "Finds road design"
     (declare (variables ?name))
     (design
(name ?name)
(type_of_construction_material ?type_of_construction_material)
      (quantity ?quantity) )  )
```

In the C1Negotiator-RawMaterial negotiation, the RawMaterial agent fires two rules in order to get the total price of the offer:

```
(defrule getTotalPrice
    "Get the total price of the proposed offer"
    ?p <- (item)
    =>
     (add  item_quantity
((?p.name str-cat _item_quantity)
(?quantity1.quantity) (?quantity1.quantity * ?p.price) (?quantity1.maximum_price) ) )
     (printout t (str-cat ?p.name _item_quantity)
", " ?quantity1.quantity  ","
(* ?quantity1.quantity  ?p.price) "," ?quantity1.maximum_price crlf) )
```

and to check if the total price against the maximum price defined the our system:

```
(defrule checkTotalPrice
    "Check price against maxPrice"
    ?p <- (item_quantity
{total_price > maximum_price} )
    =>
     (retract ?p )
(printout t "Retract: " ?p.name ", "    ?quantity1.quantity  ","
      (* ?quantity1.quantity  ?p.price) crlf))
```

If the evaluated offer has a greater price than the maximum price considered, it will be retracted from the fact base and a message will be displayed.

The negotiation protocol was parameterized with specific rules taking into account the sustainable interoperability between agents. The negotiation process has been achieved according to the designed workflow, the entire communication between the agents involved is presented in the viewer. Negotiation time is assessed and calculated for the entire process in various negotiation conditions for further optimisations.

```
 msg = 101 AcceptNegotiation|, Time = 2014-03-10 11:03:05 duration: 0 seconds
 msg = 200 SendRequest|(request (operation "construction")(structure "paved_road")), Time = 2014-03-10 11:03:05 duration: 0
 msg = 201 AckRequest|, Time = 2014-03-10 11:03:05 duration: 0 seconds
 msg = 203 WaitingForResponse|, Time = 2014-03-10 11:03:05 duration: 0 seconds
 msg = 300 SendingResponse|, Time = 2014-03-10 11:03:05 duration: 0 seconds
 msg = 301 AckResponse|, Time = 2014-03-10 11:03:05 duration: 0 seconds
 msg = 302 ConfirmedReceivedResponse|, Time = 2014-03-10 11:03:05 duration: 0 seconds
 msg = 500 EndNegotiation|, Time = 2014-03-10 11:03:06 duration: 1 seconds

 msg = 101 AcceptNegotiation|, Time = 2014-03-10 11:03:05 duration: 0 seconds
 msg = 201 AckRequest|, Time = 2014-03-10 11:03:05 duration: 0 seconds
 msg = 300 SendingResponse|, Time = 2014-03-10 11:03:05 duration: 0 seconds
 msg = 302 ConfirmedReceivedResponse|, Time = 2014-03-10 11:03:05 duration: 0 seconds
l
erial, msg = 100 StartNegotiation|, Time = 2014-03-10 11:03:05 duration: 0 seconds
erial, msg = 101 AcceptNegotiation|, Time = 2014-03-10 11:03:06 duration: 1 seconds
erial, msg = 200 SendRequest|, Time = 2014-03-10 11:03:06 duration: 1 seconds
erial, msg = 201 AckRequest|, Time = 2014-03-10 11:03:07 duration: 2 seconds
erial, msg = 203 WaitingForResponse|, Time = 2014-03-10 11:03:07 duration: 2 seconds
erial, msg = 300 SendingResponse|, Time = 2014-03-10 11:03:08 duration: 3 seconds
erial, msg = 301 AckResponse|, Time = 2014-03-10 11:03:08 duration: 3 seconds
erial, msg = 302 ConfirmedReceivedResponse|, Time = 2014-03-10 11:03:09 duration: 4 seconds
terial, msg = 300 SendingResponse|, Time = 2014-03-10 11:03:09 duration: 4 seconds
erial, msg = 500 EndNegotiation|, Time = 2014-03-10 11:03:09 duration: 4 seconds

erial, msg = 101 AcceptNegotiation|, Time = 2014-03-10 11:03:06 duration: 0 seconds
erial, msg = 201 AckRequest|, Time = 2014-03-10 11:03:07 duration: 1 seconds
erial, msg = 300 SendingResponse|, Time = 2014-03-10 11:03:08 duration: 2 seconds
erial, msg = 302 ConfirmedReceivedResponse|, Time = 2014-03-10 11:03:09 duration: 3 seconds
gotiator
iefNegotiator, msg = 101 AcceptNegotiation|, Time = 2014-03-10 11:03:12 duration: 0 seconds
iefNegotiator, msg = 201 AckRequest|, Time = 2014-03-10 11:03:14 duration: 2 seconds
iefNegotiator, msg = 300 SendingResponse|Price OK 100000:  road_construction = 100000, Time = 2014-03-10 11:03:15 duration:
iefNegotiator, msg = 302 ConfirmedReceivedResponse|, Time = 2014-03-10 11:03:16 duration: 4 seconds
```

Fig. 4. A sample of results

5 Conclusions and Future Work

This paper proposes a novel multi-agent system that adopts a rule-based approach to implement automated negotiations, to support the sustainability of interoperability and adaptation of the networked organizations. The developed intelligent system integrates responses from the collaborative networked environment and actively contributes to the harmonization of breakings through re-negotiation of various jobs and tasks in order to enhance the sustainable interoperability within the network.

The design and implementation of the proposed MAS serve to break boundaries within an automated negotiation environment.

Future work outlooks: i) the completion of the integration of the rule-based framework into our abstract negotiation environment; ii) the assessment of the generality of this implementation by extending it with several types of negotiations occurred in case of breaking downs of interoperability; iii) the allowance of the logical specification of the rules in order to asses their correctness; iv) the investigation of the e-effectiveness by system validation in several collaborative networked environments.

References

1. Pine, B.J., Gilmore, J.H.: The experience economy. Harvard Business Press (1999)
2. Ray, S.R., Jones, A.T.: Manufacturing interoperability. Journal of Intelligent Manufacturing 17(6), 681–688 (2006)
3. Jardim-Goncalves, R., Grilo, A., Steiger-Garcao, A.: Challenging the Interoperability in the Construction Industry with MDA and SoA. Computers in Industry 57(8-9), 679–689 (2006) ISSN 0166-3615
4. White, W.J., O'Connor, A.C., Rowe, B.R.: Economic Impact of Inadequate Infrastructure for Supply Chain Integration. NIST Planning Report 04-2. National Institute of Standards and Technology, Gaithersburg (2004)
5. Friedman, T.: The World is Flat. Farrar, Straus & Giroux (2005)
6. Wilkinson, I., Young, L.: On cooperating: firms, relations and networks. Journal of Business Research 55(2), 123–132 (2002)
7. Amin, A., Cohendet, P.: Architectures of knowledge: firms, capabilities, and communities. Oxford University Press (2004)
8. Beck, P.: Collaboration vs. Integration: Implications of a Knowledge-Based Future for the AEC Industry (2005), `http://www.di.net/articles/archive/2437/` (accessed September 22, 2009)
9. CompTIA: European Industry Association. European Interoperability Framework. white paper - ICT Industry Recommendations (2004)
10. Kadar, M.: Assessement of Enterprise Interoperability Maturity Level through Generative and Recognition Models. In: Proceedings of the 2013 International Conference on Economics and Business Administration (EBA 2013), Rhodes Island, Greece, July 16-19 (2013) ISBN 978-1-61804-200-2
11. Wooldridge, M., Jennings, N.R.: Intelligent Agents: Theory and practice. The Knowledge Engineering Review 10(2), 115–152 (1995)
12. Kadar, M., Cretan, A., Muntean, M., Goncalves, R.: A Multi-Agent based Negotiation System for Re-establishing Enterprise Interoperability in Collaborative Networked Environments. In: Proceedings of 2013 UKSim 15th International Conference on Computer Modelling and Simulation, Cambridge University (Emmanuel College), April 10-12, pp. 190–195 (2013) ISBN 978-0-7695-4994-1
13. Sycara, K., Dai, T.: Agent Reasoning in Negotiation. In: Handbook of Group Decision and Negotiation, Part 4. Advances in Group Decision and Negotiation, vol. 4, pp. 437–451 (2010)
14. Java Agent DEvelopment Framework, `http://jade.tilab.com/`