A Prototype for an Agent-Based Electronic Contracting using an Organizational Model

Djamel Khadraoui and Benjamin Gâteau CRP Henri Tudor. 29 Av. John F. Kennedy, L1855 Luxembourg {*djamel.khadraoui,benjamin.gateau*}@tudor.lu

Abstract

The purpose of a contract, in a B2B context, is to establish an agreement in order to decrease risks associated with the interactions between trading partners. Humans currently carry manually out the business contract processes. This mainly includes the mechanisms of negotiation, signing, validation and monitoring. In order to improve the execution of the contracts with complex scenarios the use of Multi Agent System technologies is more appropriate. This paper describes a prototype implementing the concepts of Electronic Contract dedicated for Tele-services based on MOISE^{Inst}, an organizational model for agents. This model aims to specify the rights and duties of agents in a society according to four points of view: structural, functional, contextual and normative. We use a real-world contract scenario as a test-bed for examining our e-contract architecture and for implementing our prototype.

1. Introduction

The electronic contracting has recently gained lots of interests in the business-to-business domain, as a way to improve traditional paper contracting. In fact, the high number of established and complex contracts has lead problems for Small and Medium Enterprise due to the high costs and high time requirements. New business paradigms that aim to improve the competitiveness of companies like dynamic virtual enterprises and dynamic service outsourcing appear and require an adequate contracting support [4, 6]. We actually assist to an increasing number of B2B applications handling electronic contracts and digital signatures. Depending on the level of automation pursued in e-contract enactment and management, e-contracts can be seen only as digitalized paper contracts [12]. In [5] there is a presentation of a specific Microsoft implementation of the enterprise modeling of contracts. But in most of the existing research work, less attention has been paid to the specific features of the e-Contract content on the supporting architecture. In fact, e-contracts contain specific data due to the technological environment for their enactment. For example, e-contracts can contain parameters defining workflow data exchanged between parties [11]. Additionally to this, most of the existing research works don't take into account negotiation regarding to contracts clauses and arbitration of potential conflicts [14].

In order to improve the execution of the contracts with complex scenarios the use of Multi Agent System technologies would be more appropriate, particularly in the case of dealing with the creation, the execution, the monitoring of the electronic contracts, as well as providing means for arbitration and reputation in case of conflicts [3, 2]. The main goal of the research work presented in this paper is to integrate a Multi-Agents System within an exiting application named EBSME (Electronic Business for Small and Medium Enterprises) [10] in order to allocate more autonomy to the software components that assist users. EBSME is an eCommerce application allowing activities between an employer and an employee through the creation, the signature and the execution of secured electronic contracts via a web-based interface. The main concept of the solution is the execution of an electronic contract, which is a digital object representing an agreement among some participants and describing the obligations of people involved, in terms of deliverables, costs and deadlines. Three participants are evolved in a such a scenario. A tele-worker (or service provider) is the one who seeks for a job and has specific abilities and knowledge to offer. A tele-employer (or service consumer) who seeks for employees looking for a job and having certain skills and knowledge. The third participant os the arbitrator who participates in an Electronic Contract as the trusty third party and resolves the disputes between the tele-worker and the tele-employer.

In this context a contract scenario has been developed to provide a definite setting for the implementation of the contract prototyping system on EBSME. User can trade intangible assets and translation services in particular. In this context, the user would be able to delegate a part of his control in the management and monitoring of contracts. This paper is organized as follows. We present in Section 2 a Multi-Agents organizational model and its contract modeling. Section 3 is dedicated to the implementation architecture with the contract model applied to the scenario of the employment domain. The Section 4 conclude the paper with the perspectives of the implementation results applied to the EBSME platform.

2. Agent-Based Organizational Model Proposal

Electronic Institution: These last years the Electronic Institution concept has been introduced in Multi-Agent System domain and in electronic commerce in particular. In human societies an institution defines the game rules [13]. These rules enclose all kinds of formal or informal constraints that human beings use to interact. The current multi-agent methods propose the modeling of these rules through normative systems [9]. These ones define an institution as a set of agents, which behave according to some norms taking into account their possible violation.



Figure 1. Institution model

In the same way we define our Electronic Institution for electronic contracting as an autonomous agents organization. The behavior of these agents is ruled by norms and controlled by an arbitration system that is allowed to reward or punish agents whether they respect or not their agreements. Figure 1 illustrates the architecture, which make use of this definition. The architecture is composed of two levels: (i) a Multi-Agent System dedicated to application domain in which autonomous agents evolve, (ii) an institutional multi-agent middleware named SYNAI dedicated to the norms management. The two levels render a normative organizational model based on $\mathcal{M}OISE^{Inst}$ metamodel. Thus, the agents are able to reason and to take into account the specification described within \mathcal{M} OISE^{Inst}. The institutional middleware takes into account this specification in order to supervise and control the agent level working.

 $\mathcal{M}OISE^{Inst}$ based Contract Modeling: $\mathcal{M}OISE^{Inst}$ is founded on the $MOISE^+$ organizational model (Model of Organization for multI-agent System) [8]. MOISE⁺ allows to specify the global expected functioning (functional specification) of an agents organization as well as the structure of this organization in terms of roles, groups and links (structural specification). A deontic specification expresses permissions, obligations and prohibitions of functional specification missions regarding to the structural specification roles. $MOISE^{Inst}$ extends $MOISE^+$ by improving the three existing specifications. It adds a specification which aim is to describe contexts. \mathcal{M} OISE^{*Inst*} is basically composed of (see Figure 2) four main specification. The first is s Structural Specification (SS) defining roles that agents will play and relations between theses roles as well as an additional structural level named group to which roles belong. The second is a Functional Specification (FS) defining all goals that have to be reached in the system. The third is a Contextual Specification (CS) defining the different contexts influencing roles as well as transitions between contexts. The last one is a Normative Specification (NS) extending the Moise+ deontic specification and defining clearly rights and duties of each roles on a mission (set of goals), within a specific context.

We consider that a contract must specify the functioning related to the agreement reached by agents through negotiation (internal or external to the system). This agreement concerns the distributed execution of global tasks by the different participants to the contract. In fact, a contract specifies both the functioning and the structure organizing this functioning. This is why this contract model is based on $\mathcal{M}OISE^{Inst}$. This organizational model considers two main dimensions: the organizational structure, defined in a Structural Specification (SS), and the functioning schemes, defined in the Functional Specification (FS). Moreover, it adds an explicit deontic relation, Deontic Specification (DS), among these two first dimensions to better explain how a Multi-Agent System's organization collaborates for the social purpose. The electronic contract life cy-



Figure 2. *MOISE*^{Inst} Specifications

cle is depicted in Figure 3. This model is described in terms

of roles and their relationships which together support contract establishment, execution, monitoring and enforcement stages in a contract life cycle.



Figure 3. Contract Execution Steps

Key roles in the model are mainly composed of five main phases [10]. The first one is the registration Phase. During this phase, the users identify the services. The second one is the creation phase of the contract where we establish formal relation between the different participants. The third phase is the negotiation phase. The fourth is mainly the execution phase where the services are basically provided. The last phase is for the payment where the requester validates the service and pays the providing service.

3. Prototype Development

3.1. Overall architecture



Figure 4. EBSME Agents Organization

As specified in the Figure 1, the architecture of the MAS integration to the electronic contracting application (EB-SME) is composed of the three following main layers:

Agents Layer: The agents will be basically integrated within EBSME at various levels. We first have a layer, which will contain all the agents interacting directly with the system. They take note of their goals and roles while being informed at $MOISE^{Inst}$ (see Figure 4). A second

layer is more precisely based on the constraints properties of the contract (SYNAI layer). This layer has the authority on the first layer in order to force its agents to suitably carry out their tasks. Lastly, we have the Communication layer. This layer is build up of the methods of the SACI API which allows the communication between all the agents. We basically consider that the actual agents layer is composed of passive agents, which have the role of supervising the inputs/outputs. They will be controlling the actions carried out by the users, reading/writing in the contract database. They are used as intermediaries between the system and the user. Then we have decisional agents, also forming part of the first layer of agents, which will have the possibility of making decisions regarding the fulfillments of the contracts.

Decisional Layer: This layer is more related to the action to be performed by the agent regarding the contract execution, like the arbitration procedure, as well as to the contract objects actions such as the search facilities. Lets take a look to these two implemented properties.

• Arbitration:

We consider here two kinds of arbitration: the arbitration performed by a physical user or the arbitration automatically performed by an autonomous agent. The arbitration by an individual is based primarily on the contract provided by the person having created the contract. When the intervention of the referee is required by one of the two parts (the employer or the employee), it is the physical arbitrator who intervenes. He reads the clauses of the contract signed by the three parts and makes a decision according to these considerations. The case of the arbitration by the system is based on the dates and the completion periods of deliverable. An agent created in the system constantly supervises (N time per day for example) the various existing contracts in the database and as soon as a going beyond of completion period is located, it specifies the related penalties to the concerned users. In addition, when a contract is perfectly carried out, it will allot rewards to the users.

• Contract Search:

The search for an offer (a contract) is the second decisional part of the application. This one is based on the profile of the users and their reputation (specified by the agent referees of the system). When a user desire to create a contract, the agents will look for correspondents that have a good reputation attributes and meeting the needs for the applicant (expertise, dates of availability, etc) and present them at the user in a coherent order. They are thus primarily the decisional aspects of the agents integrated in the existing system.



Figure 5. EBSME Structural Specification

Communication Layer: This layer is based on SACI [7], which is a Java API that provides all the necessary methods which facilitate to the agents, in order to organize themselves according to the $MOISE^{Inst}$ model. It offers in particular a powerful communication protocol to the agents. The communications protocol is usable thanks to a script provided with the API one, which makes it possible to make communicate the agents in a transparent way.

| (perform | ative | |) |
|----------|------------|------------|-----------------------|
| 1 | language | word | • message layer |
| 3 | ontology | word | J |
| | sender | word |) |
| | receiver | word | > communication layer |
| | reply-with | word | J |
| 3 | content | expression | content layer |
| |) | | |

Figure 6. KQML message content

The agents communicate with KQML (Knowledge Query and Manipulation Language). In fact, each agent is identified by a reference name which makes it possible to choose the recipients of a message thanks to these reference names. The KQML messages use basically some keywords like presented in [1] and resume on the Figure 6.

3.2. Contract Scenario Specification

3.2.1 Structural Specification

We have several groups in our case of figure (see Figure 5). A "Contract" is a group and is a part of an "Institution" group. In others terms, each instantiation of contracts is supervised by an only one Institution. The group "Contract" is composed of three roles played by three different agents because of non-compatibility between roles. An agent plays the role "Employer", "Employee" or "DomArbitrator". The "Institution" group is composed of supervisor roles which have all authority on *contractual* roles.

3.2.2 Functional Specification

In the roles, we can distinguish two types of roles: roles of the field (domain roles) and roles of the institution. The field is a part suitable for the application. It will vary according to this one.

Domain Roles:

| Role | Properties | |
|---------------|---|--|
| Employee | - Initiate a contract | |
| | - Sign a contract | |
| | - Carry out the part of its contract | |
| | - Ask for arbitration | |
| Employer | - Initiate a contract | |
| | - Sign and execute the part of its con- | |
| | tract | |
| | - Pay the deliverables | |
| | - Ask for arbitration | |
| DomArbitrator | - Fix another deadline and make a loop | |
| | on the cycle of the contract | |
| | - Service refused for: bad quality of | |
| | the service (activity off line), bad rep- | |
| | utation implying the decision to stop | |
| | the process. | |



Figure 7. EBSME Functional Specification

Roles of the Institution:

| Role | Description |
|----------------|--------------------------------------|
| ComManager | Mediator between domain agents |
| | and institutional agents |
| StructManager | Manage the roles adoption and |
| | groups access |
| FunctManager | Manage commitments on missions |
| | and goals state changes |
| ContextManager | Manage context changes relating to |
| | happening events |
| NormManager | Manage the norms respect relating to |
| | roles and contexts |
| InstManagerAg | Manage the organization access and |
| | sanctions application |

3.2.3 Contextual Specification



Figure 8. EBSME Contextual Specification

The contextual specification describes the whole of the contexts, a priori, in which the organization will be able to evolve/move as well as the rules of transition from one context to the other. We distinguishe here the two following contexts (see Figure 8). The "InContract" context which corresponds to some functions like: contract Setup, sign, employee execution, arbitration and payment. The second context is the 'OutContract" which is basically associated to two associated actions. The first is related to the "Registration" where all the users of the system seek to play a role (Employer, Employee, DomArbitrator) within a contract. The second is related to the "Quit" which allows to leave "EBSME"session.

Other important tasks have been elaborated in the implementation of agent within the actual prototype. The main functions are basically summarised in the following:

- Identification which has the responsibility of controlling the access to the platform,
- The registration function manages the profiling of the participants,
- The create contract module initiate the contract execution,
- The contract signing part permits to proceed to the protection of the negotiated contract object
- The contract execution module is responsible for the processing of the contract by the employer or the employee

• The arbitration function is able to process the arbitration mechanism, internally or externally.

4. Conclusion

The proposed prototype presented in this paper is mainly related to two main aspects. First we specified the agent model to be integrated within the existing EBSME application thanks to $\mathcal{M}OISE^{Inst}$. The actual model specifications permit to take into account all the reactions awaited of the agents (related to the rights and duties of the agents in the system) according to various contract case studies. Then, this model was validated by the $\mathcal{M} OISE^{Inst}$ specification thanks to SimOE graphic tool. In fact, this guarantees that the created agents meet the requirements while being coherent with $\mathcal{M}OISE^{Inst}$. We can thus simulate an agent society with regard to the various contexts of EBSME application. Secondly, we can say that the implemented solution allows an on line control of the e-contracting activities in an autonomous way thanks to integration of a Multi-Agents System. The integration of the agents in the application are organized and normalized thanks to the $\mathcal{M}OISE^{Inst}$ model. The application EBSME is used as platform which will thus make it possible to validate the $MOISE^{Inst}$ model. The new architecture offers a real autonomy thus allowing an efficient execution of the contracts. Additionally to this, the implemented GUI performed via the J2EE application and the interaction of the agents platform offers a real robustness to the application. Indeed, thanks to adopted architecture, an agent will be able to communicate with the EJBs without having any major changes on the exiting software components. In 2nd-tier of the application we have the EJBs components and the agents, which communicate with these components.

References

- T. Finin, Y. Labrou, and J. Mayfield. *KQML as an agent* communication language, chapter Agent Communication, Collaboration, and Mobility, pages 291–316. MIT Press, 1997. ISBN: 0-262-52234-9.
- [2] B. Gateau, D. Khadraoui, O. Boissier, and E. Dubois. Contract model for agent mediated electronic commerce (*Poster*). In N. R. Jennings, C. Sierra, L. Sonenberg, and M. Tambe, editors, 3rd International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS), volume 3, pages 1454–1455, Columbia University, New York City - USA, 19-23 July 2004. ACM. ISBN 1-58113-864-4.
- [3] B. Gateau, D. Mathevon, D. Khadraoui, O. Boissier, and E. Dubois. Transforming a secure ecommerce application into a multi-agent based solution for e-contracting. In M. Mohammadian, editor, *International Conference on Intelligent Agents, Web Technology and Internet Commerce -IAWTIC'2004*, pages 418–429, Gold Coast - Australia, 12-14 July 2004. ISBN 1740881893.

- [4] P. Grefen, K. Aberer, Y. Hoffer, and H. Ludwig. Crossflow: Cross-organizational workflow management in dynamic virtual enterprises. *Int. Journal of Computer Systems Science* & Engineering, 15(5):277–290, 2000. ISSN 0267 6192.
- [5] N. He and Z. Milosevic. B2b contract implementation using windows dns. In *ITVE '01: Proceedings of the workshop* on *Information technology for virtual enterprises*, pages 71– 79, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-0960-6.
- [6] Y. Hoffner, S. Field, P. Grefen, and H. Ludwig. Contractdriven creation and operation of virtual enterprises. *Comput. Networks*, 37(2):111–136, 2001.
- [7] J. F. Hubner and J. S. Sichman. Saci: Uma ferramenta para implementaão e monitoração da comunicação entre agentes. In M. C. Monard and J. S. Sichman, editors, *International Joint Conference, 7th Ibero-American Conference on AI*, *15th Brazilian Symposium on AI (Open Discussion Track)*, pages 47–56, São Carlos, 2000. ICMC/USP.
- [8] J. F. Hubner, J. S. Sichman, and O. Boissier. MOISE⁺: towards a structural, functional, and deontic model for mas organization. In *Proceedings of the first International Joint Conference on Autonomous Agents and MultiAgent Systems*, pages 501–502, Bologna, Italy, 2002. ACM Press. ISBN 1-58113-480-0.
- [9] A. Jones and J. Carmo. *Handbook of Philosophical Logic*, chapter Deontic logic and contrary-to-duties, pages 203– 279. Kluwer, 2001.
- [10] D. Khadraoui and E. Dubois. B2b econtract solution for teleservices. In International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC2003), pages 12–14, Vienna, Austria, February 2003. ISBN 1740880692.
- [11] M. Koetsier, P. W. P. J. Grefen, and J. Vonk. Contracts for cross-organizational workflow management. In EC-WEB '00: Proceedings of the First International Conference on Electronic Commerce and Web Technologies, pages 110– 121, London, UK, 2000. Springer-Verlag. ISBN 3-540-67981-2.
- [12] H. Laurikkala and K. Tanskanen. Managing contracts in virtual project supply chains. In PRO-VE '02: Proceedings of the IFIP TC5/WG5.5 Third Working Conference on Infrastructures for Virtual Enterprises, pages 93–100, Deventer, The Netherlands, The Netherlands, 2002. Kluwer, B.V. ISBN 1-4020-7020-9.
- [13] D. C. North. Institutions, Institutional Change and Economic Performance. Political Economy of Institutions and Decisions. Cambridge University Press, 1st edition edition, October 26 1990. ISBN 0521397340.
- [14] B. Padovan, S. Sackmann, T. Eymann, and I. Pippow. A prototype for an agent-based secure electronic marketplace including reputation tracking mechanisms. *International Journal of Electronic Commerce*, 6(4):93–113, 2002.