Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 1, Number 1; October, 2014 pp. 11-16
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/acsit.html

Multiagent System for Admissions in Academic Institution

Puneet Modgil[1], **M. Syamala Devi**[2]

*[1]Research Scholar, Department of Computer Science and Applications,
Panjab University, Chandigarh, India
punmod@rediffmail.com
[2]Professor, Department of Computer Science and Applications, Panjab University, Chandigarh, India
syamala@pu.ac.in*

Abstract: Admission of students is an important activity of any academic institution along with many other activities like conducting examinations, result preparation and course scheduling. All these activities are time consuming, distributed and involve number of knowledge intensive sub-activities. If these knowledge intensive sub-activities are automated using agents, the entire work can be efficiently performed without much human intervention. An Agent is a software entity that can automatically perceive environment and can behave reactively as well as proactively. A Multiagent System is distributive artificial intelligence system in which a number of agents interact with each other to solve problems cooperatively to achieve the desired objectives. In this paper a multiagent system for admission activity called Multiagent System for Admissions (MASA) is designed and developed. In MASA, the knowledge intensive sub-activities of admission such as filling the admission form of admission seeker with very little information given by seeker whereas retrieval of rest of information from relevant sources; and passing of information about admitted students to the external academic system are automated by means of agents. MASA is analyzed and designed using agenttool3 which is supported by Organization based Multiagent System Engineering (O-MaSE) framework. MASA is implemented using Java Agent Development Environment (JADE). The interface of entire system to end users is provided using Java Enterprise Edition 7 technologies. It is concluded that Multiagent Systems can automate number of activities which otherwise are performed by humans only, and improve system performance with respect to time and accuracy.

1. INTRODUCTION

An academic institution consists of many activities which are distributed and knowledge intensive in nature like admission of students, course scheduling, exam scheduling and result preparation. Admission of students is one of the indispensable activities of any academic institution especially in courses where admission is done on merit basis, a knowledge intensive and distributed activity. The first sub-activity in admission activity is filling of forms by admission seekers. Secondly, all the forms are scrutinized for checking the details filled in by seekers. Thirdly, additional weightage is assigned to forms by scrutinizer based on the course desired and previous record of the seeker. Then scrutinizer creates the merit list with respect to percentage of seeker and weightage assigned. This merit list is displayed to admission seekers and given to admission coordinators. After that on a pre-fixed day, interview of admission seekers is taken by admission coordinator according to merit list and seats are allocated. Lastly, information of all the admitted student is sent to concerned authorities like university to which institution is affiliated.

1.1 Multiagent system

An Agent is a software entity that can automatically perceive environment and can behave reactively as well as proactively. The system where a number of agents interact with each other to achieve some common goal is called Multiagent System (MAS).

More specifically Multiagent System is distributive artificial intelligence system in which a number of agents interact with each other to solve problems cooperatively to achieve the desired objectives. Agents use knowledge based reasoning approaches for solving problems assigned to them. Autonomy, reactivity, pro-activeness and social ability are important agent properties. The emphasis of work in Multiagent System is on behavior co-ordination amongst agents. A global solution arises from the co-operation between agents. Thus agents are well-suited for solving the problems which are knowledge intensive and distributed in nature.

This paper defines how various sub-activities of admission activity that are knowledge intensive, are automated with help of agents, thus producing a Multiagent System. In section 2, related research is discussed. Section 3 explains created Multiagent System for Admissions (MASA) in context of its functionality, process used to create it along with its analysis and design. Implementation is detailed in section 4. Section 5 provides conclusion and scope of future work.

2. RELATED WORK

Many systems and their activities are automated with the help of Agent-based Multiagent systems like I-MINDS: A Multiagent System for Intelligent Computer- Supported

Collaborative Learning and Classroom Management [1]. I-MINDS provides a computer-supported collaborative learning (CSCL) infrastructure and environment for learners in synchronous learning and classroom management applications for instructors meant for large classroom or distance education situations. A MAS named ExPlanTech: Multiagent Support for Manufacturing Decision Making is defined in [2]. ExPlanTech automates number of decisions taken during manufacturing process of automobiles. Another system created with MAS approach named "A multiagent knowledge and information network approach for managing research assets" is elaborated in [3]. A MAS for timetabling in university scenario is discussed in [4]. A MAS, named X.MAS, which retrieves information from multiple sources on web is elaborated in [5]. Another MAS for university is detailed in [6].

Organization based Multiagent systems (OMAS) have been evolving as one variation of Agent-based Multiagent systems. OMAS incorporates the concepts of organization in multiagent systems [7] like that of groups in correspondence to departments in organizations, agents in correspondence to humans and roles in correspondence to responsibilities of humans in organizations. Consequently applications are designed using organization based multiagent system paradigm, many of which are specified in [8]. Systems are created using different methodologies like AGR [7], ASPECS [9], Extended Gaia [10-11], INGENIAS [12], MaSE [13], Moise+ [14] and OperA [15] and O-MaSE [16]. Amongst all these methodologies O-MaSE is one which covers most of lifecycle phases of system development. O-MaSE has unique features which especially support the development of organization based Multiagent Systems. Thus MASA is created using O-MaSE framework.

Many organization based systems are developed with the help of O-MaSE methodology like "A Task-based Support Architecture for Developing Point-of-care Clinical Decision Support Systems for the Emergency Department" [17]. The system defined in [17] supports decision making in Emergency department during emergency situations. Another organizational MAS defined in [18] named "Integrative Multi-agent Clinical Decision Support System with Open Source Software" supports decision making in clinical scenario.

3. MULTIAGENT SYSTEM FOR ADMISSION (MASA)

3.1 Overview of MASA

The Multiagent System for Admissions (MASA) is an organization based multiagent system which automates admission activity of any academic institution irrespective of its size. The system consists of five agents namely: Interface agent, FormIncharge agent, Scrutiny agent, Informer agent and Admission agent. The admission activity starts with 'Form filling' sub-activity. 'Form filling' sub-activity consists of the task of filling the form (online) by admission seeker. This task is supported by FormIncharge agent of MASA. Next sub-activity is 'Scrutiny'. This sub-activity is supported by Scrutiny agent in MASA. After that seekers and admission coordinators need to be informed about merit list. This sub-activity is again taken care of by FormIncharge agent. After this, on a predefined day interview of admission seekers is taken by admission coordinator and seat is allocated. This entire sub-activity is supported by Admission agent in MASA. After completion of seat allocation, information of students admitted is sent to University (to which institution is affiliated). This sub- activity is performed by Informer agent of MASA

3.2 Knowledge Intensive and Intelligent activities of MASA

Three main knowledge intensive and intelligent activities performed by MASA are: assistance in Filling of form by FormIncharge agent, scrutiny of forms by Scrutiny agent and passing of information about admitted students to university by Informer agent.

FormIncharge agent assists every admission seeker in filling the form as it asks seeker only for his/her previous class board/university, name of course he/she is seeking admission in, previous class roll number and year of passing and then it retrieves rest of information from web site of university/board from which student had passed his previous class. Thereafter, it asks for confirmation of retrieved information from seeker and then saves the form in database for further processing. In case FormIncharge agent is unable to locate web site of specified board/university it asks the seeker to fill form himself/herself and in addition manually submit the copy of the form in institution for further processing.

The second knowledge intensive activity automated is scrutiny of forms by Scrutiny agent. Scrutiny agent supports scrutinizer by assisting him in checking marks specified in the forms and assigning additional weightage to forms based on the class in which seeker is seeking admission and his previous record. Scrutiny agent provides the scrutinizer a list of forms which are filled by FormIncharge agent that need not be further checked for marks. Forms in this list are automatically assigned weightage by Scrutiny agent using predefined rules according to course in which admission is sought and previous record of the seeker. After confirmation from scrutinizer these forms stand '*scrutinized'*. However, a second list is also displayed to scrutinizer in which the forms are not filled with help of FormIncharge agent and thus, these forms are manually checked for marks after seeker has submitted form along with duplicate documents. After these forms are also checked for marks, Scrutiny agent assigns weightage to them automatically and generates a merit list and informs the concerned agents.

The third knowledge intensive sub-activity which MASA automates is informing the university by logging in universities website and entering the information about admitted students without any human intervention. Once the Informer agent gets information that seats of particular course are fully filled or a particular deadline date is reached, it wraps up university's online system and passes information automatically.

3.3 Process for creation of MASA

Tasks specified in process for creating MASA include creation of a goal tree along with mention of precedence of one sub-goal over other. Next, 'Organization Model' is created to capture all the interfaces to external actors and systems. Then 'Goal Model' is used to create the initial 'Role Model'. Based on the 'Role Model', an 'Agent Class Model' is created. The details of the agents and protocols identified in the 'Agent Class Model' are further refined into 'Protocol Models' and 'Agent Plan Models'. Finally, implementation and testing takes place. Implementation includes creation of agents and their behaviors on some agent oriented platform.

3.4 Analysis and Design of MASA

Only major models produced during the process will be discussed here because of constraint on space. All the models are generated using agenttool3 [19, 20], which is O-MaSE compliant tool. Agenttool3 is used by integrating it to eclipse environment along with graphical framework.

3.4.1 Role Model

This model, shown in Figure 4, specifies various roles which need to be played by any admission system in order to complete the admission activity. Role is an abstraction recognized by all the major methodologies for creating organizational Multiagent Systems [7,8], as it provides the basic abstraction of the works to be done by the system with/without agents. 'Role Model' along with various roles specifies various protocols using which different roles can interact with each other.

Various roles specified in 'Role Model' are: Form Filler, Form manager, Scrutinizer, Notifier, Seat allocator and University Informer. Every role has some responsibilities associated with it that it is supposed to achieve. 'Form filler' role is given responsibility named: 'Get the form filled' which is meant to help the admission seeker in filling the form. Further 'Form manager' is given the responsibility of collecting the forms and distributing the forms where required. 'Scrutinizer' is given three responsibilities namely: 'Check Marks', 'Give Weightage' and 'Generate Merit list'. 'Notifier' is given the responsibility of notifying seekers and admission coordinators about the merit list. 'Seat allocator' is given responsibility of allocating the seat to admission seeker thus giving him/her admission according to merit list. Lastly, 'University Informer' has the responsibility of informing university about admitted students.

Further, all those roles which need to be automated especially if those roles have responsibilities which are knowledge intensive in nature are assigned to agents as agents can fulfill both knowledge intensive and procedural activities through their behaviors.

3.4.2 Agent Model

This Model shows all the agents of MASA along with the roles which need to be played by them, thus objectives need to be achieved by them. Agents follow same interaction protocols as that of roles assigned to them. Agents specified in Agent Model are: Interface Agent, FormIncharge Agent, Scrutiny agent, Admission agent and Informer agent.

Interface agent provides the interface of agent system with non agent system that is the end user. FormIncharge agent plays the roles of 'Form Filler', 'Form Manager' and 'Notifier' as responsibilities of all these roles are closely related to each other. For achieving its objectives FormIncharge agent needs to interact with other agents: with Interface agent using 'Log in' and 'Notify' interaction protocols; with Scrutiny Agent using 'Collect forms', 'Submit Scrutinized forms' and 'Submit merit list' interaction protocols; with Admission agent using 'Get scrutinized forms' and 'Submit forms' protocols and with Informer agent using 'Inform' interaction protocol. Similarly, different protocols used by agents for different interactions with other agents are shown in figure 2. All Interaction protocols are further elaborated in interaction protocol model along with performatives to be used for sending and receiving messages.

3.4.3 Other Models

Apart from the specified models, various other models are generated such as 'Plan Model', 'Protocol Model', 'Capability and policy Model'. 'Plan Model' provides detailed design by means of specifying the plans using which agents will perform roles assigned to them, thus achieve the goals of their roles. 'Plan Model' explains the system in form of states and transitions between them. 'Plan Model' provides the detailed design of all the agents specified in agent model. 'Protocol Model' specifies kind of messages along with the performatives that are allowed in interaction between two roles and/or agents. 'Capability Model' specifies various capabilities required for fulfilling various roles.

4. IMPLEMENTATION

All the agents specified agent model are created using Java Agent Development Environment (JADE) [21]. Responsibilities or roles of agents are implemented in form of behaviors of agent classes of JADE.

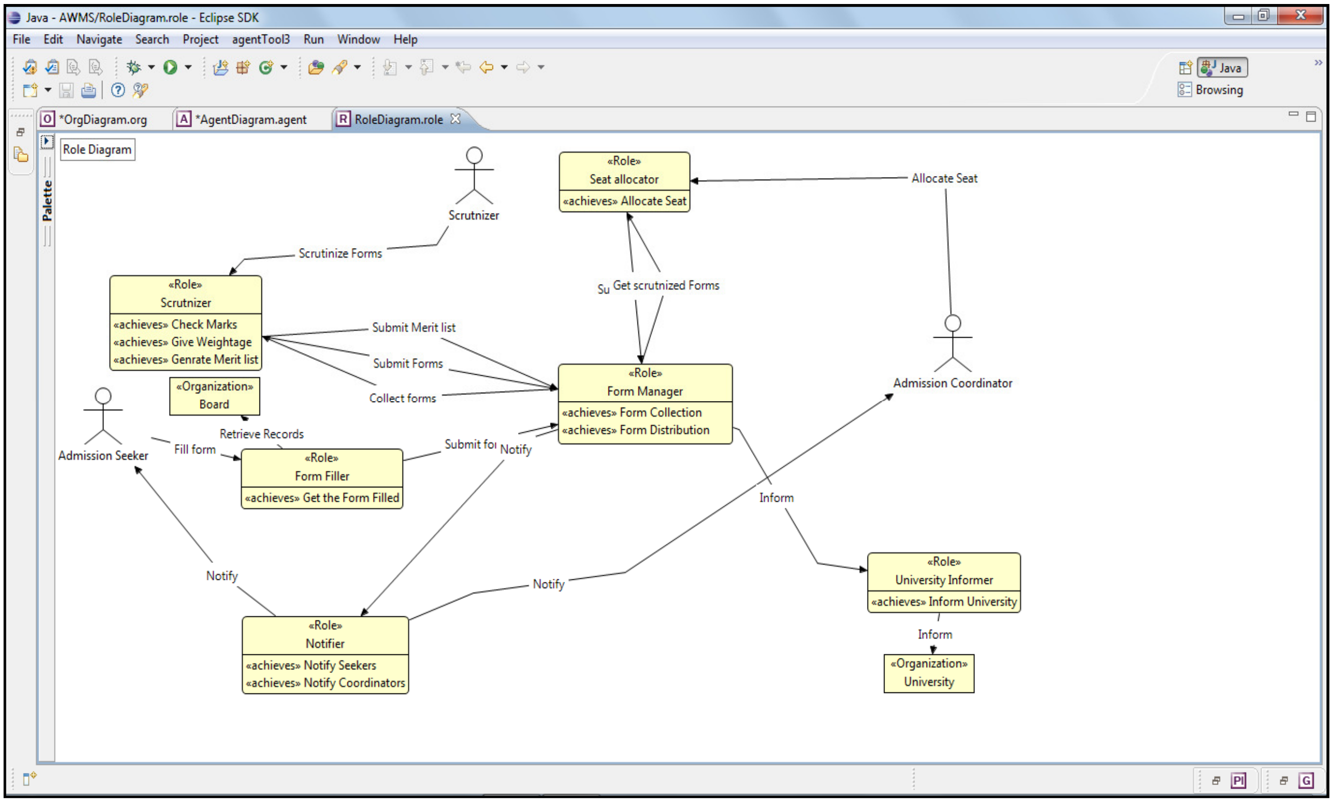


Fig. 1: Role Model

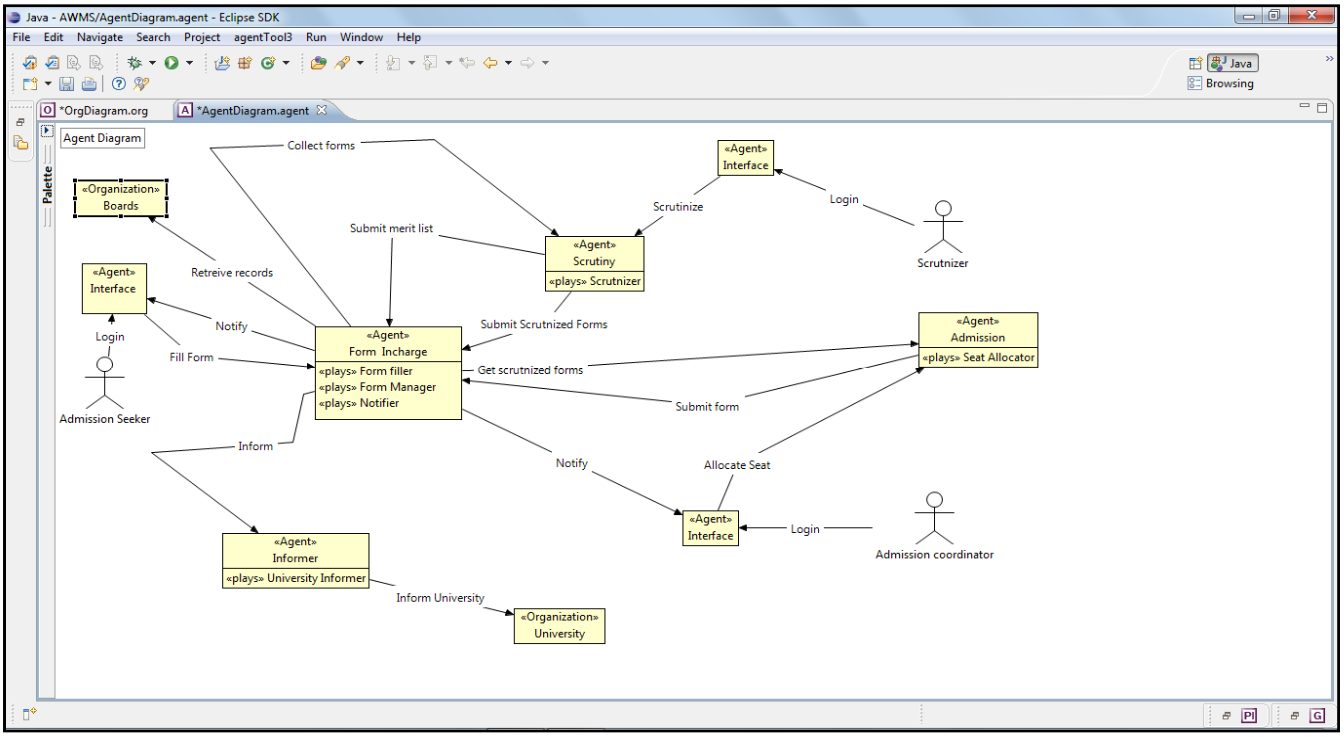
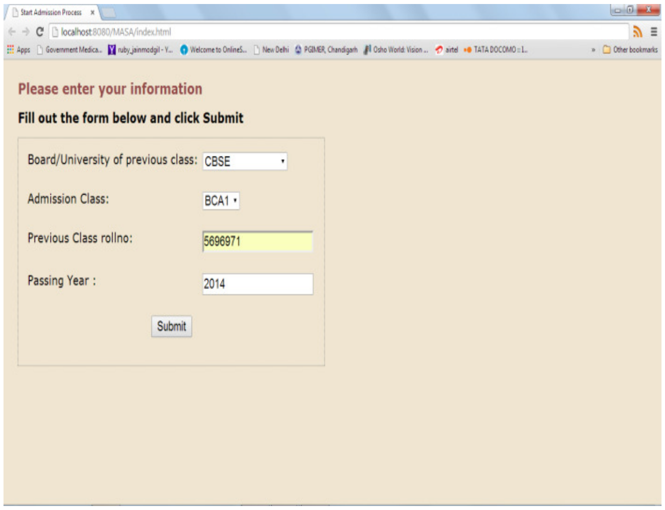


Fig. 2: Agent Model

Services are provided to users by using Java EE 7 technologies. Clients are given web interface which interact with servlets on the server and servlets further interact with agents. Interface is created between Jade Agents and Java EE 7 web servlets through jadeleap package which provides framework of gateway agent between servlets and JADE agents. Agents further interact with each other and retrieve data from database, to achieve goals assigned to them. Database is created using MySql. Though MASA is implemented and tested on Window 7 platform but it can run on any platform which is compatible JADE. Since JADE is fully made with Java which is compatible with all platforms. MASA can run on all platforms.

End users can access the system through web browsers, as they will be served by web servlets on server. First Interface for admission seeker is shown in figure 3. In this interface admission seeker needs to specify only his/her board/university, name of course he/she is seeking admission in, previous class roll number and year of passing. Rest of the form will be filled for him/her and saved in database for future use after confirmation. MASA is tested on real data of admission of course BCA in Chandigarh colleges, and results were found compatible with the present system.



Please enter your information

Fill out the form below and click Submit

Board/University of previous class: CBSE

Admission Class: BCA1

Previous Class rollno: 5696971

Passing Year : 2014

Fig. 3: Interface for Admission Seeker

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

The defined MASA automates different knowledge intensive sub-activities of admission activity that include filling of forms, scrutiny of forms and informing the university about admitted students. MASA is highly maintainable as changing of one agent's behaviors will not affect other agents as long as interfaces are kept same. MASA is distributed in nature as different agents may by executed on different machines according to work load. MASA is tested with real data and results were found compatible with present system in which all knowledge intensive sub-activities are performed by humans. Thus it is concluded that MASA has automated entire admission activity using agents.

5.2 Future Work

At present the FomIncharge agent keeps all the forms by means of records in MySql database, but future work seeks to define ontology for admission process. All agents will use same ontology for communication thus providing the ability of inferring more. The system created automates only one activity of academic institution yet. The expanding of automation of various other activities is proposed for future work. All the activities will be integrated in such a way with the help of agents and ontology, that entire institution could be represented by one Organization based Multiagent System (OMAS). At present Informer agent passes on information only to university's system but in future it will be defined as a web service so that any authorized organization would be able to retrieve information from this system.

6. ACKNOWLEDGEMENTS

This work is proposed under Rashtriya Uchtar Shiksha Abhiyan (RUSA) for Chandigarh Administration's Higher Education Institutions, under the proposal named Multiagent system for Workflow Management of Academic institution (MASWMA).

REFERENCES

[1] Soh L, Khandakar N, Jiang H. "I-MINDS: A Multiagent System for Intelligent Computer- Supported Collaborative Learning and Classroom Management" International Journal of Artificial Intelligence in Education 2008,18(2),119-151.

[2] Pechoucek M, Vokrinek J, Becvar P. ExPlanTech: Multiagent Support for Manufacturing Decision Making. IEEE Intelligent Systems 2005; 05; 154-1672.

[3] Cantú j, Ceballoss G. "A multiagent knowledge and information network approach for managing research assets" Expert Systems with Applications, ELSEVIER 2010, 37; 5272–5284.

[4] Oprea M. "MAS_UP-UCT: A Multi-Agent System for University Course Timetable Scheduling" International Journal of Computers, Communications & Control 2007, 2; 904-102.

[5] Addis A, Armano G, Vargiu E. "Multiagent systems and information retrieval our experience with X.MAS." Expert Systems with Applications, ELSEVIER 2012; 39: 2509-2523.

[6] Taskin H, Gozetepe K, Taskin M, Canay O. "Agent Based University Planning of Sakarya University: "CAWIS Project" Journal of Theoretical and Applied Information Technology 2008; 4; 753-759.

[7] P. Giorgini, Jörg Müller, James Odell, "From Agents to Organizations: an Organizational View of Multi-Agent Systems", Agent-Oriented Software Engineering (AOSE) IV July 2003, 214-230

[8] Isern D, Sanchez D, Moreno A, "Organizational structures supported by agent-oriented methodologies", The Journal of Systems and Software 84, 2011, 169-184.

[9] Cossentino, M., Gaud, N., Hilaire, V., Galland, S., Koukam, A., 2010. ASPECS: an Agent oriented Software Process for Engineering Complex Systems: How to design agent societies under a holonic perspective. Auton. Agents Multi-Agent Syst., DOI: 10.1007/s10458-009-9099-4.

[10] Zambonelli, F., Jennings, N.R., Wooldridge, M., 2003. Developing multiagent systems: the Gaia methodology. ACM Trans. Soft. Eng. Methodol. 12, 317–370.

[11] Zambonelli, F., Jennings, N.R., Wooldridge, M., 2005. Multi-Agent Systems as Computational Organizations: The Gaia Methodology. In: Henderson-Sellers, B., Giorgini, P. (Eds.), Agent-oriented Methodologies. Idea Group, pp. 136–171.

[12] Pavón, J., Gómez-Sanz, J.J., Fuentes, R., 2005. The INGENIAS methodology and tools.In: Henderson-Sellers, B., Giorgini, P. (Eds.), Agent-oriented Methodologies. Idea Group, pp. 236–276.

[13] DeLoach, S.A., 2004. The MaSE methodology. In: Bergenti, F., Gleizes, M.P., Zambonelli, F. (Eds.), Methodologies and Software Engineering for Agent Systems:The Agent-oriented Software Engineering Handbook. Kluwer Academic Publishers, pp. 107–125.

[14] Hübner, J.F., Sichman, J.S., Boissier, O., 2002b. MOISE+: towards a structural, functional, and deontic model for MAS organization. In: Castelfranchi, C., Johnson, W.L. (Eds.), First International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2002. ACM Press, Bologna, Italy, pp. 501–502.

[15] Dignum, V., 2004. A Model for Organizational Interaction: Based on Agents, Founded in Logic. University of Utrech, Utrech, The Netherlands.

[16] DeLoach, S.A. and García-Ojeda, J.C. "O-MaSE: a customisable approach to designing and building complex, adaptive multi-agent systems", Int. J. Agent-Oriented Software Engineering 2010, 4, 3, 244–280.

[17] Sz. Wilk, W. Michalowski, D. O'Sullivan, K. Farion, J. Sayyad-Shirabad, C. Kuziemsky, B. Kukawka. A Task-based Support Architecture for Developing Point-of-care Clinical Decision Support Systems for the Emergency Department. Methods of Information in Medicine, vol. 52, no. 1, 2013, 18-32.

[18] Jelber S, Szymon W, Wojtek M, and Ken F. 2012. Implementing an Integrative Multi-agent Clinical Decision Support System with Open Source Software. J. Med. Syst. 36, 1 (February 2012), 123-137.

[19] Zambonelli, F., Jennings, N.R., and Wooldridge, M.J. Organisational Rules as an Abstraction for the Analysis and Design of Multi-Agent Systems. In-ternational Journal of Software Engineering and Knowledge Engineering. 11, 3, 2001. Pages 303-328.

[20] Juan C. Garca-Ojeda, Scott A. DeLoach, and Robby, agentTool III: From Process Definition to Code Generation, Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), Decker, Sichman, Sierra, and Castelfranchi (eds.), May, 10–15., 2009, Budapest, Hungary, pp. 1393-1394.

[21] Fabio B, Giovanni C, Dominic G, "Developing Multi Agent Systems with JADE", Wiley 2007.