Chapter 20
Normative Multi-Agent Enriched Data Mining to Support E-Citizens

Stanislaw A. B. Stane and Mariusz Zytniewsk

Abstract The current trends in the development of multi-agent systems indicate the possibility to apply the concept of multi-agent systems employing ontologies for encoding the systems domain knowledge and procedural knowledge. Within such structures, the use of knowledge discovery models may represent an enhancement to the systems functionality in the context of discovering relations that can support the users activities. Distributed data mining (DDM) concepts ([19, 11, 18]) demonstrate that multi-agent systems are capable of using knowledge discovery processes in a variety of ways in the context of the process being supported by the agent as well as in order to expand its knowledge. If this is the case, the knowledge discovery becomes an intrinsic component of the agents learning process. The application of norms to support the work of agents, associated with the idea of normative multi-agent systems ([30, 2, 5, 4]), may significantly boost the performance of such systems by directing the agents actions and determining the desirable states of the agent itself as well as of the group it is part of. The chapter aims to discuss the key aspects of the development of multi-agent systems and knowledge discovery systems, and to present a proposal for an architecture of multi-agent systems supported by knowledge discovery systems.

20.1 Introduction

It could be argued that we currently see the emergence of the third generation of systems oriented on data mining processes. The first phase in their evolution was concentrated on the development of algorithms to support the data mining process, and the mainstream of research aimed to identify the possible types of analysis to be implemented under different systems. To a large extent, efforts were directed on the creation and analysis of data mining algorithms. The second stage of the evolution

Karol Adamiecki University of Economics in Katowice, POLAND
e-mail: {stanek, zyto}@ae.katowice.pl

L. Cao (ed.), Data Mining and Multi-agent Integration, DOI: 10.1007/978-1-4419-0522-2_20,
© Springer Science + Business Media, LLC 2009

focused on defining development methodologies for knowledge discovery processes that would extend the data mining process itself (such that involves the use of a kind of algorithm) to incorporate steps involving the preparation and evaluation of the data and the outcomes. Examples of such solutions include SEMMA and CRISP-DM [16, 6]. Thus defined, the second generation was related to the emergence of the concept of knowledge discovery systems that are expected to integrate diverse data mining models and support the pre-processing of data to be analyzed.

The current phase in the evolution of data mining does not simply seek to apply this or other kind support to the knowledge discovery process, but it also endeavors to provide support to business organizations, whereby the data mining process itself becomes an element of a broader business process. Within this approach – from the end user's perspective – data mining constitutes at once a tool and a process and becomes just one of the stages in the operation of the organizational information system. This being the case, the end user is not equipped with the knowledge of a systems analyst and, more often than not, is oriented on the outcome to be generated rather than on the knowledge discovery process. Yet, to be able to employ such an approach to the application of data mining within organizations, we need to have mechanisms to store and propagate data mining models across the system. The latest technology solutions make it possible to define data mining models as part of the organizational global metadata set, so that their structure and their outcomes can be accessed by different systems. The multi-agent system architecture using data mining to support customer service, discussed further in the chapter, represents an example of such a solution. Software agents, which can at the same time handle the knowledge discovery process and the delivery of pre-processed information to users, are thus capable of mediating between the end user and the knowledge discovery process.

When analyzing the current trends in the development of organizational knowledge discovery systems through the prism of multi-agent applications, two main streams of research relating to the Distributed Data Mining (DDM) concept can be identified ([19, 11, 18]). The former sees agent technology as an instance of intelligent systems designed to replace or assist humans in using information systems such as, in particular, ERP and Business Intelligence solutions. Within this class of systems, software agents may be components of an intelligent system infrastructure which is able to carry out routine activities and thus support the functioning work of the system as a whole ([24, 15, 17]). In this case, software agents can employ data mining in pursuing their objectives, such as e.g. performance monitoring or generating recommendations. Besides, software agents may by used as part of a data mining system, where software agents equipped with expert knowledge can control the knowledge discovery process by analyzing the output generated by the system. In this case, the software agent can be considered as a component of the data mining model. A deductive problem solving process within a software agent's knowledge system can be enhanced by the addition of an inductive knowledge discovery process. This makes it possible to build hybrid multi-agent systems that are capable of supporting various experts. Typical examples of such solutions are Bodhi, Padma, JAM, Papyrus [18] or DAMSA (DAta Mining System based multi-

Agent) [1]. Within architectures of this kind, software agents can assist in selecting data for analysis, choosing the data mining method, validating the results, and optimizing the data mining process.

The authors of this chapter intend to present a proposed multi-agent system architecture incorporating normative solutions as well as knowledge discovery processes, designed to support customer service at a local government office. The discussion will be structured as follows. The first sub-chapter describes sample multi-agent systems supporting knowledge discovery processes, along with the underlying concepts, pointing out the directions in which such systems currently evolve. The second sub-chapter outlines the theoretical foundations for developing normative solutions with a view to building normative multi-agent systems. The third sub-chapter discusses examples of architectures and implementations of systems that make use of norms to support system operation. The fourth sub-chapter puts forth a proposal for a multi-agent system architecture including knowledge discovery systems and supporting customer service at a local government office.

20.2 Theoretical issues – A Multi-Agent System vs. Norms

From the viewpoint of both data mining and software agent technology, a multi-agent system can be treated as a data source for the data mining process, particularly in the construction of simulation models, where software agents can generate a large amount of output. In such systems, the social behaviors of software agents can be analyzed with regard to interaction and collaboration as well as with respect to the dynamic structures that emerge during the simulation. Such solutions may be deployed in economics (Agent-Based Computational Economics) [1], [26] sociology [23], anthropology [2], etc. On the other hand, results generated by data mining systems may be part of a software agent's knowledge. In this case, the process of knowledge extraction can extend the software agent's learning process [27], [22], providing an opportunity to build self learning systems with adaptive capabilities.

Currently, business oriented knowledge discovery systems are closely integrated with business solutions such as SAS Data Miner 5.3, in which the management of data mining models is performed via metadata servers. The approach supports the use of multi-agent systems and offers new prospects for systems integration and management. The autonomy of software agents, being an essential factor in a system's operation, entails the need for the construction of mechanisms to support decision making by agents. Normative systems, which provide for the integrity and soundness of software agents' actions by allowing a possibility to pre-define these, could be seen as a viable response to this requirement. In society, norms will support the processes of task performance by members of the community through establishing standards for individual behaviors. In executing a specific task and interacting with the environment, we can rely on the accepted norms, rules and standards to help us predict what actions can be taken by other individuals as a result of our own behavior. Norms can be broken down into formal ones, which have been written down as laws and regulations, and informal ones, which do not have binding legal force but, nevertheless, hold within a specific society or social group. Social norms can also be divided into legal, religious, traditional, cultural, ethical, etc. However, in terms of their application in software agent theory, a broad distinction of norms is not required. [30] identifies three types of norms that can affect software agents:

1. norms that define the **meaning** of words, expressions, names, etc.
2. norms that define **actions** as well as their constituent activities, such as plans, rules, procedures, etc.
3. norms that define **obligations, permissions** and **restrictions**.

Another distinction which can be used in agent-based systems has been introduced in [2] and [5]:

1. **regulative norms**: obligations, permissions, prohibitions,
2. **constitutive norms**: counts-as conditionals in which are manifest the general norms defining relationships among the components of the system,
3. **procedural norms** defining procedures for the performance of specific tasks, e.g. the rules of administrative proceedings.

Within multi-agent systems, according to [3] and [29], norms are not addressed to any specific agents but to the roles they play. In case problems arise that are attributable to some agents' overlapping objectives, the relevant objectives can be altered, depending on their priorities. Normative reasoning is founded on the fact that [3] a norm is not identifiable with an objective or an obligation but it forms a set of criteria that allow an agent to choose a valid objective or adopt a suitable attitude. Besides norms that are specific to each autonomous agent and can be modified by the agent itself, there ought to exist system-wide norms that hold for all entities throughout the system. Within systems where autonomy represents a determining factor for an agent's behaviors, the application of knowledge discovery systems may enhance the process of learning and contribute to defining new norms. Prolonged examination of the behaviors of groups of agents may lead to discovering important relationships and optimizing the agents' behaviors by generating further internal norms.

In decentralized systems (i.e. those with networked, or distributed, architectures), where agents are fully autonomous, some authors [8] will employ the notion of social order, which is achieved through social control. To ensure control mechanisms in such environments [8], each agent in the system is assigned an additional role, which is to supervise another agent. This makes it possible to monitor and analyze its behavior. While a similar degree of control over the agents' behaviors can be exercised by introducing a superior agent, this latter approach will increase the number of software agents, which immediately poses issues relating to structural complexity. [13] indicates two paths for the implementation of norms into an agent system, where:

1. The former defines sets of norms as specifications of correct behaviors, which are implemented directly in a software agent's code. In this case, it is the software

engineer that decides what norms the agent will abide by. The writer of [8], however, insists that norms should not be implemented in an agent's code within systems of this sort. This is due to likely heterogeneity issues as well as to the fact that, within such an environment, agents should be able to analyze and modify their own norms.
2. The latter path is based on an assumption that norms are no more than definitions of correct behaviors which may be accepted by a software agent or not. In this case, it is the agent that decides what norms will be approved and which norms will be breached. Norms can thus become an element of the system's knowledge.

[21] argues that the approach involving constraints on actions relates to the notion of social laws which impact on the agent by determining the behaviors that are appropriate at a particular moment or in specific circumstances. In their present form, the applications of norms within multi-agent systems described in topical literature do not show a sufficient degree of formalization and do not provide convenient tools for the development of such systems [13]. In the case of software agent norms will perform functions that support its interactions with other agents, with humans, and with resources. As far as social order is concerned, norms can be perceived in terms of interaction between the normative system and its environment [4]. According to [4] and [2], a normative multi-agent system can be conceived of as a combination of a multi-agent system and a normative system where, on the one hand, agents can make decisions concerning the norms built in the system and their observance while, on the other hand, the normative system specifies how the norms can be altered by agents. What is remarkable about the approach under consideration is that a normative system may be perceived by an agent as a normative agent. If this is the case, a normative agent is equipped with mechanisms to enabling it to analyze norms. The concepts discussed in this section signify a possibility to build systems enriched with norms and supported by data mining processes. The following sub-chapter will present examples of normative multi-agent systems along with recommended system architectures.

20.3 A Normative Multi-Agent Enriched Data mining Architecture and Ontology Frameworks

[10] claims that *...artificial societies are typically characterized by agents that interact with each other in accordance with common rules or norms. Similarly to a human society, members of the artificial society must be allowed to coexist in a shared environment and to follow their respective goals in the presence of others. Here, the application of norms serves an important purpose in that they govern the rules of participation and provide important measures to achieve the desired behavior in a society.*

These concepts are founded on the use of trust deriving from the observance of applicable norms. In subject literature, trust is typically defined in the context of specific goals, where agents resident within the system can form small groups and temporarily cooperate. The recent trends in the design of hybrid multi-agent systems are headed toward the development of **heterogeneous, open** and **dynamic** systems. This is illustrated in Fig. 20.1. From the perspective of a software agent, norms

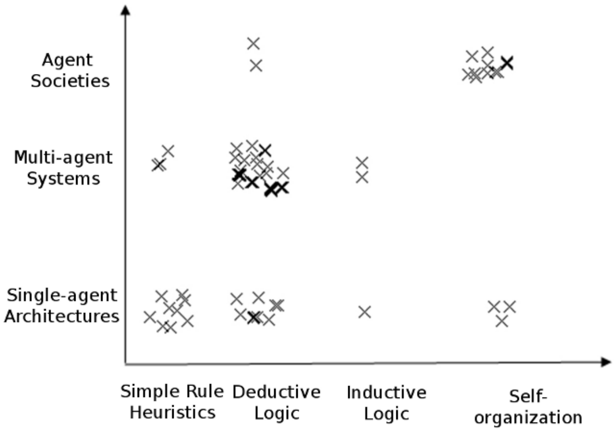

Fig. 20.1: A review of approaches to the construction of the reasoning component. Source: [24]

may determine the principles of its interaction with the user, while in a multi-agent system they will usually regulate an agent's operations within the system, and in artificial societies they will affect the rules of participation and thus support activities involving trust [12, 14] and reputation [12, 9]. Some authors view agent organizations as a space in which agent institutions can be defined. Agent organizations are then regarded as groups of agents that have been united for a cause. Such groups make up [12] social entities, have a definite structure as well as its own resources and authorities, and are created for the pursuit of emergent objectives.

Another term designating solutions of this kind, encountered in some literature, is "electronic institutions" (EI) [12]. Electronic agent institutions may be found functioning within an organization and determining social conventions [7] which are to be observed within this or other system. Social conventions adopted within a system will affect its general objectives that are translated into specific agents' goals. A system of this kind is exemplified by road traffic regulations which define the behaviors of drivers. Observance of the norms can streamline car traffic and help avoid accidents. Unless these general objectives are identified with agents' direct goals (an agent's goal may be e.g. to travel from a to b), they will support the performance of agents' goals by determining the rules for communication within the system. Software agents operating within open societies may cause a number of threats relating to the use of their knowledge and to the autonomy of their behaviors. Trust – specifically defined in the context of software agents and their interactions and communications — is therefore a concept that can largely contribute to reducing such threats in the construction of agent-based organizational support systems. In [12], the following two main types of trust are distinguished:

1. personal – which is subjective and relies on a person's beliefs, observations, reasoning, stereotypes, communication, experience; hence, this perception of trust is centered on the human;
2. impersonal – which arises from information and experience acquired from other persons, or from third-party subjects.

[12] points to three moments when agents' behaviors can be audited: **prior to task execution, in real-time, and on task completion**. In the first case, trust can decrease due to the agent's insufficient knowledge on the task being performed or on the user's intent. In the second instance, trust is built through constant monitoring of the agent's progress and performance. In the third variant, trust depends on a post factum validation of the effects of the agent's actions. The concepts being presented concerning the use of norms as an element defining an agent's role in the system and determining its behavior are closely linked to the notion of a system's knowledge which, given an adequate structure and extent, would inform an agent's actions.

The lack of standards for the model of agents' knowledge within a system employing norms to define the agents' desirable states has encouraged the authors of this chapter to propose their own model of agent' knowledge, both in terms of norms and domain knowledge, being an example of a normative knowledge meta-model in which knowledge on norms constitutes an element of domain knowledge representation. The considerations and recommendations on the design and development of multi-agent systems, knowledge discovery and the use of norms have all been incorporated to support the user's activities related to customer service at a local government office. Since it was the designers' ambition to address not just isolated problems handled by the office, but to provide citizens with the widest possible support, the application of a multi-agent solution was contemplated. It was particularly justified considering the need to adopt a comprehensive approach to problem solving and to stimulate the acquisition of experience by persons involved. The sample fragment of the multi-agent system architecture, otlined further in the chapter, emphasizes the idea of the user's involvement in defining the system's knowledge and assumes the user's participation in strengthening the multi-agent system's learning process. As a follow-up for research done, and taking account of all the considerations discussed above, the proposed multi-agent system's knowledge representation was created. Ongoing research efforts are aimed at using the OWL language to represent the knowledge of a multi-agent system being an extension of the functionality of the system presented in this chapter. Within the structure of the agents' knowledge 20 main classes of objects have been specified, including Address, Working Hours, Document, Worker, Citizen, Service, Coordinator, Contact, Payment, Legal Basis, Processing Time, Category, etc. [28].

The specification of the normative knowledge meta-model proposed in this chapter makes it possible for software agents to access knowledge on norms, rules and restrictions defined on the basis of the system's domain knowledge. To ensure that the meta-model specification is easily expandable, simple to interpret, and applicable across a range of information systems including multi-agent systems, an OWL ontology was used to build a normative meta-model of knowledge which was then employed within a hybrid multi-agent system. As a key element of the research project, it was assumed that regulatory norms would be used to determine what behaviors of agents are desirable in terms of system performance.

The proposed fragmentary structure of the normative knowledge meta-model defines the following classes of objects: Norm, Effect, Agent, Role, Recipient, Location, Restriction, Beliefs, Goals and Plans. The system knowledge models mentioned above have been used to specify the knowledge of the system described in the following sub-chapter.

20.4 Case Study – A Multi-Agent System Architecture

The introduction of electronic media to support public administration entails the use of tailor-made solutions offering intelligent guidance to at least match the level of competence that can be expected of a public officer. This challenge can be faced by systems equipped with interface agents and backed up by multi-agent approaches. A number of system modules have been defined within the multi-layer architecture of our multi-agent system to support the work of an office of public administration. The specific components (modules) of the proposed architecture, illustrated in Fig. 20.2, are described in subsequent paragraphs A through D.

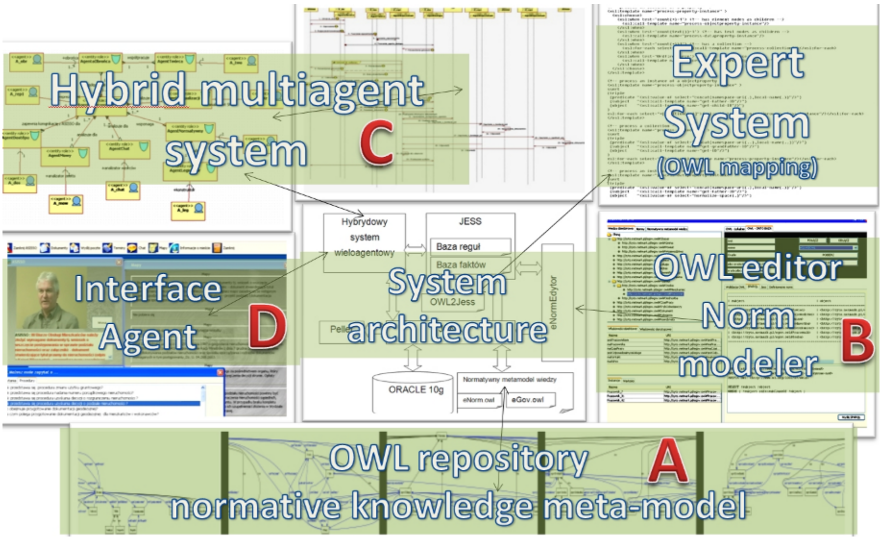

Fig. 20.2: Components of the hybrid support system for customer service at a local government office. Source: own

A. The knowledge representation and processing module

The apparent lack of insightful experiences in implementing the concepts discussed earlier in this chapter, i.e. using norms and domain knowledge to build hybrid multi-agent systems, has made it necessary to elaborate the theoretical foundations and develop a prototype solution for the management of the system's knowledge. In the proposed models, knowledge can be stored locally in OWL files or in an ORACLE 10g data base. A central element which served as the platform for the system was JENA – a semantic Web framework for Java[1]. To ensure consistency of the knowledge with the adopted model, JENA and Pellet[2] validation tools were used, both capable of reasoning based on an ontology provided.

The Jess[3] expert system was utilized to define rules, from which norms were subsequently derived. Although Jess cannot directly use knowledge saved in OWL files, a mechanism based on XSD templates was put in place to transform the knowledge into a fact base, which made inference possible and, as a result, permitted defining the rules.

B. The system knowledge definition module

Following is a list of key features required of the prototype normative knowledge meta-model editor. Namely, the module should:

1. allow storage, retrieval and validation of a model of norms and knowledge in the Oracle 10g database system in the form of OWL DL, so that the knowledge could be used by a multi-agent system;
2. allow storage, retrieval and validation of norms kept locally in the OWL DL format;
3. permit a graphical preview of the system's normative knowledge meta-model structure;
4. provide a possibility to create, edit and save norms in a normative knowledge meta-model, in keeping with the considerations discussed earlier in this paper;
5. make it possible to interactively define rules and restrictions being part of a given norm, using the knowledge stored within the normative knowledge meta-model;
6. permit operations on the system's knowledge using the SPARQL language;
7. allow a graphical preview of selected instances of any specific system norm;
8. provide a transformation mechanism to convert the system's knowledge represented in the OWL language into a fact base of the expert system;
9. enable inference using a fact base with rules and restrictions defined in an ontology.

---

[1] http://jena.sourceforge.net

[2] http://pellet.owldl.com

[3] http://herzberg.ca.sandia.gov/

C. The multi-agent system module

The first role defined in the system was that of **Agent-Protector** (agent A_obr), which was required as the entity providing access to the agent system and safeguarding observance of the norms governing the system. It is responsible for making, keeping and terminating connections with the interface agent, initiating the creation of an agent to perform the role of Agent-Representative, and supervising that agent's behaviors. Another role which was isolated was that of **Agent-Representative** (agent A_rep1). It is a hybrid agent implementation which is supported by a normative system in pursuing its goals. The role of **Agent-Legislator** (agent A_leg) is, in line with what is demanded of a normative system, supposed to analyze and update knowledge concerning norms. This sort of knowledge, defined in the system owing to a normative knowledge meta-model, is also used by an agent performing the role of **Agent-Normative** (agent A_nor), which handles queries from agents playing the role of Agent-Representative and responds to these in the context of their current tasks and goals. Other roles include **Agent-Manager** (agent A_two), which creates hybrid Agents-Representative and **Agent-Supplier** (agent A_dos), which acts as an intermediary between the interface agent and an agent performing the role of Agent-Representative. In addition, there are the roles: Agent-Chat (agent A_chat), Agent-Visualize (agent A_wiz), Agent-Speak (agent A_mow), which are responsible for automated communication with the user via different media, respectively, as well as a group of agents which handle access to external resources, e.g. a database access agent, a knowledge discovery agent, a met-data agent, etc.

For the sake of analysis of the relationships between the agents' behaviors, their knowledge, and the user support processes, the proposed system was integrated with the SAS system, specifically with the Enterprise Miner module that provides support for electronic document processing. The agents mentioned above will control the different system modules. Based on the considerations shaping agent roles within the system, elements of the architecture of the prototype hybrid multi-agent system were defined. The next step was to specify the hierarchical relationships resulting from dependencies among the types of roles which specific instances of agents perform within the system. For example a role of Representative is central, as this is the entity which controls the process of communication with the customer.

The overal structure of the system is visualized in Fig. 20.3.

D. The interface agent module

The module is supposed to separate the presentation functions of the hybrid system and to standardize the user interface so that it takes on a uniform appearance and reveals similar operating properties regardless of which of the diverse information systems handles the customer's business. The diversity of solutions applied to develop public information portals is the primary difficulty in the case of public administration offices. Problems crop up as users are forced to use several platforms at the same time, or where a system does not have an electronic interface customized

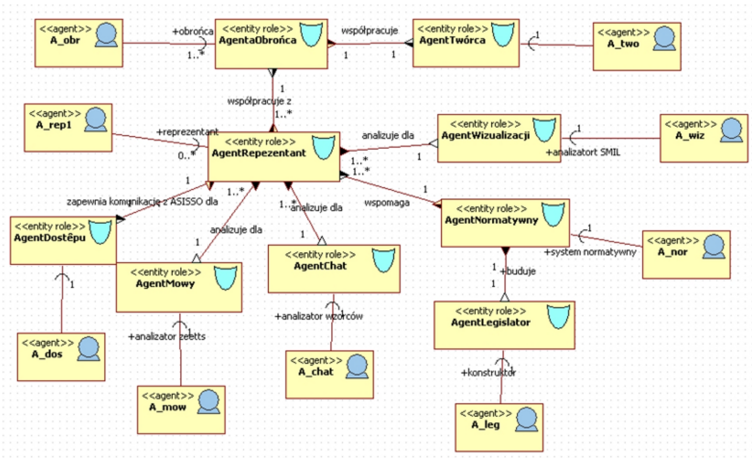

Fig. 20.3: Representation of the quantitative relationships and the types of agents operating in the system. Source: own

to cater for citizens. Within the system presented, the ASISSO agent is just an example of a “light-weight” agent which, if necessary, can be steered and controlled by a multi-agent system. The key system modules include: the advisory module which handles presentation of knowledge (base) on procedures and on frequently asked questions (viz. visualization, speech synthesis, etc.), the document processing module, the chat module, the e-mail module, the about-the-office module, the multimedia presentation module, and the update module.

The following example aims to highlight the application of a knowledge discovery module in the context of support for the process of document filling by users. To run the experiment we had prepared 135 different documents and 15 Agents-Representative. The objective was to discover relevant relationships between documents, and then work out and deliver clues for user. A simplified cycle was as follows:

1. A document selected by the user.
2. The parameters of the selected document sent to Agent-Representative along with the names of all documents available to the user.
3. A query transmitted to the multi-agent system.
4. A local matrix of relationships between fields in the user’s documents created by each Agent Representative.
5. Relationship matrices collected by the Agent Representative querying the system; output from mistrusted agents rejected, all remaining matrices processed to build up a global relationship matrix.
6. Clues derived from the global matrix and presented by the interface agent.
7. One of the relationships strengthened by the user by choosing the most relevant clue.

A sample clue generated by interpreting the matrix of relationships is shown in Fig. 20.4.

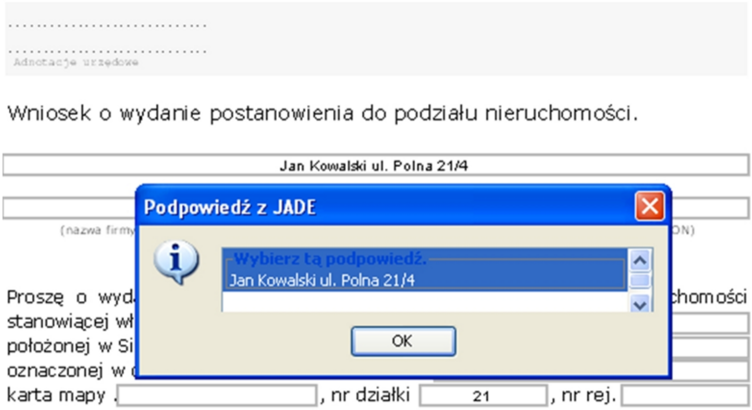

Fig. 20.4: A sample clue generated by system. Source: own

20.5 Concluding Remarks

The concepts involving the use of the OWL to represent a multi-agent system's knowledge are the starting point the first step in exploring the potential of hybrid multi-agent systems coupled with norms defining agents' desirable actions in supporting customer service at local government offices. The general structure of the knowledge model developed within this project, which has been discussed in greater detail and illustrated with a figure earlier in this paper, allows to:

1. support local citizens by supplying them with up-to-date information on what kind of services are available from a particular office;
2. support hybrid multi-agent systems by enforcing a standardized structure for knowledge storage, which facilitates interpretation of the knowledge by agents, enabling the agents to make use of it in pursuing their goals;
3. support officers by providing them with a possibility to store information on the cases processed independently and outside of the office's individual computer systems and create relationships between the stages of processing at the organizational level, i.e. in the context of the overall process of case processing.

Among others, this paper delivers findings and outcomes of research on the application of agent technologies to support citizens in filling official documents. A hybrid agent system was designed to assist the customers of a public office and to guide the process of discovering knowledge in analyzing the data fed by the customers. Once in place, the system was able to form recommendations concerning the creation of new official documents.

It is the author's belief that, in general, the paper has been successful in demonstrating that there might be a wide scope for integrating knowledge discovery methods, normative solutions and multi-agent systems.

References

1. Baazaoui Zghal, H., Faiz, S., Ben Ghezala, H.: A Framework for Data Mining Based Multi-Agent: An Application to Spatial Data. In: Proceedings of World Academy of Science, Engineering and Technology, Vol. 5 (2005)
2. Boella, G., van der Torre, L.W.N.: A Game-Theoretic Approach to Normative Multi-Agent Systems. In: Boella, G., van der Torre, L.W.N., Verhagen, H. (eds.) Normative Multi-agent Systems, 2007. Dagstuhl Seminar Proceedings 07122. Internationales Begegnungs- und Forschungszentrum fr Informatik (IBFI), Schloss Dagstuhl, Germany (2007)
3. Boella, G., van der Torre, L.W.N.: The ontological properties of social roles in multi-agent systems: Definitional dependence, powers and roles playing roles. Artificial Intelligence and Law 15, 201-221 (2007)
4. Boella, G., van der Torre, L.W.N., Verhagen, H.: Introduction to normative multiagent systems. Computational & Mathematical Organization Theory 12, 71-79 (2006)
5. Caire, P.: A normative multi-agent systems approach to the use of conviviality for digital cities. In Boella, G., van der Torre, L.W.N., Verhagen, H. (eds.) Normative Multi-agent Systems. Dagstuhl Seminar Proceedings 07122. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Germany (2007)
6. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: Cross Industry Standard Process for Data Mining (CRISP-DM) 1.0. Step-by-step data mining guide. SPSS Technical Report (2000). Available from http://www.crisp-dm.org. Cited 15 December 2008
7. Campos, J., Lpez-Snchez, M., Rodriguez-Aguilar, J.A., Esteva, M.: Formalising Situatedness and Adaptation in Electronic Institutions. In: Proceedings of The Fifth Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems COIN@AAMAS 2008, pp.103-117 (2008)
8. Castelfranchi, C.: Formalising the informal? Dynamic social order, bottom-up social control, and spontaneous normative relations. Journal of Applied Logic 1(1-2), 47-92 (2003)
9. da Silva, V.T., Hermoso R., Centeno, R.: A Hybrid Reputation Model Based on the Use of Organizations. In: Proceedings of The Fifth Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems COIN@AAMAS 2008 (2008)
10. Davidsson, P., Jacobsson, A.: Aligning Models of Normative Systems and Artificial Societies: Towards Norm-Governed Behavior in Virtual Enterprises, Dagstuhl Seminar Proceedings, `http://drops.dagstuhl.de/opus/volltexte/2007/908Cited1January2007`
11. Di Fatta, G., Fortino, G.: A Customizable Multi-Agent System for Distributed Data Mining. In: Cho, Y., Wainwright, R.L., Haddad, H., Shin, S.Y., Koo, Y.W. (eds.) Proceedings of the 2007 ACM Symposium on Applied Computing (SAC), pp. 42-47. ACM (2007)
12. Fasli, M.: Agent Technology for e-Commerce. John Wiley & Sons (2007)
13. Grizard, A., Vercouter, L., Stratulat, T., Muller, G.: A Peer-to-Peer Normative System to Achieve Social Order. In: Noriega, P., Vazquez-Salceda, J., Boella, G., Boissier, O., Dignum, V., Fornara, N., Matson, E. (eds.) Coordination, Organization, Institutions, and Norms in Agent Systems II, LNCS/LNAI 4386, Springer (2007)

14. Harbers, M., Verbrugge, R., Sierra, C., Debenham, J.: The Examination of an Information-Based Approach to Trust. Springer, Berlin Heidelberg (2008)
15. Kehagias, D., Symeonidis, A.L., Chatzidimitriou, K.C., Mitkas. P.A.: Information Agents Cooperating with Heterogeneous Data Sources for Customer-Order Management. In: Haddad, H., Omicini, A., Wainwright, R.L., Liebrock, L.M. (eds.) Proceedings of the 2004 ACM Symposium on Applied Computing (SAC), pp. 52-57. SAC (2004)
16. King, J., Linden, O.: Data Mining Isn't a Cookbook Activity. National Underwriter 106, 11-12 (2002)
17. Kirn, S., Herzog, O., Lockemann, P., Spaniol, O. (Eds.): Multiagent Engineering Theory and Applications in Enterprises. Springer (2006)
18. Klusch, M., Lodi, S., Moro, G.: Agent-Based Distributed Data Mining: The KDEC Scheme. In: Intelligent Information Agents. Springer (2003)
19. Klusch, M., Lodi, S., Moro, G.: Issues of Agent-Based Distributed Data Mining. In: Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1034-1035. AAMAS (2003)
20. Kohler, T.A., Gumerman, G.J., Reynolds, R.G.: Simulating Ancient Societies: Computer Modeling is Helping to Unravel the Archaeological Mysteries of the American Southwest. Scientific American 293, 76-83 (2005)
21. Lopez y Lopez, F.: Social Power and Norms: Impact on Agent Behaviour. PhD thesis, University of Southampton (2003)
22. Praca, I., Viamonte, M.J., Vale, Z., Ramos. C.: Agent-Based Simulation of Electronic Marketplaces with Decision Support. In: Wainwright, R.L., Haddad, H. (eds.) Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), pp. 3-7. ACM (2008)
23. Rennard, J.P.: Artificiality in Social Sciences. In: Rennard, J.P. (ed.) Handbook of Research on Nature Inspired Computing for Economics and Management, pp. 1-15. IGR, Hershey (2006)
24. Symeonidis, A.L., Mitkas, P.A.: Agent Intelligence through Data Mining. Springer Verlag, New York (2005)
25. Tesfatsion, L.: Agent-Based Computational Economics: Growing Economies from the Bottom Up, Artificial Life, Vol. 8, No. 1, 55–82 (2002)
26. Tesfatsion, L.: Agent-Based Computational Modeling and Macroeconomics. In: Colander, D. (ed.) Post Walrasian Macroeconomics: Beyond the Dynamic Stochastic General Equilibrium Model, pp. 175-202. Cambridge University Press, Cambridge (2006)
27. Tozicka, J., Rovatsos, M., Pechoucek, M.: A Framework for Agent-Based Distributed Machine Learning and Data Mining. In: Durfee, E.H., Yokoo, M., Huhns, M.N., Onn, S. (eds.): 6th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS (2007)
28. Stanek, S., Pankowska, M., Soltysik, A., Zytniewski, M.: Agent system application for geoinformation management at municipal office. Annals of Geomatics, Polish Association for Spatial Information, Vol VI, Number 2,Warsaw, 81-87
29. Vazquez-Salceda, J.: The Role of Norms and Electronic Institutions in Multi-Agent Systems Applied to Complex Domains. The HARMONIA Framework. Springer-Verlag, New York (2004)
30. Vazquez-Salceda, J., Aldewereld, H., Dignum, F.: Norms In Multiagent Systems: From Theory to Practice. International Journal of Computer Systems Science & Engineering 20, 225-236 (2005)