Online Journal of Applied Knowledge Management
A Publication of the International Institute for Applied Knowledge Management

Volume 3, Issue 1, 2015

Trust in software agent societies

Mariusz Zytniewski, *University of Economic, in Katowice, mariusz.zytniewski@ue.katowice.pl*

Mateusz Klement, *University of Economics in Katowice, mateusz.klement@ue.katowice.pl*

Abstract

Modern IT solutions, such as multi-agent systems, require the use of mechanisms that will introduce certain social elements to improve the process of communication. Such mechanisms may be trust and reputation models, which allow a very important aspect of human relations, i.e. trust, to be introduced between autonomous software agents. Models that are currently proposed usually fail to take into account openness of present systems or mobility of agents, which allows them to move across systems. According to the authors of this paper, agents from the same system should be evaluated in a different way than agents from a different multi-agent system. The concept of a trust model proposed in this paper takes into account the above mentioned factors and enables a simple evaluation of other agents depending on the system from which they come and the action they are designed to perform.

Keywords: Agent societies, agent reputation, knowledge management system, trust

Introduction

In order to increase its competitiveness, an organisation has to search for new IT solutions that are able to support its processes. Such solutions may be agent technologies (Soltysik-Piorunkiewicz, Zytniewski, 2013). Agent solutions, thanks to their features, such as autonomy, proactivity, reactivity, social properties and communication, can support various areas of an organisation - in particular, they can be used to support knowledge-based organisations as an element of knowledge management systems (Zytniewski, 2013a). In this case, software agents may be applied in the processes of gathering and codifying knowledge, processing possessed knowledge and propagating it across an organisation and its environment.

The key element of agent solutions is their autonomy, which allows agents to use their knowledge to make independent decisions based on their mechanism of artificial intelligence. This approach allows an agent to operate independently in a specific environment and enables it to perform its tasks without direct influence of a human being on its operation, but it also causes a risk that it may take actions that are unfavourable from the perspective of the system it supports. For that reason, it is necessary to search for solutions that would not limit the autonomy of agents in their operation, but at the same time ensure safety mechanisms for the system in which it resides against its possible actions. An example may be mechanisms of reputation and trust in software agents in systems, which are analysed in this paper.

The aim of this paper is to present the concept of trust in software agents and propose its model based on basic trust and social trust. The first part contains introduction to the theory of agents and the use of trust in such systems. Further, the paper presents examples of the models of trust in agent systems. The last chapter presents a model of trust based on basic trust and social trust.

Introduction to mechanisms of reputation control in software agent societies

Heterogeneity of multi-agent systems is connected with diverse functionality of agents, their organised cooperation, heterogeneity related with the possibility of building various agents representing the user in the virtual world, openness expressed in the possibility of agents moving across multi-agent systems, and dynamism resulting, among other things, from changes of cooperation rules for agents in a given multi-agent system. Research into agent solutions indicates possibility of examining their social properties, which constitute a feature of an agent system (Zytniewski, 2013b). We can distinguish two main trends in the development of multi-agent systems. The first one is agent society focus, whereas the second one - agent norm focus.

The concepts of multi-agent systems presented show that research in this area concentrates on perceiving such solutions from the angle of a social system, where cooperation among entities, their communication or roles played by them can be supported by principles or rules on the processes of communication and cooperation among agents, also in the context of their knowledge. Possibility of using such concepts in the process of building software agent societies which are linked with an organisation's IT systems and focused on supporting actions of a decision-maker makes it reasonable to undertake research into the development of autonomous entities representing a decision-maker in the virtual world.

During a review of the literature on the subject, both trust and reputation models can be encountered. In practice, both these approaches are designed to ensure that multi-agent systems have certain social properties that significantly affect the process of communication among autonomous agents, but these models are not identical to each other. Further sections will attempt to define trust and reputation in respect of software agents.

It should be noted at the beginning that there is no single universal definition of trust in multi-agent systems. However, drawing on research by Diego Gambetta for the purpose of this paper, trust in multi-agent systems can be defined as a measurable level of probability with which agent x is able to determine whether agent y will perform a specific task in a way that is satisfactory to it (i.e. agent x). It should be stressed that the probability level determined by agent x is subjective, as this agent does not have complete knowledge about how agent y will behave in the future (Gambetta, 1990).

Literature on the subject provides two main approaches to the issue of trust among software agents: cognitive approach (Castelfranchi & Falcone, 2001) and probabilistic approach (Yu & Singh, 2002).

The main assumption in the cognitive approach is the willingness to delegate tasks, where the authors assumed that agent x needs to delegate task (action) α to agent y. To do that, agent x has to evaluate its level of trust in agent y, the latter's intentions and motivation to perform the task. In their work they identified a few elements of beliefs based on which it is possible to establish the level of trust that agent x may place in agent y: competence belief, willingness belief, persistence belief and motivation belief.

The probabilistic approach does not take into account other agents' intentions, but focuses mainly on the experience gained by agent x during its interaction with agent y. The information gathered

in this way is used to predict future behaviour of an agent, and more specifically - to calculate probability of an agent behaving in a certain way in a given situation. For that reason, agents have to gather information about interactions among members of a multi-agent system. The use of the probabilistic approach seems to be more natural for open multi-agent systems, as it does not require a complicated modelling of agents' mental states.

Some models encountered in the literature on the subject are called reputation models, therefore it is necessary to define this term. Reputation (Abdul-Rahman & Hailes, 2000) means expectation that an agent will behave in a certain way based on its past behaviour and the information about it. Often, information about the reputation of an agent comes from other agents functioning in a multi-agent system and their experience from interaction with this agent. Nevertheless, some information is based on own past experience.

Both in human social relations and relations in software agent societies, we can distinguish two main sources that allow trust to be built based on the use of reputation.

1. Private information gathered during direct relations that enables building trust in an agent,
2. Reputation in a society, that is largely based on opinions of others, which also generates trust in an agent.

Trust that is built only on the basis of own (private) information is called direct trust and constitutes the basic source used to build trust. Trust that is based on both the sources mentioned above, i.e. private information and reputation in a society, is referred to as composite trust. The use of reputation may allow a given agent to be evaluated from the perspective of the rest of the society, but it also entails certain threats, resulting from, among other things, possible partiality of the other agents. It is accepted that reputation allows problems of a dynamic agent society to be overcome, but in certain situations (e.g. when society members often change or there is a huge number of mutual interactions) it may turn out useless (Burnett, 2006).

Reputation models in the theory of software agents

Some trust models are well known to users of internet platforms. An example is the trust model used by eBay service. A similar trust model is used by the most popular auction portal in Poland - Allegro. The model is based on the idea of adopting trust and reputation mechanisms known from the traditional market. In the case of Allegro, the trust model is based on a system of comments posted by the parties of every transaction. This model uses both text comments, which can be positive, neutral or negative, and graphical ones [Allegro Website, 2014]. Such a reputation-based trust model works well with people and their mutual interactions, but its implementation in a multi-agent environment is not recommended due to numerous shortcomings. The literature on the subject presents other trust models that will work much better in agent systems.

The pioneer trust model in multi-agent systems was the model proposed by Marsh in 1994 in his doctoral thesis. The author distinguished three types of trust: basic trust, general trust and situational trust. Basic trust defines the trust level of agent x (more specifically: its general capability to trust) in time t and is denoted as T_x^t assuming values from the bracket [-1; +1]. General trust refers to the trust level of agent x with respect to agent y in time t, however without

taking into account a specific situation, i.e. $T_x(y)^t$. Also in this case, the value is assumed from the bracket [-1; +1]. Situational trust refers to the trust level of agent x with respect to agent y in situation α and in time t, i.e. $T_x(y,\alpha)^t$. Here also the value is assumed from the bracket [-1; +1]. Based on appropriate calculations, agent x may decide whether to trust agent y in a given situation and interact with it: $T_x(y,\alpha) = U_x(\alpha) \times I_x(\alpha) \times \overline{T_x(y)}$, where $U_x(\alpha)$ is the utility that agent x gains from situation α, $I_x(\alpha)$ is the importance for agent x in the situation α and $\overline{T_x(y)}$ is the estimation of general trust after taking into account all information related to $T_x(y)$. [Marsh S., 1994].

A model that uses reputation to build trust is the model presented by Abdul-Rahman and Hailes (2000). This model is based on two sources from which knowledge about other agents is collected: direct experience and recommendations of other agents. Moreover, it uses four grades to evaluate trustworthiness of a specific agent: very trustworthy – vt, trustworthy – t, untrustworthy – u and very untrustworthy – vu. Individual agents store in tuples information about previous experience in the interactions with other agents in a given context: (vt, t, u ,vu). For instance, agent x may have the following information about agent y as a seller in an appropriate tuple: (0, 1, 2, 3). This means that agent x has one opinion about agent y as a trustworthy seller, two opinions about it as an untrustworthy seller and three opinions about it as a very untrustworthy seller. The value of trust is calculated by using the maximum contained in the tuple (in our example, agent x will be very untrustworthy as a seller for agent y - vu). In disputable situations, e.g. when agent x has the following information about agent y: (4, 4, 0, 0), the system will categorise agent as mostly trustworthy, i.e. U^+. In an analogous situation, when agent x has the following information: (0, 0, 4, 4), the system defines agent y as mostly untrustworthy, i.e. U^-. In any other disputable situation, the model will return a neutral result: U^0 (Pinyol & Sabater-Mir, 2013).

The trust models that have been synthetically described above belong, according to the authors, to more interesting and often encountered models in the literature on the subject. The actual list of models is much more extensive and includes many other interesting concepts. Models that are worth looking at include, among other things FIRE (Huynh, Jennings, Shadbolt, 2006), LIAR (Muller & Vercouter, 2005), Regret (Sabater & Sierra, 2001), czy Repage (Sabater-Mir, Paolucci, Conte, 2006). Unfortunately, most of these models do not take into account differences between trust of agents within one multi-agent system and trust of agents that come from different systems (this concept is included in two models: Regret and FIRE, but without general capability to trust). According to the authors of this paper, this factor should have a significant influence on how agents are perceived and affect the level of their trust as well as decision making processes that are based on trust.

Proposal of a trust model for the purpose of software agent societies

For the purpose of this publication, as well as the proposed trust model, the authors will use the definition of a software agent offered by Jacques Ferber (1999). He defines a software agent as a physical or virtual entity that is capable of operating in a specific environment and at the same time able to communicate with other agents, is guided by specific objectives, and has its own resources, limited ability to perceive its environment and provide services, and ability to

reproduce. According to Ferber, as well as the authors of this paper, one of key capabilities of a software agent should be capability of communicating with other agents.

When building a model of such a system, it is necessary to first refer to the model of operation of an agent. Software agents are created based on various concepts, commonly referred to as architectures. The literature on the subject distinguishes reactive agents, prudent agents and hybrid agents, which combine the properties of the former two concepts (they may have sensors, effectors, symbolic representation of the environment and may operate based on the BDI paradigm). For that reason, the architecture of a hybrid agent was proposed for the purpose of control and coordination of the operation of autonomous agents in a multi-agent environment (Stanek, Sroka, Paprzycki, Ganzha, 2008).

For the purpose of this research, the mechanism proposed in the work (Zytniewski, 2010) was used. The mechanism for analysing the operation of agents referring to the proactive and reactive levels of processing norms can be extended to include elements of evaluation of agents' actions. The idea behind this mechanism is that in the case of using a solution that controls the operation of an agent, the reactive layer may be activated to analyse decisions or - directly - actions of an agent. The proactive layer, which performs actions resulting from agents' tasks and their objectives, is supported by a reactive layer, which monitors and controls these actions from the perspective of the system in which the agent resides. Every action of an agent may cause the need to analyse the actions taken by the agent and its behaviour, and can be monitored by the system in the form of an agent entity in a given society.

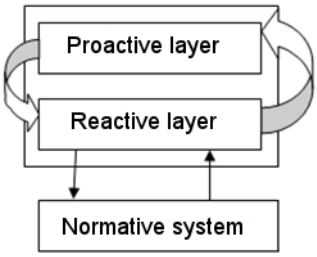

Figure 1. Architecture of a hybrid agent
Source: Zytniewski M. (2010). Mechanizmy reprezentacji wiedzy w hybrydowych systemach wieloagentowych In: Wiedza i komunikacja w innowacyjnych organizacjach. WiK 2010 Systemy ekspertowe – wczoraj, dzis i jutro, AE Katowice

An important issue is how the authors view multi-agent systems (MAS). The fundamental difference between multi-agent systems and single software agents is that in MAS the operation environment is subject to dynamic and uncontrolled changes as a result of other agents' actions. In practice, a single agent does not have (and is not likely to have) full knowledge about its environment, the agents that comprise it, or chances to solve all problems (Ebadi, 2012). In a multi-agent system, possibilities of a single agent are very limited and without implementing an appropriate trust model to support communication among agents and their cooperation they will remain such. According to the authors, an important property of modern multi-agent systems is openness, which allows the mobile agents comprising such systems to move across different platforms.

For creating multi-agent systems, dedicated platforms are used, including the open-source JADE platform (Java Agent Development Kit). After conducting a comparative analysis of available solutions that support the development of multi-agent systems, the authors rated the JADE platform as the best and most comprehensive solution (Klement & Zytniewski, 2015). The proposed model was developed specifically to be used in a multi-agent system based on the JADE platform.

The authors assumed that in agent societies, direct trust will be used to build reputation of an agent in a given agent society. When an agent interacts with another agent to perform its objectives and does not have trust in it, its decisions about its behaviour towards it may be based on the reputation of the system and it may strengthen or weaken it in the course of actions. Such approach fits the probabilistic approach, where the system learns proper behaviour without control. To meet these demands, the proposed model distinguishes **basic trust** and **social trust**, which draw on work by Marsh [Marsh S., 1994], Regret model (Sabater & Sierra, 2001) and the concept of software agent society [Zytniewski M. (2013)].

Basic trust comprises three elements: native system trust, other multi-agent systems' trust and trust of the action in which an agent has already participated. Native system trust affects capability of agent x to trust agents that come from the same system (society). This trust will be denoted as $T_x(NMAS)$, where NMAS is a native system and x is an agent. Like the other parameters in the model, it will assume values from the bracket [-1; +1]. Value -1 will mean complete distrust, and value +1 blind trust, which is not a desired value, as such a high value would mean that no other agent, irrespective of its environment or performed action, would be more trustworthy. Other systems' trust affects the capability to trust agents that come from a different multi-agent system than agent x and constitutes an important factor for agents characterised by mobility. It will be denoted as $T_x(OMAS)$, where OMAS is other multi-agent systems. Action trust is the trust of agent x in itself as a performer of a specific action (task). An agent may be a good performer of certain actions, but it may fail in the case of other actions. This trust will be denoted as $T_x(a)$, where a is an action which belongs to set A ($a \in A$). This value will be used by agents to evaluate the usefulness of a given agent in performing a specific action and decide whether to interact with it.

Social trust will comprise analogous elements but with respect to a different agent, e.g. agent y. Native system trust in agent y will be the value of the trust of agent x in agent y, assuming that both the agents come from the same multi-agent system. This trust will be denoted as $T_x(y, NMAS)$, where y is another agent. Other systems' trust is the value of the trust of agent x in agent y assuming that both the agents come from different systems. This trust will be denoted as $T_x(y, OMAS)$. Action trust in agent y will mean the trust that agent x places in agent y as a performer of a specific action. It will be denoted as $T_x(y, a)$.

We can establish the total value of trust of agent x in agent y when they both come from the same system using the following formula: $TT_x(y, NMAS) = T_x(NMAS) \times T_x(y, NMAS)$. Thus, total trust within one system is affected by two factors: capability of agent x to trust expressed as $T_x(NMAS)$, and trust that it may place in agent y, i.e. $T_x(y, NMAS)$.

Analogous calculations can be made in a situation where agents come from completely different environments. In this case, total trust in other systems is also affected by two factors: $TT_x(y, OMAS) = T_x(OMAS) \times T_x(y, OMAS)$. The use of the formula below allows agent x to

determine whether for a specific action it's worth cooperating with agent y: $TT_x(y,a) = T_x(a) - T_x(y,a)$. If $TT_x(y,a) < 0$, then agent x regards agent y as more appropriate to perform a given action a. In this case, it's worth starting cooperation, as this agent should perform the evaluated action a in a more effective way. If $TT_x(y,a) > 0$, then the cooperation may not bring expected effects. In the situation when $TT_x(y,a) = 0$, agent should determine whether it's worth starting cooperation based on the other factors constituting basic and social trust. Agent x may also have dilemma whether to interact with agent y or agent z. In such a situation, the agent should compare the value $T_x(y,a)$ with the value $T_x(z,a)$ and choose the one that is closer to 1. Agent should make calculations in accordance with the formula of total trust in a specific action.

Summary

Interest in multi-agent systems as solutions that may impact competitiveness of an organisation is systematically growing. New tools, concepts and models appear that are designed to improve these solutions and remove problems they still struggle with. The concept of the model presented in this paper does not solve all problems of multi-agent systems, but it raises the issue connected with mobility of agents. It constitutes a strong foundation for further research and improvements. In the future, it will be extended to include more elements, such as a method for evaluating interactions and knowledge conveyed by agents. The model should also take into account the level of an agent's trust in the user. All these elements will be the subject of further research, and the concept presented here will be subject to experiments.

Acknowledgement

The issues presented constitute authors' research into the aspect of modeling software agent societies in knowledge-based organizations. The project was financed from the funds of National Science Centre 2011/03/D/HS4/00782.

References

Abdul-Rahman, A., & Hailes, S. (2000). *Supporting trust in virtual communities.* In: *Proceedings of the 33rd Hawaii International Conference on System Sciences*, vol. 6. IEEE Computer Society Press.

Allegro Website (December 25, 2014). http://allegro.pl/country_pages/1/0/user_agreement.php#rule11

Burnett, Ch. (2006). *Trust Assessment and Decision-Making in Dynamic Multi-Agent Systems* (Doctoral dissertation), pp 24 – 25, Departament of Computing Sciences, University of Aberdeen.

Castelfranchi, C. & Falcone, R. (2001). *Social trust: A cognitive approach*. In: C. Castelfranchi and Y. Tan (Eds.), *Trust and Deception in Virtual Societies*, pp 55 -90. Kluwer Academic Publishers.

Ebadi, T. (2012). *Facilitating Cooperation in Multi-agent Robotic Systems* (Doctoral dissertation), pp 15 – 16. University of Otago, Dunedin, New Zeland.

Ferber, J. (1999). *Multi-Agent Systems – An Introduction to Distributed Artificial Intelligence*, Addison-Wesley Longman Publishing. Boston, USA.

Gambetta, D. (1990). *Can we trust trust?*. In D. Gambetta (Ed.), *Trust: Making and Breaking Cooperative Relations*, pp 213 – 237. Departament of Sociology, University of Oxford.

Huynh, T., Jennings, N., Shadbolt, N. (2006). *An integrated trust and reputation model for open multi-agent systems*. J AAMAS 2(13), pp 119–154

Klement, M., & Zytniewski, M. (2015). *Analiza porownawcza wybranych platform wieloagentowych.* In: W. Chmielarz, J. Kisielnicki, T. Parys (Eds.), *Informatyka 2 Przyszlosci*, pp. 88 – 100. Wydawnictwo Naukowe Wydzialu Zarzadzania Uniwersytetu Warszawskiego, Warszawa.

Muller, G., & Vercouter, L. (2005). *Decentralized monitoring of agent communications with a reputationmodel*. In: R. Falcone, K.S. Barber, J. Sabater-Mir, M.P. Singh (Eds.) *Trusting agents for trusting electronic societies, theory and applications in HCI and e-commerce*, volume 3577 of lecture notes in computer science. Springer, Berlin.

Pinyol, I., & Sabater-Mir, J. (2013). *Computational trust and reputation models for open multi-agent systems: a review*. *Artificial Intelligence Review*, vol. 40.

Sabater, J., & Sierra, C. (2001). *Regret: A reputation model for gregarious societies*. In: *Proceedings of the 4th workshop on deception, fraud and trust in agent societies*, pp 61–69, Montreal.

Sabater-Mir, J., Paolucci, M., Conte, R. (2006). *Repage: reputation and image among limited autonomous partners*. JASSS 9(2)

Soltysik-Piorunkiewicz, A., & Zytniewski, M. (2013). *Software Agent Societies for Process Management in Knowledge-Based Organization*. In: *Proceedings of the 14th European Conference on Knowledge Management*, vol. 2, pp 661-669, ACPI, UK.

Stanek, S., Sroka, H., Paprzycki, M., Ganzha, M. (2008). *Rozwój informatycznych systemów wieloagentowych w środowiskach społeczno-gospodarczych*, Placet, Warszawa.

Yu, B., & Singh, M. P. (2002). *An evidential model of distributed reputation management*. In: *Proceedings of First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, vol. 1, pp 294 – 301. ACM Press.

Zytniewski M. (2013a). *Aspects of the knowledge management system's life cycle with the use of software agents' society*. In: M. Pankowska, S. Stanek, H. Sroka (Eds.) *Cognition and Creativity Support Systems*, pp 191-201. Publishing House of the University of Economics in Katowice.

Zytniewski M. (2013b). *Rozwoj koncepcji spolecznosci agentow programowych*. In: J. Buko (Ed.) *Europejska przestrzen komunikacji elektronicznej*, pp 481-493. Zeszyty Naukowe Uniwersytetu Szczecinskiego.

Zytniewski, M. (2010). *Mechanizmy reprezentacji wiedzy w hybrydowych systemach wieloagentowych* In: *Wiedza i komunikacja w innowacyjnych organizacjach WiK 2010 Systemy ekspertowe – wczoraj, dzis i jutro*, AE Katowice

Authors' Biographies

Mariusz Żytniewski, Ph.D. is employed at the University of Economics in Katowice as lecturer on Faculty of Informatics and Communication, at Department of Informatics. He is taking part in the research into computer science, systems analysis and computer system design, management information systems, software agents and knowledge-based organizations.

Mateusz Klement, MSc. is a graduate student and an assistant lecturer at the University of Economics in Katowice on Faculty of Informatics and Communication. As an employee of the Department of Informatics he is taking part in the research into computer science, with particular emphasis on software agents.