Uniwersytet Ekonomiczny w Katowicach

Studia Ekonomiczne. Zeszyty Naukowe
Uniwersytetu Ekonomicznego w Katowicach
ISSN 2083-8611 Nr 355 · 2018

Informatyka i Ekonometria 13

Mariusz Żytniewski

University of Economics in Katowice
Faculty of Informatics and Communication
Department of Informatics
mariusz.zytniewski@ue.katowice.pl

USE OF A REPUTATION INDICATOR IN ASSESSING THE PERFORMANCE OF AN AUTOPOIETIC SYSTEM'S COMPONENTS[1]

Summary: Distributed IT systems require the application of mechanisms that support the monitoring and coordination of their operation. Such systems, characterised by self-organization and self-adaptation, can be perceived in terms of autopoietic systems, which are capable of self-production and defining the relationships among the system components. Self-organization as a bottom-up mechanism refers to the aspect of communication among entities of a structure for achieving certain outcomes. Adaptation, on the other hand, is a top-down mechanism initiated by a system's control mechanisms which show how the individual entities should behave. An example of such a system is an organization knowledge management system supported by agent technologies. Such systems, equipped with autonomous agents, allow to model their self-organization and adaptability in response to changing environmental conditions. The aim of this paper is to analyse the concept of autopoiesis and to propose a model for assessing the reputation of autopoietic elements and regulating the behaviour of agents.

Keywords: software agent, autopoiesis, agent societies, knowledge management.

JEL Classification: D83.

Introduction

The purpose of building distributed systems is to diversify their functionality, which is distributed between subsystems that constitute their components.

[1] Note: This paper is an extended version of my previous paper: *Autopoiesis of knowledge management systems supported by software agent societies* [in:] Refereed Paper Proceedings – KM Conference 2017, A Publication of the International Institute for Applied Knowledge Management, Novo Mesto, Slovenia, 2017.

This requires not only separating the functions of the system into subsystems, but also defining the relations between its components. These relations can rely on interaction relating to the communication between subsystems. The relations built in the system and the communication among its components are important for the proper operation of distributed systems focused on self-organization and self-adaptation. The key aspect in such systems is creation of one- or multilevel structures based on defined relations, and the exchange of data, information and knowledge about the environment in which they reside. As a result, when defining distributed systems one must not only indicate their structure, but also their dynamism by defining the relations between the components of such systems. In order to analyse the dynamism of a distributed system which is focused on the autonomy of its elements, relations among its components and communication, it is necessary to use mechanisms designed to assess the activities undertaken by the components. One of the approaches that can be adopted when analysing the behaviour of distributed systems is to view these systems in terms of autopoietic systems which are focused not only on self-organisation and adaptive activities.

Maturana and Varela [1980, p. 78-79, see also: 135] defined an autopoietic machine as "organized (defined as a unity) as a network of processes of production (transformation and destruction) of components that produces the components which: (i) through their interactions and transformations continuously regenerate and realize the network of processes (relations) that produced them; and (ii) constitute it (the machine) as a concrete unity in the space in which they (the components) exist by specifying the topological domain of its realization as such a network".

Thus, autopoiesis is the ability to organize a system, connected with the processes it realizes which involve production of the system's elements, changes in these elements and removal of unnecessary elements. It is extension of a system's self-organization, which only assumes modification of the relations among the existing elements/components without any interference in the scope of the elements that exist in the system being analysed.

N. Fernandez, C. Maldonado and C. Gershenson [2013] analysed autopoiesies in terms of system complexity. C. Gershenson [2015, p. 870] argued that: "autopoiesis considers systems as self-producing not in terms of their physical components, but in terms of their organization, which can be measured in terms of information and complexity". Thus, they pointed out that autopoiesis cannot be considered only in terms of self-production of system elements, but it must also refer to their organisation. The theories discussed further in the paper will concern the analysis of the relations that are built in an autopoietic system.

A self-organized system is composed of many locally functioning and interacting components [Kasinger et al., 2010]. Local functions describe the relations among the different elements of a system. This is very important, as it allows a given relation to be described qualitatively, showing how it is perceived in the system. The measure of reputation as proposed below is an example of such a local function. From the outside, the system may seem complicated, it is controlled by rules that define the interaction of subsystems in a quite simple way. This is due to defined relations between system components, which must be ordered and organized. The interaction of components allows to see the synergistic effect in such systems [Żytniewski, 2010]. Distribution of system functions into its subsystems supports its analysis and changes. They relate to each of its components, which makes them easier to implement and control.

Self-organization requires communication. Communication of system components is, however, the basis for the construction of social systems [Paetau, 1996]. The basis of social systems is defined relations between its components that are established by communication. Communication of system components is what differentiates social systems from technical systems. A technical system, which may be implemented as part of an organization, constitutes a bounded totality. It is implemented and utilized. A social system, on the other hand, assumes that changes within its structure and function result from a change in the relation of its components that have a dynamic nature. Examples include normative systems, which can be applied in social systems.

The author's previous research showed that multi-agent systems can be considered in terms of social systems, especially assuming their openness or partial openness, autonomy, decentralization, local view of the system, security connected with the protection of the system's borders, and communication. However, the theory of software agent society does not address the issues of autopoiesis, which the author thinks is vital, as it extends the capabilities of such systems and can prove useful in the process of knowledge processing by such systems.

Definition of the standards and principles of the operation of a system may lead to new relations between the elements of the system as a result of the changing standards and principles. An example of such a mechanism is the model proposed further in the paper. It may also cause removal of a fragment/component of the system, e.g., a software agent from the society. This enforces change in the relations between its elements.

The present article focuses on the issues of self-organization and adaptability associated with the concept of autopoietic systems and automorphosis

[Yolles, 2006], considered as an element of multi-agent systems and knowledge management. The aim of this paper is to analyse the concept of autopoiesis and to propose a model for assessing the reputation of autopoietic elements and for regulating the behaviours in the autopoietic system. Chapter 1 will present the theory of autopoiesis analyzed from the angle of knowledge management systems supported by agent technologies. Chapter 2 will feature a proposal for a model of an autopoietic agent system supported by mechanisms regulating agent behaviour. Chapter 3 will present an example of the operation of a software solution developed.

1. Autopoiesis in a software agent society against the background of knowledge management processes

Social systems considered in the context of technical solutions can be perceived as autopoietic solutions, i.e. [Letelier, Marin, Mpodozis, 2002; Bourgine, Stewart, 2004; Razeto-Barry, 2012]. Furthermore, "a system is autopoietic, if it is able to reproduce itself as an autonomous and self-organising unit only by interaction of the internal elements of the system" [Paetau, 1996, p. 5]. The key element in this definition is reproduction (assuming the change in the structure of a system without changing its organization). This view of autopoiesis seems incomplete, as it does not assume the improvement of a system by changing its organisation, but only reproduction of the existing elements. This is due to the fact that although autopoiesis includes the aspect of reproduction of system components, reproduction of a system's existing components may result in new relations being revealed among its elements. As indicated by Thannhuber, Tseng and Bullinger [2001], autopoiesis is a cycle in which a system-executed process defines the structure of the system being created. The system structure is determined by possible self-organization of an autopoietic system and impacts the execution of the process itself.

This process can be viewed in two ways. In the first approach, the system's self-organization operations may trigger the mechanism of its autopoiesis. As a result, during the process of organizing its components, it may become necessary to extend the local society of the system's components to include additional elements or to remove the existing ones. Also, the triggering of the autopoiesis mechanism connected with the appearance of new system components may cause self-organization operations, if the qualitative parameters of the new component show its advantage over the already existing elements. In both these cases, an important element is the earlier-indicated local functions. Such a function will be proposed further in the paper.

It should be mentioned that self-organization is often mistakenly perceived as the concept of self-adaptation. The difference between the two approaches is related to the process of structuring a group of subsystems. In the case of autoadaptation, we are dealing with a top-down approach [Cheng et al., 2009].

In such cases, self-organizing and autopoietic activities of the system are initiated top-down through the system control mechanism. However, for that to be possible, the mechanism must possess the full knowledge about the current structure of the system it is affecting.

Autopoiesis can also be considered in terms of social systems. A knowledge management system shares a range of characteristics with an autopoietic system. Within accepted assumptions, T.W. Jackson indicated that a learning organization could be perceived as an autopoietic system. At the same time, he pointed out a range of features that an entity being considered should have to be an autopoietic system [Jackson, 2007]:

1) entity must have a boundary,
2) entity must have distinct components of a system,
3) components must be capable of satisfying relations that determine interactions and transformations – the system is made up of the interactions of its parts,
4) components creating boundaries must do so as a result of interactions with other components of the system,
5) components of the boundary must be produced from inside the system,
6) all other components must be produced from inside the system.

The above-defined characteristics of an autopoietic system indicate functional dispersion of such a system resulting from its division into components, necessity of modelling relationships between its elements and significance of defining boundaries between components understood as sets of rules and regulations. What's important, these characteristics also apply to the earlier-discussed technical systems.

Maturana and Valera [1980] point out that an autopoietic system should have the following characteristics: autonomy, individuality, organizational closure and self-specification of boundaries. These characteristics can be examined at the level of the whole system or its elements. In the latter case, these characteristics can also be noticed in the concept of software agent [Żytniewski, Klement, 2015]. A software agent society has defined boundaries in the form of the society within which agents reside {1}. The second characteristic shows that the system is divided into separate components in the form of agents {2}. Such a distinct component can be an agent or a group of agents comprising an agent community. Then, the autopoiesis process will involve not only the aspect of reproduction, but it must also involve the relations that already exist in the community.

The third assumption is that the components interact with each other based on defined relationships {3}, which has already been pointed out earlier. From the perspective of the fourth characteristic {4), it is necessary to define additional mechanism which refers to the earlier-indicated local function describing the behaviour of components. Then, it is possible to define the operating principles of such a system {4}.

The last two characteristics {5} and {6} indicate a strong link between the concept of autopoietic systems considered here in terms of software agent societies and knowledge management systems. Assuming that a software agent society is created as an element of a knowledge management system, the mechanisms for defining the principles governing the operation of agent societies should be built based on the knowledge defined in an organization's knowledge management system {5} and refer to the knowledge bases defined in an organization {6}. As a result, the interface of software agent societies and knowledge management systems in the autopoietic approach presented is the aspect of the sharing of organizational knowledge and rules and regulations governing an organization by an agent-based system.

Modelling of a software agent society as an element of a knowledge management system requires addressing a range of issues. The first of these is openness and autonomy. Multi-agent systems, and in particular a society of software agents, must adapt their structure or behaviour to the purposes of a given system. These purposes can be time-varying, which means that the state of agents, their knowledge and relations that they build will also change. As a result, achievement of the same objective may vary at two different points of time and bring different effects. The second problem is the lack of observation of the entire system by agents and the lack of ability to control the activities of other agents. As a result, it is not possible to optimally control such a society, but only to search for a local optimum associated with a particular action of the agent. One must be aware that software agents act in a certain time horizon. Even the mechanism of a multi-agent platform, which supervises the activities of agents, does not complete the knowledge about agents, because the time it takes to change the state of an agent, its knowledge, generate a message and receive it can result in subsequent events. The activities of agents should be monitored. The supervising component is the multi-agent platform or agents themselves. The problem that arises when using a monitoring and control mechanism is its failure rate. In the case of failure of the multi-agent platform, the operation of agents will not be possible, because the platform itself provides the environment in which agents are located. The situation is different in the case of a single agent unit. If it ever fails, the society will continue to function but without the possibility to use the services that it offers.

In the case of an agent society, self-organization can be realized through the mechanisms of a multi-agent platform or by direct interaction of agents. The first type of self-organization is associated with the specificity of agents treated as software entities. Agents wishing to reside in a given environment are dependent on the prevailing rules imposed by the environment in which they are located. Such an environment may be the operating system or multi-agent platform. These restrictions affect the aspect of controlling agents and their behaviour. The multi-agent platform can impose rules prevailing in a given society or provide helpful information, used by agents in interactions with other individuals. In this case, information from the platform can be seen as the impulse which influences the agent society. As a consequence, the agent society starts the adaptation process.

The second type of self-organization is related to the interactions between agents. Agents observing individual agent units have the ability to collect information about them; e.g. the lack of cooperation on the part of the agent may cause its exclusion from the society. This kind of self-organization is related to the Peer-to-Peer communication or interaction of robots.

In order to analyse autopoiesis in the context of software societies that can be used in knowledge management systems, it is necessary to define the typology of software agents and the criteria indicating when a given society represents a specific type.

Referring to the indicated types of societies, it was pointed out [Sayama, 2014] that it is legitimate to refer to five main elements that define the society. These are: the agent's state, its observations, taken actions, the function of observing the agent and the function of changes in the agent. The agent's state refers to the current state of the agent. The agent can have multiple pre-defined states it can assume in specific situations. The agent's observations constitute its knowledge resource and can include the knowledge about its environment in the form of the society in which it resides or the knowledge about the tasks it performs. Actions refer to activities in which the agent engages or may engage. They may affect its relations with its environment and refer to the local function describing its relations with its environment.

Based on that it is possible to specify four types of agent collectives [Sayama, 2014]:

– Homogeneous collectives – this approach assumes that the behaviour of specific agents is determined by their observations with reference to the environment. The agent cannot determine its state. It only responds to stimuli from the environment and is considered in terms of a reactive system. The key aspect of the construction of this arrangement is the function that transforms the stimuli from the environment to the behaviour of the agent.
– Heterogeneous collectives – this type presupposes the existence of a mechanism defining the current state of the agent, changing under the influence of the observation concerning functions defining the behaviour of the agent. As a result, between successive iterations resulting from the change of time, the agent is able to process information about itself and about the changes of the environment in which it is located. This approach does not imply changes in the agent's state in subsequent iterations.
– Heterogeneous collectives with dynamic differentiation/re-differentiation – this is extension of the previous approach. The agent analyzes its environment and takes action based on the collected knowledge. In addition, it makes continuous changes to its state (something that was not present in the previous approach).
– Heterogeneous collectives with dynamic differentiation/re-differentiation and local information sharing – this approach assumes the possibility of sharing information between agents about their states and observations. In addition, the assumptions from previous approaches are realized.

The typology indicated is hierarchical in character. Each subsequent type of society has the characteristics of the previous one, therefore, it is appropriate to make an attempt to build a model that has the features of the last type of society, as with the use of certain simplifications it will be able to be used in other types.

The model that will be proposed in the next section provides specifications of a software agent society in line with the above-defined concept of "Heterogeneous collectives with dynamic differentiation/re-differentiation and local information sharing" [Sayama, 2014, p. 2], which has been extended with the reputation elements proposed in the article [Żytniewski, Klement, 2015]. The practical elements shown in this article were defined by using the JADE platform extensions developed by the author [Żytniewski, 2017].

2. Model proposition

Let $m \in N$, where m is the number of agents, and $l \in N$, where l is the number of agents' actions (N is the set of natural numbers). D signifies a set of elements d. Upper case indicates the type of this element, e.g. A is an agent, O – an observation, S – a state, RA – reputation of an action, RT – reputation of a task. Let (1):

$$D_{k_j}^{(A)} = \{d_{1_{k_j}}^{(A)}, d_{2_{k_j}}^{(A)}, ..., d_{x_{k_j}}^{(A)}\} \tag{1}$$

where $k \in \{1, 2, \dots, m\}$, $j \in \{1, 2, \dots, l\}$ and $D_{k_j}^{(A)}$ is a set of behaviours of agents' k-th concerning j-th actions which constitute an element of the entire society of agents.

Let $m \in N$ be the number of agents, $t \in N$ be the number of agents observations, D signifies a set of elements, (O) represents an observation context and (2):

$$D_{k_i}^{(O)} = \{d_{1_{k_i}}^{(O)}, d_{2_{k_i}}^{(O)}, ..., d_{y_{k_i}}^{(O)}\} \tag{2}$$

where $k \in \{1, 2, \dots, m\}$, $i \in \{1, 2, \dots, t\}$and $D_{k_i}^{(O)}$ is a set of expected behaviours for the k-th agent concerning the i-th observation of the agent. As a result, the approach proposed on the "homogeneous collectives" can be described as (3):

$$d_{y_{k_i}}^{(O)\prime} = F_t(d_{x_{k_j}}^{(A)}) \tag{3}$$

where F_t is a function of observing agent (based on [Sayama, 2014]).

Let $m \in N$ be the number of agents, $h \in N$ be the number of agents states and (4):

$$D_{k_l}^{(S)} = \{d_{1_{k_l}}^{(S)}, d_{2_{k_l}}^{(S)}, ..., d_{z_{k_l}}^{(S)}\} \tag{4}$$

where $k \in \{1, 2, \dots, m\}$, $l \in \{1, 2, \dots, h\}$and $D_{k_l}^{(S)}$ is a set of expected behaviours for the k-th agent concerning the l-th state of the agent. By defining a set of states of an agent in accordance with the "Heterogeneous collectives" approach, the function enabling the transformation of the state and action of the agent in the new action will be defined as (5):

$$d_{y_{k_i}}^{(O)\prime} = F_t(d_{x_{k_j}}^{(A)}, d_{z_{k_l}}^{(S)}) \tag{5}$$

where F_t is a new function of observing agent (based on [Sayama, 2014]).

In the third approach “heterogeneous collectives with dynamic differentiation/re-differentiation”, one can define another function that makes it possible to change the state of an agent defined as (6):

$$d_{z}{}_{k_h}^{(S)\prime} = G_t(d_{x}{}_{k_j}^{(A)}, d_{z}{}_{k_l}^{(S)}) \tag{6}$$

where G_t a function of changes in the agent state (based on [Sayama, 2014]).

However, functions modelled in this way do not take into account the issue of self-organization resulting from the aspect of building a society of agents, because they do not indicate the mechanism that should define the change of agents’ state. As a result, the proposed model of self-organization needs to be developed in the context of the mechanisms which can dominate in the society of agents. One of the mechanisms indicated in the literature is the use of trust and reputation of agents [Żytniewski, Klement, 2015]. The reputation of agents will be built at the lowest level of the structure, i.e. an agent’s reputation in the society as a performer of a specific action. Reputation of the action will be built, based on the feedback from other agents that constitute their average. Based on (1), we can specify (7):

$$D_{k_j}^{(RA)} = \{d_{1_{k_j}}^{(RA)}, d_{2_{k_j}}^{(RA)}, ..., d_{y}{}_{k_j}^{(RA)}\} \tag{7}$$

where $D_{kj}(RA)$ is a set of reputation for agent k-th concerning j-th action (behaviours) in the entire society of agents. Let (8):

$$\mid D_{k_j}^{(RA)} \mid =: y_{k_j} \tag{8}$$

where $\forall k \in \{1, 2, \ldots, m\}$, $\exists j \in \{1, 2, \ldots, 1\}$, $y_{kj} > 0$. A set of indicators y_{kj} of reputation located in the society of agents must be greater than zero, so that one could determine the reputation of the agent, in their absence the value of reputation is set to 0. To maintain such changes we need to specify function G_{RA} defined as (9):

$$d_{y}{}_{k_j}^{(RA)\prime} = G_{RA}(d_{x}{}_{k_j}^{(A)}, d_{z}{}_{k_l}^{(S)}, d_{y}{}_{k_j}^{(RA)}) \tag{9}$$

Function G_{RA} allows the system to specify a new agent reputation for a specific agent’s action. As a result, the established indicator concerning reputation of agents’ actions (taken in a given society) is expressed by the formula (10):

$$RA_k = \frac{\sum_{y=1}^{N} d_{y}{}_{k_j}^{(RA)}}{y_{k_j}} \tag{10}$$

The use of such an indicator requires the use of a mechanism to assess actions taken by agents located in the multi-agent platform, where each action is assessed. Another indicator is the indicator concerning the reputation of a task's performance. Agents chosen to perform a given business process are assigned specific tasks. Each task requires an agent to perform a set of actions. One can, therefore, specify that a subset of a given set of indicators is a set concerning a specific task which the agent has previously performed (11):

$$D_{k_j}^{(RT)} = \{d_{1_{k_j}}^{(RA)}, d_{2_{k_j}}^{(RA)}, ..., d_{\xi_{k_j}}^{(RA)}, d_{(\xi+1)_{k_j}}^{(RA)}, ..., d_{(\mu)_{k_j}}^{(RA)}\} \tag{11}$$

where $d_{1_{k_j}}^{(RA)}, d_{2_{k_j}}^{(RA)}, ..., d_{\xi_{k_j}}^{(RA)}$ is a set of indicators concerning the reputation of actions related to a particular task. Then, similarly to the formula (10), the reputation of a given task will be defined by the formula (9), (10) and (11):

$$RT_k = \frac{\sum_{\xi=1}^{N} d_{\xi_{k_j}}^{(RT)}}{\xi_{k_j}} \tag{12}$$

The indicators concerning reputation for the *RPk* process of the *k*-th agent and the overall indicator concerning reputation are calculated analogously. In the case of the last indicator concerning general reputation, if it concerns only one society of agents, its value will be equal to the indicator concerning the reputation of tasks or actions depending on the adopted assumptions.

The indicators *RAk*, *RTk* and *RPk* assume values from the range <0,1>. Value 0 means very low reputation of the agent, which indicates a lack of trust in its activities as part of the actions, tasks and the process in which it participates. Value 1 means high reputation of the agent with regard to its actions, tasks and processes. What's important, a high value of one of the indicators does not have to entail a high value of the other ones. In some cases, an agent executing tasks as part of a given process will have a high *RP* indicator and a low *RT* or *RA* indicator. This may be due to the fact that the agent does not work correctly only with a specific task or action, while correctly executing the other tasks or actions connected with the performance of a given process. Thanks to the gradation of trust indicators, such a situation can be detected. As a result, over a longer period certain actions or tasks can be assigned to other agents.

The indicated model allows for evaluation of purposeful behaviour in a given society based on the mechanism of reputation. This model makes it possible to analyze the actions of individuals and gives the system an opportunity to self-organize and adapt. According to the proposed model, the knowledge on the agents' reputation kept by the society enables the selection of agents to undertake actions, tasks and processes in which they may participate. The multi-agent JADE platform does not have this functionality, which is why the next chapter will concentrate on the developed software solution elements implementing the specified model.

3. System example

For the realization of agents society, it was necessary to develop an adaptation mechanism and a mechanism to control the behaviour of agents. The latter was developed by extending the mechanism of agents behaviour through its ability to monitor and evaluate its activities, expanding the defined Behaviour class. The agents created on the platform implement specific actions induced by the mechanism of an agent's artificial intelligence. This means that every behaviour of the agent is analyzed and memorized by the multi-agent platform.

The second key element of the society is the mechanism of society adaptation. According to the concept of application of a society of software agents to support the activities of the organization, the structure of an agent-based society will be subject to changes by indicating its responsibilities. In the presented example, the society's task is to evaluate business processes that need to be performed and dynamically adapt its structure in order to achieve them. The JADE platform does not have this functionality, and thus, it was necessary to extend its mechanisms by adding new agents to the platform.

The defined tasks of a business process undergo decomposition into actions of the society of agents. Then, they are analyzed by the process agent, which determines whether they can be realized with the current configuration of agents. On the basis of reputation indicators, the agents are invited to the selected society. The decision to join the society comes out of social confidence of the agent in relation to other agents in the society. After forming the society, the agents have to perform tasks.

The solution proposed is consistent with the presented concept of using the concept of software agent societies as a solution designed to support a knowledge management system. An agent system, during internalisation of its parameters, receives knowledge on the rules, relations and regulations connected with the business process being performed. On this basis, in accordance with the cycle of the operation of an autopoietic system, it performs the business process using self-organization mechanisms. As a result, new knowledge is provided to the knowledge management system and can be used in subsequent iterations of its operation.

The example process consists of three tasks. First, the selected agent saves the document (task 1), then the information on saving the document is recorded in the database (task 2), and the user is informed of this action (task 3). For each task, two agent actions need to be performed. Based on the prepared simulator, the tasks were assigned to a set of three agents, each of which can perform a selected task. Their implementation is monitored by the control system. After completing the tasks, the society is dissolved. This example attempts to implement the following business process (Fig. 1).

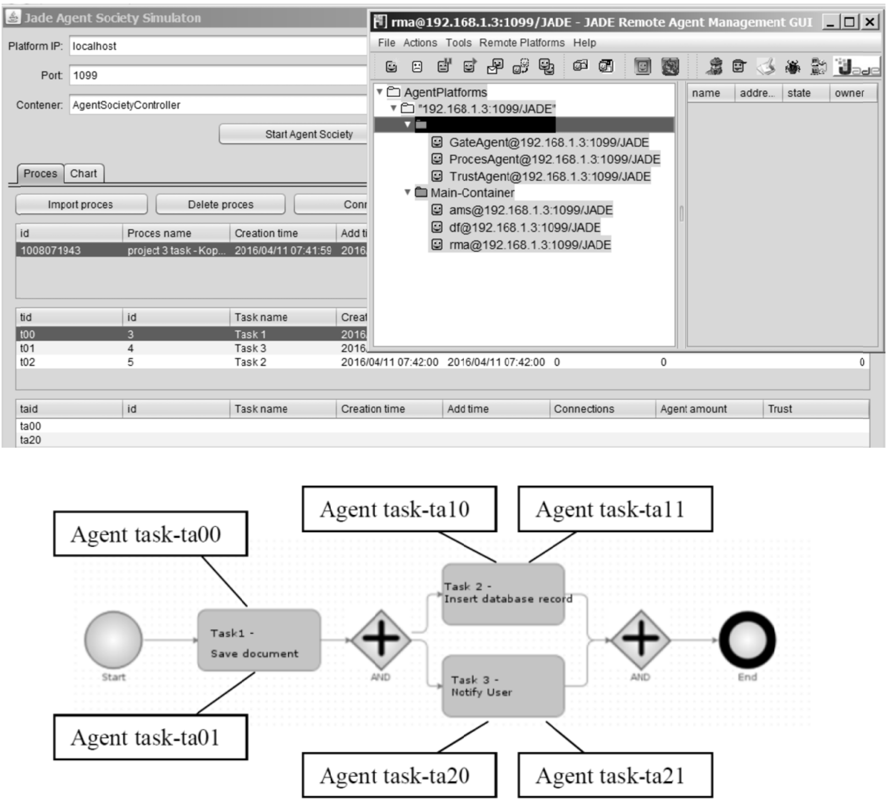

Fig. 1. Agent Society Simulator interface

Source: Own research.

Table 1 shows a set of initial parameters for a single process, assuming a cold start. In this approach, only with the result of the activities of agents it is possible to indicate the value at a given point of time. The following figures will present the changes in the value of reputation for the individual agents and their actions.

Table 1. Initial parameters of simulation

Level/Type	Reputation of agent n (initial)	Possible agent	Success probability
Action ta00	0,5	ServiceAgent01	0.3
Action ta01	0,5	ServiceAgent01	0.7
Action ta10	0,5	ServiceAgent02	0.7
Action ta11	0,5	ServiceAgent02	0.7
Action ta20	0,5	ServiceAgent03	0.7
Action ta21	0,5	ServiceAgent03	0.7

Source: Own research.

It can be stated that some agents have high level of success probability, whereas other entities demonstrate low level. It has been assumed that the simulation will cover 10 iterations of the indicated process. The effect of the developed simulator is demonstrated in Figure 2.

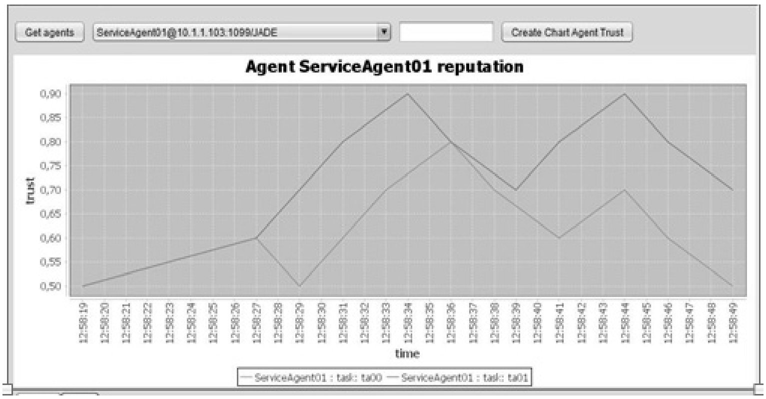

Fig. 2. The effects of society activities

Source: Own research.

As shown in Figure 2 – based on equation (10), (12) – the correct execution of actions arising from the initiated process resulted in enhancement of reputation of the respective agent. In the case of ServiceAgent01, the likelihood of correct execution of action t00 (red color) was very low. In a situation like that, the reputation mechanism in the system should detect that and with time indicate a decrease in the agent's reputation (RA). It did happen. After the simulation, the value of the agent's RA reputation with regard to action t00 decreased to the baseline 0.5 in contrast to action t01 (blue color), where its reputation remained at 0.7. Thus, the developed model and the mechanism programmed on JADE platform worked correctly.

The developed model may find application in building autopoietic systems. In the case of autopoiesis indicated in the paper, the developed local function can indicate which of the system components in the form of agents should be subject to reproduction. Agents with low reputation in the system can be omitted due to their small contribution to the correct operation of the system. As was indicated in the paper, autopoiesis does not involve only the self-production of the system, but is also connected with the relations among the system components. The developed local function allows to determine agents' reputations in the context of the business process in which they participate. In order to build a correctly functioning agent community, assigned to a given business process, it is necessary to analyse not only the reputation indicator at the level of the agent's action, but also the process in which it participates. Then, autopoiesis will involve reproduction of not one agent, but the whole group responsible for a given business process and the relations in which they enter during performing the process. This aspect was presented in a different paper of the author: "Gossip and ostracism in modelling automorphosis of multi-agent systems" [Żytniewski, 2017].

Conclusions

This paper addresses the subject of building autopoietic systems in the context of the theory of software agent societies. The issues of autopoiesis of systems and the characteristics they should demonstrate as discussed in the paper show that software agent societies can be considered in terms of autopoietic systems, as they provide mechanisms connected with self-organisation and adaptability capabilities of a system.

One of the indicated elements of the theory of autopoiesis is local functions that enable assessment of the performance of system components. The model and reputation indicators proposed in the paper can be used in autopoietic systems to assess how their components work. This allows for triggering autopoietic mechanisms connected with the development of new system components based on new system elements. The gradation of indicators enables detailed assessment of the performance of a system component, not only with regard to single actions, but also the whole process in which it participates, which is connected with the relations that are built as part of a group of co-operating agents. As a result, the components indicated in the definitions of autopoiesis can be considered not only in the context of single agents, but also whole groups of agents with certain relations.

The concept of using a software agent society as an element of knowledge management system has a range of advantages. The first one is the possibility of using individual agents' knowledge about the operation of other units. Such knowledge may come from the information possessed by an agent as a result of its presence in another society. In the adopted model, reputation has a local character, which refers to a specific multi-agent platform. The second advantage is shortening of the time during which an agent will negatively affect a specific society and its operations. The third one is speeding up of the moment when an agent ceases to be a part of a given multi-agent platform. As a result, it releases its resources, contributing to a general improvement of the performance of the operation of the platform and agents. Further research of the author will address the issues of using the model in building autopoietic systems to support selected business processes. The research will involve the evaluation of the impact of the model on the operation of the system.

References

Bourgine P., Stewart J. (2004), *Autopoiesis and Cognition*, "Artif Life", Vol. 10(3), p. 327-345.

Cheng B.H.C., Lemos de R., Giese H., Inverardi P., Magee J., Andersson J., Becker B., Bencomo N., Brun Y., Cukic B., Marzo Serugendo di G., Dustdar S., Finkelstein A., Gacek C., Geihs K., Grassi V., Karsai G., Kienle H.M., Kramer J., Litoiu M., Malek S., Mirandola R., Müller H.A., Park S., Shaw M., Tichy M., Tivoli M., Weyns D., Whittle J. (2009), *Software Engineering for Self-adaptive Systems: A Research Roadmap* [in:] B.H.C. Cheng, R. de Lemos, P. Inverardi, J. Magee (eds.), *Software Engineering for Self-Adaptive Systems*, Vol. 5525, Springer-Verlag, Berlin–Heidelberg, p. 1-26.

Fernandez N., Maldonado C., Gershenson C. (2013), *Chapter 2: Information Measures of Complexity, Emergence, Self-organization, Homeostasis, and Autopoiesis* [in:] M. Prokopenko (ed.), *Guided Self-organization: Inception*, Springer-Verlag, Berlin–Heidelberg, p. 19-51.

Gershenson C. (2015), *Requisite Variety, Autopoiesis, and Self-organization*, "Kybernetes", Vol. 44, p. 866-873.

Jackson T.W. (2007), *Applying Autopoiesis to Knowledge Management in Organizations*, "Journal of Knowledge Management", Vol. 11(3), p. 78-91.

Kasinger H., Bauer B., Denzinger J., Holvoet T. (2010), *Adapting Environment Mediated Self Organizing Emergent Systems by Exception Rules*, Proceedings of the Second International Workshop on Self-Organizing Architectures, ACM, p. 35-42.

Letelier J.C., Marin G., Mpodozis J. (2002), *Computing with Autopoietic Systems* [in:] R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, F. Hoffmann (eds.), *Soft Computing and Industry*, Springer, London, p. 67-80.

Maturana H.R., Varela F.J. (1980), *Autopoiesis and Cognition: The Realization of the Living*, Boston Studies in the Philosophy and History of Science, Vol. 42, Springer, Netherlands.

Paetau M. (1996), *Self-organization of Social Systems – A New Challenge for Organization Sciences and Systems Design*, "ACM SIGOIS Bulletin", Vol. 17(1), p. 4-6.

Razeto-Barry P. (2012), *Autopoiesis 40 Years Later. A Review and a Reformulation*, "Origins of Life and Evolution of Biospheres", Vol. 42(6), p. 543-567.

Sayama H. (2014), *Four Classes of Morphogenetic Collective Systems* [in:] H. Sayama, J. Rieffel, S. Risi, R. Doursat, H. Lipson (eds.), Proceedings of the 14th International Conference on the Synthesis and Simulation of Living Systems, MIT Press, US, p. 320-327.

Thannhuber M., Tseng M., Bullinger H. (2001), *An Autopoietic Approach for Building Knowledge Management Systems in Manufacturing Enterprises*, "CIRP Annals – Manufacturing Technology", Vol. 50(1), p. 313-318.

Yolles M. (2006), *Organizations as Complex Systems: An Introduction to Knowledge Cybernetics*, Information Age Publishing, Greenwich, UK.

Żytniewski M. (2010), *Perfecting Synergic Effect in Hybrid Multi-agent Systems / Doskonalenie efektu synergicznego w hybrydowych systemach wieloagentowych* [in:] J. Korczak (ed.), *Business Intelligence and Data Mining*, Publishing House of the University of Economics, Wrocław, Poland, p. 114-127.

Żytniewski M. (2017), *Gossip and Ostracism in Modelling Automorphosis of Multiagent Systems* [in:] J. Goluchowski, M. Pańkowska, H. Linger, C. Barry, M. Lang, C. Schneider (eds.), *Complexity in Information Systems Development*, Lecture Notes in Information Systems and Organisation, Vol. 22, Springer, Switzerland, p. 135-150.

Żytniewski M., Klement M. (2015), *Trust in Software Agent Societies*, "Online Journal of Applied Knowledge Management", Vol. 3(1), p. 93-101.

ZASTOSOWANIE WSKAŹNIKA REPUTACJI W OCENIE DZIAŁANIA KOMPONENTÓW SYSTEMU AUTOPOJETYCZNEGO

Streszczenie: Rozproszone systemy informatyczne wymagają zastosowania mechanizmów wspomagających monitorowanie i koordynację ich działania. Systemy charakteryzujące się samoorganizacją oraz autoadaptacją postrzegane mogą być w kategoriach systemów autopojetycznych, posiadających możliwość samoprodukcji i związanej z nią umiejętności definiowania relacji między komponentami systemu. Samoorganizacja jako mechanizm realizowany „z dołu do góry" dotyczy aspektu komunikacji między jednostkami struktury dla celu osiągnięcia określonych rezultatów. Autoadaptacja działa odgórnie i jest inicjowana przez mechanizmy kontrolne systemu, które wskazują, jak poszczególne jednostki powinny się zachowywać. Przykładem takich rozwiązań mogą być systemy zarządzania wiedzą wspomagane przez technologie agentowe. Celem tego artykułu są analiza koncepcji autopojezy oraz propozycja modelu oceniającego reputację elementów autopojetycznych i regulującego zachowanie agentów.

Słowa kluczowe: agent programowy, autopojeza, społeczność agentów, zarządzanie wiedzą.