

## MODELLING AGENTS COOPERATION THROUGH INTERNAL VISIONS OF SOCIAL NETWORK AND EPISODIC MEMORY

Michał WRZESZCZ, Jarosław KOŹLAK

*AGH University of Science and Technology  
Faculty of Computer Science, Electronics and Telecommunications  
Department of Computer Science  
al. A. Mickiewicza 30, 30-059 Krakow, Poland  
e-mail: {wrzeszcz, kozlak}@agh.edu.pl*

Jacek KITOWSKI

*AGH University of Science and Technology  
Faculty of Computer Science, Electronics and Telecommunications  
Department of Computer Science  
al. A. Mickiewicza 30, 30-059 Krakow, Poland  
&  
ACC Cyfronet AGH  
ul. Nawojki 11, 30-950 Krakow, Poland  
e-mail: kito@agh.edu.pl*

**Abstract.** Human societies appear in many types of simulations. Particularly, a lot of new computer games contain a virtual world that imitates the real world. A few of the most important and the most difficult society elements to be modelled are the social context and individuals cooperation. In this paper we show how the social context and cooperation ability can be provided using agents that are equipped with internal visions of mutual social relations. Internal vision is a representation of social relations from the agent's point of view so, due to being subjective, it may be inconsistent with the reality. We introduce the agent model and the mechanism of rebuilding the agent's internal vision that is similar to that used by humans. An experimental proof of concept implementation is also presented.

**Keywords:** Social networks, behaviour modelling, simulation of human societies, multi-agent systems, social context, artificial intelligence, computer games

**Mathematics Subject Classification 2010:** 68T42

## 1 INTRODUCTION

Simulations of human societies may be used to create virtual worlds, e.g. environment for a role playing computer game (RPG). We have chosen computer games as a test area because the development of video games constitutes a highly dynamic branch of the software industry. According to “Global entertainment and media outlook: 2012–2016”, the total revenue of this segment in 2011 was 58.7 billion US\$. During the last five years this branch had one of the highest increase rates among the global entertainment and media market segments (only Internet access and Internet advertisements were higher) and it is forecasted that its increase rate will remain in this position, with predicted average annual increases of 7.2% in the period from 2012 to 2016. Nowadays, profits from video games significantly surpass those of the music and radio industries, and it is assumed that it will even exceed consumer magazine publishing and be close to the film industry in 2016.

The players expect that the decisions made by their allies and opponents will be rational and that these decisions will be a consequence of logical reasoning. The behaviour of simulated individuals must be as similar as possible to human behaviour. In this article we focus on behaviour during the cooperation. We define realism as a replication or imitation of human decision-making process where each human is independent but may be influenced by external factors, including other people.

Classically, the simulation of individuals requires definition of four main ingredients [35]: high-level behaviour, perception, animation and graphics. Not all simulations require advanced visualization, so the third and fourth aspects will not be considered in this article. To model the perception of individuals we can adopt software agents interacting with the environment [25], thus, in this article, we treat each individual as a software agent. A great challenge in all simulations is high-level behaviour modelling. Research proved that the agents based on PECS model (Physical conditions, Emotional state, Cognitive capabilities and Social status [27]) and ontology can be configured to simulate specific scenarios [15]. User reputation in social network is also a field of study (e.g. [9]). Furthermore, Nazir, Prendinger and Seneviratne [21] show that their pattern based mobility model reproduces day-to-day human activities of people well. However, the difficulty of modelling of high-level behaviour grows significantly when the behaviour of a simulated individual depends on social context. For this reason, we focus our research on the social context. Nevertheless, it should be noticed that the issue of context in simulations is broad hence this paper limits the concept of context to the influence of one individual on another individual/s.

In this article we present a model that provides a social context by equipping each agent with its internal, conceivably subjective, vision of social relations with other agents. The internal vision is a representation of social relations from the agent's point of view, so it may be inconsistent with the reality. The model also contains elements that allow to use this social context to cooperate. We also show that the appropriately defined functions that update an agent's internal vision of social relations on the basis of observations of interactions of other agents allow the agent to build its internal vision in similar way to that of human's (i.e., the social network built by introduced algorithms is similar to the social network built by humans manually based on observation of people). A short version of this article has already been presented in [36].

The rest of the paper is organized as follows. In Section 2, we outline the state of the art of the Artificial Intelligence (AI) algorithms, agent systems and methods of social context provision. An extension of the classic agent model is proposed in Section 3. Next, in Section 4 we verify our context provision algorithms. In Section 5 we verify our agent model experimentally. Afterwards, we conclude this paper in Section 6.

## **2 STATE OF THE ART**

At the beginning of this section, we introduce agent systems because we are adopting software agents interacting with the environment for our simulations. We have chosen computer games as a test environment, so next the relation between AI and computer games is shown. Afterwards, we discuss methods of social context provision to show how the influence of one individual on another is modeled by other researchers. At the end of the section we describe AI techniques used by test agents (during the decision making, the path finding and moving), although our agent model is generic and can use other AI techniques as discussed at the end of this section and introduced here for the illustration of results.

### **2.1 Multi-Agent Systems**

The group of techniques that can be used to model interacting autonomous units is called multi-agent systems. The multi-agent domain researchers have already designed many algorithms that are useful during the development of AI for computer games. There are a lot of definitions of the agent, e.g.: "An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors" [25]. However, most of the scientists attribute similar features to agents [5]: communication, perception (environment's observation), knowledge, reasoning ability, aims and abilities (actions set).

According to Wooldridge [34] the agent system is a system that contains many computable elements (agents) that influence each other. Most of the systems have similar features [1]: distribution, autonomy (of each agent), decentralization, knowledge exchange, interactions, organization (agents can be organized into groups),

localization in environment (which is shared by agents), openness (results of operations impact the system, an agent can any time join or leave the group which is resolving a problem), emergency (global behaviour is not programmed, it is the reason of interactions between processes), delegation (of control from user to agents), personalization and intelligibility.

## **2.2 Definition of AI and Its Relation with Computer Games**

Norvig and Russel [25] defined four categories of definitions of the term “Artificial Intelligence”: systems that think like humans, systems that act like humans, systems that think rationally, systems that act rationally. When AI in computer gaming is considered, it is expected that AI will be able to replace human opponents. The Turing test is very useful when assessing whether an agent thinks like humans and when it can be described as an artificial intelligence. In this approach it is assumed, that the machine can be called intelligent if we are not able to differentiate the machine’s answers from human answers.

The architectures of most computer games are similar. According to Rollings and Morris [24], the computer game contains the following subsystems: user interface, event handler, data engine, dynamics system, logic engine, graphics engine, sound engine, and hardware abstraction layers. It can also contain: game configuration system, menuing system, online instructions and help system, and music system. The logic engine contains artificial intelligence. AI interacts also with other modules: physics engine (information about collisions, moving objects etc.), game data (rules used by AI) and event handler (use of information included in messages).

## **2.3 Provision of Social Context**

Several social context models have been introduced. In [19] a model that describes how agents influence each other within one organization is shown. The basis of the model are beliefs and organizational code. The beliefs are elements of the agent’s internal representation of the world while the organizational code is a string of values that represents the organization’s approximation of beliefs about that reality. In each period, every agent alters any given belief to conform to that of the organizational code with some probability, reflecting the rate of socialization of individuals in the organization. The organizational code also alters any given belief based on the dominant belief of the set of agents – the superior group, defined as those agents whose individual beliefs correspond better with reality than does the code. The model was later extended by Kane and Prietula [12]. They let individuals learn from (be influenced by) one another. The probability that a given individual would learn from the other individuals rather than from the code was represented by an additional parameter. However, it should be noticed that the agents easily develop erroneous beliefs [6].

Another interesting approach is the tag-based computational model [3], which uses tags to describe agents features. In this model, only similar agents can influence each other. Similarity of agents is calculated by comparison of their tags.

One of the latest approaches that use virtual world to show how people behave in particular situations was developed by the EUSAS project (European Urban Simulation for Asymmetric Scenarios) [14, 16]. In this project, the social influence of agents, that were divided into groups, was modelled according to the Latane formula of strength, immediacy and number of other agents [17]. The agent's internal state was changed on the basis of observation of actions of other agents that belong to the same group, e.g., aggression of the agent was growing when it observed aggressive actions of other agents that belonged to the same group. The strength of this effect depended on the social position of the observed agent.

The presented models are not able to reproduce some phenomena from the real world where a group of people often includes pairs of close friends as well as pairs of people who do not know each other personally. The influence of a good friend is often higher than the influence of other people even with a higher social position, hence the social context model should contain a social network, which describes relations between each pair of agents. Nevertheless, a single social network might be insufficient. In the real world, an attitude to people depends on the past interactions and information received from others. When somebody is not known personally, our attitude to them depends on their relations with the people we know and his/her behaviour that we can observe. Unfortunately, we cannot always evaluate relations between other people correctly. If we observe two friends arguing, we can assume subjectively that they do not like each other. Each human being usually has his/her own internal vision of social relations that link people around, so each agent should have its own internal vision of the social network.

After equipping each agent with its internal vision of the social network we can easily improve the social context models. For instance, a similar model like in the EUSAS project may be used, however, the strength of the influence could depend on strength of the social relation (in internal social network) between the observer and the observed agent, not obligatorily on the social position of the observed one. Therefore, a key issue is providing a mechanism that updates an agent's internal vision of the social network.

Summarizing the importance of the context and value of the social relations it is worth to mention one of the classical books on the topic [4], useful in both every day activities and scientific research.

## **2.4 AI Techniques Used During Decision Making**

Each autonomous agent decides which action it should execute. There are many AI techniques that can be very useful during the decision making process. Three techniques were selected for the tests: game trees, influence maps and state machines. These techniques were chosen because they are popular and provide solutions of elementary problems which may appear. The selected techniques offer substantive

features necessary for successful functioning in the game environment, such as identifying the attractiveness of goals, prediction of opponents decisions, identification of the current situation of the agent and appropriate choice of kinds of activities and adaptation to the changing conditions.

Game trees [30] can show future states of the game. An agent can analyse the tree and decide which future state will be most profitable. Afterwards, it can do actions (represented by tree edges) that lead to a chosen state of the game. The main problem during strategy creation is that some edges in the tree may represent actions of enemies, so the agent does not have full knowledge of real consequences of its decision. The agent should choose actions that will provide a profit irrespective of the opponents' decisions. Game trees work best with games where profit for one player indicates a loss for the others. They can also be used to model some aspects of the game, e.g. evolution of the agent.

Influence map [20] shows which areas of the map are the best ones for the agent. The influence map is a mesh. Values at this mesh represent the profit that the agent can receive if it goes to a particular area. The higher value, the greater chance for a better profit. Figure 1 shows an example of the influence map. The map includes one strong aggressor (A1), one weak aggressor (A2), one weak prey (P1) and one strong prey (P2). The highest profit the agent can receive at the areas where a prey can be found. The weaker the prey, the more probable the success of killing (receiving a profit), so the influence of a weak prey is higher than the influence of a stronger one. The agent should avoid areas where it can be killed by an aggressor so the influences of aggressors are negative. The more dangerous the aggressor, the higher the absolute value of its influence.

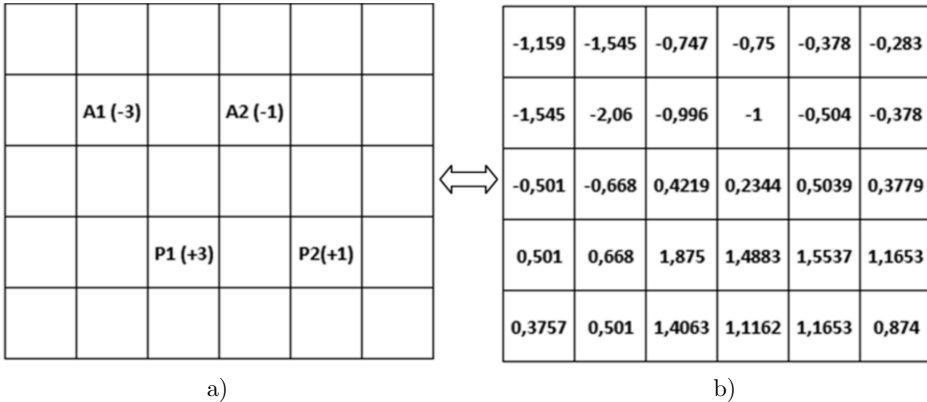


Figure 1. a) Map with two aggressors (A1, A2) and two preys (P1, P2) – values in brackets represent influences of the agents b) an influence map calculated for the presented situation

The state machine [32] is an easy but powerful technique. Each agent that uses a state machine can only be in one state at any time. The state can represent the current aim of an agent. State changes can be automatic or forced by an event or

a message. It is a universal technique and can always be used when we can represent something as a set of independent elements.

## **2.5 AI Techniques Used During the Path Finding and Moving**

Techniques and algorithms used during path finding and moving are as important as techniques used during decision making because they allow the execution of actions chosen by decision algorithms. The most popular algorithm used during path finding is A\* [40]. It guarantees that the path will be found if it exists [29]. Moreover, some modifications of A\* allow for finding of an optimal path from the tactical point of view, e.g., the safest path – it only depends on the type of information that the edges of the graph represent. The path found by the A\* algorithm is often not smooth. However, there are a lot of algorithms that can improve it, for instance the visibility test [28]. To smoothen the path, an agent can check if the view between two points on the path is clear (a visibility test). If the view is clear, all the points on the path that are between the tested points may be deleted, which reduces the number of turns during movement.

Another issue is the representation of roadblocks and areas where players can or cannot move. In the presented system, navigation meshes [20] were used. A navigation mesh is made of polygons. Each agent must be placed inside one of these polygons. Roadblocks are outside of the mesh, ensuring that an agent cannot stand on them.

## **3 AGENT ARCHITECTURE**

In this section an agent model is shown (Subsection 3.1) and compared with other agent architectures (Subsection 3.2).

### **3.1 Agent Model**

The agent model is shown in Figure 2. The agent observes the environment using sensors and stores information about important events in the memory. In our case, the agent stores information about possible interactions of other agents. On the basis of this information it builds its internal representation of the world, which describes social relations of known individuals. The agent uses this internal vision of environment, together with the information about itself, to plan its actions (it is done by a logic engine) that are executed by effectors. During the planning, the agent can choose possible allies (on the basis of the internal vision of the environment and the information about the past cooperation) and then communicate with them to propose cooperation. With this solution each agent is fully independent but may dynamically create temporary groups that will bring profits to all participants.

The internal social network may be multi-layered [13] to describe various types of relations, e.g. professional relations, friendship, etc. The Multi-layered Social

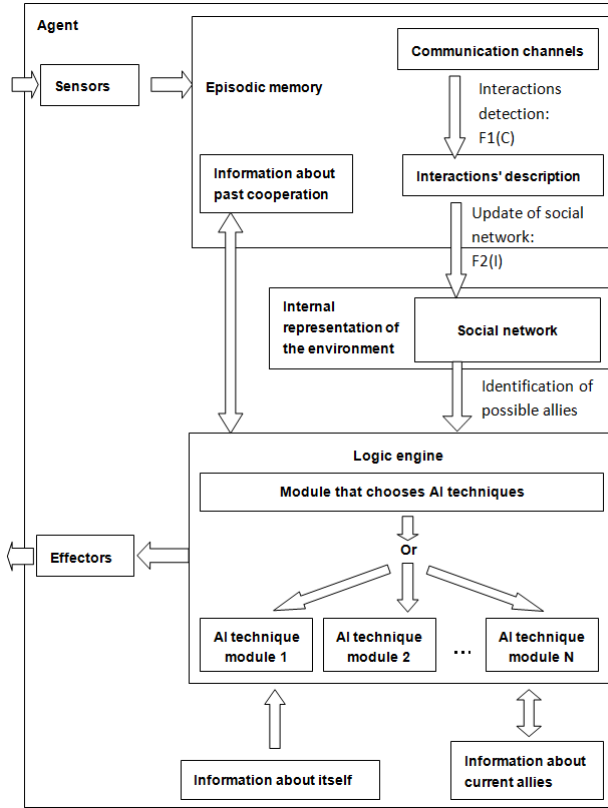


Figure 2. Agent model

Network is defined as a tuple  $\langle V; E; L \rangle$  where  $V$  is a non-empty set of nodes,  $E$  is a set of edges and  $L$  is a set of layers [2]. The edge is a tuple  $\langle x, y, l \rangle$  where  $x, y$  are different nodes and  $l$  is a layer ( $x, y \in V, l \in L, x \neq y$ ). The layer is a set of relations of the same type (e.g. a trust layer, family ties layer). Maximum two relations between particular nodes ( $x$  to  $y$  and  $y$  to  $x$ ) belong to each layer:  $\langle x, y, l \rangle \in E \wedge \langle x', y', l' \rangle \in E \wedge x = x' \wedge y = y' \Rightarrow l \neq l'$ .

An update of the social network is carried out through the episodic memory. The agent observes the environment and represents observed social events as a set of communication channels. The representation of the environment as the set of communication channels was described in [37]. The agent creates the communication channel in its memory when it observes the possibility of communication between a pair of other agents and destroys the channel when it states that this pair is not able to communicate any longer. The communication channel contains information about the period of time in which the described agents were able to communicate. It may also contain some additional information if available, e.g. an attitude of one



observed agent to another evaluated on the basis of its gestures. The communication channels are analysed by function  $F1$  to provide an interactions description. The  $F1$  function decides if the channel describes interactions or not. If the channel describes interactions, the function assigns a type to them. Afterwards, on the basis of the interactions types, the relations in the social network are updated using function  $F2$ . The interaction's type is used to choose which layer of network should be updated and decides how the value of the relation in a chosen layer should be changed, e.g., observation of kind gestures should increase the value of the friendship relation while observation of an argue should decrease this value. More formal definitions of  $F1$  and  $F2$  are presented below:

- $F1(C) \rightarrow \{I1, I2\} \vee NULL$  where:
  - $C = \{A1, A2, D, O\}$  – a communication channel between pair of agents where:
    - \*  $A1, A2$  – agents connected by the channel,
    - \*  $D$  – duration of the channel functioning,
    - \*  $O$  – other information about the observed event if available (e.g. attitude of agents  $A1$  and  $A2$ ),
  - $\{I1, I2\}$  is a tuple that describes interactions –  $I1$  represents actions of  $A1$  while  $I2$  actions of  $A2$ ; function may return  $NULL$  when an event described by the communication channel is not an interaction (e.g., when one agent walked by another and they did not see each other); the  $I1$  and  $I2$  interactions may be different when additional information is included in the communication channel (e.g., if  $A1$  attacks  $A2$  and  $A2$  only defends itself,  $I1$  will represent aggressive interaction initialized by  $A1$  while  $I2$  – a not aggressive response of  $A2$ ); an interaction is defined as a tuple  $I = \{A1, A2, T\}$  where:
    - \*  $A1, A2$  – agents involved in the interaction,
    - \*  $T$  – type of interaction,
  - $NULL$  – ignore this communication channel this time, no interactions are produced (function  $F2$  will not be used),
- $F2(I) \rightarrow [U]$  where:
  - $I$  – an interaction,
  - $[U]$  is a list that describes updates of the social network that should be done on the basis of the interaction; function returns a list because updates of more than one layer can be performed on the basis of a single interaction. Update is a tuple  $U = \{A1, A2, L, V\}$  where:
    - \*  $A1, A2$  – agents whose relation should be updated,
    - \*  $L$  – a layer of social network that should be updated,
    - \*  $V$  – the amount of changes of relation (numerical value).

Sample functions  $F1$  and  $F2$  are shown in Section 4.

The quality of each layer of the social network can be evaluated using control data provided by human and the following formulas:

- calculate the output value of each node:

$$- \text{node\_out\_value} = \sum_{\text{outgoing\_relations}} \text{relation\_strength},$$

- normalize the relations' strengths:

$$- \text{normalized\_strength} = \frac{\text{relation\_strength}}{\text{start\_node\_out\_value}},$$

- *start\_node\_out\_value* is the *node\_out\_value* of node from which the relation starts,

- calculate the quality of node:

$$- \text{node\_quality} = \sum_{\text{outgoing\_relations\_included\_in\_control\_data}} \text{normalized\_strength},$$

- control data contains relations that indeed exist in the real world (the social network created by the agent is subjective while the control data is objective),

- calculate the quality of layer:

$$- \text{layer\_quality} = \frac{\sum_{\text{all\_nodes}} \text{node\_quality}}{\text{number\_of\_nodes}}.$$

The quality of a layer equal to 1.0 means that this layer describes only relations included in the control data, so the obtained results fit perfectly to the reality as ones evaluated independently by the individuals. The evaluation of the friendship layer in the exemplary social network is shown in Figure 3. The input data and the control data were artificially created to illustrate the algorithm.

The logic engine may contain many AI techniques because different algorithms can work best in different situations/environments. Moreover, some AI techniques may work better during the cooperation with other agents, while others may work better when the agent is working alone. For this reason, the agent first uses information stored in episodic memory to choose an AI technique that will be used during the decision making and only then it uses a chosen technique to make a decision. During the selection of AI technique, information about the past cooperation as well as information about possible allies are used to evaluate if it is worth to cooperate.

The way of use of the information included in the internal social network depends on the AI technique currently used by the logic engine. For example, the agent may evaluate a type of relation between particular agents checking which layers of the social network contain the relation between the considered agents. An other possibility is choosing a layer using some algorithm (e.g., the friendship layer may be used during evening while the professional relations layer may be used during working hours) and then an analysis of the relations' strengths in the chosen layer.

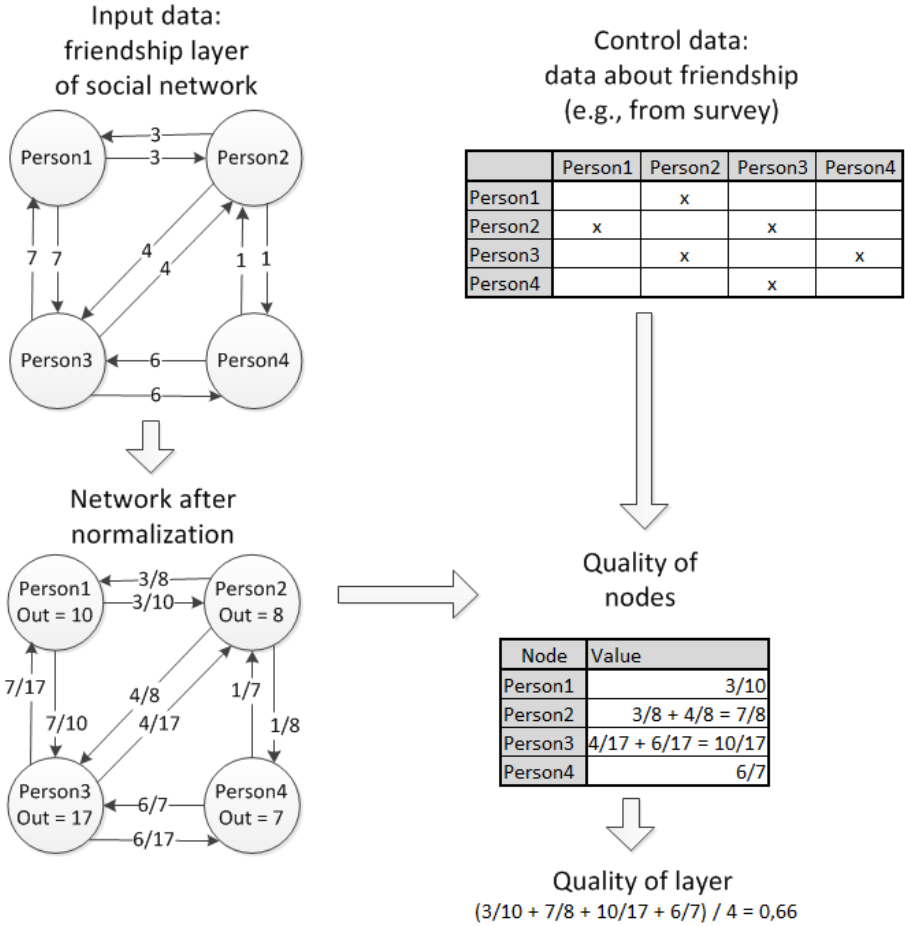


Figure 3. Evaluation of the friendship layer in the exemplary social network

### 3.2 Comparison with Other Agent Architectures

The proposed agent model uses the classical approach. The classical approaches introduce a deliberative agent [35] that contains a symbolic model of the world and in which decisions are made via logical reasoning, based on pattern matching and symbolic manipulation. The agent also gathers information about the past events. This type of information is stored in so-called episodic memory. It was introduced by Vere and Bickmore [33] in an agent called HOMER.

The idea of decomposition of logic engine is similar to the concept introduced by Maes [18]. She defined an agent as a set of competence modules. When this agent is executing, various modules may become more active in some given situations. In this

case, each module has an activation level that is related with the probability that this module will influence the behaviour of an agent. The choice which AI technique will be used by an agent proposed by us is also dependent on the situation. However, we do not enforce a choosing technique. The agent creator may use a random number generator as well as deterministic algorithms.

The need to process large amounts of information about the world during the decision making can be considered as a drawback of the model. The opposite to the proposed solution are reactive agent architectures. They do not use any kind of central symbolic world model and do not use complex symbolic reasoning [35]. If the speed of decision making will be too low, it is possible to add a module that will react to events that happen quickly (too quickly to process them in a standard way) and allow for propagation of information from sensors to this module directly. In such a case, the agent model would be classified as a hybrid architecture [35]. This type of solution was proposed earlier by Ferguson [8] in the Touring Machines hybrid agent architecture. This architecture consists of perception and action subsystems (that correspond with sensors and effectors in the proposed architecture), and three control layers, embedded in a control framework. The reactive layer generates responses for events that happen too quickly for other layers and the planning layer constructs plans and selects actions to execute. The modelling layer of the Touring Machines hybrid agent architecture contains symbolic representations of the cognitive state of other entities. It is also responsible for identifying the situations when an agent is not able to achieve its goals (as a result of an unexpected interference).

As shown above, the described agent uses many well-known techniques. The idea of use of sensors, effectors, episodic memory and internal representation of the world is not new. The translation of the social network to a form useful for the logic engine has also been described, e.g., in [26] a mechanism of calculation of social reputation on the basis of social relations was described. Reputation of the agent can be used by the logic engine to verify the reliability of messages sent by it. Thus, we propose a novel method of building and updating agent's internal vision of social network what is, in our opinion, a key to increasing realism of the simulations where agents are independent but can create temporary alliances to achieve their goals.

#### 4 SOCIAL CONTEXT PROVISION ALGORITHMS TESTS

The aim of creation of the introduced agent model was to increase the realism of the behaviour of individuals in simulations with independent agents being influenced by external factors including other agents. To verify that, we compare the social network built using the proposed approach with that constructed manually by human on the basis of data collected in the real world. The results conformity would verify the ability of the approach to mimic well some human processes during the decision making. Hence, the verification of the use of functions ( $F1$  and  $F2$ ) for the creation of two separate layers of the social network – one showing a friendship and another professional relations of a group of the observed people – were performed.

During the experiments we used the Reality Mining Dataset [7] that included the data collected in the real world. The Reality Mining Dataset incorporates 94 subjects that had completed the survey conducted in January 2005. Of these 94 subjects, 68 were colleagues working in the same building in a campus (90% graduate students, 10% staff) while the remaining 26 subjects were incoming students at the university's business school.

For our experiments, we have used three types of data included in the dataset. The first type was Bluetooth data from subjects' telephones. Using MAC addresses of the Bluetooth devices discovered on each Bluetooth scan and times of Bluetooth scans we could reproduce possible interactions of subjects. The second type was a survey where each subject indicated his/her friends. This data was used to verify, whether the constructed friendship layer was correct. The third type was survey data that described which subjects saw each other every day in office. This data was used to identify the coworkers of each subject.

#### 4.1 Experiment 1

In our experiment, the social network was evolving over a time, from the first time of Bluetooth scan to the last time of Bluetooth scan. Communication channels were updated at each time of scan. Additionally, at each time of scan, the interactions were identified (use of  $F1$  functions) and the social network was updated (use of  $F2$  function). Simple  $F1$  and  $F2$  functions were used. Function  $F1$  was creating a pair of interactions when a communication channel between agents existed at the time of scan. The type of each interaction was "general". For each interaction, function  $F2$  was incrementing the value of the relation from  $A1$  to  $A2$  (see the definition of  $F2$  in the previous section) in both layers if the relation existed. If not, the relation was created with value one. After the last Bluetooth scan, the qualities of the layers have been calculated using the instruction described in Section 3. The control data for the friendship layer contained relations between the people that marked each other in the survey as a friend while the control data for the coworkers layer contained relations between the people identified as coworkers (on the basis of the survey). The quality of the friendship layer equal to 1.0 means that this layer describes only relations between friends, the quality of the coworkers layer equal to 1.0 means that this layer describes only relations between coworkers.

The quality of the friendship layer was 0.18 while the quality of the coworkers layer was 0.5 (see Table 1 in Section 4.2). This is consistent with the expectations because people spend more time at work than with friends, and this method treated all meetings equally.

#### 4.2 Experiment 2

Experiment 2 was similar to Experiment 1. We have only redefined  $F1$  and  $F2$  functions on the basis of our everyday experience. People meet friends mainly in the evenings during weekends while coworkers meet during office hours from Monday

to Friday. Additionally, the meeting duration should be taken into account. Meeting with close friends is usually longer than meeting with people we know but they are not our friends. Moreover, during office hours, the time that we spend near coworkers is usually very long – much longer than the time of visit of a person that has only some business with us. Therefore we have defined  $F1$  and  $F2$  functions as follows:

- $F1$  – create a pair of interactions between the subjects connected by a communication channel when the following requirements are fulfilled:
  - If the communication channel has existed during the last 10 scans, and the day of week is Saturday or Sunday, and the time is between 9 and 12 pm., create friends' interactions.
  - If the communication channel has existed during the last 60 scans, and the day of week is not Saturday or Sunday, and the time is between 8 am. and 4 pm., create coworkers' interactions.
- $F2$  – update the value of a relation between a pair of the subjects participating in an interaction when the following requirements are fulfilled:
  - If the interaction type is a friends' interaction, increment the value of the relation from  $A1$  to  $A2$  (see the definition of  $F2$  in the previous section) in the friendship layer if the relation exists. If not, create it with value one.
  - If the interaction type is a coworkers' interaction, increment the value of the relation from  $A1$  to  $A2$  (see the definition of  $F2$  in the previous section) in the coworkers layer if the relation exists. If not, create it with value one.

The results of the experiment are shown in Table 1. The quality of the friendship layer was 0.80 while the quality of the coworkers layer was 0.95. It means that the quality of the friendship layer is more than four times better than in the Experiment 1 and the quality of the coworkers layer is almost two times better.

Layer	Layers' Qualities	
	Experiment 1	Experiment 2
Friendship	0.18	0.80
Coworkers	0.5	0.95

Table 1. Results of the experiment

However, it should be noticed that the functions  $F1$  and  $F2$  proposed by us may not be optimal. The aim of the experiment was not to find the best  $F1$  and  $F2$  functions, but to show that the proposed solution works. Probably, more complicated functions would give better results but even such simple functions have shown that the introduced technique works well and can be used to simulate societies. On the other hand, use of simple functions allows for reduction of the simulation time or use of a greater number of agents.

## 5 AGENT MODEL TESTS

The experiments shown in the previous section verified the social network creation algorithm because this algorithm is the key element of the proposed agent model. In this section we verify the whole agent model in an environment similar to a computer game. We implemented such an environment and then analyzed the behaviour of agents using the proposed model and algorithm tested in the previous section with F1 and F2 adopted properly. During the test we created a food chain and gave the agents an ability to kill other agents. The aim of each agent was to stay alive and kill as many preys as possible. Some agents had the same prey, e.g., agents A and B could hunt both for agent C so they could cooperate during the hunting if they were able to identify a possible ally. To make this possible we have provided the agents with algorithms that support group cooperation. The internal social network of each agent was used to identify possible allies.

### 5.1 Test Environment and Senses of Agents

The world is a rectangular area. Within this area, agents can move. This area is surrounded by walls that cannot be overcome. In the game area (between walls), roadblocks can be located.

The world is filled with smells and sounds. Each agent produces odour. The value of a smell decreases when the distance from the smell source increases. We selected simple form of this relation:

$$v = \frac{o}{fd} \quad (1)$$

where:

- $v$  – value contributed by the agent at the considered point,
- $o$  – value of agents odour (at the point where agent stands),
- $f$  – speed of the smell fading away,
- $d$  – distance to the agent (from the considered point).

Smells cannot infiltrate roadblocks so the distance, in the formula above, represents the distance that has to be covered by a smell between roadblocks (see Figure 4 a)). This distance can be calculated using any path finding algorithm, e.g. A\*. The odour values, from the agents that smell similarly<sup>1</sup>, are added together and one stronger odour is noticed by other agents instead of many weaker ones (see Figure 4 b)). The flavours from the agents that smell differently<sup>1</sup> do not influence one another – they can be distinguished by the receiver.

The agent does not know the values of smell on the whole map. It can only check the value of the smell at the point where it is standing. Continuous checking

---

<sup>1</sup> We were able to configure the game so that some agents had a similar odour.

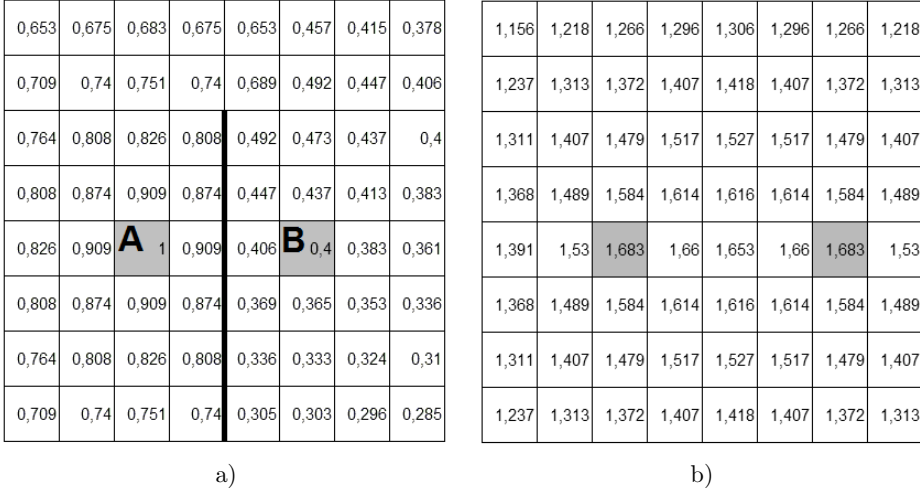


Figure 4. Smells propagation. a) propagation of the odour of an agent that stands at square A is shown. Bold line represents a roadblock. The agent that stands on square B will perceive the most intensive rise of smells intensity if it moves up and left. b) shows the addition of smells produced by two agents.

of the smell value as the agent moves, can provide information on whether the value of the smell is growing or falling, which shows if the agent is coming nearer to its enemies or further from them. This information is especially useful when the agent cannot see its opponents.

Each agent produces a sound when it moves. The sound is louder when the agent moves faster. Additionally, the agents use sounds to communicate with others. When there are no roadblocks near the sound source, the sound is propagated equally in all directions. The roadblocks can muffle sound partially or completely. The muffling rate can be configured. The sound can reach the agent crossing over the roadblocks or passing around so sounds sometimes can reach areas that are inaccessible by smells. The sound propagation is shown in Figure 5.

The sound is heard by the agent when Equation (2) is fulfilled:

$$rDist \leq l \cdot s \vee sDist \leq l \cdot s \cdot \prod_{rb} a(r) \quad (2)$$

where:

- $rDist$  – real distance from the source to the agent,
- $sDist$  – straight line distance from the source to the agent,
- $l$  – sound loudness (at the point where the sounds source occurs),
- $s$  – agent's hearing sensitivity,
- $rb$  – a set of all roadblocks that block sounds' path,



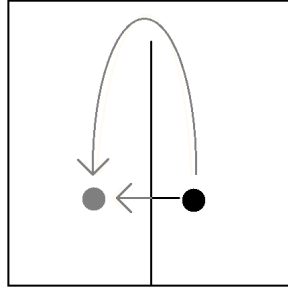


Figure 5. Sound propagation. The curve represents the sound propagation between roadblocks, the straight line represents the roadblocks penetration by the sound

- $a(r)$  – sounds absorption of roadblock  $r$ , absorption  $[0, 1]$ ,  $0$  = total absorption,  $1$  = no absorption.

A distance equal to 1 is the amount of space occupied by a single agent. Odour, speed of the smell fading away, sound loudness, and agent's hearing sensitivity are arbitrarily set by the creator of the simulation/environment, e.g., if some agent should be heard from a long distance, its loudness should be a large number.

## 5.2 Implementation of Agents

Implemented agents use various AI techniques. Lines of sight [23] were used to check what can be seen by an agent. The usage of this technique is shown in Figure 6.

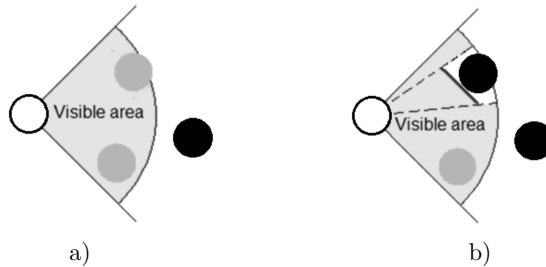


Figure 6. Visible area for a white agent. a) open area and b) area near roadblock. Grey agents are seen by the white agent while the black ones are not. The dashed lines represent lines of sight.

Fuzzy logic [39] is used to provide the agent energy management (the amount of agent's energy is limited but regenerated after a period of time). Five speed levels of agent's movement are defined and an aggression level is introduced. The agent is more aggressive when a lot of opponents are around. If the agent estimates that it can be killed by an enemy or that it can kill an opponent, its aggression grows.

If it evaluates that the opponents that are around are not a threat for it and it is not able to kill anybody, its aggression drops. A relationship between aggression, energy level and agents speed is shown in Table 2.

		<i>Aggression Level</i>		
		<i>None</i>	<i>Medium</i>	<i>High</i>
<i>Energy Level</i>	<i>Low</i>	Very slow	Slow	Normal
	<i>Medium</i>	Slow	Normal	Fast
	<i>High</i>	Normal	Fast	Very fast

Table 2. Agent speed depending on aggression and energy level

The agents use combinations of several well known techniques to move and find paths between roadblocks. A\* is the base for path finding. The graph used by this algorithm is created on the basis of a navigation mesh. Areas in the mesh correspond with graph nodes (see Figure 7). The edges' weights are calculated on the basis of the weights of areas. The weight of an edge is equal to the multiplicative inverse of the weight of the area that corresponds with the target node. The agents can change the characteristics of the algorithm by changing the weights of areas, e.g. they can find the safest path by assigning to each area a weight that means "safety" of this area. A visibility test is used to smoothen the path found by the A\* algorithm.

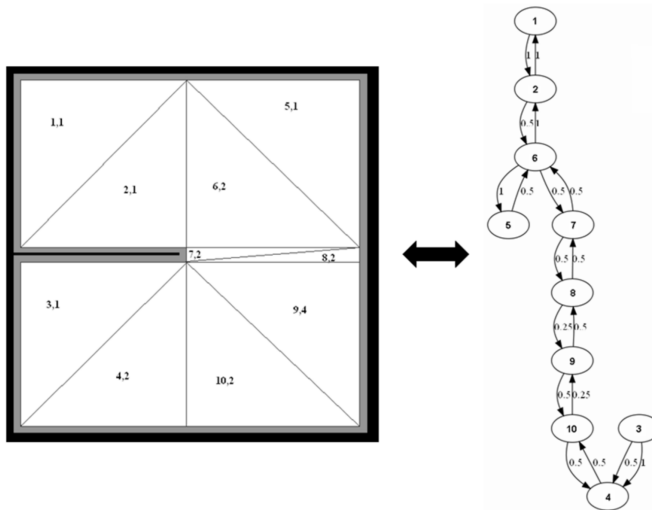


Figure 7. Relationship between the navigation mesh and graph used by the A\* algorithm

**The state machines** are used to model the main activities that can be performed by the agent (see Figure 8). It allows agents to remember their main aims at the moment. The states support procedures that allow for hunting, getting

information and escaping into a group. During group hunting, the agents decide if each agent should hunt for its own prey (to maximise the number of killed opponents) when the enemies are weak, or to hunt together for one prey when the prey is strong (some agents try to catch the prey while others flank it). When they are collecting the information together, they divide the area between them to find the enemies faster. During the escape, they decide if they should disperse (when the aggressor is strong) or to escape together (when the aggressor is weak) to perform other group actions when they will be safe.

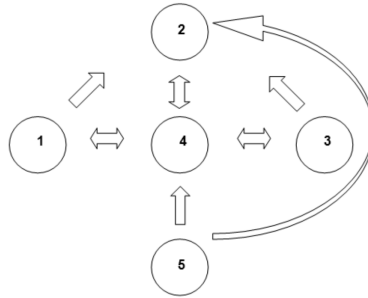


Figure 8. State machine used by the agent. States names: 1. Hunting, 2. Escaping, 3. Getting Information, 4. Thinking, 5. Patrol

**The influence map** is used to find the areas that are most beneficial for the agent.

The agent uses sensors to mark information about possible or current allies, aggressors, preys and agents that hunt for aggressors. It also slowly increases the value of areas that were not seen for a long time – it allows to explore new areas when nothing interesting happens. When the agent does not know what to do, it can simply move towards the point which has the highest value on the map. Additionally, the agents use the information from the influence map to evaluate the values of areas of the navigation mesh. Each area of the navigation mesh obtains a value that is the average of the weights of the zones of the influence map that are inside this area. It allows agents to find the optimal path.

**The game tree** (the tree nodes correspond with the navigation mesh areas) is used to show possible paths from a chosen area on the map (see Figure 9). It allows the agents to foresee the moves of the opponents to catch them.

### 5.3 Results

The tests have shown that the agents are able to create temporary groups to increase their chance for success. In this section we show an interesting behaviour observed during the tests. The first interesting action took place near roadblocks and is shown in Figure 10. The agents depicted as dark circles hunt for the agent shown as a grey circle. The grey agent is faster than hunters but the location of

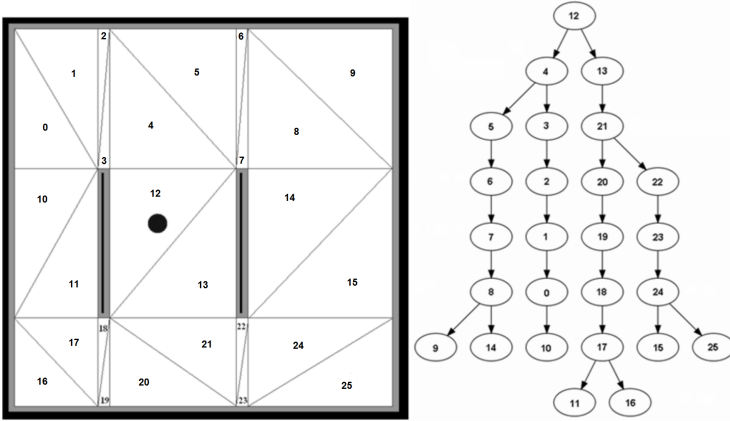


Figure 9. Navigation mesh and the corresponding game tree that have the root in area 12

roadblocks (represented by lines) gives the hunters a possibility to catch the prey if they cooperate. At the beginning, each agent evaluated his chance to catch the prey on his own. They realized that the prey could easily escape in the open terrain so they communicated to create a temporary group. Afterwards, they tried to foresee all possible escape paths of the prey. Some agents tried to secure all possible escape paths while one of the agents attacked the prey directly.

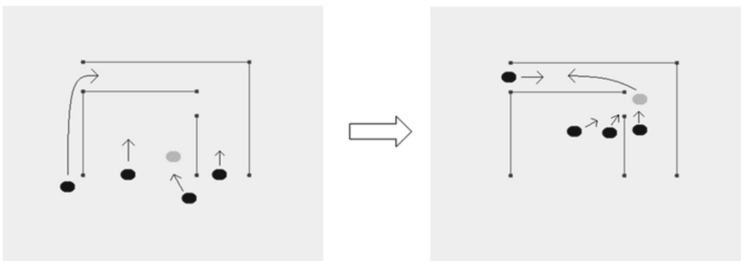


Figure 10. Behaviour of created AI when the location of roadblocks gives hunters an advantage

In Figure 11, cooperation during the exploration of the world and hunting is shown. The agents depicted as dark circles were not able to kill the prey (the grey circle) for some time because it was much faster. They identified that they had common goals and formed a temporary group to find and kill the prey jointly. At the beginning, they divided the areas to explore among themselves (step 1). When an agent saw a prey during exploring the world (step 2), it got back to hunt together with its allies (steps 3 and 4). It was possible because the prey had a smaller seeing range than the hunters.

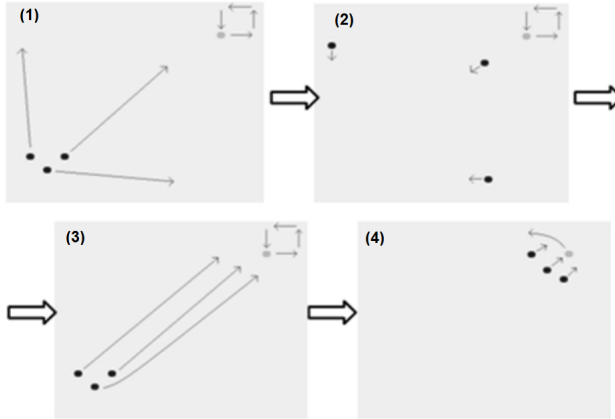


Figure 11. Exploring the world and returning for allies when the prey is very strong

Due to the fact that the agents are able to identify the relations between other agents, they are also able to evaluate a danger during the path finding (see Figure 12). If there is no danger, the agent chooses the shortest path (Figure 12 a)) but if the path is very dangerous, the agent can choose a longer path (Figure 12 b), where the dark agent hunts for grey one). However, knowing the relations that link others, the agent is able to evaluate whether other agents can help it to pass by the aggressor. If there are a lot of agents that hunt for its aggressor on the path (Figure 12 c)), the risk is acceptable and a shorter path will be better (hunting for the agent will be too dangerous for the aggressor).

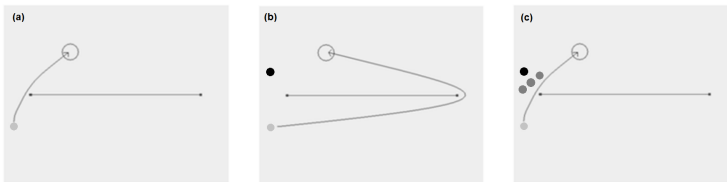


Figure 12. Path finding dependent on situation (the dark grey agents hunt the black one, the black one hunts the light grey one)

Figure 13 shows hunting in the open terrain. The arrows show movements of the prey (marked by a grey circle) and the hunters (dark circles). When there are no roadblocks near the prey, the agents can form a temporary group to limit possible escape paths of the prey by flanking.

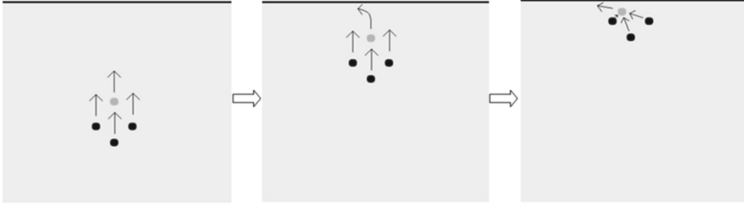


Figure 13. Hunting in the open terrain

## 5.4 Result Analysis

We can classify our tests as solving the pursuit problem for which many methods have been proposed [22]. Some typical examples are Q-learning [11], genetic programming [10] and self-organizing neural models, e.g. TD-FALCON [38] (which is a specific class of FALCON<sup>3</sup> [31] models). Previous research enabled the formulation of some useful conclusions. First, an appropriate state representation is needed to produce agents that cooperate efficiently [38]. Our tests confirm that conclusion. Without the social network and the episodic memory used to create this network, our agents would not be able to find allies.

Although, it is very important for agents to possess information about all allies and preys, it is usually impossible to provide all information to all agents because the problem of combinatorial explosion arises [11, 38]. However, this problem can be solved by the application of an appropriate representation of information [11, 38]. In our case we can reduce the amount of information gathered by each agent choosing the layers of a social network reasonably.

It should be also noted that the system, when the pursuit problem is considered, should be treated as a heterogeneous multiagent system because even if the agents are homogeneous at the beginning, their abilities become heterogeneous in the learning process [11]. For example, when some agents gather knowledge about relations of other agents faster, they are able to cooperate better than others at the moment. Ishiwaka et al. [11] also observed elements that imply the way in which the agents organize themselves. An agents' organization depends on: the target movement rules, differences between the hunter and target agents (in particular their speeds) and the initial location of each agent. We could observe the same situation during our tests. The faster/stronger agents behaved differently from the slower agents because they were able to catch a prey on their own while weak agents had to cooperate. The initial location was also very important because when an agent started near a group of other agents, it was able to build its internal vision of the social network faster than in the case when it started isolated.

---

<sup>3</sup> FALCON is an abbreviation from Fusion Architecture for Learning, COgnition, and Navigation.

All in all, our test confirmed the conclusions about the pursuit problem formulated by other researchers. However, finding a perfect pursuit algorithm was not the aim of our work. The most important innovation of our work is the way of gathering the knowledge about the world by the agents. We have tested the designed agent model with the pursuit problem to show that the proposed agent model allows for implementation of effective agents despite of the fact that the internal representation of the world constructed by the agent may not be entirely correct as in the case of humans.

## 6 CONCLUSIONS

We have shown that we were able to build a reliable social network only on the basis of observing interactions of others. The experiment shown that defining the functions  $F1$  and  $F2$  in an appropriate way results in a successful identification of various types of social relations. The proposed agent model increases the realism of the social simulations because the simulated individuals have their internal vision of social relations exactly like humans (agents can have different opinions about relations that link other individuals in the system in a similar way like different people may have different opinions about social relations that link the people around). Moreover, the proposed mechanism of updating the internal vision was successfully verified using test data from the real world. The proposed agent model can be used in computer games. The knowledge about social relations may be used to form temporary alliances that help to realize agents' goals. All in all, the novelty of this article lies in

1. the method of constructing the social network based on the observations of interactions between agents,
2. use of this method to create an agent's internal vision of a social network, which represents an agent's subjective vision of relations between others,
3. use of this internal vision of social network to choose allies dynamically.

In the near future we want to test the behaviour of the introduced agents in computer games that contain a much more complicated society. The identification of relations between the agents in the presented environment was simple due to their simplicity. The successful verification of the approach on the basis of the real world data allows to believe that the proposed agent model will work well also in the games with complex societies where agents have much more sophisticated goals than killing and are able to form such social relations as friendship or hatred.

## Acknowledgments

M. Wrzeszcz is grateful for European Union project UDA-POKL.04.01.01-00-367/08-00. J. Koźlak and J. Kitowski are grateful for AGH-UST grant No. 11.11.230.124.

## REFERENCES

- [1] BOISSIER, O.—GITTON, S.—GLIZE, P.: *Caracteristiques des Systems et des Applications. Systemes Multi-Agents*. Paris: Observatoire Francais des Techniques, Avancees, OFTA, 2004 (in French).
- [2] BRODKA, P.—FILIPOWSKI, T.—KAZIENKO, P.: *An Introduction to Community Detection in Multi-Layered Social Network*. WSKS 2011, The 4<sup>th</sup> World Summit on the Knowledge Society, 2011.
- [3] CHEN, Y.—PRIETULA, M.: *To Deceive or Not to Deceive? Mimicry, Deception and Regimes in Tag-Based Models*. Intra-Organizational Networks (ION) Conference, 2005.
- [4] CHRISTAKIS, N. A.—FOWLER, J. H.: *Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives*. Little, Brown and Company, Hachette Book Group, New York, 2009.
- [5] DEMAZEAU, Y.—MULLER, J.-P. (eds.): *Decentralized AI*. Elsevier, 1990.
- [6] DORAN, J.: *Social Simulation, Agents and Artificial Societies*. Third International Conference on Multi-Agent Systems, 1998, pp. 4–5.
- [7] EAGLE, N.—PENTLAND, A.—LAZER, D.: *From the Cover: Inferring Friendship Network Structure by Using Mobile Phone Data*. *Proceedings of the National Academy of Sciences*, Vol. 106, 2009, pp. 15274–15278, doi: 10.1073/pnas.0900282106.
- [8] FERGUSON, I. A.: *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*. Ph.D. thesis. Clare Hall, University of Cambridge, UK, 1992.
- [9] HAN, Y.-S.—KIM, L.—CHA, J.-W.: *Computing User Reputation in a Social Network of Web 2.0*. *Computing and Informatics*, Vol. 31, 2012, pp. 447–462.
- [10] HAYNES, T.—SEN, S.: *Evolving Behavioral Strategies in Predators and Prey*. *Adaptation and Learning in Multi-Agent Systems*, 1996, pp. 113–126, doi: 10.1007/3-540-60923-7\_22.
- [11] ISHIWAKA, Y.—SATO, T.—KAKAZU, Y.: *An Approach to the Pursuit Problem on a Heterogeneous Multiagent System Using Reinforcement Learning*. *Robotics and Autonomous Systems*, Vol. 43, 2003, pp. 245–256, doi: 10.1016/s0921-8890(03)00040-x.
- [12] KANE, G.—PRIETULA, M.: *Influence and Structure: Extending a Model of Organizational Learning*. Twelfth Annual Organizational Winter Science Conference, 2006.
- [13] KAZIENKO, P.—BRODKA, P.—MUSIAL, K.—GAWORECKI, J.: *Multi-Layered Social Network Creation Based on Bibliographic Data*. *SocialCom/PASSAT*, IEEE Computer Society, 2010, pp. 407–412, doi: 10.1109/socialcom.2010.65.
- [14] KRYZA, B.—KROL, D.—WRZESZCZ, M.—DUTKA, L.—KITOWSKI, J.: *Interactive Cloud Data Farming Environment for Military Mission Planning Support*. *Computer Science*, Vol. 23, 2012, No. 3, pp. 89–100.
- [15] KVASSAY, M.—HLUCHÝ, L.—KRYZA, B.—KITOWSKI, J.—ŠELENĚ, M.—DLUGOLINSKÝ, S.—LACLAVÍK, M.: *Combining Object-Oriented and Ontology-Based Approaches in Human Behaviour Modelling*. *IEEE 9<sup>th</sup> International Symposium*



- on Applied Machine Intelligence and Informatics (SAMI), 2011, pp. 177–182, doi: 10.1109/sami.2011.5738871.
- [16] LACLAVÍK, M.—DLUGOLINSKÝ, S.—ŠELENĚ, M.—KVASSAY, M.—SCHNEIDER, B.—BRACKER, H.—WRZESZCZ, M.—KITOWSKI, J.—HLUCHÝ, L.: Agent-Based Simulation Platform Evaluation in the Context of Human Behavior Modeling. In: Dechesne, F., Hattori, H., ter Mors, A., Such, J. M., Weyns, D., Dignum, F. (Eds.): *Advanced Agent Technology (AAMAS 2011)*. Springer, Lecture Notes in Computer Science, Vol. 7068, 2011, pp. 396–410.
- [17] LATANE, B.: Dynamic Social Impact. Robust Predictions from Simple Theory. In: Hegselmann, R., Mueller, U., Troitzsch, K. G. (Eds.): *Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*. Springer, Theory and Decision Library, Vol. 23, 1996, pp. 287–310.
- [18] MAES, P.: *Situated Agents Can Have Goals. Designing Autonomous Agents*. MIT-Bradford Press, 1991.
- [19] MARCH, J. G.: Exploration and Exploitation in Organizational Learning. *Organization Science*, Vol. 2, 1991, No. 1, pp. 71–87.
- [20] MILLINGTON, I.: *Artificial Intelligence for Games*. Elsevier Inc., 2006.
- [21] NAZIR, F.—PRENDINGER, H.—SENEVIRATNE, A.: Participatory Mobile Social Network Simulation Environment. 2010 IEEE International Conference on Communications (ICC), IEEE, 2010, pp. 1–6, doi: 10.1109/ICC.2010.5502197.
- [22] ONO, N.—OHIRA, T.—RAHMANI, A. T.: Emergent Organization of Interspecies Communication in Q-Learning Artificial Organisms. In: Moran, F. et al. (Eds.): *Proceedings of the Third European Conference on Advances in Artificial Life*. Springer, Berlin, 1995, doi: 10.1007/3-540-59496-5\_314.
- [23] PRITCHARD, M.: A High-Performance Tile-Based Line-of-Sight and Search System In: DeLoura, M. A. (Ed.): *Games Programming Gems 2*, Charles River Media, 2001.
- [24] ROLLINGS, A.—MORRIS, D.: *Game Architecture and Design: A New Edition*. New Riders Publishing, 2004.
- [25] RUSSELL, S.—NORVIG, P.: *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 1995.
- [26] SABATER, J.—SIERRA, C.: Reputation and Social Network Analysis in Multi-Agent Systems. Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02). ACM, 2002, pp. 475–482, doi: 10.1145/544741.544854.
- [27] SCHMIDT, B.: Modelling of Human Behaviour: The Pecs Reference Model. 14<sup>th</sup> European Simulation Symposium, 2002.
- [28] SNOOK, G.: Simplified 3D Movement and Pathfinding Using Navigation Meshes. In: DeLoura, M. A. (Ed.): *Games Programming Gems*, Charles River Media, 2000.
- [29] STOUT, B.: The Basics of A\* for Path Planning. In: DeLoura, M. A. (Ed.): *Games Programming Gems*. Charles River Media, 2000.
- [30] SVAROVSKY, J.: Game Trees. In: DeLoura, M. A. (Ed.): *Games Programming Gems*, Charles River Media, 2000.

- [31] TAN, A. H.—XIAO, D.: Self-Organizing Cognitive Agents and Reinforcement Learning in a Multi-Agent Environment. Proceedings of IEEE/ACM/WIC International Conference on Intelligent Agent Technologies, 2005.
- [32] THOMAS, D.—HUNT, A.: State Machines. *IEEE Software*, Vol. 19, 2002, No. 6, pp. 10–12, doi: 10.1109/ms.2002.1049380.
- [33] VERE, S.—BICKMORE, T.: A Basic Agent. *Computational Intelligence*, Vol. 6, 1990, pp. 41–60, doi: 10.1111/j.1467-8640.1990.tb00128.x.
- [34] WOOLDRIDGE, M.: *An Introduction to Multiagent Systems*. John Wiley & Sons, 2002.
- [35] WOOLDRIDGE, M.—JENNINGS, N. R.: Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, Vol. 10, 1995, No. 2, pp. 115–152, doi: 10.1017/s0269888900008122.
- [36] WRZESZCZ, M.—KITOWSKI, J.: Creation of Agent's Vision of Social Network Through Episodic Memory. *Lecture Notes in Computer Science*, Vol. 8385, 2014, pp. 741–750, doi: 10.1007/978-3-642-55195-6\_70.
- [37] WRZESZCZ, M.—KITOWSKI, J.: Mobile Social Networks for Live Meetings. *Computer Science*, Vol. 13, 2012, No. 24, pp. 87–100.
- [38] XIAO, D.—TAN, A. H.: Cooperative Cognitive Agents and Reinforcement Learning in Pursuit Game. Proceedings of Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS '05), Singapore, 2005.
- [39] ZADEH, L. A.: Is There a Need for Fuzzy Logic? Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2008), 2008, doi: 10.1109/nafigs.2008.4531354.
- [40] ZOU, H.—ZONG, L.—LIU, H.—WANG, C.—QU, Z.—QU, Y.: Optimized Application and Practice of A\* Algorithm in Game Map Path-Finding. Proceedings of The 10<sup>th</sup> IEEE International Conference on Computer and Information Technology, 2010, pp. 2138–2142, doi: 10.1109/cit.2010.365.



**Michał WRZESZCZ** is a Ph.D. student of computer science at the AGH University of Science and Technology in Cracow, Poland. He is an author or co-author of over 20 scientific papers and conference contributions. His research interests are artificial intelligence and social networks as well as distributed computing and remote data access.



**Jarosław KOŹLAK** works as Assistant Professor at the Department of Computer Science of the AGH University of Science and Technology in Krakow, Poland. He received his Ph.D. degree in computer science from the Institut National Polytechnique de Grenoble (INPG), France and AGH University of Science and Technology, Krakow, Poland in 2000 and D.Sc. (habilitation) from AGH University of Science and Technology, Krakow, Poland in 2014. His research interests include multi-agent systems, transportation modelling and optimization, social network analysis. He is an author or co-author of about 120 scientific publications.



**Jacek KITOWSKI** is Full Professor of computer science and Head of Computer Systems Group at the Department of Computer Science of the AGH University of Science and Technology in Cracow, Poland and senior researcher at the Academic Computer Centre CYFRONET-AGH, responsible for developing high-performance systems and grid/cloud environments. He is the Head of International Affairs at ACK CYFRONET AGH. He is author or co-author of about 240 scientific papers. His topics of interest include large-scale computations, Grid services and cloud computing, distributed storage systems, high availability systems, network computing, knowledge engineering. He is a member of program committees of many conferences, participant of many international and national projects, funded by the European Commission, European Defense Agency, Polish National Centre for Research and Development and Polish National Science Centre. Director of PLGrid Consortium running PLGrid e-infrastructure in Poland for scientific computing. Member of Ministry Expert Body for Scientific Investments.