A New Approach to the Design of Collective Behavior in Multi-Agent Systems

Dániel László Kovács

Department of Measurement and Information Systems, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, P.O.box 91, H-1521 Budapest, Hungary
dkovacs@mit.bme.hu

Abstract: The problem of designing a given social behavior in a multi-agent system is a well known issue, yet there is still no general concept to solve it. In fact, there is still no theory, that connects the individual behavior of agents with the collective behavior of the multi-agent system in general. Nonetheless there are theories, which capture some profound aspects of the problem. One of the foremost is the theory of implementation of social choice rules. However the roots of the theory lie in social sciences, so its approach is not universally suitable. This article presents a new approach to the problem: a high-level agent-model for description, design and analysis of collective behavior in multi-agent systems.

Keywords: game theory, implementation of social choice rules, multi-agent systems

1 Introduction

The problem of designing a given social behavior (e.g. cooperative, optimal) in a Multi-Agent System (MAS) is a well known issue, yet there is still no general concept to solve it. In fact, there is no general theory, that connects the individual behavior of agents with the collective behavior of the MAS. Nonetheless there are theories, which capture some profound aspects of the problem. One of the foremost is the theory of implementation of social choice rules. However the roots of the theory lie in social sciences, so its approach is not universally suitable for MAS design. This article presents a new approach to the problem: a high-level agent-model for description, design and analysis of collective behavior in MAS.

MAS are usually considered from the perspective of intelligent agents [1]. An *agent* “can be anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.” [2]. This means, that if an agent’s actions depend on its senses, then it must have some representation of the environment, i.e. some kind of a *percept*. A percept is typically not equivalent to the environment, because the environment is usually not fully accessible to the agent. Using percepts an agent is able to compute its next action. Moreover, all the preceding percepts (the complete percept history) can have an effect on that choice. Consequently we may speak of *two levels* of environmental representation: an *outer representation* exterior to the agent, and an *inner representation*, inside the agent. It is the latter, upon which the agent's decision mechanism – choosing among its possible actions – may be placed. It is the task of the Designer to design this mechanism appropriately given the outer representation of the environment, and the agent's architecture (sensors, effectors, etc). This decision mechanism may depend on some special features of the environment to allow the agent to act effectively, e.g. there may be other agents, which make the environment dynamic. Such *multi-agent* situations require individual agents to consider other agents' activity for effective operation. Not only the past, or the present activity should be considered, but also events, which may occur in the future. Thus it is advantageous for an agent to *plan* its actions in advance, and to consider other agents' *planning* activity too.

Obviously the *goodness* (utility, payoff, etc) of such agents depends not only on the plan they execute, but also on the plans executed by others. This kind of strategic interaction is commonly modeled by *game theory* [3], where agents are called *players*, and their plans are called *strategies* [4]. Although game theory provides an elaborate description framework, it does not specify how the decision mechanism works. This makes game theory inappropriate for the design of collective behavior in MAS, where agents should act according to a specified (possibly optimal) rule of behavior. *Theory of implementation of social choice rules* [5] (a new branch in game theory) proposes a solution to this problem. However, it considers agents to be given. Therefore it specifies the decision mechanism not inside, but outside of them. This causes fundamental difficulties, which may be overcome, if the mechanism is specified within the agents.

This article introduces a new game theoretic approach to implementation of social choice rules: *virtual games*. Virtual games specify the mechanism within the agents, thus enabling the design of provably optimal collective behavior in MAS. The next sections will introduce fundamentals of game theory, and implementation theory. Then they'll proceed to the definition of virtual games. After the most important definitions, some essential results [10] are stated, followed by a conclusion and an outline of future research.

2 A Common Approach to Design of Collective Behavior in Multi-Agent Systems

Theory of implementation of social choice rules is used to handle problems of designing optimal social behavior. The population of agents is considered a *society*, which – as a collective entity – acts according to a *social choice rule* (SCR), a mapping from relevant underlying parameters to final outcomes. Thus, a SCR produces social alternatives (outcomes) depending on the private information (e.g. type, individual preferences) of the agents in the society. A single-valued SCR is called a *social choice function* (SCF). The implementation problem is then formulated as: “under what circumstances can one design a mechanism so that the private information of agents is truthfully elicited and the social optimum ends up being implemented?” [5]

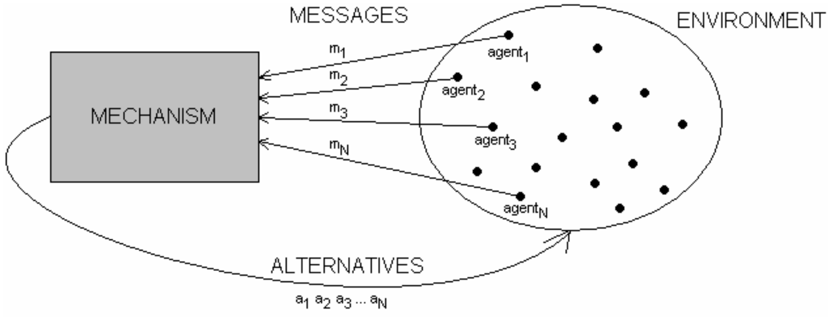

Figure 1
The implementation problem

Fig. 1 shows the implementation problem in more detail: *a Designer must construct a mechanism that implements a given SCR* by producing the same outcomes a_1, a_2, a_3, ..., a_N, supposing that the agents *1, 2, 3, ..., N* choose their messages (e.g. actions, strategies) m_1, m_2, m_3, ..., m_N according to a given game theoretical solution concept *S* (e.g. dominant strategies, Nash equilibrium). If it is possible to design such a mechanism for a given SCR, then the SCR is called *S-implementable*.

The above approach holds many advantages, since mechanisms can model social institutions, outer enforcement or even mutual agreement between agents. For instance it is shown [5], that if *S* is dominant (i.e. if each agent chooses its dominant strategy regardless of what the other agents choose), then only dictatorial SCFs are implementable[1].

[1] An SCR is dictatorial if it follows the preferences of one particular agent.

Despite its constructive results, the approach has also its weaknesses. In non-economical situations, e.g. in informatics, the Designer of an intelligent system (software agent, robot, etc) has *explicit control* over the system's decision mechanism (e.g. program [6]), unlike to a game theoretical solution concept, where the assumption about agents' decision mechanism is *implicit*. Why should every agent in a MAS act according to a given solution concept *S*? It is also a weakness, that agents are forced to act "through" a *central mechanism*, which has *global access* to the environment. This assumption is generally unrealistic when designing MAS, because agents mostly act in a decentralized way, and the Designer, or any mechanism – apart from trivial cases – has only *local access* to the environment (e.g. Internet, deep sea, surface of Mars). Moreover, it is also a drawback, that the approach guarantees implementation only when certain special conditions hold for the SCR (e.g. monotonicity, ordinality, incentive compatibility). Generally only *approximate implementation* is possible, i.e. *generally* an SCR is implementable only with some error. This type of implementation is called *virtual implementation* [7].

3 A New Approach: Virtual Games

To solve the above mentioned problems a new, high-level model of agent decision mechanism, called *virtual games*, is proposed. To give a detailed description of the concept, let us first introduce the fundamental notions of game theory: agents; pure and mixed strategies; agent-types; payoff functions; static Bayesian games; social choice functions; and finally, the notion of Bayesian Nash-equilibrium.

3.1 Game Theoretic Fundamentals

Let $N = \{1, 2, \ldots, n\}$ denote a finite, non-empty *set of agents*, S_i is the finite, non-empty *set of strategies* available to agent *i* $(i = 1, 2, \ldots, n)$. Now $s_i \in S_i$ denotes an arbitrary member of this set. A strategy associates an elementary action with every possible contingency of an agent. Let $s = (s_1, s_2, \ldots, s_n) \in \times_{i=1}^{n} S_i = S$ denote an arbitrary *strategy combination*. A strategy combination $s \in S$ prescribes a strategy $s_i \in S_i$ to every agent *i*. Agents choose their strategies simultaneously, without knowing each other's choice.

For the description of the uncertainty agents may face in MAS environments (deficient sensors; dynamic, non-deterministic behavior of other agents, etc), let us introduce types [8]. Types of an agent can be used to represent the type of private information, resources, processing abilities, etc, it may possess. Thus the uncertainty of an agent about other agents (e.g. because of the imperfection of its sensors) can be modeled as the uncertainty about the types of other agents. Let T_i denote the finite, non-empty *set of types* of agent *i*, and $t_i \in T_i$ an arbitrary type of agent *i*.

Now we can define the payoff of agents. The payoff of an agent describes its success (optimality, efficiency, etc) in the environment. Let $u_i : S \times T_i \to \Re$ denote the *payoff function* of agent *i*, where $u_i(s_1, s_2, \ldots, s_n; t_i) = u_i(s; t_i)$ is the payoff to agent *i* if the agents choose strategies $s = (s_1, s_2, \ldots, s_n) \in S$, and the active type of agent *i* is $t_i \in T_i$. This means, that the payoff of an agent *i* depends only on the strategy $s_i \in S_i$ it selected, its active type $t_i \in T_i$, and the strategies $s_{-i} = (s_1, s_2, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n) \in S_{-i}$ chosen by other agents.

The active type $t_i \in T_i$ of the agent *i* is supposed to be chosen by *Nature* with a probability $p_i(t_i)$, where $p_i \in \Delta(T_i)$ denotes a *probability distribution* over T_i. Every agent *i* knows only its own active type $t_i \in T_i$, but is uncertain about the active types $t_{-i} = (t_1, t_2, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) \in T_{-i}$ of others. To model this uncertainty, let us introduce a $p \in \Delta(T)$ joint probability distribution over $T = \times_{i=1}^{n} T_i$. Now the probability that the types of the agents are really $t = (t_1, t_2, \ldots, t_n)$ can be calculated as $p(t) = p_1(t_1) \cdot p_2(t_2) \cdot \ldots \cdot p_n(t_n)$, assuming that $p_1, p_2, \ldots, p_n$ are independent. The probability $p_i(t_{-i} | t_i)$ is called agent *i's belief* about other agents' types, t_{-i}, given its knowledge of its own type, t_i. Assuming, that $S_1, S_2, \ldots, S_n$, $T_1, T_2, \ldots, T_n$, $u_1, u_2, \ldots, u_n$, and $p_1, p_2, \ldots, p_n$ are *common knowledge* among the agents (i.e. everybody knows, that everybody knows, that…), the belief $p_i(t_{-i} | t_i)$ can be calculated by any of the agents using Bayes' rule:

$$p_i(t_{-i} | t_i) = \frac{p(t_{-i}, t_i)}{p(t_i)} = \frac{p(t_{-i}, t_i)}{\sum\limits_{t_{-i} \in T_{-i}} p(t_{-i}, t_i)}, \text{ where } p(t_{-i}, t_i) = p(t), \text{ and } t = (t_{-i}, t_i) \qquad (1)$$

Types enabled us to transform any incomplete information game to a game with imperfect information [8]. Incomplete information games are games, where some players are uncertain about the structure of the game (e.g. strategy sets, or utility functions of others), while imperfect information games are essentially the classic games introduced by von Neumann [3]. Collecting all of this information together, we have:

Definition 1 The ***normal-form representation of an n-player (static Bayesian) game*** specifies agents 1, 2, …, n, their strategy spaces $S_1, S_2, \ldots, S_n$, their type spaces $T_1, T_2, \ldots, T_n$, their payoff functions $u_1, u_2, \ldots, u_n$, and the probability distributions

$p_1, p_2, \ldots, p_n$. At the beginning of a play of the game Nature chooses agent types according to the independent probability distributions, and reveals type $t_i \in T_i$ only to agent *i*. After that agents choose their strategies simultaneously and execute them in parallel. Agent *i* gains a payoff depending on the chosen strategy-combination, and its active type $t_i \in T_i$. Such a game is denoted by a 5-tuple: $\Gamma = \left(N, \{S_i\}_{i\in N}, \{T_i\}_{i\in N}, \{u_i\}_{i\in N}, \{p_i\}_{i\in N}\right)$.

If agents are allowed to choose their strategies according to a probability distribution $q_i \in Q_i = \Delta(S_i)$, where $\sum_{s_i \in S_i} q_i(s_i) = 1$, and $q_i(s_i) \geq 0$ for every $s_i \in S_i$, then the strategies $s_i \in S_i$ are called *pure strategies*, while the probability distributions q_i are called *mixed strategies*. Now $q_i(s_i)$ denotes the probability, that agent *i* plays a given pure strategy s_i by playing the mixed strategy q_i. Thus mixed strategies generalize pure strategies. The *set of mixed strategy combinations* is constructed as $Q = \times_{i=1}^{n} Q_i$.

Utility functions also need to be generalized to support mixed strategies. Let $u_i : Q \times T_i \to \Re$ denote agent *i's payoff function*, where $u_i(q; t_i)$ is the payoff to agent *i* if agents choose mixed strategies $q = (q_1, q_2, \ldots, q_n) \in Q$, and agent *i*'s type is $t_i \in T_i$. With a slight abuse of notation, this utility can be written as the expectation above the payoffs of all pure strategy combinations:

$$u_i(q; t_i) = \sum_{s=(s_1,s_2,\ldots,s_n)\in S} q_1(s_1) \cdot q_2(s_2) \cdot \ldots \cdot q_n(s_n) \cdot u_i(s; t_i), \text{ where } q = (q_1, q_2, \ldots, q_n) \in Q \quad (2)$$

Before proceeding to the definition of the Nash equilibrium [9], let us first define strategy profiles $\{f_i(t_i)\}_{t_i \in T_i}$ of agent *i* $(i = 1, 2, \ldots, n)$*, and social choice functions. A* *strategy profile* is a mapping $f_i : T_i \to Q_i$, which associates a mixed strategy q_i to every type $t_i \in T_i$ of an agent *i*. Let $f = (f_1, f_2, \ldots, f_n) \in F = \times_{i=1}^{n} F_i$ denote a strategy profile combination, i.e. a *social choice function (SCF)*, and let $f(t) = (f_1(t_1), f_2(t_2), \ldots, f_n(t_n)) \in Q$ denote the mixed strategy combination provided by SCF f, given the agents' types are $t = (t_1, t_2, \ldots, t_n) \in T$. Now the expected payoff of agent *i* in case of an SCF f is:

$$u_i(f; t_i) = \sum_{t_{-i} \in T_{-i}} p_i(t_{-i} | t_i) \cdot u_i(f(t_{-i}, t_i); t_i), \text{ where } t = (t_{-i}, t_i) \quad (3)$$

In (3) the payoff function $u_i : F \times T_i \to \Re$ of agent *i* was redefined again (with a slight abuse of notation) to support SCFs. Because of the uncertainty about other agents' types, this is the payoff, that agent *i* with type $t_i \in T_i$ tries to maximize, not $u_i(f(t);t_i)$. The belief $p_i(t_{-i} \mid t_i)$ in (3) should be calculated according to (1), and the expected payoff $u_i(f(t);t_i)$ in case of a mixed strategy combination $f(t) \in Q$ should be calculated according to (2). Now we can define Bayesian Nash equilibrium:

Definition 2 In a static Bayesian game $\Gamma = \left(N, \{S_i\}_{i \in N}, \{T_i\}_{i \in N}, \{u_i\}_{i \in N}, \{p_i\}_{i \in N}\right)$ a SCF $f^* = \left(f_1^*, f_2^*, \ldots, f_n^*\right) \in F$ is a ***Bayesian Nash equilibrium*** if for each agent *i* and for each $t_i \in T_i$, $f_i^*(t_i) \in Q_i$ solves $\max_{q_i \in Q_i} \sum_{t_{-i} \in T_{-i}} p_i(t_{-i}|t_i) \cdot u_i\left(f_1^*(t_1), f_2^*(t_2), \ldots, f_{i-1}^*(t_{i-1}), q_i, f_{i+1}^*(t_{i+1}), \ldots, f_n^*(t_n); t_i\right)$.

3.2 Virtual Games

Section 3.1 introduced the fundamentals of game theory. Now we can proceed to discuss the solution of the problem outlined in Section 2. A new approach for implementation of social choice rules is proposed, called *virtual games*. This concept enables the construction of mechanisms, which provably implement any SCF exactly. Roughly speaking a virtual game is a part of this mechanism. Fig. 2 illustrates the concept:

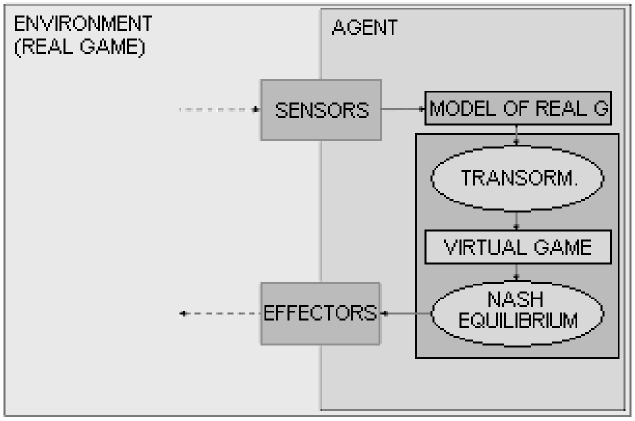

Figure 2

A new approach to the implementation problem

The mechanism is distributed among the agents. Every agent has a *decision mechanism*, which has three parts: a *transformation*, a *virtual game*, and a *function for selecting a Nash-equilibrium*. First the agent senses the outer representation of the environment: the *real game*. From that percept it creates an inner representation of the real game: the *model of the real game*. This is the input for the decision mechanism choosing among strategy profiles. Finally, the agent acts according to that profile.

Thus, virtual games are artificial constructs built from the model of the real game. They are not models of the real game, they are components of the decision mechanism of agents, and as such, they may be arbitrarily "far" from the model of the real game. Technically they differ from the model of the real game only in that they have different pure strategy spaces, called *pure virtual strategies*, and payoff functions, called *virtual payoff functions*. Formally this means, that every agent *i* has a finite, non-empty *set of pure virtual strategies* $V_i \subset Q_i$, a subset of the set of mixed strategies. These are the feasible strategies for agent *i*. Now the *virtual payoff function* of agent *i* is denoted by $v_i : V \times T_i \to \Re$, where $V = \times_{i=1}^{n} V_i$. Virtual payoff represents an agent's private valuation of the feasible strategic outcomes. A *virtual game* is then a normal-form static Bayesian game $\Gamma^* = \left(N, \{V_i\}_{i \in N}, \{T_i\}_{i \in N}, \{v_i\}_{i \in N}, \{p_i\}_{i \in N}\right)$. In this game the concepts of mixed strategies, mixed strategy combinations, their payoff, strategy profiles, social choice functions, their payoff, and Bayesian Nash equilibrium are defined similarly to the concepts introduced in Section 3.1.

A *mixed virtual strategy* of agent *i* is denoted by $r_i \in R_i = \Delta(V_i)$, where $r_i(q_i)$ denotes the probability, that agent *i* plays the pure virtual strategy $q_i \in V_i \subset Q_i$ by playing the mixed virtual strategy $r_i \in R_i$. The *set of mixed virtual strategy combinations* is denoted by $R = \times_{i=1}^{n} R_i$. The *virtual payoff function* for them is denoted by $v_i : R \times T_i \to \Re$, and the *virtual payoff* is calculated similarly to (2). Let $g_i : T_i \to R_i$ denote a *virtual strategy profile* of an agent *i* in a virtual game. An SCF *g* of the virtual game is called a *virtual social choice function (VSCF)*. The *virtual payoff* for a VSCF is calculated similarly to (3). A mixed virtual strategy $r_i \in R_i$ in the virtual game is *equivalent* to a mixed strategy $q_i \in Q_i$ in the model of the real game, and denoted $r_i \equiv q_i$, if $q_i(s_i) = \sum_{q_i^{(j)} \in V_i} r_i\left(q_i^{(j)}\right) \cdot q_i^{(j)}(s_i)$ holds for every $s_i \in S_i$. A mixed virtual strategy combination $r \in R$ is *equivalent* to a mixed strategy combination $q \in Q$, and denoted $r \equiv q$, if $r_i \equiv q_i$ holds for every $i = 1, 2, \ldots, n$. A VSCF g is *equivalent* to a SCF f, and denoted $g \equiv f$, if $g(t) \equiv f(t)$ holds for every $t \in T$.

Corollary 1 If given a mixed virtual strategy $r_i \in R_i$ and a mixed strategy $q_i \in V_i \subset Q_i$ which is also pure virtual strategy, where $r_i(q_i) = 1$ holds, then $r_i \equiv q_i$.

Now it is possible to state the result, which is a key step in showing that with decision mechanisms based on virtual games any SCF is exactly implementable.

Theorem 1 If in a virtual game $\Gamma^* = \left(N, \{V_i\}_{i\in N}, \{T_i\}_{i\in N}, \{v_i\}_{i\in N}, \{p_i\}_{i\in N}\right)$ constructed for a static Bayesian game $\Gamma = \left(N, \{S_i\}_{i\in N}, \{T_i\}_{i\in N}, \{u_i\}_{i\in N}, \{p_i\}_{i\in N}\right)$ for every $t = (t_1, t_2, \ldots, t_n) \in T$ exists a $q^{(t)} = \left(q_1^{(t)}, q_2^{(t)}, \ldots, q_n^{(t)}\right) \in V \subset Q$ pure virtual strategy combination such that for every $i = 1, 2, \ldots, n$ $v_i(q, t_i) = \begin{cases} 1, & q = q^{(t)} \\ 0, & q \in V \setminus \left\{q^{(t)}\right\} \end{cases}$ holds, then the only Bayesian Nash equilibrium of the virtual game Γ^* that yields maximal virtual payoff for every $i = 1, 2, \ldots, n$ is the VSCF $g^* = \left(g_1^*, g_2^*, \ldots, g_n^*\right)$, where for every $t = (t_1, t_2, \ldots, t_n) \in T$ $g^*(t) = \left(g_1^*(t_1), g_2^*(t_2), \ldots, g_n^*(t_n)\right) \in R$ is a mixed virtual strategy combination such that $g_i^*(t_i)\left(q_i^{(t)}\right) = 1$ holds for every $i = 1, 2, \ldots, n$, i.e. $g^*(t) \equiv q^{(t)} = \left(q_1^{(t)}, q_2^{(t)}, \ldots, q_n^{(t)}\right) \in Q$ for every $t \in T$.

Theorem 1 guarantees a unique Bayesian Nash equilibrium in virtual games, where the virtual payoff functions of agents are such, that for every type $t \in T$ their value is zero for all except one pure virtual strategy combination $q^{(t)} = \left(q_1^{(t)}, q_2^{(t)}, \ldots, q_n^{(t)}\right) \in V \subset Q$, where it is one. The theorem proves this proposition by first showing, that if every agent *i* plays according to a virtual strategy profile g_i^*, where $g_i^*(t_i)\left(q_i^{(t)}\right) = 1$, then the VSCF $g^* = \left(g_1^*, g_2^*, \ldots, g_n^*\right)$ is a Bayesian Nash equilibrium of the virtual game. Second, it proves (by contradiction) that this is a unique Bayesian Nash equilibrium of that virtual game in a sense that it is maximal for every agent. A proof of the theorem can be found in [10].

To use this result, the notion of game theoretical solution concepts and implementation need to be defined. Let S be a *game theoretical solution concept*. Given a game Γ we denote by $S(\Gamma) \in 2^F$ the set of strategy profiles (SCF's) that are recommended by S in game Γ. An SCF f in Γ is S*-implementable* if there exists a virtual game Γ^* constructed for Γ, such that $f \equiv S\left(\Gamma^*\right)$. Now the main result of the article can be stated as follows:

Theorem 2 Any SCF of any static Bayesian game is Bayesian Nash-implementable.

Theorem 2 uses Theorem 1 to prove its statement in the following way: first it takes an arbitrary Bayesian game Γ, and an arbitrary SCF f in Γ. Then it constructs a virtual game Γ^* such, that for every agent *i* the set of virtual pure strategies V_i is the set of mixed strategies $q_i = f(t_i)$ recommended by the SCF f, i.e. $V_i \in \{f(t_i)\}_{t_i \in T_i}$, and the virtual payoff function v_i is zero for all except one virtual pure strategy combination $q=f(t)$, where its value is one. In this case Theorem 1 guarantees, that the only Bayesian Nash equilibrium of the virtual game Γ^* is the VSCF, where every agent *i* with type $t_i \in T_i$ plays $f(t_i)$, i.e. $f \equiv B(\Gamma^*)$, where $B(\bullet)$ denotes the game theoretical solution concept of Bayesian Nash equilibrium.

A proof of the Theorem 2 can be found in [10]. In a game theoretical sense the result is independent of the accuracy of agents' modelling abilities. Theorem 2 states only that any SCF of any static Bayesian game can be implemented (even by virtual games with special binary payoffs) in case when agents act according to the maximal Bayesian Nash equilibrium of the virtual game constructed for the given static Bayesian game. Nonetheless, when players are considered agents, there is no guarantee, that they will use the same virtual game, because – by definition – they construct virtual games upon their model of the real game (see Fig. 2), and this model may be different among the agents. Thus, the results in Theorem 2 apply only to situations, when agents have the same virtual game. I assume that it is the task of the Designer to construct agents that way. Any relaxation of the assumptions is the task of future research.

Conclusions

The results in this article enable a high-level description, design and analysis of agents' decision mechanism in MAS. The results overcome the weaknesses of the theory of implementation of social choice rules. It is shown, that arbitrary collective behavior can be achieved exactly and in general. Consequently optimal (e.g. Pareto-optimal, bounded optimal [6]) SCFs are implementable, e.g. to optimize agents' communication protocols (strategic interaction); resource usage (in connection with the utility of agents); or the quality of various services of MAS (in connection with the optimality of the SCF). A uniform framework is provided to describe, design and analyze social behaviour. Elaborate distinctions can be made in the incentives, private valuation and preferences of agents if modelling their decision mechanism via virtual games. However, only virtual games with binary payoffs were discussed. The examination of virtual games with non-binary payoff functions is the task of future research. This research will mainly concentrate on connecting the concept of virtual games to existing low-level agent architectures (e.g. [11], [12]) and integrating it into a unified theory of designing and analysing intelligent multi-agent systems.

References

[1] Weiss, G.: Multiagent Systems. MIT Press (1999)

[2] Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall (1995)

[3] Neumann, J., Morgenstern, O.: Theory of games and economic behavior. Princeton University Press (1947)

[4] Bowling, M., Jensen, R., Veloso, M.: A Formalization of Equilibria for Multiagent planning. In: Proceedings of IJCAI'03 Workshop (2003) 1460-1462

[5] Serrano, R.: The Theory of Implementation of Social Choice Rules. In: SIAM Review, Vol. 46, (2004) 377-414

[6] Russell, S., Subramanian, D.: Provably bounded-optimal agents. In: Journal of AI Research, Vol. 2, (1995) 1-36

[7] Abreu, D., Sen, A.: Virtual Implementation in Nash Equilibrium. In: Econometrica, Vol. 59, (1991) 997-1021

[8] Harsányi, J. C.: Games with incomplete information played by Bayesian players I-II-III. In. Management Science, Vol. 14, (1967-1968) 159–182, 320–334, 486–502

[9] Nash, J. F.: Non-cooperative games. In. Annals of Mathematics, Vol. 54, (1951) 286–295

[10] Kovács, D. L.: Virtual Games: A New Approach to Implementation of Social Choice Rules. In Proceedings, M. Pěchouček, P. Petta, L. Z. Varga, editors, Multi-Agent Systems and Applications IV, 4th International Central and Eastern European Conference on Multi-Agent Systems, (CEEMAS 2005), Lecture Notes in Artificial Intelligence, Springer Verlag (2005)

[11] Ferguson, I. A.: TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents. Ph.D. Thesis, Clare Hall, University of Cambridge, UK (1992)

[12] Kovács, D. L.: Evolution of Intelligens Agents: a new approach to automatic plan design. In. Proceedings of IFAC Workshop on Control Applications of Optimization, Elsevier (2003)