České vysoké učení technické v Praze,
Fakulta elektrotechnická

Czech Technical University in Prague,
Faculty of Electrical Engineering

Dr. Ing. Michal Pěchouček, M.Sc.

Sociální uvažování ve výpočetních multi-agentních systémech

Social Reasoning in Computational Multi-Agent Systems

Summary:

The concept of multi-agent system contributes to the fields of computer science and artificial intelligence by a societal and distributed approach to computing. Non-trivial problems can be solved, complex systems can simulated and controlled by a community of autonomous computational entities. These entities can be geographically distributed across wider network infrastructures.

The computational entities in multi-agent systems need to perform specific types of social interaction in order to achieve collective behavior and collective decision making. Socially oriented reasoning, the reasoning process that underlies rational interaction, is currently a subject of a deeper theoretical investigation and practical deployment in industrial applications.

This habilitation thesis provides an introduction to the field of multi-agent systems, required knowledge of a formal system for reasoning about an agent and a unified view on agents' social knowledge and reasoning processes maintaining validity and accurateness of social knowledge. In this thesis we present three principal approaches to handling agents social knowledge: the acquaintance model, stand-in agents and meta-agents. Experiments with socially oriented behavior and comments on practical applicability of the socially oriented reasoning in practical situations are presented also in the thesis.

Souhrn:

Koncept multi-agentních systémů přispívá do počítačových věd a umělé inteligence distribuovaným a na sociálním uvažování založeným přístupem. Netriviální problémy lze řešit, komplexní systémy lze simulovat a řídit pomocí komunity autonomních výpočetních jednotek. Tyto jednotky mohou být geograficky distribuované napříč širokou počítačovou sítí.

Autonomní výpočetní jednotky v multi-agentních systémech vykonávají specifické typy sociální interakce za účelem dosažení kolektivního chování a společného rozhodování. Model sociálního uvažování – výpočetní proces, který je založen na racionální interakci, je v současné době předmětem studia hlubších teoretických zkoumání a praktického vývoje průmyslových aplikací.

Tato habilitace předkládá úvod do oblasti multi-agentních systémů, nezbytné znalosti formálního systému pro modelování uvažování o ostatních agentech a unifikovaný pohled na sociální znalosti agentů a uvažovací procesy, které udržují pravdivost a přesnost sociální znalosti. V této práci jsou prezentovány tři základní přístupy ke správě a manipulaci se sociální znalostí – sociální modely, zástupní agenti a meta-agenti. V této práci jsou rovněž prezentovány experimenty se sociálním chováním a jsou diskutována praktická aplikovatelnost sociálního uvažování v praktických situacích.

Keywords:

multi-agent systems, agent technologies, social knowledge, meta-reasoning, coordination

Klíčová slova:

multi-agentní systémy, agentní technologie, sociální znalosti, meta-uvažování, koordinace

Contents

1 Introduction

An **agent** is an encapsulated computational (or physical, even human) system, that is situated in some environment, and that is capable of flexible, autonomous behavior in order to meet its design objective [1]. An agent has to be **autonomous** (have free will, responsible for choosing its actions, can cheat, can leave/join the community), **reactive** (be able to reconsider its activity according to the change of the environment in timely fashion), **proactive** (ability to maintain agents long term intention) and **sociable** (be able to collaborate, communicate, form coalitions and teams).

The architecture of a single agent usually consists of the **agent's body** and the **agent's wrapper**. We can also say that the body – a functional core of an agent – is encapsulated by the wrapper in order to put together an agent. The wrapper accounts for the inter-agent communication and real-time reactivity. The body is an agent's functional component, responsible for carrying out the main functional performance of the agent locally. It is usually not acquiring the information about the other members of the community, their capabilities, duties etc. This is the wrapper, which is responsible for communicating with the other agents, for collecting information about the intents, goals, capabilities, load, reliability etc. The agent body processes these information and makes a use of them.

A **multi-agent system** is a collection of fully autonomous rational agents that act independently on their owner's/designer's behalf (fulfilling the design objective), are able to communicate, interact, coordinate their activities, collaborate, negotiate with other agents situated in the environment.

The agents can be (i) physically distributed or all running on a single machine (ii) cooperative or self-interested (iii) all agents designed by one developer (or a company) or different agents developed by different designers.

In the case of a singular agent architecture, where we talked about the 'style' of operation, methods of reasoning or approaches to implementation of a singular agent. However, when we talk about a multi-agent architecture we mean specific aspects of the multi-agent community organization, ways of performing distributed decision making, the level of cooperation or sharing the goals. In this section we will introduce various categories of classification the multi-agent architectures and methods of collaboration and coordination.

2 Social Knowledge in Multi-Agent System

Agents, in order to perform rational behavior and sensible decision making, maintain knowledge in their memories. Knowledge is of various kind. It can be classified according to its source (empirical or theoretic), orientation (object-level, meta-level), use (propositional knowledge, intentional knowledge, knowledge of reference, knowledge of procedure). When we talk about the subject of the knowledge we introduce two special classes of knowledge:

– **self-knowledge** – beliefs that the agent maintain about itself,

– **social knowledge** – beliefs that the agent is aware of about the other members of the community.

It is clear that an agent may be aware of knowledge that is neither self-knowledge nor social knowledge. This can be e.g. **background knowledge** – knowledge that is true a priorty, knowledge about the environment, or various **problem solving knowledge** or case-specific expert knowledge. The latter is the case especially of the legacy systems integration. In such situations the agent's computational core is unaware of the rest of the multi-agent community, while the agents' wrapper implement all the interaction. All knowledge that the agent's core is processing when implementing intelligent reasoning is regarded as neither social nor self-knowledge.

Identification of the social/self knowledge in agents' knowledge bases is an uneasy task indeed. In order to do so, let us introduce the concept of **social-belief formula**. Let us have a special predicate $\beta_A(\varphi)$ that is telling about the formula φ that it gives an agent A some information about the agent A and is believed by some other agent. If $\beta_A(\varphi)$ is true, φ is regarded as social-belief formula. The following is this valid:

$$\beta_A(\varphi) \Leftarrow \exists B : (\mathsf{Bel}\ B\ \varphi) \wedge A \in \arg(\varphi) \tag{1}$$

Not only that. We also need to include formulas that give the agent some information about a collective of agents where the subjective agent is a member.

$$\beta_A(\varphi) \Leftarrow \exists B : (\mathsf{Bel}\ B\ \varphi) \wedge \exists \theta : A \in \theta \wedge \theta \in \arg(\varphi) \tag{2}$$

Intuitively, we would define agent A's **self-belief formula** by the use of the same predicate $\beta_A(\varphi)$. Very often it can happen that some piece of information would be self-knowledge and social knowledge at the same time. We define the self-belief formula $\overline{\beta}_A(\varphi)$ as piece of knowledge that provides an information about the agent itself. Formula that provides information about the 'self' agent and its relation to yet another agent is understood as a social-belief formula. Formally:

$$\overline{\beta}_A(\varphi) \equiv \beta_A(\varphi) \wedge (\mathsf{Bel}\ A\ \varphi). \tag{3}$$

Agent's social knowledge is defined as a set of formulas that are valid, the agent believes that they are valid and at the same time they are classified as agent belief formulas. Formally:

$$\mathsf{sok}_A \equiv \{\varphi | \exists B, B \neq A : \beta_B(\varphi) \wedge \varphi \wedge (\mathsf{Bel}\ A\ \varphi)\} \tag{4}$$

At this point we only categorize agent's knowledge into separate sets – social knowledge (sok), self knowledge (sek) and the remaining e.g. background knowledge(bk). We will discuss their properties, methods of maintenance.

For implementing formal systems of automated reasoning about agents' social knowledge in multi-agent system we would need to extend the modal logic framework to yet another order. This needs to be done anyway provided that we want to represent the nested beliefs or agent's capability to reason about other agent's mental positions. In the similar way the system of modal logic has been extended when defining the extreme properties of the $\mathsf{M{-}Bel}$ operator [1]. This is why we may say that the agent's social

level	social knowledge	example
0	minimal social knowledge	IP address, port, ACL
1	YP social knowledge	capabilities, services
2	agent-properties	load, trust, relations
3	models of behavior	intents, preferences

Table 1: Level of social knowledge – examples

knowledge is a subset of agents belief, defined as a higher order $\mathsf{KD45}$ modal operator. This is why all the axioms that are valid in the system of belief cannot be made invalid in any of its subsets. Consequently, the system social belief in multi-agent system is a $\mathsf{KD45}$ higher order modal system and agent social knowledge is represented as $\mathsf{KDT45}$ higher modal system.

Social knowledge represents both **necessary** and **optional** information which an agent needs for its efficient operation in the multi-agent community. The social knowledge is mainly used for reduction of communication, acceleration of agents' internal reasoning processes, but also provides self-interested agents with a competitive advantage and allows agents to reason about the others in environments with partial communication accessibility. Processing social knowledge replaces voluminous computation among many agents.

Social knowledge is an inevitable knowledge structure of the agent systems operation. Agents really need to know one about the other otherwise at least the simplest possible interaction is not be possible. On the other hand, complicated reasoning mechanisms that result in a speculative knowledge of the future course of operation of the alien agents makes the agent powerful, while it is not an inevitable characteristics of an autonomous agent. This is why, we categorize the agent's social knowledge as depicted in Table 2.

The simplest possible while inevitable instance of social knowledge are pieces of information that facilitate agents interaction – knowledge of symbolic name, physical address, the appropriate instance of the agent communication language (ACL). More sophisticated social knowledge is the *yellow-page* (YP) list that collects the services the particular agent provides the community with. Second level social knowledge provides the agents with the information about other agents nonpermanent properties, e.g. computational load, trust and relations with other members of the multi-agent community. The higher level types of social knowledge, the more sophisticated models of agents behavior that are used for modelling agents intent, predicting future course of behavior can be represented. For more details about various types of the social knowledge see Section 3.

From the point of the knowledge maintenance perspective, we distinguish between the several levels of sophistication/complexity of the social knowledge maintenance algorithms:

– **centralized maintenance** – various facilitators and platforms components

– maintenance by dedicated **middle agents** – brokers, mediators, matchmakers

– **individual maintenance** – periodical revisions or subscriptions-based contraction, and

– maintenance by **meta-reasoning** – by monitoring, meta-agents, reflection.

For more details about different mechanisms for social knowledge maintenance see Section 4. Maintenance mechanisms and types of social knowledge are very closely linked.

3 Types of Social Knowledge

In the following, we will discuss the various types of social knowledge in more details. We will comment on the nature, type and other properties of the social knowledge.

3.1 Minimal Social Knowledge

The minimal social knowledge in multi-agent systems is implemented by the *white-page* (WP) list. The white page list is a collection of information about all the members of the agent's total neighborhood – α. Here the agents maintain the information about their IP physical addresses, port number, their ACL language they use for communication. The white-page list represents minimal and mandatory requirements for interaction among the agents.

The white-page list information is very often maintained by a special registration agent who the new created agents register with and it in turn provides all the registered agents with the information about a newcomer. In a classical multi-agent integration environment (such as JADE [2]) this information is maintained by a special platform component AMS (Agent Management System).

In an open multi-agent system it is hardly possible to keep one central repository of all the white-page information. An open multi-agent system very often integrates several platforms, where each administers the information about the agents independently. Therefore, the platform identification is very often a part of the agent's physical name. The platforms may, while need not, to share the information across the whole system.

3.2 First Level Social Knowledge

The first level social knowledge in multi-agent systems is represented by the *yellow-page* (YP) type of information. YP related knowledge contains information about the services and skills that the particular agent provides to the community. Sophisticated mechanisms for YP knowledge maintenance improve the collaborative properties of the multi-agent system.

An instance of the first level social knowledge is knowledge representing agents' ability to perform a task, achieve a goal or provide a service. This kind of knowledge is maintained very often centrally. For example in the FIPA Agent Abstract Architecture this knowledge is maintained by the *Directory Facilitator* component that is integrated at each of the participating platforms [3].

In loosely coupled and more flexible multi-agent systems the first level social knowledge is collected and maintained by the dedicated agents, e.g. *brokers*, *matchmakers* or *mediators* (see 4.2). Once an agent needs a specific service, the dedicated special agent is contacted in order to identify the best possible match.

In the non-collaborative environment the first level social knowledge may be regarded as a private information with respect to a part of the community. The agents excluded from the part of the community, where this social knowledge is shared, have no access to the information about agents' capability to meet goals and provide services. Advanced concepts of meta-reasoning can be deployed as social knowledge maintenance mechanisms in non-collaborative communities.

The first level social knowledge tends to be little dynamic. The facilitators of dedicated agents maintain a simple table matching agents and offering services. If agents change services frequently, each update needs to be communicated between the respective agent and the social knowledge adminstration component. This becomes very inefficient in situations with frequent social knowledge changes and infrequent requests for social knowledge. This is why the first level social knowledge does not contain any information about quantitative attributes of the provided service (e.g. price, due time).

3.3 Second Level Social Knowledge

In order to achieve rational and efficient collaboration between the agents, a richer form of social knowledge needs to be communicated. Agents need to be aware of agents' reliability, maintain and manipulate mutual trust, investigate each other communication, computational and operational load. The information about the nature and the amount of resources required for a successful fulfilment of the offered services is vital for efficient team action planning. In collaborative, but primarily in self-interested communities the information about price and a completion time is critical. All these pieces of information are regarded as the second level social knowledge.

In the most of currently deployed multi-agent systems little or no support for maintenance of the second level social knowledge is available. This is replaced by negotiation process, where the required information (e.g. price) is communicated directly between the agents. In the single-criteria decision making, classical methods such as contract-net-protocol [4], [5] or advanced methods of combinatorial auctions [6], [7] are adopted.

In complex communities, when solving rather complex planning problems or in the real-time operations, these methods are not sufficient due to high requirements for communication resources. With an increasing amount of social knowledge maintained by the agents and decreasing requirements for communication traffic, the operation of the multi-agent system as a whole shall be made more efficient. However, with all the agents knowing everything one about the other, the requirements for computational resources for the agent's internal reasoning process (e.g. matching services and finding the optimal provider) may become very high. The requirements for communication resources shall be in certain balance with the requirements for computational resources required for internal reasoning.

The second level social knowledge is usually stored in agent's acquaintance models (see Section 5.1). The knowledge stored in acquaintance model can be maintained primarily by individual methods of knowledge maintenance. However some can be also maintained by the concept of mobile stand-in agents (see Section 5.2). In self-interested and adversarial communities the second level social knowledge needs to be maintained by special methods of meta-reasoning, more thoroughly described in Section 5.3.

3.4 Higher Level Social Knowledge

Higher level social knowledge is an essential component in non-collaborative multi-agent systems. Higher level social knowledge, also referred to meta-knowledge, allows agents to reconstruct private knowledge, model agents' intentions and future course of behavior. In many real life situations reconstruction of information about available resources provides agents with an important competitive advantage when participating in a negotiation process.

The central part of the higher level social knowledge is a model of the multi-agent community – a *community model*. Community model is a collection of formulae of a different kind. We have formulae that are either **social believes** or **auxiliary lemmas**. The community model consists of three elements:

- **Background knowledge** – that is the set of default, a priori knowledge known to the agent about the community (either lemmas or social belief formulas). Background knowledge is assumed to be always true and to be known to all agents before any event in the community happens. Background knowledge can have a number of different forms, while in our experiments we have been using the first-order logic.

- **Event set** – that is a collection of formulas (only social belief formulas) describing the events that have happened within the community (such as contract-net-protocol, sending a team allocation request, accepting or rejecting a team allocation request and informing about actual resources). The event set is empty at the beginning of monitoring process. When a new event is observed, the corresponding event formula is added to the event set.

- **Assumed model** – the set of formulas produced by the meta-reasoning processes (either lemmas or social belief formulas). Assumed model is the most dynamic component of the community model and it represents the current knowledge about the community (or other agent). It is repetitively revised during the meta-reasoning process according to the observed events.

For mechanisms how the community model can be maintained by the meta-reasoning processes see Section 4.4.

4 Social Knowledge (SK) Maintenance

Now let us discuss the different approaches to social knowledge maintenance. In the following we will distinguish between two specific instances of agents

- **social knowledge provider** (SKP), the agent that the respective social knowledge is about and

- **social knowledge requestor** (SKR), the agent who 'knows' and uses the respective social knowledge.

4.1 Centralized SK Maintenance

Centralized social knowledge maintenance and exploitation is very common in many situations. It is easy to implement in multi-agent systems and possible duplication and redundancy can be avoided. The difficult side of central maintenance is communication fragility. Agents increase their communication traffic and computational load of the central component. The central component may become a bottleneck in large scale or real-time applications. In practical systems, one cannot go without any centrality completely. Central components are inevitable at least for the registration phase of the life-cycle of multi-agent systems. An example of a centralized social knowledge maintenance is e.g. a facilitator, who is a communication interface among collaborating agents [8].

4.2 SK Maintenance by Middle Agents

A lot of research and development attention has been paid to the concept of specific agents that provide social knowledge maintainable as a service to the other agents. Social knowledge can be administered by loosely coupled agents such *brokers*, which are responsible for finding the best possible addressee of the transmitted message [9], *matchmakers* who also suggests cooperation patterns that may be equally used in the future [10], or *mediators*, who besides facilitating, brokerage and matchmaking coordinate the agents by suggesting and promoting new cooperation patterns among them [9]. If these agents are tightly connected to the platform, they have been classified as *middle agents* [11].

4.3 Individual SK Maintenance

Even higher degree of agents' independence is implemented by various kinds of individually organised social knowledge administration. In such cases social knowledge is maintained either by SKP or SKR. The agents' social knowledge is stored in a special knowledge structure denoted as an agent's acquaintance model. There is a number of ways how the acquaintance model can be maintained. In the collaborative environment, we distinguish between two principal approaches:

– a **pull** model of the knowledge maintenance is often implemented by *periodical revisions* [12] when the SKR periodically queries the SKP for updates of the relevant information.

– a **push** model of the knowledge maintenance can be implemented by e.g. *subscribe-advertise* protocol. The SKR subscribes the SKP for specific information and the SKP reports on updates of the relevant pieces of information upon changes.

4.4 SK Maintenance by Meta-Reasoning

Until now we have been discussing the role of social knowledge in a collaborative environment, where the SKP is ready to share the respective information with the SKR. In all the above described cases social knowledge has been maintained by communication. In many complex situations the agent is required to construct and maintain the acquaintance model autonomously – mainly by monitoring and higher forms of reasoning. Capability to reason about other agents (not only to use the social knowledge) is referred to as meta-reasoning.

We can distinguish between the two types of meta-reasoning activities. In **collaborative meta-reasoning** the agents that are subject of the meta-reasoning activity (SKP in this case) actively collaborate with the meta-reasoning agent (SKR) by e.g. providing copies of the communicated messages, while in **intrusive meta-reasoning** the SK providers do not support the meta-reasoning activity; the meta-reasoning agent needs to employ sophisticated meta-reasoning mechanisms in order to collect social knowledge.

The meta-reasoning processes in multi-agent system are based on manipulation of the *community model* introduced in Section 3.4. Meta-reasoning comprises three mutually interconnected computational processes:

- **monitoring** process makes sure that the agent knows the most it can get from monitoring the community of agents. Observed and recorded events serves as an input or a trigger for revision processes.
- **reasoning** process manipulates the model of the community so that true facts (other than directly observed) may be revealed; two key operations of the reasoning process are *model revision* and *model inspection*.
- **community revision** mechanism utilizes the community model (via the model inspection operation) to influence operation of the agent community.

5 Social Reasoning Implemented

Event though there is a number of different algorithms and techniques for implementing the appropriate social reasoning algorithms, in the following we will be discussing the three different key approaches: acquaintance model, stand-in agent and meta-agents.

5.1 Acquaintance Models

As mentioned earlier, the acquaintance model is a very specific knowledge structure that is usually located in the agent interaction wrapper. The acquaintance model is a knowledge and computational model of agents' mutual awareness. The acquaintance model stores all the relevant information that the agent knows about its collaborators and other agents that belong to its monitoring neighborhood [13]. Social knowledge stored in the acquaintance model is often structured according to the levels listed in Section 2. Besides the social knowledge structures the acquaintance model also need to contain the appropriate knowledge maintenance mechanisms that assures validity of the collected knowledge.

There have been several acquaintance model architectures suggested in the past, such as ARCHON, an Architecture for Cooperating Heterogeneous On-line Systems [14] [15], Coverage [16], Pleiades architecture of collaborative agents decision making over the collection of internet-based heterogeneous resources [17] or twin-base model [18].

We have designed a general acquaintance model architecture – 3bA (Tri-base Acquaintance Model) and validated its appropriateness in the manufacturing applications.

– **co-operator base** - maintains permanent information on co-operating agents (i.e.: their addresses, communication languages, and their predefined responsibilities). This type of knowledge is expected not to be changed very often.

– **task base** - stores in its *problem section* the general problem solving knowledge - (i) information on possible decompositions of the tasks to be coordinated by the agent and (ii) in its *plan section* it maintains the actual and most up-to-date plans on how to carry out those tasks, which are the most frequently delegated to the agent

– **state base** - stores in its *agent section* all information on the current load of co-operating agents. This part of the state base is updated frequently and informs the agent who is busy and who is available for collaboration. In the *task section* there is stored information on the status of tasks the agent is currently solving.

The agent is supposed to select an optimal plan from the plan section, where an appropriate number of plans prepared in advance is stored. By this it does not need to contract peer agents in order to find out the most appropriate (optimal) offers for further problem delegation. The model maintenance algorithms are based on a simple `subscribe/inform` mechanism. After parsing the problem section, each agent identifies possible collaborators and subscribes these for reporting on their statuses. The subscribed agent advertises its load, capabilities and task completion times and costs estimates either periodically or when either of these changes. This mechanism facilitates the agent to make the best decision with no further communication in the critical moment of the agent's decision making.

It is obvious that with an increasing quality of the acquaintance model, the quality of cooperation improves. A real challenge is how to make the acquaintance model compact in the sense of required memory space, efficient to be maintained and exploited, while still providing very useful and relevant information that optimizes the interaction between the agents.

One of our major contribution is a design of the new contraction mechanism – **iterative acquaintance models based contraction** (IAM) [19], that is very efficient in large communities negotiating about a very complex issue (such as supply chain management, complex logistics, non-trivial project driven production planning, etc.). The acquaintance model here is approximated, inexact information about the collaboration neighborhood. Each time the agent works with the acquaintance model, it produces the most optimal task decomposition and resource allocation among the agents, given the knowledge stored in the acquaintance model. If the acquaintance model is far too inexact, the SKP reject the suggested resource allocation. Such information improves the quality of the acquaintance model and new decomposition and resource allocation is computed.

The Figure 1 illustrate effectiveness of the suggested algorithm. Here we compare classical uniform sampling of the acquaintance model with IAM contraction. The solid straight line indicates optimal contraction (completely exact contraction value). In the logistic delivery negotiation among two agents we can see that while IAM needs 4 iterations to find the optimum, classical uniform sampling requires more than 20 samples in order to find a contract for both the right final duration and the right amount to be provided.

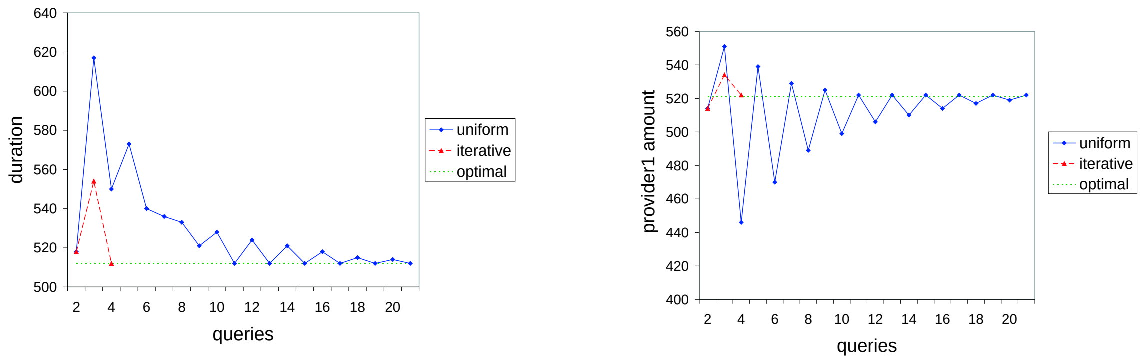

Figure 1: Partially linear distribution - comparison of algorithms - durations, amounts

5.2 Middle Agents

As previously explained the middle agents are typical examples of dedicated social knowledge mediators. These agents try to match the requirements of the social knowledge requestors with the appropriate social knowledge providers. Middle agent play an active role in the operation of the multi-agent community and are inevitable for assuring agents' rational cooperation. Examples of the middle agents are listed in Section 2.

Our original contribution to the research community investigating middle agents is by suggesting methods for middle-agents-based social knowledge maintainable in the disruptive environment, especially with temporal communication inaccessibility. Such situations can occur especially in the domain of ad-hoc networking.

We have designed the architecture of **stand-in agent** as a dedicated middle agent that represent a single agent or a community of agents in the situations when it/they become inaccessible from the rest of the multi-agent community. When designing stand-in agents one need to model an appropriate **level of meta-representation** of the missing agent or a community that stand-in agent needs to carry on. It is important that only the relevant pieces of information (both declarative and procedural) are represented. Besides, the knowledge maintained by the stand-in agent needs to be kept updated. The appropriate **mechanisms for updating** the knowledge needs to be also designed.

The most complicated problem in the stand-in agent design regards the appropriate allocation of the stan-in agents in the ad-hoc network. In addition to the stand-in agent architecture we have been also investigating the methods of **optimization of stand-in agents allocation**. There two different classes of methods, based on different assumptions:

- **uninformed** – when there is no information about how inaccessibility and the overall connectivity of the network can be in the future and
- **informed** – when probability of future communication encounters as well as future possible inaccessibility situations can be modelled and predicted.

In the latter case the stand-in agents swarm to the locations where there is a higher probability of future requested interaction and possibility of disconnection. The required

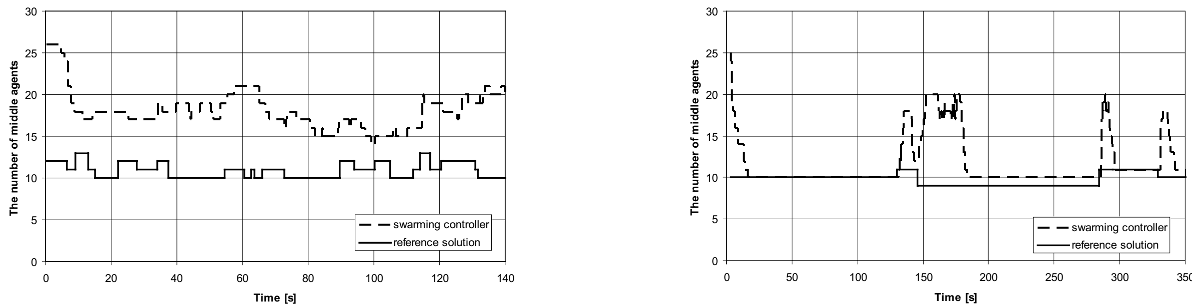

Figure 2: Adaptation of stand-in agent network to changing environment.

information needed for such an optimization is stored in agents' acquaintance models. Such a swarming control strategy is called *forward swarming strategy*. In the uniformed situations, we need to progress in backward fashion – *backward swarming strategy*. The stand-in agents need to swarm first in all relevant segments of the network and they need to die-out if unneeded for an extensive period of time.

Each of the approaches has its pros and cons. The forward swarming control is computationally efficient, as it tries to minimize the number of middle agents in the system and prevent the possible swarming explosion. This is why that approach seems to be particularly suitable for domains with high scalability and operational efficiency requirements. On the other hand, the backward swarming control has got an important advantage. This approach is substantially more domain independent, demands less knowledge about the environment nature and is more robust, as it doesn't explicitly use any prediction about the future of the community.

Two different mechanisms for backward swarming have been designed tested (i) social dominance and altruism models, where stand-ins will adapt their social role to the new configuration of the environment and (ii) and utility-based, where the concept of micro-payments have been used for an information update, indicating the usefulness of the received information. The Figure 2 presents adaptation of the stand-in agent network to frequently (left) and infrequently (right) changing environment.

5.3 Meta-Agents

Socially oriented reasoning provides the agents with higher level of intelligence and an increased problem solving and coordination capabilities. This very special quality of agents intelligence that allows reasoning about itself and about the collaborating or adversarial agents is referred to as capability of *reflective reasoning*. Unlike in classical computational systems, in the multi-agent system reflection is based on higher forms of socially oriented reasoning (both, each agent in a multi-agent system reasoning about itself and about the other agents).

Meta-reasoning, as a higher level social knowledge maintenance mechanism, is an important and inseparable part of reflective behavior of the multi-agent system [20]. Reflective reasoning and reflective behavior provide a unifying framework for meta-reasoning and self-learning in multi-agent system.

If we require any computational system to be reflective, it needs to be able to manipulate with symbols and perform computation in order to perform behavior that will meet its designed objective – *primary reasoning* and to reason about itself, its knowledge, problem solving strategies, scope of competence or record of past behavior – *reflective reasoning*.

There are three different types of reflective behaviour in multi-agent systems: *individual reflection* – reflection of a single agents knowledge, services, properties and behavioural capabilities, *collective reflection* – allows reflection of behaviour of the multi-agent system as a whole and *mutual reflection* – where one agent may want reason about one or many other members of the community.

The individual reflection is implemented by reflection on the level of the single agent and allows an agent to adapt and learn from experience of its past course of decision making. A classical self-learning mechanism (unlike collective reflection) needs to work in the 'on-line mode'. Firstly, it needs to react quickly and comply with the calculative rationality requirements. At the same time in needs to be very lightweight so that it will not slow down the agents primary functionality (such as planning in the case of planning agents, information retrieval in the case of database agents, etc.). In the field of artificial intelligence there have been many techniques for self-learning agents (such as multi-layer perceptron networks with backpropagation algorithm [21], or decision trees [22]). However these were by the major part case specific and there is still a need for a unified, lightweight and fast self-learning architecture.

Mutual reflection allows an agent to form hypothesis about another agent, either collaboratively or in the self-interested communities. Once an agent reasons about another agent in the self-interested fashion it uses meta-reasoning in order to form and maintain higher level social knowledge (such as believes, commitments, individual plans, etc.).

Collective reflection can be implemented in two quite different ways: (i) by the **reflective component** – a single specific agent or a community of dedicated agents, or (ii) **emergently** – all agents in the community contributing to reflective behavior.

In either way the agents are capable of meta-reasoning and forming hypothesis of the higher level social knowledge about individual agents and also about communities of agents. This allows reasoning about members of the collective, their history, beliefs, joint commitments, shared plans and collective organizational relationships.

If there is a reflective component in the multi-agent system that is instantiated by a single agent (or community of agents), we will refer to this agent as a **meta-agent**. The meta-agent manipulates the model (a self representation of the multi-agent system) by meta-reasoning. The introspective integrity is implemented by monitoring and the introspective force is implemented by the community revision process. The collective reflection can be used for improving the overall behavior of the multi-agent system, e.g. appropriate roles allocation, efficient task decomposition and delegation, bottleneck identification, trust assignment, etc.

Provided that there is a primary functionality of the multi-agent system specified, the **meta-agent** is defined as an independent agent that plays no role whatsoever in the primary functionality of the multi-agent system, and implements by meta-reasoning reflection of the multi-agent system.
This definition implies that the meta-agent cannot become a bottleneck of the system

primary operations. It works mainly with copies of communicated messages (if available), environmental observations, or the information gathered from communication between the agents and the meta-agent.

We have been working with different AI technologies supporting the meta-agent reasoning process. In principle the meta-reasoning algorithm can be divided into deductive and inductive mechanisms. We have been comparing behavior of the classical theorem proving mechanisms based on the resolution principle (as an example of deductive meta-reasoning) and version space algorithm and inductive logic programming (as an example of inductive meta-reasoning). The Figure 3 illustrates that the inductive logic programming outperforms deductive meta-reasoning. The main advantage of deductive meta-reasoning is that it never (unlike inductive) misclassifies.

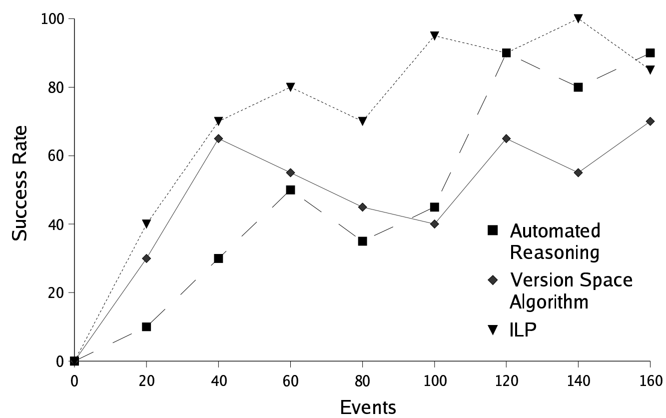

Figure 3: Comparison of deductive and inductive meta-reasoning algorithms

Not only the quality of the meta-reasoning output is important when designing the meta-agent, but one need to take care about the computational resources requirements. Different technologies can be used for different meta-reasoning tasks. In the applications such as real-time diagnostics visualization, intrusion detection applications the real-time aspects are very important, while in the applications targeted towards prediction, explanation or off-line learning the applications are less restrictive.

6 Conclusion

This lecture introduced the concept of social knowledge in the multi-agent systems. It has been explained that with different levels of social knowledge various maintenance mechanisms need to be employed. Table 2 specifies our experience in practical deployment of social knowledge in multi-agent systems. It shall be also noted that with an increasing complexity of agent knowledge (including social knowledge) the reasoning processes became more computationally demanding. While using the minimal social knowledge is very easy and does not put any extra computational burden on the agents, using higher level social knowledge would require very specific reasoning algorithms that would be able to carry out computationally rational reasoning in bounded domains.

mech./level	min.	1st ord.	2nd ord.	higher
centralized	××			
special agents		××	×	
individual		×	××	
meta-reasoning		×	×	××

Table 2: Level/Maintenance relation between the level of social knowledge and required mechanisms for social knowledge maintenance: ×× – typical, × – possible.

The lecture introduced three different approaches to social knowledge representation and socially oriented reasoning. Either of the approaches have got specific advantages and disadvantages.

The acquaintance model is an important source of information that would have to be repeatedly communicated otherwise or is not available in the situations of agents' short term inaccessibility. The nice advantage of acquaintance model is that they supports fully agent autonomy and independence. However, the acquaintance models provides rather *"shallow"* knowledge, that does not represent a complicated dynamics of agent's decision making, future course of intentions, resource allocation or negotiation preferences. This type of information is needed for inter-agent coordination in situation with longer-term inaccessibility.

The stand-in agents allow representation of the *"deeper"*, knowledge by migrating the parts of its programme in the network. This approach is particularly suitable for the situations where agent are not happy to share their knowledge as such and they can disclose some of the information within specific agent-to-agent interactions. Also the agents' knowledge may be dynamic and keep changing with the changing environment. This can be also hard to achieve by the sole acquaintance models, maintained by the client. On the other hand the concept of the stand-in agent is rather heavy-weight and can cause important slowdown of the multi-agent community if designed poorly. With backward swarming, there can be situations (a reaction to the update of the environment) where the community gets flooded by an enormous amount of stand-in agents. This may cause important operational bottlenecks in some phases of the lifecycle of the community. Forward swarming on the other requires an important amount of a' priori knowledge.

Meta-agent represent yet another approach to social knowledge maintenance. This approach is suitable particularly for autonomous construction of social knowledge, primarily from observation. Meta-agent approach can be beneficial in real-time situations where there is no time to communicate and share complex social knowledge among the agents. Similarly in non-trusted and adversarial communities the meta-agents can be used for detection of intruders or adversaries detection. Meta-agent can observe the community behavior and try to construct non-trivial social knowledge, information about roles and mental states of the agents. Unlike the previous two approaches that are designed primarily for social knowledge sharing, meta-agent are tailored primarily for the use of social knowledge detection in fast response or non-trusted communities.

This lecture has been rather abstract and the potential applications of the socially oriented reasoning and social knowledge in multi-agent system has not been discussed in greater details. However the research presented here has been primarily motivated by the industrial requirements that has been formulated in numerous industrial projects we have been involved in previously.

The acquaintance models have been in deployed and tested in the domain of production planning and supply chain management, particularaly in collaboration with Modelarna Liaz, SkodaAUTO, CertiCon, a.s. and gedas, sro. Acquaintance models have been also used in the coalition support prototype for planning humanitarian relief operations. This project was supported by the European Office for Aerospace Research and Development, UK. The concept of stand-in agents have been studied and investigated within the support of the project funded by Air Force Research Laboratory, Rome, NY. The meta-agents design and development was supported among others by the Office for Naval Research, Arlington. Meta-agents, stand-in agents and acquaintance models have been jointly applied in the underwater minesweeping exercise funded by the Office for Naval Research.

References

[1] Wooldridge, M.: Reasoning about Rational Agents. Intelligent robotics and autonomous agents. The MIT Press (2000)

[2] JADE: Java Agent Development Framework. `http://jade.tilab.com` (2004)

[3] FIPA: Foundation for intelligent physical agents. `http://www.fipa.org` (2004)

[4] Smith, B.: Reflection and semantics in lisp. In: Proc. 11th ACM Symposium on Principles of Logic Programming, Salt Like City, Utah, Also Xerox PARC Intelligent Systems Laboratory (1984)

[5] Sandholm, T.: Distributed Rational Decision Making. In: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge, MA. (1999) 201–258

[6] Sandholm, T.: Algorithm for optimal winner determination in combinatorial auctions. Artificial Intelligence **135** (2002) 1–54

[7] Rothkopf, M.H., Pekec, A., Harstad, R.M.: Computationally manageable combinatorial auctions. Technical Report 95-09, Certer for Discrete Mathematics & Theoretical Computer Science (1995)

[8] McGuire, J., Kuokka, D., Weber, J., Tenebaum, J., Gruber, T., Olsen, G.: Shade: Technology for knowledge-based collaborative engineering. Concurrent Engineering: Research and Applications **1(3)** (1993)

[9] Shen, W., Norrie, D., Barthes, J.: Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing. Taylor and Francis, London (2001)

[10] Decker, K., Sycara, K., Williamson, M.: Middle-agents for the internet. In: Proceedings of the 15th International Joint Conference on Artificial Intelligence, Nagoya, Japan (1997)

[11] Sycara, K.: Multi-agent infrastructure, agent discovery, middle agents for web services and interoperation. In: Multi-Agent Systems and Applications. Lecture Notes in Computer Science. 2068, Heidelberg : Springer-Verlag (2001) 17–49

[12] Štěpánková, O., Mařík, V., Lažanský, J.: Improving Cooperative Behaviour in Multiagent Systems. In: Advanced IT Tools. Chapman and Hall London (1996) 293–302

[13] Mařík, V., Pěchouček, M., Štěpánková, O.: Social knowledge in multi-agent systems. In Luck, M., Mařík, V., Štěpánková, O., eds.: Multi-Agent Systems and Applications. LNAI, Springer-Verlag, Heidelberg (2001)

[14] Wooldridge, M., N., J.: Intelligent agents: Theory and practice. The Knowledge Engineering Review **10** (1995) 115–152

[15] Wittig, T.: ARCHON: An Architecture for Multi-agent System. Ellis Horwood, Chichester (1992)

[16] Lamb, N., Preece, A.: Verification of multi-agent knowledge-based systems. In: Proceedings of ECAI-96 Workshop on Validation, Verification and Refinement of KBS. (1996)

[17] Sycara, K., Decker, K., Pannu, A., Williamson, M., Zeng, D.: Distributed intelligent agents. IEEE Expert **11** (1996) 36–46

[18] Cao, W., Bian, C.G., Hartvigsen, G.: Achieving efficient cooperation in a multi-agent system: The twin-base modeling. In Kandzia, P., Klusch, M., eds.: Cooperative Information Agents. Number 1202 in LNAI, Springer-Verlag, Heidelberg (1997) 210–221

[19] Lerch, O., Pěchouček, M.: Iterative acquaintance models based contraction. In: Joint IEEE/ACM Confernce on Intelligent Autonomous Agents (submitted for publication). (2005)

[20] Pěchouček, M., Norrie, D.: Knowledge structures for reflective multi-agent systems: On reasoning about other agents. Research Report 538, Calgary: University of Calgary (2000)

[21] Rumelhart, D.E., Hinton, G.E., Williams, R.J. In: Learning Internal Representations by Error Propagation. MIT Press, Cambridge, MA (1986) 318–362

[22] Quinlan, J.R.: Induction of decision trees. In Shavlik, J.W., Dietterich, T.G., eds.: Readings in Machine Learning. Kaufmann, San Mateo, CA (1990) 57–69

Dr. Ing. Michal Pechoucek, M.Sc. – curriculum vitae

born June 29, 1972 in Kolin Czech Republic, nationality Czech, married, two children

Present affiliation:

– Faculty of Electrical Engineering, Czech Technical University in Prague, senior lecturer in artificial intelligence and head of the research group, (full-time),

– CertiCon, a.s. senior consultant in agent technology (part-time)

Education:

– Engineering degree (equivalent to M.Sc.) in Technical Cybernetics at Czech Technical University in Prague (1995),

– M.Sc. in IT: Knowledge based systems, at Department of Artificial Intelligence, University of Edinburgh (1996),

– PhD in Artificial Intelligence and Biocybernetics at Czech Technical University in Prague (1998)

Experience in lecturing and training:

– since 1998 lecturing and responsibility for Artificial Intelligence II course at Czech Technical University, complemented by various tutoring, diploma theses and dissertation projects supervision;

– in 2003 visiting research professor at State University of New York - University of Bingahmton, lecturing a full semester course on Agent Technologies and Multi-Agent Systems;

– in 2000 delivering an intensive 1 week course on Foundations of Artificial Intelligence at Hanoi University of Technology

Track Record in Research and Technology Transfer:

– Member of the management committee of AgentLinkIII, responsible for an industrial take up of agent technology;

– Expert reviewer to EC in several research and technology transfer projects (WIDE, SpiderWin);

– Principal Investigator of the following research contracts to Air Force Research Laboratory, NY, AFOSR

– Principal Investigator of the research of the research contracts to Naval Research Laboratory, ONR;

– Principal Investigator of the research contract to Army Research Laboratory, NJ;

– Consulting to Rockwell Automation Research Center in Prague, design of the agent-based reconfiguration shipboard automation for the chilling system;

– Coordination of the consulting project to GEDAS, design of the agent based solution for engine manufacturing in SKODAAuto;

– Local Coordinator of the ExtraPLANT IST project, part of EUTIST-AMI cluster; design and exploitation of the agent-based solution for supply chain management;

– Local coordinator of ExPlanTech IST project, an international of development and take-up of a production planning multi-agent system;

– Local coordination of the MPA (modular planning architecture) project, development of JADE-based planning and simulation system for BEHR company;

– Local coordinator of the HYRES project funded by NASA - development of an agent based root-cause detection, subcontracted by IHMC, University of West Florida

Selected Publications:

– Marik, V. Pechoucek, M. Vrba, P. Hrdonka, V. FIPA Standards and Holonic Manufacturing. In: Agent-Based Manufacturing. New York : Springer, 2003,

– Pechoucek, M. Decision planning knowledge representation framework: A case-study. In: Annals of Mathematics and Artificial Intelligence. 2003, vol. 39, no. 1-2;

– Pechoucek, M. Marik, V. Bárta, J. A Knowledge-Based Approach to Coalition Formation. In: IEEE Intelligent Systems. 2002, vol. 17, no. 3;

– Pechoucek, M. Ríha, A. Vokrínek, J. Marik, V. Pražma, V. ExPlanTech: applying multi-agent systems in production planning. In: International Journal of Production Research. 2002, vol. 40, no. 15;

– Tate, A. Bradshaw, J. Pechoucek, M. (ed.): Special Issue on Knowledge Systems for Coalition Operations. Editorial to: IEEE Intelligent Systems. 2002, vol. 17, no. 3;

– Marik, V. Müller, J. Pechoucek, M. (ed.) Multi-Agent Systems and Applications III. Heidelberg: Springer, 2003.

Membership and Awards:

– Best paper award at EMSCR 1998; System Innovation Award at CIA2004; AAMAS (Autonomous Agents and Multi-Agent Systems) industry track co-chair; KSCO2002 (Knowledge Systems for Coalition Operation), KSCO 2004 co-chair; HoloMAS (Industrial Application of Holonic and Multi-Agent Systems) 2000 - 2002 co-chair; CEEMAS (Central and Eastern European Conference on Multi-Agent Systems) 2003, 2005 co-chair; EUMAS AB (European Workshop on Multi-Agent Systems Advisory Board) chair; programme committee member of ESAW, ECAI (European Conference on Artificial Intelligence), CIA (Cooperative Information Agents), AAMAS (Autonomous Agents and Multi-Agent Systems), Excalibur Alumni - Association of Czech Graduates of British Universities, founding member and chairman; - CSKI member; Memeber of Academic Senate, Faculty of Electrical Engineering, Czech Technical University