Towards Reducing Communication Traffic In Multi-Agent Systems

Michal Pěchouček, Vladimír Mařík, Olga Štěpánková

Gerstner Laboratory for Intelligent Decision Making,
Czech Technical University in Prague
(pechouc, marik, step@labe.felk.cvut.cz)

This paper is aimed at acquaintance-based methodology for reducing communication traffic in multi-agent systems. We present a tri-base acquaintance model as a computational model of agents' mutual awareness focused at reduction of the communication traffic in the system's critical time. The model contains three separate knowledge structures for representing agents' permanent, semi-permanent and temporary knowledge, respectively, and mechanism for administering, maintenance and exploration of the knowledge. A simple measure helping to evaluate volumes of communication traffic needed in different communication phases of a multi-agent community has been proposed. The paper explains how utilisation of an acquaintance model contributes to communication savings and to reduction of overall distributed problem solving complexity. We report results of experiments carried out on the community of decomposition agents, all members of ProPlanT multi-agent system for project-oriented production planning.

Keywords: multi-agent systems, acquaintance models, communication, production planning

1. Introduction

The concept of agent-based computing and multi-agent systems provides a revolutionary paradigm to modelling complex systems and solving problems of high computational complexity. Multi-agent systems consist of a set of non-centralised, mutually co-operating elements – agents which act autonomously. They exchange information and knowledge in order to achieve desired instance of collaboration. Multi-agent decision making is not only processing information from various sources of diverse nature. Multi-agent systems allow its agents to act proactively: any agent is able to try to persuade the rest of the community members to adopt the agent's local goal as their joint motivation. Inter-agent communication is crucial for understanding and sharing common goals as well as for coordinating activities to achieve them. Agent based approaches provide frameworks for handling uncertainty, inconsistency, security and optimisation requirements in communication among agents [Wooldridge 1995, 2000].

The extent to which possibilities of the multi-agent approach are exploited is usually given by how much of system's rationality and decision making is distributed within the community of decentralised and autonomous agents. Each central element such as yellow pages agent or communication facilitator drives the community fragile and less flexible. However, multi-agent systems with no or little centrality have to face the problem of explosive communication complexity caused by absence of some sort of central, global-view element [Sandholm 1999].

The original **tri-base acquaintance (3bA) model** [Štěpánková 1996] has been formalized and designed in order to limit explosive communication space in multi-agent systems. As a formal model of social behaviour knowledge in the form of agents mutual awareness, the 3bA model approach provides MAS designers with knowledge structures and knowledge maintenance mechanism. The 3bA models are to be encoded in agents' wrappers. Instead of communicating with the collaborating agents in order to find out certain information about the community, an agent equipped with the acquaintance model consults this social knowledge stored in his wrapper instead. The trade-off for this lavishness is a need to maintain the model and keep the stored knowledge up-to-date. One of the major advantages of the tri-base philosophy is the fact that the knowledge maintenance processes are carried out in community idle times.

A typical problem requiring intensive communication is that of resource allocation. We have experimented with a community of **decomposition agents** who have been in charge of optimal decomposition of a task into set of subtasks and appropriate delegation of these subtasks within a team of collaborating agents. Agents of this capability have been essentially used in the production planning multi-agent system ProPlanT [Mařík 2000]. Similarly, the agents equipped with an acquaintance model may contribute to reducing the message traffic in communication networks with comparatively large number of agents. The concept of computational model of agents' mutual awareness finds its relevance in areas which require forming of coalitions of peer agents without any supervisory or central element [Pěchouček 2000].

After giving briefly the rationale behind doing research in formalizing a computational model of agents' mutual awareness, we will present the concept of acquaintance model in detail and will comment on its role in multi-agent system. Thereafter the formalization of the 3bA model will be presented, including its knowledge structures and knowledge maintenance mechanism. Substantial part of this contribution will be devoted to description of experiments testing the communication traffic within the community of agents equipped with acquaintance models. Therefore we will give a brief account on how does the communication traffic differ in the community of agents with the 3bA model and without it, and will introduce the concept of *degree of freedom* of the community. Two separate sets of experiments and their results will be presented. The paper concludes with comments on open issues and relations of this research effort to ongoing projects.

2. Tri-base Acquaintance Model

In principle we can distinguish among three simple brokering mechanisms, by means of which an agent can find the best suitable collaborating partner for delivering a required service:

- **broadcasting a request** – where an agent sends a request for service to all members of its collaboration environment and the best collaborator is selected from the subsequent replies, or
- **facilitator** – where a request for service is send to a certain central agent (facilitator) who administers all the data about the community members and forwards the request to an appropriate agent, or
- **acquaintance model** – where each agent maintains certain amount of social knowledge about the collaborating agents and thus is aware of their actual capabilities – the agent to be contracted is then selected without any further communication.

While the first approach demands substantial communication traffic, the second approach lowers the amount of sent messages. On the other hand the second approach, unlike the former one, depends on a central agent and the community is therefore very fragile.

By using agents' acquaintance models we may find a compromise solution. Each agent maintains in his acquaintance model that part of the information administered by the facilitator (in the second approach) which he needs and which is accessible to him. Therefore each agent takes over an appropriate part of the facilitator's knowledge. This paper introduces briefly an original methodology for agents' mutual awareness administration – **Tri-Base Acquaintance Model** (3bA) which was designed as an extension of a twin-base model [Cao 1997]. In the following paragraph we will explain two key aspects of the tri-base model:

- **3bA model knowledge structures** – data collections that will appropriately represent an agent's collaboration space and
- **3bA model knowledge maintenance mechanism** – algorithms used to provide the closest possible actuality of the acquaintance model.

2.1 Knowledge Structures of the 3bA Model

Prior to formalising the knowledge structures of the model let us introduce several primitives we will use throughout the course of explanation. Let Θ be set of all agents within the community and S a set of all tasks the community members are able to take responsibility for. Results of our work will be illustrated on a very typical communication scenario – a *request for decomposition*. If an agent is requested to decompose a task it shall detect the best possible collaborators (based on his knowledge of decomposition) and contract these with parts of the original request. Hereafter we will talk about such agents. This is why we may refer to a set S as a collection of all tasks the agents are able to decompose.

For each agent $A \in \Theta$ let

- $\alpha(A) \subseteq \Theta$ be an *agent's total neighbourhood*, a set of agents an agent A is aware of**,**
- $\beta(A) \subseteq \mathsf{S}$ be the set of all tasks the agent A is able to decompose,
- $\gamma(T)$, contains all possible plans for decomposing the task $T \in \mathsf{S}$. Plan for the task T is in the form $\langle T, S, \mathsf{O}, \mathsf{C} \rangle$, where S is a set of subtasks which ensure completion of the task T provided that their processing meets precedence constraints O and applicability constraints C.
- $\omega(A,T) \subseteq \gamma(T)$ contains those plans for the task T an agent A knows about (if $T \notin \beta(A)$ then $\omega(A,T) = \varnothing$).

The following sets provide time dependent information. Let

- $\varepsilon^t(A) \subseteq \alpha(A)$ be the agent's current *cooperation neighbourhood* , a set of agent's A collaborators at the time instant t,
- $\tau^t(A) \subseteq \beta(A)$ contain the tasks being solved by the agent A in a time instance t and the set
- $\pi^t(A) \subseteq \beta(A)$ be a collection of tasks an agent A is supposed to have pre-prepared in advance in time instance t

Within the tri-base model each agent maintains three knowledge bases where all the relevant information about the rest of the community is stored. We can distinguish among:

Co-operator Base (CB) maintains permanent information on co-operating agents (i.e.: their address, communication language, and their predefined responsibility). This type of knowledge is expected not to be changed very often. CB(A) is then defined as

$$\mathsf{CB}(A) \equiv \{\langle B, \mathsf{Addr}(B), \mathsf{Lang}(B), \beta(B)\rangle\}_{B \in \alpha(A)} \text{ where}$$

Addr(B) specifies agent's the address, Lang(B) language it communicates, as already mentioned $\beta(B)$ is a set of tasks the agent accounts for and the set $\alpha(A)$ denotes members of the agent's A scope of the community.

Task Base (TB) stores in its *problem section* (PRS) general problem solving knowledge – (i) information on possible decompositions of the tasks to be solved by the agent and (ii) in its *plan section* (PLS) 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 - the owner of the task base, those denoted as $\pi^t(A)$. Formal definition of the TB(A) is then

$$\mathsf{TB}(A) \equiv \langle \mathsf{PRS}(A), \mathsf{PLS}(A)\rangle, \text{ where}$$

$$\mathsf{PRS}(A) \equiv \{\omega(T,A)\}_{T \in \beta(A)} \text{ and}$$

$$\mathsf{PLS}^t(A) \equiv \{\langle T, \langle\{\langle s, B\rangle\}_{s \in \mathsf{S}}, \mathsf{O}, \mathsf{C}, \mathsf{Trust}(T)\rangle\rangle\}_{T \in \pi^t(A)},$$

where for any $\langle T, \langle\{\langle s, B\rangle\}_{s\in S}, O, C, \mathrm{Trust}(T)\rangle\rangle \in \mathrm{PLS}^{t}(A)$ where exist O_1, C_1 such that following constraints are met $\langle T, S, O_1, C_1\rangle \in \mathrm{PRS}(A)$, $B \in \varepsilon^{t}(A)$, $s \in \beta(B)$ and as is a specialisation of C_1 reflecting the considered allocation of the asks $s \in S$, O is refinement of O_1 and both O and C are valid.

State Base (SB) stores in its *agent section* (AS) all information on 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* (TS) there is stored information on statuses of tasks the agent is currently solving. Formal description of the SB(A) of agent *A* is thus

$$\mathrm{SB}(A) \equiv \langle \mathrm{AS}(A), \mathrm{TS}(A)\rangle, \text{ where}$$

$$\mathrm{AS}(A) \equiv \{\langle B, \mathrm{Cap}(B), \mathrm{Load}(B), \mathrm{Trust}(B)\rangle\}_{B\in \varepsilon^{t}(A)} \text{ and}$$

provided that agent's *B* capability has the form of $\mathrm{Cap}(B) \equiv \{\langle T, \mathrm{Cost}(T)\rangle\}_{T\in \beta(B)}$, overall agent load is $\mathrm{Load}(B)$, and trust in this information in $\mathrm{Trust}(B)$.

TS(A) contains relevant information on all the tasks agent *A* agreed to supervise recently. This set is denoted by $\tau^{t}(A)$. Formally

$$\mathrm{TS}(A) \equiv \{\langle T, \mathrm{Dec}(T), \mathrm{State}(T), \mathrm{Trust}(T)\rangle\}_{T\in \tau^{t}(A)},$$

where decomposition $\mathrm{Dec}(T)$ is taken from the $\mathrm{PLS}^{t}_{1}(A)$ at the moment of contract (time t_1). $\mathrm{State}(T)$ partitions subtasks from $\mathrm{Dec}(T)$ into three parts: subtasks finished, actually processed, and the rest. The record is complemented with the trust value $\mathrm{Trust}(T)$ denoting trust in the plan of the task *T*.

2.3. Knowledge Maintenance of the 3bA Model

Let us first comment how the knowledge is maintained in the **cooperator base**. As we have already mentioned, this base collects knowledge of rather permanent nature and we do not expect to update it very often besides the register phase. Once a new agent registers with a community (by means of contacting the **facilitator** agent), facilitator replies the newcomer by providing information about the community members. In addition to this, facilitator informs other agents about the newcomer and thus invokes an update of the CB (in the form of a record append).

The **state base**, which is supposed to model the actual state of the collaborating agents, is maintained by a simple **subscribe/advertise** mechanism. After parsing the problem solving knowledge (in PRS), each agent identifies possible collaborators and subscribes them for reporting on their statuses. Let us denote the subscribing agent **subscriber** and an agent who was subscribed as a **subscribee**. There are two ways how to maintain knowledge in the bases of all agents in the community. The subscribee can keep advertising its load, capabilities, task completion times and costs estimates

- either periodically or
- whenever either of these changes.

This mechanism facilitates the subscriber to make the best decision with no further communication.

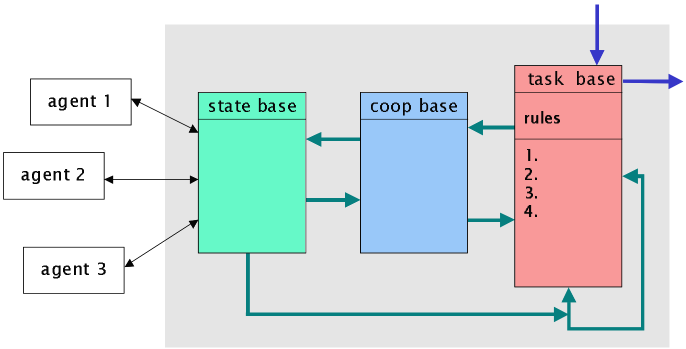

Figure 1 – Tri-base Acquaintance Model

Hereafter we will refer to a **contractor** as an agent who contracts another agent with a request. A **contractee** is an agent who was contracted by another agent with respect to the request. The **contractor** is supposed to select an optimal plan from the PLS$^{t}(A)$, where an appropriate amount 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. Knowledge stored in the PLS will help the agent to decide by itself. It is obvious that limiting the communication among agents will in its own way decrease the computational complexity of the entire problem. The price we have to pay for this, is a communication increase among agents when updating the entire model.

3. Methods of Communication

Hereafter we will refer to *broadcasting based task decomposition* as a communication convention consisting of three phases – (i) **broadcast phase** where a decomposing agent (*contractor)* broadcasts a tender for cooperation, (ii) **bidding phase** where interested agents reply their offers and (iii) **contract phase** when the contractor selects the best possible contractees and contracts it with a request for collaboration. We will refer to *acquaintance model based decomposition* as a communication protocol that involves two phases – (i) **contract phase** when the contractor contracts the best possible agents for collaboration (decision is based on knowledge stored in the agent's acquaintance model) and (ii) **model maintenance phase** when the contracted agent advertises any change of its status (caused by the contract) within the group of its subscribers, i.e. the agents who subscribed the appropriate agent for this information.

There is another quite important communication phase in the life of the community, which we call a **register phase**. In this phase agents get constructed and mutually acquainted. This process is inevitable for further utilization of an acquaintance model. Naturally, the process of

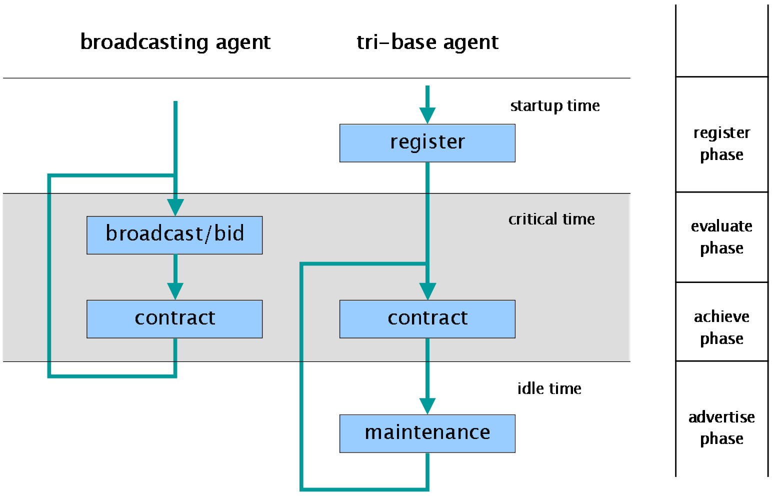

Figure 2 – Models of Communication

mutual exchange of information among the agents is expensive in terms of communication and may become a bottleneck in the case of large communities. The terminology based on KQML performatives corresponding to each of the phases is used in further explanations. The contract phase is referred to as an **achieve phase**, the broadcast and the bidding phases are jointly named as an **evaluate phase** and the model maintenance phase is called an **advertise phase**.

Utilisation of the tri-base acquaintance model for optimising the communication flow will be justified through comparison of the **broadcasting agent** – an agent who does not maintain an acquaintance model and carries out broadcasting based task decomposition and the **acquaintance based agent –** an agent who decomposes requirements by using social knowledge stored in its acquaintance model and thus carries out acquaintance model based decomposition.

2.5. Degree of Freedom

An important aspect of how much of the communication complexity may be saved by an acquaintance model is an amount of freedom within the community of agents. Let us consider an **acquaintance based agent** A to be **free** to decide about its collaborators if his knowledge base (PRS in the TB) provides several alternatives how to decompose at least one requirement and how to delegate responsibility (provided that the predicate $\mathsf{free}(A)$ will denote this property of the agent A and $|M|$ denotes the cardinality of the set M):

$$\mathsf{free}(A) \text{ iff } \exists \mathrm{T} \in \beta(A)\ (|\omega(A,\mathrm{T})| > 1).$$

A *degree of cooperation freedom* of the community Θ, denoted as $\varphi(\Theta)$, is the number of agents who are free to make their choice whom to contract and whom to delegate with decomposed subtasks:

$$\varphi(\Theta) = |\{\, A\colon A \in \Theta \wedge \mathsf{free}(A)\}|$$

Any free agent either broadcasts a request and waits for offers from possible candidates or uses some piece of social knowledge to make the rational decision. On the other hand, if agent's knowledge provides only fixed decomposition assignments, the agent does not have any freedom to select his collaborators. It is obvious that in the latter case the communication complexity is substantially smaller. The degree of cooperation freedom of the community is thus equivalent to number of agents free to form the team collaborators for a specific project.

However for many purposes the introduced metric is not precise enough. For each particular request the community has different degree of cooperation freedom (as each specific agent can choose between two variants when solving certain task, but for another task there is only one decomposition possible). We say, that the agent B is free for the task t iff B knows about more than one decompositions for t (provided that the predicate $\mathsf{free}(B,t)$ will denote this property of an agent A):

$$\mathsf{free}(B,t) \text{ iff } |\omega(B, r)| > 1.$$

Let $\varphi_t(\Theta)$ denote the number of agents which are free for the task t. Then $\varphi(\Theta)$ as defined above is an upper bound estimate of $\varphi_t(\Theta)$ for all particular tasks t. A more precise metric may be the sum of all degrees $\varphi_t(\Theta)$ per each request t,

$$\sum_{t \in S} \varphi_t(\Theta),$$

or the average degree of freedom per single request.

It is possible to introduce a more complex measure which is based e.g. on information entropy calculations. Let's consider e.g.

$$\varphi'(\Theta) = -\sum_{i=1}^{n} \frac{1}{k_i} \ln k_i$$

Where n is the number of free agents, ki means the number of options for choice which the i-th agent has. In the case, the agent's trust into different choice options differ, the corresponding measure formula can be extended in a natural way.

Complexity of the decision making process is also significantly influenced by "substitutability" among agents. Thus another interesting measure corresponds to the maximal number of different agents, which are able to solve a single task. In the rest of the report we will stick to the first measure $\varphi(\Theta)$, referred to as the degree of cooperating freedom. This seems to be good enough to validate the approach.

3. Testing the Tri-base Acquaintance Model

Utilization of the 3bA model has been justified by means of two separate sets of experiments carried out on different agent communities within the frame of the ProPlanT multi-agent system.

3.1 PROPLANT Multi-Agent System

A prototype of the ProPlanT multi-agent system has been implemented for planning and modeling of the production processes in a project-driven manufacturing facility (this system is e.g. experimentally running in the TESLA-TV Czech company producing TV transmitters, many experiments were carried out with the simulation of planning processes for U.K.based company producing compressors, etc.). Resulting from a thorough production process analysis we have identified specific information units the general production process is based on. In principle we cluster agents into two fundamental super-classes: *intra-enterprise agents* (IAE) and *inter-enterprise agents* (IEE). We distinguish among the following basic classes of IAE agents:

- **Production Planning Agent** (PPA) is in charge of project planning. It is supposed to construct an exhaustive, partially ordered set of tasks that need to be carried out in order to accomplish the given project. It contracts PMA agents.
- **Production Management Agent** (PMA) is responsible for the project management in terms of contracting the best possible PA agents (in terms of operational costs, offered delivery time, and current capacity). PMA delegates its responsibility either to another PMA or it conducts work of a group of PA agents contracted for the considered task. In this manner, a multi-level managing structure can be modelled.
- **Production Agent** (PA) belongs to the lowest level production units that simulate or encapsulate shop floor production processes on the IAE level. PA carries out the parallel-machinery scheduling of given tasks and manages resources allocation via a special type of database agents. On the IEE level, the PA agent may encapsulate contracted suppliers offering either services or components participating in the manufacturing process. Appropriate optimisation within the community will result in the cheapest (or shortest) production plan.
- **Meta Agent** (MA) is a special monitoring agent who visualises information, material and work flows across the agents' community and advises on optimal system's efficiency. It shall be noted that the community of agents will survive well with no meta-agent. 'Ordinary' agents are able to communicate in peer-to-peer manner, but the meta-agent is able to induce specific efficiency considerations from observation of the community workflow.

The PPA agent constructs a component list with the team of PMA agents and delegates further responsibilities to PPAs who contract the best possible PA agents. Final costs and deadlines, a meta-representation of the distributed production plan, will be then back-propagated to the customer.

For our experiments, each testing community consists of two key classes of agents – **decomposition agents** (DA) and **resource agents** (RA). While resource agents provide other agents with some service or commodity, the decomposition agents are responsible for optimal decomposition of a task into a set of subtasks and further delegation among collaborating agents. Each DA agent can contract either an RA agent or another DA agent. This classification of agents in terms of their capability to decompose and delegate bears a resemblance to the Pleiades architecture of collaborative agents architecture that consists of **task-specific** agents (TA) and **information-specific** agents (IA) [Sycara 1995]. Similarly, we can match this classification to architecture of ProPlanT multi-agent system [Marik 2000]. As the flow of task decomposition goes from PPA thought PMA to PA agents, the cluster of PPA and PMA agents is viewed as class of DA agents, whereas class of RA agents consists of ProPlanT's PA agents.

As we did not want to experiment with entirely data-of-thumb, the entire simulation was carried out on the community of agents planning manufacturing of compressors. The task of the community is to come to a general agreement on how to allocate given pieces of production resources when considering an internal structure of the manufacturing process.

In order to confirm the communication savings estimates we have designed two different experiments:

(i) Communication traffic of the tri-base agent and the broadcasting agent were compared with respect to **different degrees of freedom**. Modifications of the degree of freedom resulted from changes of different agents' knowledge stored in their task bases. While the *architecture of the community* (number of RA and DA agents and their relationships) remained unchanged, the DA agents were simulated to have different number of options to choose their collaborators as the real life community. This is why we say that the *functionality of the community* (set of tasks the community can solve and number resulting allocations) is different for either community.

(ii) Communication traffic of the tri-base agent and the broadcasting agent were also measured with respect to **different architectures** of the community. Whereas we made the functionality of the community to remain unchanged (identical to the real life case), the architecture of the community was modified by increased number of DA and later RA agents.

Let us describe each of the above-mentioned experiments in more details. It has been already said that the testing communities were created by modification of the original, real-life community **c-0**. The **c-0** community consists of 17 agents of different nature, 6 of which are free (have a free choice to contract the other agents). Moreover, for any free agent holds that all the tasks he treats have at least 2 different decompositions.

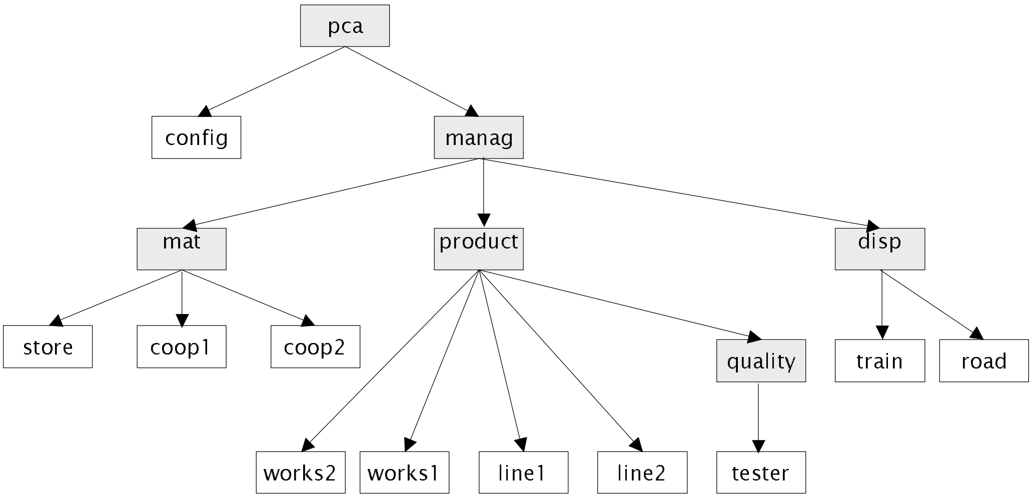

Figure 3 – **c-0** Community

- **c-0 community:** The original community consisting of 6 decomposition agents (management, quality, production, dispatching, material and pca) and 11 RA (coop1, coop2, store, ass1, ass2, works1, works2, tester, road, train, config). This is the original community containing 17 agents in total. All the DA agents are free to decide on collaborators, therefore $\varphi(\text{c-0}) = 6$.

Talking about the first experiments (different degrees of freedom), the other testing communities were constructed by constraining the decomposition possibilities in the community.

- **c-(i).1 community:** The structure of this community is identical to c-0. There is no cooperation freedom in this community and the agents have fixed decomposition knowledge that does not involve any mediation about possible collaborators, therefore $\varphi(\text{c-(i).1}) = 0$.
- **c-(i).2 community:** Like in the previous, this community has also a structure identical to c-0. However, here we have only two agents (production and material) who are free to decide on subtask delegation, therefore $\varphi(\text{c-(i).2}) = 2$.

Talking about the second experiment (different architectures), each new configuration is derived from the original one by adding (eventually simplified) "copies" of selected agents being present already in the **c-0** community (see Table 1):

- **c-(ii).1 community:** Higher communication traffic has been recognised in the case of this community. We have extended the community with several other DA agents. As the overall functionality of the community shall not be modified, a separate DA agent has been constructed for each single task, which any of decomposition agents from the first community accounts for. This is why the number of decomposition agents has risen up to 36 (pca 1, management 9, quality 5, production 9, dispatching 3, material 9), while all of them remained free. The total number of agents for this case is 47 and the degree of freedom is 36. Shall be noted that while $\varphi(\textbf{c-0}) = 6$ and $\varphi(\textbf{c-(ii).1}) = 36$, degree of freedom per request $\varphi_r(\textbf{c-0}) = \varphi_r(\textbf{c-(ii).1}) = 36$.
- **c-(ii).2 community:** The community with even higher communication traffic has been created from the community c-(ii).1 by separating the resource agents so that there is a single RA accountable for a single task. By doing so, we have got 78 RA agents (coop1 12, coop2 12, store 11, ass1 9, ass2 12, works1 6, works2 9, tester 5, road 2, train 2, config 1). Total number of agents is thus 115 while the degree of freedom still remained to be 36.

Community	c-0	c-(i).1	c-(i).2	c-(ii).1	c-(ii).2
|RA|	11	11	11	11	78
|DA|	6	6	6	36	36
Total	17	17	17	47	115
$\varphi(\Theta)$	6	0	2	36	36
$\varphi_r(\Theta)$	36	0	18	36	36

Table 1 – properties of tested communities

3.2 Broadcasting Agent vs. Tri-base Agent with Different Degree of Freedom

We have carried out simple tests in order to compare communication requirements for a single decomposition request sent to the community of broadcasting agents and tri-base agents with different degrees of cooperation freedom. The degree of cooperation freedom has been set by interventions into agents' decomposition knowledge stored in their knowledge bases by disabling or adding some alternative decompositions for specific tasks. We have experimented with both the marginal cases – no cooperation freedom (community **c-(i).1**) and full cooperation freedom of all decomposition agents (community no. **c-0**), as well as with the case of only two free agents (community **c-(i).2**). For results see Table 2.

number of messages	c-0 /$\varphi(\Theta)=0$	c-(i).1 /$\varphi(\Theta)=2$	c-(i).2/ $\varphi(\Theta)=6$
acquaintance-based	30	30	30
broadcast-based	30	46	180
Maintenance	123	123	123

Table 2 – test no. 1 results

The first row of the Table 2 (acquaintance-based) shows number of messages sent within the community during the **achieve** phase. As the degree of freedom affects communication only in the **evaluate phase** (in the case of broadcasting), there is still the identical number of messages measured for each $\varphi(\Theta)$. Although communication in the maintenance phase – the third row of the table – does not depend on $\varphi(\Theta)$. It is rather related to the architecture of the community, as it will be illustrated in the next experiment. Therefore there is the same value in each column. The second column gives the total number of messages sent by the broadcasting agent. The first value is the same as in the tri-base agent: for $\varphi(\Theta)=0$ there is no decision making process about whom to delegate the decomposition with and therefore there is no evaluate phase. The last number illustrates the case where each agent A who is contracted with a request broadcasts a call for cooperation within all the agents. These consequently reply with their offers and the agent A contracts the optimal one.

The higher the degree of cooperation freedom of the community, the more intensive communication traffic is required and the acquaintance model rises its impact on the communication savings. In the marginal case with fixed task decomposition and job delegation ($\varphi(\Theta) = 0$), the acquaintance model has no utilization while substantial communication needed for the model maintenance is required. The broadcasting agent does not have to carry out any bidding (evaluate phase) and directly contracts (achieve phase) the single relevant agent (of which the broadcasting agent knows from the communication structure). Broadcasting based decomposition consists in this case from the achieve stage communication only and therefore communication requirements for either approaches shall be identical (m(acquaintance-based) = m(broadcast-based) = 30). However, the tri-base agent needs to spend certain amount of communication resources for maintenance of their models (m(maintenance) = 123). This is why utilization of the tri-base models is not very likely to pay off in this case. On the other hand, in the case of the total cooperation freedom ($\varphi(\Theta) = 6$) in the considered community no. 1, the broadcasting agent would have to exchange six times more messages (m(broadcast-based) = 180) than the tri-base agent (m(acquaintance-based) = 30) in the critical time. The total number of messages sent in the whole communication cycle of tri-base agent is also smaller (m(acquaintance-based)+m(maintenance) = 153 < m(broadcast-based) = 180). In the mentioned example, reflecting the real organizational structure in the plant (see [Pechoucek 00a]), there are only two agents (production, material) making their own choices in terms of optimal allocation of the collaborating agents workload. An interesting observation is that in this case with rather low degree of collaboration freedom ($\varphi(\Theta) = 2$) the tri-base agent saves already more than one third of communication needs (m(acquaintance-based) = 30 < m(broadcast-based) = 46) in the critical times.

3.3 Broadcasting Agent vs. Tri-base Agent in Different Communities

The principal subject of the second experiment was the **c-0** community of 17 agents with six free agents ($\varphi(\Theta) = 6$). Communication savings were tested also in other two communities: **c-(ii).1** and **c-(ii).2**, having architectures based on functionality of the **c-0** community. Although with increasing number of agents the degree of freedom of the community may increase, for a specific request there are still two free agents within the available part of the involved community ($\varphi(\Theta)$ increases with increasing number of agents in our case, because new DA agents were constructed so that each DA agent accounts for one particular task, while still maintaining the same number of decomposition alternatives as in the community no. 1.). This is why the degree of freedom was not an issue when experimenting with the different communities.

Tables 3 and 4 summarize the test results. The Table 3 gives a number of communication messages per decomposition cycle for acquaintance based decomposition. Each row gives number of messages in an appropriate communication phase of one decomposition cycle. In addition to this we illustrate communication requirement in the registration phase. The Table 4 provides results of measurements in the case of broadcasting agents.

Community	c-0	c-(ii).1	c-(ii).2
Register	173	1871	4517
Contract	30	30	30
Maintenance	123	206	31
Total per request	153	236	61

Table 3 – acquaintance based decomposition – number of messages

Community	c-0	c-(ii).1	c-(ii).2
Register	27	58	196
Evaluate	150	285	788
Achieve	30	30	30
Total per request	183	315	818

Table 4 – broadcasting based decomposition – number of messages

As expected, experiments showed that with increasing number of agents in the community the 3bA model provides more of communication savings. The overall degree of freedom rises with increasing number of agents. Consequently this observation is not in contradiction with the results of the previous experiments.

Another important observation regards sharing resources. Sharing resources brings difficulties to the tri-base agents. While the community of 47 agents needs 206 messages for model maintenance 115 agents community requires 31 maintenance messages. The reason for this is that in the former **c-(ii).1** community there are 36 DA agents contracting 11 RA agents. Resource agents have to be therefore inevitably shared. The latter community no. 3 is organized in a tree and this is why no RA agent may be contracted by two different DAs and thus it advertises to single agent only. However, the organization of the community does not have any impact on communication load of the broadcasting agent. The increased number of RA agents will economize the flow of advertise messages sent from the level of RAs towards DA agents. As the community is organized in a tree-like manner with a high branching factor in the close-to-leaves levels of the tree, it is easy to back-propagate an update within the community (maintenance – 31) than searching for an optimal configuration (evaluate – 788).

The drawback of all the acquaintance-model-based approaches is the register phase. Unlike broadcasting agent, the tri-base agent has to set subscribe-advertise links with its pears prior to functioning of the community. This is what is rather time consuming and may become a bottleneck with large communities (4517 messages among 115 agents). Fortunately this process has to be gone through only once when the community is created.

3.4. Frequency

As already mentioned the communication savings offered by this approach may be seen as a specific communication load shift. Tri-base minimizes communication load in the agent's contract phases while a new advertising activity appears in the agent's maintenance phase.

Each request is answered quickly but it brings substantial communication flows following the request fulfilment. The contract phase is usually placed in community *critical time* of operation, when agents are required to answer promptly. The virtue is that the maintenance phase is carried out in the community *idle times*, where agents are not required to provide any assistance to the user. Successful operation of such a multi-agent system depends on the community lifecycle. In order to utilize this acquaintance model based mechanism and to guarantee its communication savings for the frequency f of requirements to plan, the following condition should be met

$$\frac{1}{f} \geq t_c + t_i ,$$

where t_c is maximum amount of time spent in the *critical time planning* and t_i is maximum amount of time needed for processing the subscription/advertise mechanism in the agents' *idle time*.

This observation suggests exclusion of possible processing of parallel requests within the community (as minimal time slot between two consecutive requests is required). If there are two completely independent requests arriving at the same time, the quality of the decomposition, the time of the response and communication requirements are unaffected by this irregularity.

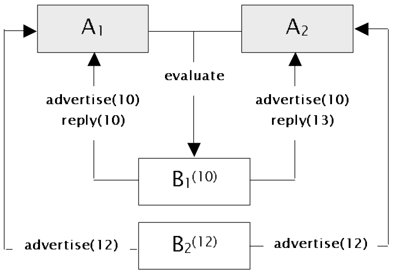

Figure 4 – conflicts caused by shared resources

However sharing resources causes conflicts. Let us assume that two different requests will need to share resources of a single agent. In the simplest possible case of the previous two agents (A_1, A_2) try to contract either of their two subscribees (B_1, B_2) for carrying out task t (time(t)=3). Each of subsribees advertised its offer – $B_1(t)$ = 10 and $B_2(t)$ = 12. The agent A_1 makes a sound decision and contracts agent B_1 with a task t. Provided the agent A_2, who did the same decision as the agent A_1, contracts the agent B_1 prior to agent B_1 advertising a new offer (thus failing to meet the above mentioned condition) A_2 will obtain a reply that will be in conflict with the social information the agent A_2 maintains about the agent B_1 in its wrapper.

In the current implementation of the ProPlanT tri-base, the community fails to provide an optimal answer due to the conflict mentioned above and a global failure (sorry KQML message) is announced. In another philosophy the community will ignore the conflict and will reply with an answer that will not be optimal (such as agent A_2 accepting collaboration with B_1 although the task will be completed in 13 times unit where the optimal solution would be provided by agent B_2 in 12 time units). The best possible approach suggests refusing such a contract, updating the tri-base model according to the conflict nature and carry out local re-planning. It is easy to understand that this approach brings an optimal answer but at the cost of some extra communication. This is what we wanted to emphasize in this paragraph.

3.5. Open Issues

When applying the tri-base acquaintance model in different domains one would need to economize several aspects of the knowledge maintenance mechanisms.

Register-Phase: Firstly (and maybe most importantly) the communication load in the register-phase of the community shall be brought to minimum. We have carried out separate research in the areas of efficient community configuration. The current configuration mechanisms are based on an intensive broadcast within the entire community, where each decomposition agent is fishing for possible future collaborators.

An elegant solution is to utilize a facilitator, who the agents have to register with if they desire to participate in the community. Apart from their symbolic name, IP address and port number, the agents will register also their capabilities and services. Such a community component will resemble functionality of a FIPA (www.fipa.org) normative directory facilitator (DF) who provides "yellow pages" services to other agents. Even though this is a standard way, we wanted to test our approach in the environment with minimal centrality and maximum autonomous rationality of participating agents.

Cooperation Neighbourhood: As defined [Pechoucek 2000], the agents' cooperation neighbourhood $\varepsilon^t(A)$ is a set of agent's A collaborators at the time instant t. More precisely we talk about those agents, the agent A subscribed for reporting on their actual statuses. These agents are listed in the **agent section** (AS) of the agent's **state base** (SB).

In the previous section we said that although members of the ProPlanT community can join someone's cooperation neighbourhood, there are no means for subscribers to restrict the cooperation neighbourhood, or for subscribee to discontinue advertising to a subscriber. Introduction of a new FIPA-like performative – **unsubscribe** – will allow subscribers to restrict their cooperation neighbourhoods as well as the performative **subscribe** expands it. Subscribees do not necessarily need to be benevolent. Subscribed agents may easily discontinue advertising their statuses in spite of the fact they are subscribed. They have to have only a good reason for doing so.

We have shown that implementing technical means for maintenance of the agents' cooperation neighbourhood is not a big research challenge. What is much more interesting, it is detection of when and how the cooperation neighbourhood would be revised. Let us call a subscriber-subscribee link to be **unexploited** if and only if subscriber did not contract the subscribee for certain threshold period of time (t_{old}). Restriction of the cooperation neighbourhood may be thus driven two-fold:

- either the **subscriber** agent may analyse record of its communication history and easily detect unexploited links (with respect to t_{old}) and unsubscribe appropriate subscribees,
- or the **subscribee** agent will analyse its record of past communication and stop advertising to agents who did not contract it for longer than t_{old} (the subscriber shall be informed of this).

If the subscriber needs to contract unsubscribed subscribee, it subscribes it first and then does usual decision making. The communication requirements of this process will not be worse than local broadcasting communication. This is why, careful specification of the threshold value t_{old} will affect the overall system efficiency.

4. Relations to Projects

The work carried out by the authors is related to several larger projects carried out at the Gerstner laboratory. Let's mention the most relevant of them.

ProPlanT: The tri-base acquaintance model has been used in implementation of the PMA (project managing) agents of the ProPlanT multi-agent system. ProPlanT (production planning tool) is a working prototype of a multi-agent system, which through simulation of the production process assists in quoting and planning of the entire course of project-driven manufacturing. Due to the fact that each new manufacturing project has been treated separately, while other already planned production activities are regarded, ProPlanT suits very much to the problem of project-driven production planning. As already mentioned in the paper, ProPlanT consists of three classes of agents: class PPA (project planning) agents represented by a quotation expert system that is supposed to transform users demands into a specific project specification, PMA agents who are supposed to manage the most optimal decomposition of a task into subtasks and appropriate delegation, and, finally, PA agents who represent a front-end to the production process (schedulers, models of machines, company databases, …) The requirements for the PMA agent's functionality imply appropriateness of utilisation of the tri-base acquaintance model as the agent's software model. The Java implementation of the agent has been used for experiments described in this article.

ExPlanTech: The research carried within implementation of the ProPlanT multi-agent system (funded by EUREKA PVS'98) is being exploited in an ExPlanTech EC-funded project, that is aimed at applying ProPlanT technology on the production planning problems in two different in their nature SMEs. In Chatzopoulus, S.A. (GR) producing flexible packaging materials the ProPlanT-like system will play a role of a component that will gather information from the company IS, will maintain its own models of actors involved in manufacturing process and will carry out intelligent reasoning aimed at providing an optimal production plan. In contrary, The Liaz Pattern Shop, s.r.o. (CZ) is administered by non-standard, heterogeneous and distributed information system. Here we expect ProPlanT to play a key integrating role in the company information system. The IS components will be represented as unique agents in the multi-agent community.

CPlanT: Another research area of utilisation of the tri-base acquaintance model to reduce the communication traffic is to consider it as a formal model of agents' mutual awareness is the problem of **coalition formation** [Shehory 1998]. This issue has been investigated within the CPlanT – a project funded by the U.S. Air Force Research Laboratory (AFRL). The application domain of this coalition formation research belongs to the area of **war avoidance operations** such as peace-keeping, peace-enforcing, non-combatant evacuation or disaster relief operations. Unlike in classical war operations, where the technology of control is strictly hierarchical, **operations other than war** (OOTW) are very likely to be based on cooperation of a number of different, quasi-volunteered, vaguely organized groups of people, non-governmental organizations (NGO's), institutions providing humanitarian aid but also army troops and official governmental institutions. **Collaborative**, unlike hierarchical, approach to operation planning allows greater deal of flexibility and dynamics in grouping optimal parties playing an active role in the operation. New entities shall be free to join autonomously and involve themselves in planning with respect to their capabilities. The main reason why we can hardly plan operations involving different NGO's by a central authority is given by their **reluctance to provide information** about their intentions, goals and resources. Consequently, besides difficulties related to planning and negotiation we have to face problems how to assure sharing detailed information. Many institutions will be ready to share resources and information within some well specified community, whereas will refuse to register their full capabilities and plans with a central planning system and will not follow centrally distributed commands. They may agree to participate in executing a plan, on forming of which they played an active role. Actual information may become **unavailable** also due to **unreliable or overloaded** communication channels. It may happen that a collaborative entity gets cut off the communication links for certain period of time and the rest of the community still wishes to be able to form/execute plan relying upon a missing player (and the same vice-versa). For this reason, each participating entity shall be able to maintain approximate model of the collaborating members of the coalition in the form of the 3bA model..

The 3bA model technique has been used as a formal framework in design of a multi-agent system for the coalition formation process with the goal to study above mentioned problem of **information privacy** and **agent inaccessibility**.

4. Conclusions

Communication traffic among the agents is a very important issue when evaluating global functionality and efficiency of a multi-agent system. It is clear that the more knowledge is posessed locally, the less messages containing lower volume of knowledge and data are needed to be communicated. But – as a rule – knowledge and especially data can evolve dynamically in a multi-agent system. This fact leads, on the other hand, to an increase of the needs in communications. Thus, appropriate balance between the volume of knowledge kept locally and the volume of messages to maintain it should be achieved.

The 3bA acquaintance model helping to classify and organize knowledge and data into three bases was designed. The portions of knowledge stored locally can be changed and optimized dynamically applying this model. FIPA-like communicative acts have been used for this purpose. The main advantage of the proposed model is that the knowledge/data is maintained by messages exchanged in the idle time.

Optimisation of the communication traffic cannot be achieved without defining corresponding criteria measuring this traffic. In the paper, we made a first attempt towards this direction proposing to use a very simple measure – degree of freedom. This can be of course extended into more complex, e.g. information entropy based measures easily. The experiments with differently organized agents‘ communities performing the same global functionality documented the usefulness of the proposed approach. Despite its simplicity, the proposed measure can help the designers to understand the essence of the traffic and to optimize it.

The optimisation of the communication traffic in multi-agent systems is currently a hot topic. It is crucial for finding tractable-in-real-time solutions for processes essential mainly (but not only) for industrial applications. These include e.g. automatic community reconfiguration, replanning/rescheduling and optimal coalition formation. More intensive research in these directions is expected.

5. References

Cao W., Bian C.-G., Hartvigsen G.:(1997). Achieving Efficient Cooperation in a Multi-Agent System: The Twin-Base Modelling. In: *Co-operative Information Agents* (Kandzia P., Klusch M. eds.), LNAI 1202, Springer-Verlag, Heidelberg, 1997, pp. 210-221

Mařík V., Pěchouček M., Štěpánková O., Lažanský J.(2000). ProPlanT: Multi-Agent System for Production Planning. *Applied Artificial Intelligence,* Taylor & Francis, vol. 14 no. 7, pp. 727-762

Pěchouček M., Mařík V., Štepánková O. (2000). Role of Acquaintance Models in Agent-Based Production Planning Systems, In: *Cooperative Information Agents IV* (M. Klusch L. Kerschberg, editors), LNAI No. 1860, Springer Verlag, Heidelberg, pp.179-190

Sandholm W. (1999). Distributed Rational Decision Making, In *Multi-Agent Systems: Modern Approach to Distributed Artificial Intelligence* (Weiss G., ed.), The MIT Press, Cambridge

Shehory O. and Kraus S.(1998). Methods for Task Allocation via Agent Coalition Formation, *Artificial Intelligence*, vol. 101 (1-2), pp. 165-200.

Sycara K. (1995) Intelligent Agents and Information Retrieval. Unicom Seminar on *Intelligent Agents and their Business Applications*, 8-9 November, London, pp. 143-159

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

Wooldridge M., Jennings N. R. (1995) Intelligent Agents: Theory and Practice, *The Knowledge Engineereing Review*, 10(2), ss. 115-152

Wooldridge M.(2000) *Reasoning about Rational Agents*, The MIT Press, Cambridge, MA,