Improving the Scalability of Multi-agent Systems

Phillip J. Turner and Nicholas R. Jennings

Intelligence, Agents, Multimedia Group. Department of Electronics and Computer Science, Southampton University, SO17 1BJ. United Kingdom
{pjt,nrj}@ecs.soton.ac.uk

Abstract. There is an increasing demand for designers and developers to construct ever larger multi-agent systems. Such systems will be composed of hundreds or even thousands of autonomous agents. Moreover, in open and dynamic environments, the number of agents in the system at any one time will fluctuate significantly. To cope with these twin issues of scalability and variable numbers, we hypothesize that multi-agent systems need to be both *self-building* (able to determine the most appropriate organizational structure for the system by themselves at run-time) and *adaptive* (able to change this structure as their environment changes). To evaluate this hypothesis we have implemented such a multi-agent system and have applied it to the domain of automated trading. Preliminary results supporting the first part of this hypothesis are presented: adaption and self-organization do indeed make the system better able to cope with large numbers of agents.

1 Introduction

When designing or building a multi-agent system (MAS), a designer has to ensure that the agents and overall collective provide the facilities prescribed by users. However in multi-agent systems that consist of large numbers of agents, current design methodologies are often unable to ensure such provision. This is not because these methodologies have inherent limitations, but because the practice and the theory on which they are based contain very few studies directly concerned with the scalability of MASs. Yet, MASs for use in open systems (e.g. the Internet and corporate intranets) are not only likely to require large numbers of agents, but also present designers with the problem of dynamic agent numbers too. Even worse, if the expertise to design large MASs was indeed developed, it may still prove impossible to redesign and upgrade software before its environment, and thereby the demands on it, evolve. Thus, the current state of the art is challenged by MASs that are large or where the magnitude or speed of agent population variability confounds one overall design. To tackle both these problems we hypothesize that MASs should be *self-building* (able to determine the most appropriate organizational structure for the system by themselves at run-time) and *adaptive* (able to change this structure as their environment changes).

The majority of MAS work deals with systems in which agents are peers of each other. However, it seems unlikely that such structures are the most appropriate when hundreds or thousands of agents are required. For this reason, MAS designers are increasingly using metaphors from human social and economic organizations (e.g. [8])—where we are used to dealing with large numbers of (human) agents—to help structure their systems. Human organizations operate by enforcing avenues of communication and control between individuals in order for the overall grouping to achieve its goals. Of rough equivalence to organizational structure, MASs use acquaintance topologies to perform the same function of defining and constraining interaction. These topologies may be mesh, fully connected, star, hierarchic, etc., or hybrids, and the inter-agent relationships between them may be master, slave or peer, or anything else deemed suitable. Although it is tempting to mirror human socio-economic organizational structures or the roles therein, it is not known if this is the most appropriate way to achieve the goals of MASs. A further disparity between organizational theory and practice for humans and agents arises because the emphasis in MASs may not be on the whole 'organization' achieving its goals at the expense of the goals of individuals if necessary (as in human organizations). In MASs, the achievement of the goals of its users, and thereby their agents, is the prime focus of attention, not the overall emergent behavior. Thus, MASs that have imposed organizational structure, or have the ability to dynamically construct their own, should not be forced to limit (or inherit) their techniques, rationale and structures to those of anthropic organizational theory and practice.

Against this background, the primary focus of this paper is in presenting how a MAS can adapt its structure for various population sizes.[1] In so doing, it is suggested that herein lies a method for: 1) the creation of MASs that can better deal with variable numbers of agents, and 2) compensating for the difficulties of building large-scale MASs by building at a sufficiently manageable small size with reliance upon increased scale tolerance for later scaling to the required size. This work advances the state of the art in the following ways. First, presentation of empiric evidence to support the case for dynamic organizational structure. Second, an implemented method for allowing agents to dynamically change organizational structure. Lastly, evidence to support the proposition that MASs that have fixed organizational structures are less scalable than those that can adapt to population size. (For a discussion of fixed versus flexible form in the context of human organization theory see [11]).

We believe that a MAS that can both operate with different population sizes and deal with dynamic changes to population during operation, is more scalable. To disambiguate *scalable* (noting that the term has several different meanings in computer science), the facets we are concerned with are those that refer to the relationship between the collective computational resource needs of the agents and the population size. However, in order to say that a particular system is scalable, or compare scalability as a property, a measure must be determined. In this case we use a measure of the processing requirements for agents (collectively) which roughly equates to the number of machine instructions they incur when achieving their goals. Despite using a metric to compare several methods of

[1] Space limitations prevent us from reporting on the self-building hypothesis.

agent operation, we avoid the pitfalls of defining absolute requirements for a MAS to be termed *scalable*, and instead concentrate on the reduction of growth in computational resource demands whilst maintaining utility. (If a definition of a scalable MAS is to be forced, we would loosely define it as the ability of a MAS operating with given computational resource availability limits to achieve required minimum levels of utility to users for target ranges of population size. Thus, higher scale tolerance implies that larger population sizes may be successfully dealt with.)

The remainder of the paper is structured as follows: section 2 introduces the e-commerce application we have developed for investigating the issues of scalability. Section 3 presents data obtained for a MAS in various organizational forms. Section 4 discusses organizational adaptability. Section 5 covers related research. Finally, section 6 presents our initial conclusions and discusses future work.

2 A Trading Agents Scenario

Electronic commerce is a natural domain for testing hypotheses about scalability. It involves large numbers of end-users and online businesses and it constitutes a highly dynamic environment—both in terms of interactions and in terms of membership. Our particular scenario involves agents that encapsulate the basic needs of the end-user (e.g. automated product location and purchase) and supplier agents (automated query processing and order placement facilities). (See [7] for other goals of the scenario.)

End-users' goals to purchase commodities are presented to the customer agents according to a probability distribution over a specifiable period of simulated-time. The specifics of a request include a commodity identifier, the required volume, and a deadline. Generation of these requests is achieved through probability distributions for each request parameter.[2] Customer agents are also capable of dynamically building preferences for suppliers on a per commodity and general basis and propagating information about suppliers and their preferences to other customers (i.e. they can make recommendations), forming co-operative groups to make collective bulk (discounted) purchases, and finally, forming and maintaining models of suppliers' wares and prices.

In contrast to customer agents, suppliers are currently limited in terms of dynamics. Their character within the market is defined in terms of the products they sell and the prices they charge. Although their product ranges do not vary periodically, each supplier has the ability to model and generalize customer commodity requirements, and offer 'good' customers discounts and/or bulk deals. They also monitor the requests they receive and analyze their responses over time. Reaction to such data may be to start selling often requested commodities or drop prices slightly to improve the number of sales of given items.

[2] The probability distributions referred to in this paragraph can either be uniform or Poisson. However, the results presented herein were generated with uniform probability distributions for query generation and query parameters.

Both types of agent in this scenario are implemented so that, effectively, they do not have computational or storage resource bounds. That is, the behavior of and interaction between agents are not affected by hardware concerns—neither absolute or relative computational speeds, nor memory requirements (disk space being plentiful). Furthermore, each agent processes each of its goals, its internal models and the messages it receives until no further processing can occur without communication to the other agents. When *all* the agents have reached this state all messages are exchanged. The agents resume processing once all messages have been delivered. These two steps are referred to as a system *tick*, and their repetition provides continuous operation. Note that ticks do not constitute a fixed amount of time, but they do represent the passage of time.

The agents operate in this manner because it means that any variance in the behavior of the individuals, and ultimately the whole MAS (due to the non-deterministic order in which agents are executed or receive and transmit communications) can be eliminated. A further advantage is that since no agent receives information before any other, none can be said to have an advantage over the others. For example, in a MAS where there are too few resources, faster acting agents may gain an unfair advantage over others. We have also found this method of operation provides reproducible MAS behavior.

Moreover, this scheme also fulfills a subtle requirement of these experiments: that the aim is *not* to develop agents whose primary concern is goal prioritization (or any other issues generally dealt with by resource-bounded research), but achievement of the desirable properties of a MAS whilst reducing as far as possible the resource needs. In other words, the experiments do not seek to determine the goal achievement levels possible in a resource bounded environment, but to predict the minimal resource requirements needed to attain a given level of goal achievement. The most important property of this MAS is that of maintaining agents' utility to their users. The agents therefore undertake whatever actions are necessary to ensure that their responsibilities to users are met. (For example, searching all the suppliers for a cheaper deal even when they have already located an acceptable price, or co-operating with other customers to get bulk discount). This is done, however, in a manner which tries to minimize the total computational resource needs of the MAS.

The agents are implemented as interpreted rules with mental and (speech act based) message-content based guards on their firing (similar to agent0 [18]). Upon firing, each rule calls one or more action handlers which are Prolog predicates.

2.1 Performance Measures

The specific metrics used herein to compare scalability are: 1) the number of ticks (process/ communication cycles) that have passed, t, 2) the total number of agents in the system, n, and 3) the sum of the number of (Prologs') logical inferences[3] required for each tick, for all agents (ΣLI_t). (The ratio of the number of customer agents to supplier agents is ten to one, the minimum number of suppliers being five.) Also, as new agents are added to the system, so too are the goals that they receive from their users. Thus, the total of simulated user goals remains at a fixed ratio to the number of agents and therefore there is no need to introduce MAS throughput into the comparisons.

3 Organizational Forms and Agent Interactions

In this section we present several organizational forms of MAS for our trading scenario. These forms can be distinguished by the constraints within which the agents interact with each other. For example, whether they can share information, form co-operative groups or take action to combat inefficiency. The purpose of examining these various forms are firstly to show that different organizational forms place different resource requirements on agents, and secondly, to determine what the relationship between the resource requirements of a given form and the number of constituent agents is.

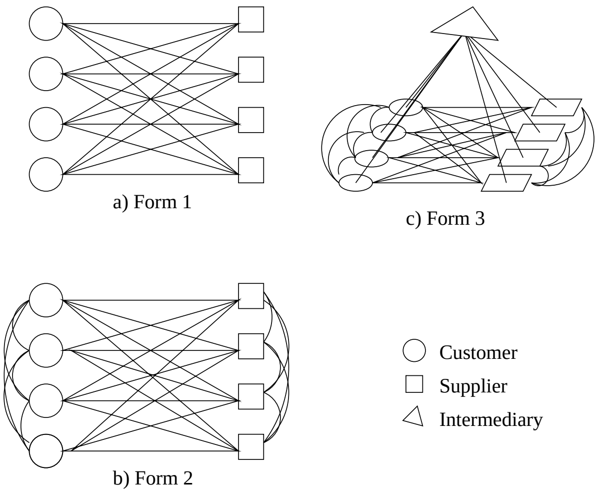

Fig. 1. Acquaintance topology forms

In the first (and most simple) organization form (figure 1a), each customer agent can reason about and communicate with each supplier (and vice versa). However, customers are unaware of other customers and suppliers are unaware of other suppliers. Consequently, agents cannot form groups, share information, or

[3] The number of logical inferences can be equated to a measure of the number of machine instructions the agents execute. More precisely, it is in fact the total number of passes through the Prolog call and redo ports.

undertake co-operative behavior. This form represents a trading scenario where there are multiple intelligent agents— but they do not exhibit properties expected of *MAS agents* since the organizational form forces the agents to be asocial in this respect.

The second form (figure 1b) additionally allows customers to be aware of other customers (and suppliers, other suppliers). Therefore, customer agents are able to form groups with other relevant customer agents, allowing them to co-operate by sharing tasks that are commonly undertaken. For example, sharing information about product availability of suppliers and formation of purchasing groups. Suppliers are similarly enabled. Topologically, this form is a fully connected mesh and represents a standard fully connected peer MAS.

The third form is identical to the second, with the exception that an intermediary agent that undertakes collective tasks is present. For example, centralization of the formation of a supplier-to-product catalogue and modeling and dissemination of user-preferences. This form represents and facilitates intermediary functions (brokerage, matchmaking, recruitment, facilitation, etc [14]). Rather than allowing agents to (s)elect one of their number to take on these tasks (for which its utility to its user may suffer greatly), an extra agent is introduced.

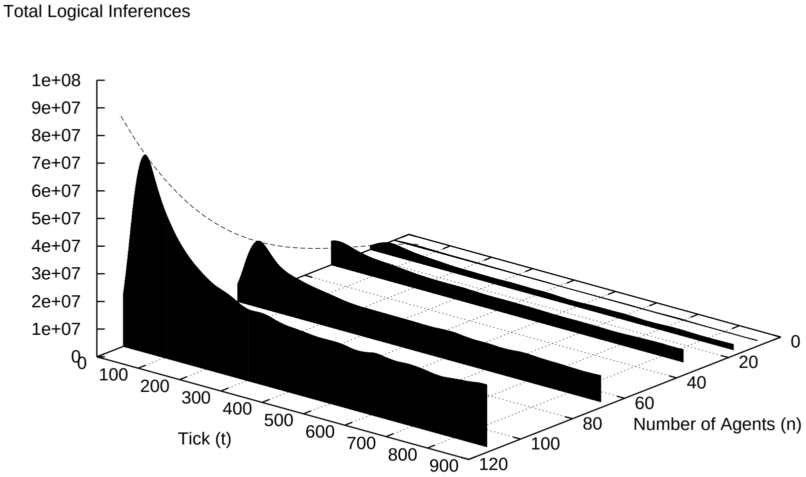

Fig. 2. Logical Inferences versus number of ticks versus number of agents for *Form 1* MAS

3.1 Resource requirements for form 1

Figure 2 shows the relationship between ΣLI_t, t and n.[4] Regardless of the number of agents, the amount of processing resources required can be split into two separate stages. Initially, the agents do not have models of each other so they require higher amounts of processing resources to achieve their goals. This is because they need to locate commodities and services and undertake model formation. As t increases, the completeness of the customers' models of the suppliers increase and they are able to more efficiently achieve their goals. Eventually the initial extra resource requirements needed whilst forming the models lessens sufficiently for the agents to assume a more stable resource requirement. We refer to this first (initial model formation) stage as the *convergence phase*. In the second stage—which we refer to as the *converged phase*— agents require resources to maintain the accuracy of these models (since users may change their buying habits and suppliers may alter product ranges or prices). (In figure 2 the convergence phase ends (and the converged phase begins) at approximately tick 400.) Models are maintained with the aim of preventing (or lessening) agents propagating incorrect information to the other agents (e.g. for recommendations), or wasting time approaching inappropriate acquaintances for group formation.

When using this organizational form, both customer and supplier agent groups have more or less the same functionality and are invariably trying to satisfy goals which are largely common (e.g. forming the same models of each other or buying the same commodities). However, because the acquaintance topology limits them, they cannot co-ordinate their activity, nor share information that could benefit one another.

When viewed from a system's level perspective this is a ridiculous position. There are massive amounts of replicated functionality, data and redundant computation and agents' experiences and discoveries do not benefit others. To this end, it is clear that sharing information and co-operating are necessary to increase efficiency and decrease resource requirements. Therefore, form 2 allows full acquaintance.

3.2 Resource requirements for form two

Figure 3 shows the situation where agents are able to detect and interact with all other agents. Hence, they can and *do* share information and co-operate for common goals. In comparison to the previous graph, we see that the graph is both quantitatively and qualitatively different. In addition to the resource requirements being very much higher, both its growth factor (against agent numbers) and shape differs.

The higher levels of logical inferences required reflect the fact that the agents with the same purpose (purchase/supply) model each other and reason with

[4] The graphs in this paper have been smoothed to show their general form. Specifically, they were smoothed with a compound windowed running maximum and windowed running average (in that order). To clarify the relationship between n and the shape of individual MAS runs, we show selected values of n rather than a surface.

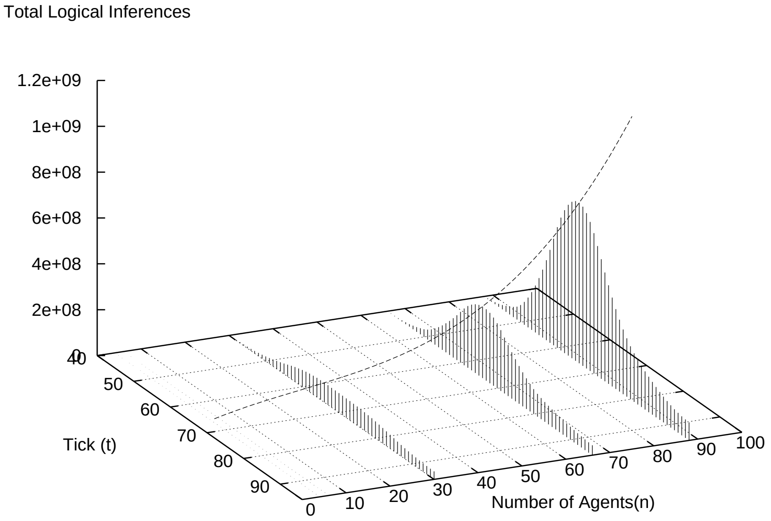

Fig. 3. Logical Inferences versus tick versus number of agents for *Form 2* MAS

these models to form groups. Formation, operation within and disbanding of joint activities is a relatively costly behavior to incorporate, compared to the act of modeling between customers and suppliers for purchasing purposes. However, the number of processing/message delivery cycles that are taken to reach the converged phase is very much reduced (convergence at approximately tick 100 - versus tick 400 for form 1) since the agents *do* share tasks and information. This is manifest by the shape of resource requirements for the convergence phase being roughly parabolic rather than asymptotic. The relationship between ΣLI_t and n remains of the same *order* of complexity (cubic). Any benefits that the agents derive because they co-operate are lost to the overheads of co-operation. (The agents in these tests perform relatively few tasks specific to the scenario compared to those necessary for MAS operation— i.e. most of the complexity and resource needs of the agents resides in general MAS interaction.)

The application scenario also illustrates an important point which can easily be overlooked when using order notation: that its information value is very limited. On paper, the difference between form 1 and form 2 is not great. Equations capturing the maximum of total resource requirements (during the convergence and converged phases) for a range of agent populations (valid only for $5 \leq n \leq 120$) are given in Table 1. [5]

As implemented systems, however, these minor differences may be crucial. Given a particular computational resource availability **limit**, form one may function perfectly well, whereas form two may not. Conversely, given a system where information changes rapidly, form two may be better (to ensure higher temporal

[5] These equations have been numerically derived to reproduce the maxima measured experimentally (correlation co-efficient not less than 0.9999).

Table 1. Growth Functions for Organizational Forms 1 and 2

	Max for Convergence Phase
Form 1	$\approx 4 \times 10^5 n - 5000n^2 + 60n^3 - 1 \times 10^6$
Form 2	$\approx 5 \times 10^6 n - 1 \times 10^5 n^2 + 1000n^3 - 3 \times 10^7$

	Max for Converged Phase
Form 1	$\approx 2 \times 10^5 n + 3000n^2 + 25n^3 - 1 \times 10^6$
Form 2	$\approx 7 \times 10^6 n + 10000n^2 + 133n^3 - 2 \times 10^6$

accuracy of the models). It is for this reason that we believe that scalability studies for MASs should be based on practical observation and measurement.

3.3 Resource requirements for form three

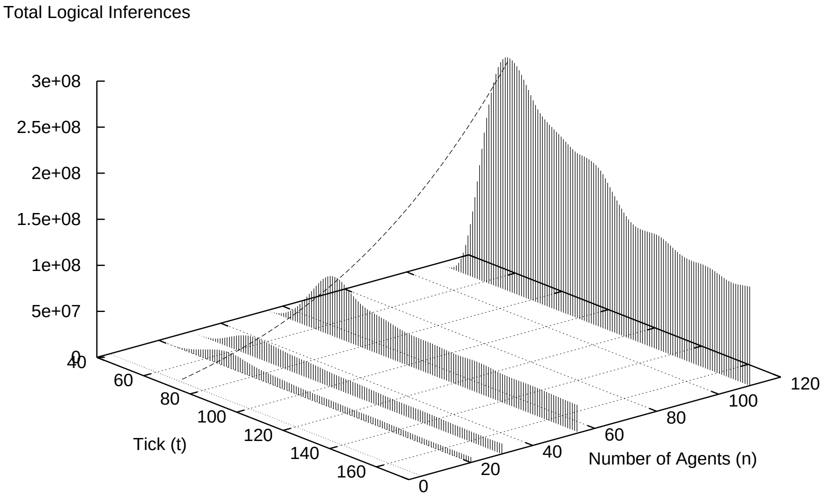

Fig. 4. Logical Inferences versus tick versus number of agents for *Form 3* MAS

Figure 4 shows the resource requirements of agents in the MAS where common tasks are centralized. As an example of a task that can be centralized, and to illustrate the irregular shape of these resource profiles, consider a common goal of customer agents: to keep each other updated with supplier commodity availability and prices. When customers obtain a new piece of information about a supplier (as a result of buying or failing to buy a commodity), this information is forwarded to other customer agents for whom it is judged relevant information. A simple way of determining which other customers will find this information relevant is to retain statistics on the number of times each other customer attempts to form a collective purchasing group for this item, or alternatively, sent an update (cf [12]). In form one, this behavior was impossible. In form two, each customer maintained a model for every other and reasoned with new pieces of information against these models, finally sending appropriate messages. Since every customer agent undertakes this task, delegating the modeling and reasoning to a single agent will require lower overall resources— instead of n models of n other agents, the intermediary needs only maintain n models. Clearly, the number of messages that need to be sent remains the same (n agents send a message to a maximum of $n-1$ others, which gives a collective total of n^2-n, conversely, n agents send one message to the intermediary agent who sends a maximum of $n-1$ in response to each— also resulting in a total of n^2-n). Therefore, the benefit of centralizing this task is a lower total of models (and processing overheads) and customer agents spend more time on other goals. The cost of centralization is that the information update arrives one tick later. This is reflected by the fact that the convergence/converged phase boundary occurs at approximately tick 170 for n=110.[6]

Hence, compared to form 2, collective centralized delegation results in lower overall resource needs, but increased latency in information sharing for those tasks. Therefore form 3 has a higher convergence time than form 2. The number of ticks required to reach convergence remains lower than form 1, but higher than form 2, and the total resource needs of all agents are lower than form 2 but higher than form 1.

4 Achieving adaptable organization

A gross characterization of the relative forms studied to date is shown in Table 2. Based on these generalizations, it would seem natural to select whichever form was pertinent to a given environment. For example, in a domain where the information that is modeled is stable for longer periods of time than is needed to reach the converged state, the form which has the lowest impact on users' agents should be used (form 3). Alternatively, it may be appropriate to select form 2 initially (to converge as quickly as possible) and then revert to a less demanding form (form 3), and so on.

Although the forms shown constrain all the interactions of the agents, it is intuitive that any given form may be inappropriate for a particular shared goal. Therefore agents need to be able to select individual forms for individual shared goals (cf Galbraith's contingency theory [9]). For example, keeping track of the existence of agents (if they have transient membership of the MAS) may be based on detecting a relatively infrequent event, so form 1 may be appropriate. Conversely, modeling suppliers' commodity availability may demand fast (and frequent) updates, so form 2 may be chosen.

[6] Unlike form 1, the point at which convergence occurs for forms 2 and 3 (which are co-operative) is a function of the number of co-operating agents. This is most readily seen in form 3 (figure 4).

Table 2. Relative comparison of organizational form attributes

	LISt	Ticks for Convergence	Relative Load on (User) Agents
Form 1	Low	High	Mid
Form 2	High	Low	High
Form 3	Mid	Mid	Low

Being able to distinguish these three basic forms sets the foundation for creating self-building and adaptive MASs. With appropriate definitions of acceptable performance specified by designers (such as maximum resources available to individual agents or the collection, or minima for model accuracy), agents can begin to change their organization to meet these demands.

In order to allow agents to dynamically change between organizational forms they have been augmented with several extra abilities: 1) the ability to remove (and add) knowledge of the existence of particular acquaintances to global or specific task data structures, 2) the ability to create intermediaries or destroy them, and 3) the ability to transfer model information and delegate tasks to other agents (and, in turn, become delegatees).

Since changing organization form has repercussions for the collective, it is necessarily a collective choice. For example, a sufficient number need to agree to create an intermediary. Agents therefore need a distributed method for triggering re-organization. In this scenario, triggers for re-organization are related to utility measures and relative resource requirements of tasks per tick. The parameters of agent operation that are appropriate for triggers (and their relative importance) should be dictated by the domain characteristics and application scenario. In this scenario, for example, achieving low purchase costs for commodities (cf utility) has a higher ranking than reducing processing resource requirements (cf efficiency).

As a worked example, consider a group of ten customer agents modeling each other (form 2), and further assume that agents' own models of their users' commodity preferences change rapidly. An agent may find that it has used most of its resources modeling and updating other agents but has benefitted little from doing so. For example, it has spent 8×10^6 logical inferences in the past 100 ticks on this task, but has utilized the resultant models less than 5 times during the same period. The agent therefore notes that this task has a low payoff and increments a counter to reflect the fact. (There being a similar decrease when utility improves.) Each agent has two counters: one being a tally of utility for its own operation, and a second representing the same information from acquaintances. Agents are therefore able to distinguish their own desire for re-organization from that of their acquaintances.[7] The agent then updates the modeled agents with this value. On receipt of this message, each agent updates

[7] The private and social counters and their trigger activation functions are used to decide whether to join groups undertaking re-organization.

its 'social' counter (which may trigger the signal to re-organize). If either the social or private trigger reaches its activation level, the agent transmits another message to *its* acquaintances upon receipt of which they take similar action if they also reach this point. If re-organization trigger updates continue to increase or a sufficient number of agents have passed their threshold for reorganization, a cascade of updates occurs between the agents and they enter a group task to reorganize. (Note that not all agents have to agree to re-organize— currently the minimum is at least 50%.)

When re-organization has been triggered and a group has formed for this task, the agents may take several actions based on the identified problem and their current organizational form. *Identifiable problems* for customers (i.e. they each have separate triggers) include 'supplier model errors high' (agent approaches supplier for commodity it does not sell), 'supplier model low coverage' (agent has to broadcast purchase requests to all suppliers and incurs a high amount of resources tracking requests and resultant model updates), and 'ratio of resources for modeling acquaintances to payoff low' (as above example).

In such situations, the actions that the agents can take are a) centralize a task, b) de-centralize a task/create a commitment to share common information/task, or c) end a commitment to share common information/task. Option *a* moves the agents into form 3, *b* moves the agents into form 2, and option *c*, into form 1. The choice of which form to adopt is based on the identified problem (i.e. the trigger that fired), the relative advantages and disadvantages of each form (as in the previous table) and the current form.

CENTRALIZATION OF A TASK

The agents vote to select an intermediary agent or an agent is randomly chosen to create one. (If no intermediary exists, or those that do refuse to take on new tasks, one is created.) When an intermediary agrees to take on another delegated task, the agents take the following actions independently: 1) all agents extract the rules relevant to the task from their Prolog database[8], un-instantiate them, and send them to the intermediary. 2) on receipt of a message to proceed from the intermediary, the agents send all the relevant data from their databases to the intermediary. The intermediary then responds confirming receipt of the rules and data. During this process, the intermediary checks that all the agents have sent the same rules (i.e. verifies that all the agents about to delegate a task to it are going to stop performing the same task), after the success of which, the signal to proceed is sent to each agent in the group. When the intermediary has received all the data from the agents, it removes a guard on the processing of the relevant rules and sends a message to each agent informing them to remove the relevant rules and replace them with another that represents a commitment to undertake that task for them. Additional rules may be sent which cause the agents to forward runtime data to the intermediary. (All the rules the intermediary receives and/or sends back are part of the package of rules that the agents send it.)

[8] Additional meta-information is included in the agents which described the rules relevant to particular tasks.

Ending commitments to shared tasks
A rule for a shared (information) task contains a query to the agents' belief database from which it retrieves the agents to whom it has a commitment to consider forwarding information. Such rules are modified by removing commitment facts from the appropriate belief structures.

Entering a commitment to a share task
Same as for ending commitments, except that commitments are added.

4.1 Resource requirements for form four

Figure 5 shows the resource requirements of agents when utilizing organizational adaption (form 4) as compared to forms one to three. Despite the qualitative difference from the other forms, form 4 initially requires less resources than form 2, but more than forms 1 and 3. Eventually, it requires less resources than form 3.

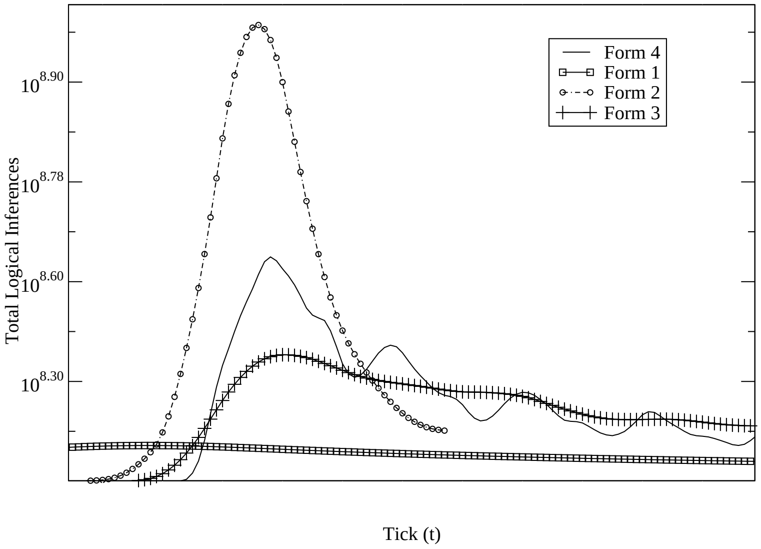

Fig. 5. Total Logical Inferences of Forms 1 to 4. $n = 110$ for forms 1 to 3, $n = 111$ for form 4 (+1 intermediary)

Although form 4 also shows a gradual decline in resource usage (on average), it does have two unique features. First, it undergoes major perturbations, and second, it does not appear to reach a stable resource level (i.e. show signs of being in the converged phase).

The reason that this form does not show the smoothness of forms 1 to 3 is because individual agents do not utilize the same form for all tasks, and not all agents are identical. That is, since the graph is a summation of many agents employing forms 1 to 3 on a per task basis, it is necessarily irregular. The perturbations in the graph and the apparent lack of evidence for convergence show an emergent feature of form 4— that it is periodic. Finally, forms 1 to 3 are isomorphic with regard to n $\geq$10. That is, that any two graphs of resource requirements for any number of agents, $n \geq 10$, can be linearly scaled to fit each other. Form 4 does not have this property. For example, the distance between the perturbation maxima varies with n. Unfortunately this means that accurately projecting the resource requirement curves for higher (untested) numbers of agents is impossible for form 4.

Finally, the figure shows form 4 being a suitable amalgam of forms 1 to 3 in terms of resource requirements. The maximum resources required falls some-way below that of form 2, and it also (eventually) falls below that of form 3. We do not consider the agents of form 1 to constitute a multi-agent system since they are asocial, therefore of the MAS forms presented here, we suggest that form 4 (which is organizationally adaptive) is more scalable than form 2 (because the maxima remain lower) and also form 3 (because the converged phase maxima are lower).

5 Related Research

Publications dealing with scalability and multi-agent systems are relatively few (but gaining in number). In order to place this work in the context of others, we broadly partition MAS scalability research into categories thus: 1) formal and experimental scalability analyses of particular co-operation and resource distribution protocols (e.g. [5] [6]); 2) creation of scalable protocols (e.g. [17]); 3) tools for (pre-implementation) predication of performance and design evaluation (e.g. [16]); 4) clarification of scalability issues (e.g. [15]); 5) scalable architectures and applications (mostly residing in the information retrieval arena); and 6) algorithms and techniques for increasing scalability through change of the MAS environment.

We view this paper as relevant to the last category. Insofar as the research deals with dynamic changes to the structure of the problem solver environment— rather than the protocols they employ. That is, techniques that indirectly affect agents and MASs by altering the agents (but not their mechanisms) or the environment in which they reside. Examples, of this are [1] ("a framework for Dynamic Reorganization"), [13] ("An Organizational Approach to Adaptive Production Systems"), and [10] ("Self-Adaption and Scalability in Multi-Agent Societies").

The creation and destruction of intermediaries as extra (parallel) processing capability is indirectly based on [13]— where not only do Ishida *et al* introduce organization self-design (OSD), but also allow two or more agents to *'compose'* (combine with each other) and *'decompose'* when organizational load is heavy or light respectively. Furthermore, a mechanism is given whereby composition and decomposition can occur at runtime without affecting the operation of the 'agents'. Explicitly included in their scheme is an organization self-designer in each agent (cf [3]). Our approach differs in that lightly loaded agents do not automatically take on extra tasks (since they must maintain a minimal utility for their owners). Also, whereas their agents effectively cease to exist after a composition into a hybrid, our (customer and supplier) agents can not. The equivalent of reduction in total system load occurs because of the centralization/ decentralization of a task, or delimitation (through disbanding and reforming) of a social group which places too large a load on its members.

In [1], André *et al* refer to dynamic reorganization (DR) through which their centralized MAS (viewed as a hierarchic graph of service decomposition) can swap to alternative decompositions. The new decomposition is retained if the alternative performs worse than the original. The scenario we presented does not allow the agents to form compound services by the combination of existing facilities. In as much as our agents can disband and reform groups, and centralize/de-centralize tasks, no explicit representation of the previous state is retained.

Lastly, in [10], the main thrust of the argument is that the problems of scalability can be managed by changing the relative amounts of processing time and other resources to the various aspects that operate within agents. For example, increasing the amount of time in manipulating knowledge representations. The configuration of the agents is represented as a search space so that when the agents' ability to perform drops below an acceptable level it will move within the search space. The first move in this search space is random. Locally, agents perform hill climbing to achieve local optima of parameters. However, a centralized agent constructs a picture of this search space using the information local to agents (i.e. local agents provide fragments of the picture). The centralized agent uses this information to prevent the agents from settling on local maxima when better solutions are 'nearby'. Viz., the centralized 'monitor' agent is able to direct the agents in their search (e.g. beyond their own fragment of the search space). However, it can not be certain that the search space is smooth and therefore the hill climbing search methods used by the agents and central monitor agent may not be appropriate in all domains. (His approach is presented as a general framework.) Critically, the monitor (as a permanent centralized entity) may constitute a bottle-neck itself and prevent scalability.

Furthermore, Gerber uses the term *Holon* and the notion of abstract resources. Abstract resources are roughly equivalent to the compound services of [1], and holons are essentially, collections of agents (which may be considered as a group or as a single agent) with partially limited autonomy (see [10] for the full extent of his hypothesis). Taken as a whole, Gerber's framework and mechanisms for adaption are both microscopic and macroscopic in nature. Despite having been tested in several application scenarios, none are represented in a form where population size is paramount (so no direct comparison can be made). We suspect that Gerber's framework is better able to fine tune response to small changes in scale than our approach, which we believe may be coarser in comparison. However, we note that our approach is less invasive (since it only requires the ability to change acquaintance topologies and group commitments rather than interfere with internal operation of agents).

Combining our work with that of other researchers, we summarize the approaches that have successfully been employed to achieve organizational adaption thus: a) agent/process de/composition and creation/destruction, b) alternative decomposition/ recomposition of compound abstract resources and dependencies, c) centralization/ decentralization of common goals and alteration of acquaintance (and thus group) topology, and finally d) re-allocation of relative processing power of agents internal processes. Clearly, all these methods could be combined. Moreover, in addition to agents having access to knowledge of their performance and limitations, they could benefit greatly from being able to form and reason with (formal) models of their dynamic organization [4] and [2].

6 Conclusions and Future work

Allowing agents to build and maintain their own organizational structure requires that they are able to decide for themselves what tasks should be shared, delegated, or individually pursued, and which acquaintances are of little benefit. This means they must be aware and be able to meta-reason about their own internal efficiency/efficacy and goals, infer or question that of their acquaintances, and the goals of the system as a whole. It also entails that agents are able to create and annihilate other agents, delegate and surrender tasks and information, modify their own operation and influence that of others (because organizational changes are taken by the collective, not the individual). The algorithm sketched above does show that organizational flexibility has advantages. In summary, although we believe that large scale multi-agent systems should be self-building and self-organizing to allow for a higher degree of scale tolerance, we believe that much greater testing is required in several domains of application. One limitation of the scenario implementation is that the market dynamics is relatively stable. How re-organization copes with fluctuations in user's demands remains unclear, and so too a complete analysis of the influx (and exodus) of population. Secondary in importance to this omission are the effects of the trigger levels for re-organization, and the similarity metrics used by the (customer) agents to decide on group formation. The values to which these are set may have a major effect of the benefit of re-organization. Indeed, determining these values remains a matter of experimentation. Also, relaxation of the synchronized method of operation of agent execution needs to be undertaken since MASs are almost exclusively asynchronous. Answering the questions raised by these issues, constitutes our future work. Finally, we note that creating an algorithm to increase the scalability of a MAS is not the same (and does not imply) that the algorithm is *itself* scalable.

References

1. J.-M. André, J.-M., Mouginot, A., Venet, M.: A Framework for Dynamic Reorganization. Ninth European Conference on Artificial Intelligence (1990) 31–37
2. Baligh, H.H., Damon, W.W.: Foundations for a Systematic Process of Organization Structure Design. Journal of Information & Optimization Sciences **1**(2) (1980) 133–165
3. Brazier, F.M.T., Jonker, C.M., Treur, J., Wijngaards, N.J.E.: Deliberate Evolution in Multi-Agent Systems. Third Annual International Conference on Autonomous Agents (1999) 356–357
4. Costa, A.C.R., Demazeau, Y.: Toward a Formal Model of Multi-Agent Systems with Dynamic Organization. Second International Conference on Multi-Agent Systems (1996) 431
5. De Wilde, P., Nwana, H.S., Lee, L.C.: Stability, Fairness and Scalability of Multi-Agent Systems. International Journal of Knowledge-Based Intelligent Engineering Systems **3**(2) (1999) 84–91
6. Foner, L., Crabtree, I.B.: Multi-agent matchmaking. British Telecom Technology Journal **14**(4) (1996)
7. Foss, J.D., Garcha, K., Turner, P.J., Jennings, N.R.: Brokerage in an Information Economy. INET 2000, 10th Annual Internet Society Conference, Yokohama, Japan (2000)
8. Fox, M.S.: An Organizational View of Distributed Systems. IEEE Transactions of Systems, Man, and Cybernetics **11**(1) (1981) 70–79
9. Galbraith, J.: Designing complex organizations. Addison-Wesley Publishers (1973)
10. Gerber, C.: Self-Adaption and Scalability in Multi-Agent Societies (PhD thesis). Universität des Saarlandes (1999)
11. De Greene, K.B.:, The Adaptive Organization, anticipation and management of crisis. John Wiley & Sons, Inc. (1982)
12. Huhns, M., Mukhopadhy, U., Stephens, L.M., Bonnel, R.D.: DAI for Document Retrieval: The MINDS Project. Chapter 9 of Distributed AI Volume II, Pitman Publishing Ltd (1987)
13. Ishida, T., Yokoo, M., Gasser, L.: An Organizational Approach to Adaptive Productive Systems. Eighth National Conference on Artificial Intelligence (1990) 52–58
14. Kuokka, D., Harada, L.: Matchmaking for Information Agents. Fourteenth International Joint Conference on Artificial Intelligence (IJCAI) (1995) 672–678
15. Lee, L.C., Nwana, H.S., Ndumu, D.T., De Wilde, P.: The stability, scalability and performance of multi-agent systems. British Telecom Technology Journal **16**(3) (1998) 94–103
16. Rana, O.F., Stout, K.: What is Scalability in Multi-Agent Systems?. Fourth Annual International Conference on Autonomous Agents (2000)
17. Shehory, O.: A Scalable Agent Location Mechanism. Lecture Notes in Artificial Intelligence, Intelligent Agents VI (2000) 162–172
18. Shoham, Y.: AGENT0: A Simple Agent Language and Its Interpreter. Ninth National Conference on Artificial Intelligence **2** (1991) 704–709