# Adaptive Brokering in Agent-Mediated Electronic Commerce

Timothy J Norman, Derek H Sleeman, Nial Chapman*

Dept of Computing Science, University of Aberdeen

Aberdeen UK

{tnorman,sleeman}@csd.abdn.ac.uk, n.chapman@jkna.com

**Abstract**

In this paper we advocate an approach that extends models of trust and reputation to take into account the competence of agents. The argument is that such an approach will lead to more reliable agent-mediated electronic commerce environments than those in which agents are simply considered to have cooperated or defected. If there is a mismatch between the advertised and actual competence of an agent and the agent fails to complete a task as a consequence of this mismatch, then the description of this agent's competence should be refined in addition to any loss in reputation. Consequently, this agent is less likely to be employed for an inappropriate task in the future. Two models of adaptive brokering are presented in this paper that illustrate the use of refinement techniques in developing effective brokering mechanisms for agent-mediated electronic commerce.

## 1 Introduction

In both managed and open agent-mediated electronic commerce environments trust in and reputation of agents are important issues. For example, in a managed auction house it is important for the participants to trust in the integrity of the auctioneer. This trust is typically based on the reputation of the institution that the auctioneer represents. In traditional marketplaces, or auction houses, it is common to maintain that reputation, at least in part, by requiring all goods offered for sale to be given over to that institution for inspection prior to them being offered for sale. However, in electronic commerce environments such controls are rarely possible — the agent operating as a broker must assess the reliability, quality of service and other aspects of a potential service provider on the basis of past encounters. For example, a service provider that has consistently provided a reliable service of a specific type in the past, as reported by consumers of that service, may be considered to have a good reputation for the delivery of that specific service. This is an instance of the more general problem of modelling the reliability of an agent with respect to a specific task (or problem solving activity) and hence how trustworthy that agent is [1]. The more frequently an agent fails to deliver on its agreements, the

*Currently Director of JKNA E-Business Solutions Ltd.

more likely that it will be considered untrustworthy and the less likely it will be employed in the future. It is this that provides the context of the research reported in this paper. We argue, however, that existing models of reputation and trust in multi-agent systems are inadequate, principally because they do not address the important question of modifying a model of the competence of an agent as a result of past encounters.

Schillo *et al.* [7], for instance, investigate the use of learning in the generation of a model of the trustworthiness of agents within electronic commerce scenarios. The question addressed is that, given there are a number of agents offering the same or equivalent services, which agent is the most trustworthy on the basis of past experience. All other things being equal, the most trustworthy agent should be the best choice. In determining the trustworthiness of an agent, evidence is taken from experience (past encounters with that agent) and the testimonies of others. This idea of using both evidence from past encounters and recommendations from others [5] has been further studied in the context of trust and reputation in multi-agent systems. Yu and Singh [9] present a decentralised reputation system based on an acquaintance model to help in the identification of appropriate witnesses, and on Dempster-Shafer theory for combining evidence. Sabater and Sierra [6] extend this use of combining evidence and social (or acquaintance) networks by representing the trustworthiness of an agent in more than one dimension. A seller agent could, for example, be judged on the basis of whether it can be trusted to deliver on time, or whether it can be trusted not to over-charge for the service concerned. In all of these models of trust and reputation, all failures are assumed to be due to the agent being either *unwilling* or *unable* to perform the task concerned; i.e. no distinction is made between defection and a lack of competence. The question is reduced to: which agents is least likely to fail in this encounter?

In contrast, we would like to assume, for the sake of this discussion, that all failures are due to the agent being *unable* to complete the agreed task rather than unwilling; i.e. agents always cooperate, but fail due to circumstances outside of their control. We then focus on a different aspect of the same problem: which agent is most competent at this task? Consider an agent that represents a wholesaler of some commodity to clients within a supply chain; e.g. supplying bananas to supermarkets. Suppose that this agent has signed a contract to supply some quantity of the commodity within a specific deadline, and suppose that the deadline is passed and the delivery has not arrived. Is it appropriate to simply record this as a failure and modifying a representation of the reputation of this supplier? Certainly, but it may also be reasonable to determine the cause of this failure, thus increasing the likelihood of there being fewer future problems of this kind with respect to this supplier. Suppose that the supplier can meet the quantity required, but needed additional time in this instance to complete the delivery. According to the model proposed by Sabater and Sierra [6], this will be recorded as a failure to meet the deadline (a more fine-grained analysis of the agent's past performance than that possible in models such as that proposed by Yu and Singh [9]). There is, however, an alternative. Suppose that the broker records the fact that the delivery time required for this quantity of bananas was, perhaps, an underestimate. If, over a number of encounters, this problem occurs with some regularity for this agent under similar circumstances, it may be better to refine the model of the tasks that this agent can competently perform.

Evidence from such encounters should be recorded, so that appropriate modifications to a representation of the competence of this supplier may be made. This may be done by, for example, changing the constraints on delivery time for this supplier, commodity pair (see section 2). If a broker is able to adapt its models of the competencies of agents in this way, it will be able to develop a more accurate picture of the services available from agents, and provide a more reliable service to its customers.

We advocate an approach that extends models of trust and reputation to take into account the *competence* of agents. The argument is that such an approach will lead to more reliable agent-mediated electronic commerce environments than those in which simple cooperate/defect level information is recorded, but it should be noted that the work reported in this paper is complementary to these existing models of trust and reputation. In this paper we present two simple mechanisms whereby a broker may use the minimum of information of successes and failures in service delivery to adapt its models of the competencies of agents, and use supply chain scenarios to illustrate them (see section 2.1). In the first model, learning techniques are employed to enable the broker to adapt competence models through experience (section 3). In the second model, we investigate a simple mechanism whereby the broker and supplier may agree on a modification to the model of the competence of that supplier (section 4). Before these mechanisms are discussed, however, we present the broker-based multi-agent architecture that provides the context of this research.

## 2 A broker-based architecture

In this paper, we make a number of simplifying assumptions. First, we assume that there is a single broker that provides the service of putting together a coalition of agents that can, together, complete the task set by an agent. The use of a single broker introduces a single point of failure for the system and a potential bottleneck. An alternative solution would be to have a number of agents that provide this brokering service (or even allow any agent to take on the role of a broker) and introduce a mechanism whereby information about the reputation and competence of agents within the community to be propagated through social connections [6, 9]. In this work we are not concerned about how information on the reputation or the competence of an agent is shared within a community. Our focus is on the recognition of a conflict between the advertised competence of an agent and its perceived performance, and the resolution of this conflict. For this reason, the use of a single broker in this analysis is a reasonable simplifying assumption.

Second, as discussed in the introduction, we assume that if an agent has advertised a competence, then it is willing to perform this problem-solving activity for any other agent within the system at any time. This has been done to simplify the interaction between the broker and potential suppliers so that the broker makes a local decision on which agents to involve in the completion of a task. We do not, for example, use the Contract Net or any other contracting mechanism for the distribution of tasks. This is because we are interested primarily in the identification and resolution of conflicts between advertised and perceived competence, and not in the use of auction mechanisms.

Further, we use a simple model of agents' competencies. Each competence is modelled in terms of the competence type, and constraints on its inputs and outputs. For example, an agent may advertise that it is competent at providing a transportation service. The set of constraints on the inputs to a competence, $\mathrm{CI_j}$, are the preconditions of the competence. For example, the agent providing a transportation service may have a maximum weight for each unit, which would correspond to the capacity of the largest truck in the fleet available to that delivery company. The set of constraints on the outputs of the competence, $\mathrm{CO_j}$, are the postconditions of the competence. For example, the agent providing the transportation service may constrain the time at which the task may be expected to complete.

The process of completing a task is assumed to be coordinated by the broker in the following stages: advertising, problem specification, the production of a solution to the problem (a plan of action), the coordination of task execution, and the resolution of any conflicts between advertised competence and observed behaviour.

Service providers advertise their competencies to the broker. These adverts consist of the competence type, constraints on the inputs to and on the outputs from the competence as described above.

Agents that wish to have a problem solved — e.g. a service delivered — submit a request to the broker in the form of a task description, T, and a set of constraints, S which the solution must satisfy. For example, the task may be to have bananas delivered and the constraints may include the quantity, quality and deadline by which they must be received. This constitutes the problem to be solved by the broker.

The broker may then construct a plan to solve this problem with the information it has about agents' competencies. Let us assume that the task, T, can be solved by the application of a series of processes applied to the intermediate results produced. The plan is, therefore, an ordered set of problem-solving activities:

$$\mathrm{T : PS_1\,[CI_1, CO_1] \rightarrow PS_2\,[CI_2, CO_2] \rightarrow \ldots \rightarrow PS_N\,[CI_N, CO_N]}$$

Here, the task, T, is shown as having been refined into a total ordering of problem solving activities, $\mathrm{PS_1 \ldots PS_N}$, each with input, $\mathrm{CI_1 \ldots CI_N}$, and output, $\mathrm{CO_1 \ldots CO_N}$, constraints. In general, however, this may be a partial ordering of problem solving activities, and hence some may be executed in parallel. Furthermore, we have assumed that only a single plan would be generated, but the approach could easily be extended to deal with a solution tree.

During the planning phase only the competence information is available to the broker; i.e. the $\mathrm{PS_j[CI_j, CO_j]}$ terms in the example given the previous section. This information is used during the planning stage to make predictions about the outputs that will be generated during execution, and hence enable the construction of a plan. The actual outputs, $\mathrm{O_1 \ldots O_N}$ shown below, only become available during execution. For example, the output generated by problem solver $\mathrm{PS_1}$ on completion of this operation is shown below as $\mathrm{O_1}$.

$$\begin{aligned}\mathrm{T: PS_1\,[CI_1, CO_1] \rightarrow O_1 \rightsquigarrow PS_2\,[CI_2, CO_2] \rightarrow O_2 \rightsquigarrow \ldots}\\ \mathrm{\ldots \rightarrow PS_N\,[CI_N, CO_N] \rightarrow O_N}\end{aligned}$$

This, typically distributed, execution of the plan for solving task, T, provides the broker with information about how the agents assigned problem solving activities performed. This provides useful information for the broker to check and possibly refine its model of agents' competencies.

During execution the broker acts as a co-ordinator. That is the broker will pass $\mathrm{I_j}$, or $\mathrm{O_{j-1}}$, to $\mathrm{PS_j}$ and will check that this satisfies the constraints, $\mathrm{CI_j}$. Similarly the output $\mathrm{O_j}$ will be checked against $\mathrm{CO_j}$ and passed to the next agent if the tests are satisfied. However, we wish to investigate the situation where an agent's observed behaviour does not match its advertised competence. There are two cases that we consider. The first is where the inputs, $\mathrm{I_j}$, to agent $\mathrm{PS_j}$ satisfy the constraints specified in the competence description, but the outputs $\mathrm{O_j}$ do not. Here, the agent does not appear as competent as had been thought and the competence description may need to be *specialised*. The second is where the the inputs to the agent do not satisfy the constraints specified in the competence description and the outputs do. In this case it appears that the agent is more competent that its descriptions suggest, and so these may need to be *generalised*.

The agent execution framework being used here also provides the opportunity to experiment with agents to see if they are indeed more competent than their descriptors suggest. Suppose $\mathrm{PS_j}$ provides the expected output when provided with $\mathrm{I_j}$ ($\mathrm{O_{j-1}}$), then this input can be modified say to include a wider range of values or an object suggested by background knowledge. So for instance suppose $\mathrm{PS_2}$ executed satisfactorily with an input value of 6, then we might try values 5 and then 7, if both were successful, and 4 and 8 were unsuccessful, then the associated input constant 6 would be replaced by the range 5–7. Alternatively, if $\mathrm{PS_3}$ operated satisfactorily with one of its values being "apple" then background knowledge might suggest we try "banana", "orange" etc, and if they were both successful then the competencies would be rewritten to include the descriptor "fruit".

Failure within the execution phase of this process may lead the broker to refine its model of the competencies of specific agents within the coalition. Note, however, that there may be many causes of failure during the execution of a plan: a resource upon which the agent depends becomes unavailable, the completion of the problem solving activity has a 'likelihood of success' associated with it, and so on. As we have seen, failures that are of interest here are those in which there is a mismatch between the expected and actual outputs of a problem solving activity.

In discussing the execution phase of the process, the sorts of discrepancies which we might encounter were outlined. These can be summarised as agents being unable to do some of the tasks which would be expected of them given the description of their competencies; in a certain sense these agents are *incompetent* and their descriptions need to be specialised to exclude the tasks that the broker now knows they are unable to do. The other situation identified is where the agents are able to do a wider range of tasks than suggested by their descriptors and so their descriptors need to be generalised to include these additional situations. (In a sense these agents are being *modest* about their capabilities.) These situations (some more difficult to detect than others) provide case-based evidence for possible revision, or refinement, of the broker's model of an agent's competence.

In the following sections we discuss in detail two strategies through which a broker may refine its model of the competencies of agents, so that it may develop and maintain a more accurate picture of the tasks that agents can perform. In the first model presented, the focus is exclusively on the use of refinement techniques by the broker to modify its model of agent competencies from experience (see section 3). This is then extended into a model that takes into account both the refinement of agent competencies and the agents' reputation (see section 4). Before describing these mechanisms in more detail, however, we present a couple of supply-chain-like scenarios for the purposes of illustration — these being the scenarios in which the implementation of these adaptive brokering mechanisms have been tested.

## 2.1 Two Scenarios

In this supply chain scenario the task set by the customer is for the broker to arrange delivery of a quantity of bananas by some deadline. This customer agent is assumed to be a representative of an organisation such as a supermarket, and supplier agents are assumed to represent banana wholesalers. The transportation agents are representatives of haulage firms who provide the transportation of goods from suppliers to customers. The broker agent, as discussed above, coordinates the completion of the task set by the customer (see figure 1).

The broker will construct a plan (in this case the planning task is rather trivial) by choosing a supplier and transportation agent that can carry out the task within the constraints provided. The plan will be executed, the supplier and transporter informed, and the bananas will, if all goes well, be delivered to the customer.

The first participating agent is the customer agent: the representative of a supermarket that is attempting to purchase bananas. Let us assume that the

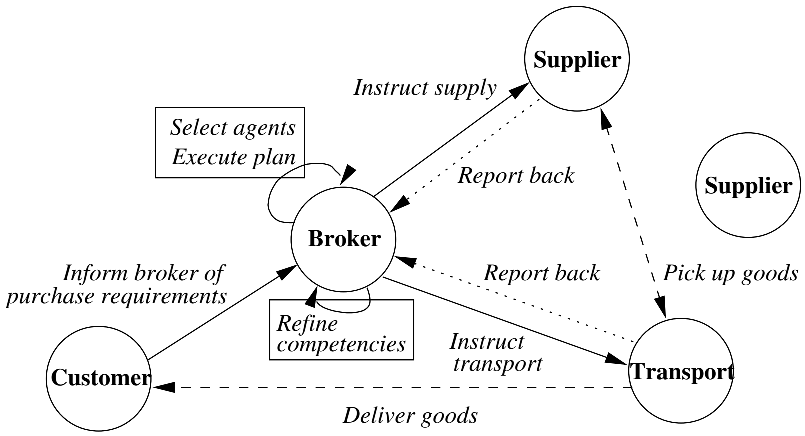


Figure 1: A supply chain scenario.

customer agent may specify three constraints on the performance of the task of banana supply: the number of pallets of bananas required, the unit cost of supply (this includes the purchase price plus delivery costs) and the deadline to be met. Note that in this paper we are not concerned with the negotiation of a contract to supply; we assume that if the combination of these constraints as specified by the customer cannot be met, the broker simply reports failure. In a complete solution to this type of scenario there will be trade-offs to consider between cost and delivery time, for example, that would require some form of bartering between the customer and broker.

There are two types of service provider in this scenario: banana suppliers and transportation service providers. Banana suppliers advertise their competencies in terms of the minimum and maximum number of pallets of bananas that they can supply within a specific period of time and to a specific unit cost. Each supplier can advertise a number of instances of this competence type: for example, one supplier may advertise that it can supply up to 50 pallets within one day at £20 per pallet, and it can also supply between 50 and 200 pallets within three days at £19 per pallet. Transportation service providers advertise their competencies in terms of the number of pallets that they can deliver, the time required for delivery and the delivery cost (for the sake of simplicity we ignore distance). For example, one delivery agent may advertise that it can deliver up to 60 pallets within one day for £5 per pallet, and that it can deliver up to 60 pallets within two days for £3 per pallet.

The second scenario considered concerns the management of the production of vehicle suspension. The task set by the customer is for the broker to arrange the delivery of the necessary parts to make the suspension for a vehicle that has certain characteristics. The customer is assumed to be a representative of an organisation such as a car manufacturer. The supplier agents are representatives of shock absorber suppliers or spring suppliers (the two main components of a suspension system). The broker, as discussed above, coordinates the completion of the task set by the customer.

In this scenario the customer will submit a request for suspension to the suspension broker indicating the type of vehicle required (off-road or on-road), the weight of the vehicle and the type of performance required (high performance or normal performance). The broker uses knowledge of suspension systems to then make a decision on the type of shock absorber and spring that is required. With this information and information on the competencies of shock absorber and spring suppliers, it arranges for the supply of these parts. If all goes well the required parts will be gathered and the suspension system built to the customers requirements.

## 3 Adapting Competence Models by Experience

The first strategy is for the broker to use refinement techniques, after every encounter, to develop an accurate picture of the competencies of agents. The broker agent holds a view of each of the agent's competencies in a table. It uses these values to determine which agent it will choose to perform a task — which agent it considers, at the present time, to be most suited to the task. If there are two or more suitable agents the broker selects one to ensure fair distribution of task instances. Following task execution, the model of that agent's competence may be revised.

Following the receipt of an order to supply bananas, the broker searches for supplier/transport agent combinations that can meet the criteria specified. Furthermore, for the sake of this investigation, we allow the broker to consider supplier/transport agent combinations where their combined constraints do not meet those specified by the customer. The following two types of case are then considered as information on which to refine the model of an agent's competence: when an agent fails to meet its advertised competence, and when an agent succeeds in fulfilling a task that is outside its advertised competence.

Let us suppose that the supplier and transport agents provide accurate information about unit cost and delivery time; i.e. the only variable that is to be considered here is the number of pallets of bananas that a supplier can supply and a transport agent can deliver. In the following we denote the number of pallets of bananas that the customer requires as $n$. We denote the minimum and maximum number of pallets that an agent, $i$, has advertised that it may supply/transport under competence $c$ as $c^i_{min}$ and $c^i_{max}$ respectively, where $i$ is a supplier, $i \in S$, or a transport agent, $i \in T$. We also denote the set of all supplier agents' competencies as $C^S$ and the set of all transport agents' competencies as $C^T$. Now, we can define the set of all competencies for suppliers and transport agents such that the number of pallets of bananas required is either met by that competence or exceeds the competence by some $\delta$; these are denoted by $C^S_\delta$ and $C^T_\delta$ respectively. The definition of $C^S_\delta$ is as follows, and $C^T_\delta$ is defined in a similar way. $C^S_\delta = \{c^s \in C^S \text{ s.t. } c^s_{min} - \delta \leq n \leq c^s_{max} + \delta\}$

In selecting a set of candidate supplier/transport agent pairs, we first create those pairs such that the advertised competence satisfies the required number of pallets of bananas for at least one of the agents involved; i.e. the task is outside the advertised competence of at most one of the agents involved. The degree to which the task may be outside the advertised competence of an agent can be varied by varying $\delta$. We define the set of pairs of competencies $P_{\delta_s,\delta_t}$ that meet the cost and time criteria exactly and meet the constraint on the number of pallets within $\delta_s$ for the supplier and $\delta_t$ for the transport agent with the aid of a predicate function that checks cost and time constraints, `costAndTime`(): $P_{\delta_s,\delta_t} = \{(c^s, c^t) \text{ s.t. } c^s \in C^S_{\delta_s},\ c^t \in C^T_{\delta_t},\ \texttt{costAndTime}(c^s, c^t)\}$. Now, the set of pairs of agents that are candidates for the task concerned with a variation of $\delta$ is candidates $= (P_{0,\delta} \cup P_{\delta,0})$

The actual pair employed will be selected according to the number of times each agent has been selected to ensure that tasks are fairly distributed among all able supplier and transport agents. If there is a tie (i.e. more than one pair has been employed an equal number of times) a winner is selected at random. Let the pair that is actually selected be denoted by $(c^s, c^t)$.

Following the execution of these tasks the broker checks whether the output of each subtask (supply and delivery) meets the criteria specified in the task set by the customer. The competence of one of the participant agents is then refined if one of the following conditions are met:

- If $c^s$ succeeds and $(c^s, c^t) \in P_{\delta,0} \setminus P_{0,0}$; i.e., the supplier succeeded in satisfying a constraint that is outside its advertised competence. In this case, competence $c^s$ is generalised.
- If $c^s$ fails and $(c^s, c^t) \in P_{0,0}$; i.e., the supplier failed to satisfy its advertised competence. In this case, competence $c^s$ is specialised.
- If $c^t$ succeeds and $(c^s, c^t) \in P_{0,\delta} \setminus P_{0,0}$; i.e., the transport agent succeeded in satisfying a constraint that is outside its advertised competence. In this case, competence $c^t$ is generalised.
- If $c^t$ fails and $(c^s, c^t) \in P_{0,0}$; i.e., the transport agent failed to satisfy its advertised competence. In this case, competence $c^t$ is specialised.

The learning techniques used in this system are well understood, and are basically extending a range of values to include an additional data point (i.e. generalisation), and refining a range to exclude a particular data point (i.e. specialisation). Additionally, in an earlier section we discussed replacing banana by the class fruit. This is generalisation against a predefined hierarchy. Details of appropriate algorithms capable of the specialisation and generalisation tasks discussed above are given in Mitchell [4]. These techniques have been used in the REFINER+ system which has the capability of inferring class descriptions from class-labelled instances; it can also suggest how overlaps between classes might be removed, Winter and Sleeman [8].

Perhaps the most important questions about the use of refinement techniques are determining how much to refine the failing/successful agent and when to refine the failing/successful agent. In the case of a failing agent, the system will automatically refine the agent's competence by an amount that is determined by the extent of the failure. That is the algorithm applies a minimum specialisation. Indeed, it is generally a conservative algorithm and the generalisations which are applied are also minimal. Additionally, as presently implemented these learning actions are applied immediately after a "mismatch" is detected, and such refinements take place regardless of the agent's past behaviour record. In a real life situation it may not be fair, or indeed sensible, to adopt this approach.

The second question that should be asked is how much an agent should be refined (effectively punished or rewarded). The mechanism discussed here refines the competence description of an agent on the extent of the failure or success. An agent that fails a task by a small margin will have its value refined less than an agent that fails a task by a large margin. The converse is true with successful agents. However, there may be situations in which a normally successful agent fails a task by a very large margin, or a normally unsuccessful agent also fails a task again by a large margin. Should the successful agent be punished (have it value refined) as much as the unsuccessful agent or should the successful agent's past record be taken into account when determining the level of punishment?

These questions, and many more like these, are extremely important questions that must be addressed in the design of systems that employ techniques such as those discussed in this section. Rather than addressing these issues here, we investigate an extension to this model that moves toward a coherent model that distinguishes between defection and lack of competence.

# 4 Adapting Competence Models by Agreement

One of the principal limitations of the mechanism discussed in the previous section is that the model of an agent's competence recorded by the broker is not available to the supplier. A supplier advertises what it believes it can do, and then the broker, following the experience that it has with the supplier fulfilling requests, adapts its beliefs about the competence of that supplier. Furthermore, the broker uses simple episodic information on which to base any adaptations of an agent's competence; it may be more effective to obtain explanations for a failure from the supplier. Here, we extend the model discussed in section 3 to allow the supplier to have some input into how the broker deals with a possible mismatch between its model of the supplier's competence and its behaviour.

The broker records the reputation of suppliers, and under certain conditions (detailed below) it will seek the agreement of a supplier to revise the constraints on the services/task that the supplier has advertised. The broker does not use refinement techniques to directly modify its model of an agent's competence, but to create a revised competence description that is suggested to the supplier. It does, however, record the performance of the agents concerned on the basis of: (i) the number of times that the service type has been used by the agent (N); and (ii) the number of times that the agent has failed that particular service type (F). This provides the broker with an evaluation of the agent's performance $\mathrm{R} = \frac{\mathrm{F}}{\mathrm{N}}$.

The broker agent uses R for two purposes. Firstly, to select an appropriate agent (the agent deemed to be the most appropriate at the present time) for the service based on the service records of all agents. Secondly, it uses R to trigger an attempt to gain the agreement of the supplier for it to revise the model of that supplier's competence. The broker operates as follows. When the broker decides that a task is to be performed, it obtains a list of all competencies that meet the requirements of the task within some $\delta$ as before: $C_\delta$. The broker then selects a competence (and hence an agent to be delegated the task) on the basis of the following rules.[1] If the agent has never been used ($\mathrm{N} = 0$), then use that agent (or select a random agent from all those that have not been used). This ensures that all agents in the system will be given the opportunity to carry out a service that they have advertised competence for. If there are no such agents, select the agent/competence pair with the lowest failures/attempts ratio, R. This ensures that the broker chooses the most suitable, and arguably, the most reliable agent. If there is more than one agent/competence pairs with a minimum R value, then the system will select the agent/competence pair from this set with the lowest number of invocations (minimum N), and then pick one at random in the case of a tie. This ensures that the system remains fair and doesn't favour one particular agent.

Following the execution of each task, the broker, as before, checks the output, and proceeds in the following manner. First, $\mathrm{N} = \mathrm{N} + 1$ for the selected agent/competence pair. If the agent succeeds and its competence, $c$, is in the set of competencies $C_\delta \setminus C_0$ — i.e., the supplier succeeded in satisfying a constraint that is outside its advertised competence — then the broker calculates a proposed generalisation and asks the supplier whether or not it agrees to this modification. If the supplier agrees, the broker refines competence $c$ accordingly. If the supplier disagrees no action is taken. If the agent fails, then the broker will attempt to find an alternative agent to perform the task, and if $c \in C_0$ — i.e., the supplier failed to satisfy its advertised competence — then $\mathrm{F} = \mathrm{F} + 1$ for that agent/competence pair.

On completion of the transaction, the broker performs the following checks for each agent/competence pair that were involved in the transaction. If the new ratio, R, for that agent/competence pair is above some threshold, $\tau$, the broker calculates a proposed specialisation and asks the supplier whether or not it agrees to this modification. If the supplier agrees, the broker refines the competence as specified, and $\mathrm{F} = \mathrm{N} = 0$ for this agent/competence pair.

The main advantages of this system are that the competence model held by the broker is transparent to the supplier agent concerned, and the selection

[1] There are many ways in which an agent may be selected given a set of candidate agents. A detailed discussion and analysis of different policies is, however, outside the scope of this paper. Here we present one reasonable strategy that takes into account the information available to the broker.

| | | |
|---|---|---|
| 1 | SuspensionBroker: | Received REQUEST from da0, content is (road,normal,1200) |
| 2 | SuspensionBroker: | Calculation of Values (Resistance=3,Type=nonprogressive,Size=2) |
| 3 | SuspensionBroker: | Agents that can deliver spring: SpringAgent1 or SpringAgent |
| 4 | SuspensionBroker: | Agents that can deliver shock absorber are ShockAgent |
| 5 | SuspensionBroker: | Send INFORM to SpringAgent1, content is (nonprogressive, 2) |
| 6 | SuspensionBroker: | Send INFORM to ShockAgent, content is (3) |
| 7 | SpringAgent1: | Broker INFORM received, content is (nonprogressive,2) |
| 8 | ShockAgent: | Broker INFORM received, content is (3) |
| 9 | ShockAgent: | Processing Order; Sending CONFIRM to broker; shocks ready |
| 10 | SpringAgent1: | Sending FAILURE To Broker; can't supply size 2 springs |
| 11 | SuspensionBroker: | CONFIRM received from ShockAgent; shocks ready |
| 12 | SuspensionBroker: | Update ShockAgent success ratio |
| 13 | SuspensionBroker: | FAILURE received from SpringAgent1 |
| 14 | SuspensionBroker: | Update SpringAgent1 success ratio |
| 15 | SuspensionBroker: | Attempt to find replacement; Send INFORM to SpringAgent, content is (nonprogressive,2) |
| 16 | SpringAgent: | Broker INFORM received, content is (nonprogressive,2) |
| 17 | SpringAgent: | Processing Order; Sending CONFIRM to broker; springs ready |
| 18 | SuspensionBroker: | CONFIRM received from SpringAgent; springs ready |
| 19 | SuspensionBroker: | Update SpringAgent success ratio |
| 20 | SuspensionBroker: | Transaction complete; INFORM sent to da0 |
| 21 | SuspensionBroker: | Send INFORM to SpringAgent1, content is (resubmit) |
| 22 | SpringAgent1: | Received INFORM, content is (resubmit) |
| 23 | SpringAgent1: | Send AGREE, content is (resubmit) |
| 24 | SuspensionBroker: | AGREE received from SpringAgent1 |
| 25 | SuspensionBroker: | Update SpringAgent1 competencies and N = F = 0 |

Figure 2: Extracts from a simulation of the refinement-by-agreement broker.

strategy outlined above provides a fair distribution of tasks between registered agents. We now illustrate the operation of this broker using the second scenario introduced in section 2.1; extracts from the simulation are presented in figure 2.

Agent da0 initiates the procedure by submitting a request to the broker. As shown in figure 2, the broker has the choice of selecting two spring agents (line 3) both claiming to be capable of performing the task. The broker selects SpringAgent1 because both agents have the same R at this time, and N for this agent/competence pair is less than N for the other option. Unfortunately, SpringAgent1 is unable to complete the transaction (line 10) and so indicates this to the broker who then updates its success ratio (line 14). The broker then attempts to complete the transaction using the alternative spring agent (SpringAgent). This time the transaction can complete and confirmation is sent to the purchasing agent (line 20). Once the transaction has terminated the broker checks the success ratios of all agents and offers a chance to resubmit competencies to any agent that falls below a predetermined threshold value. As SpringAgent1 now falls below this value it is asked to resubmit (line 21). SpringAgent1 is programmed to agree to a resubmit request and so resubmits more realistic competencies (line 23). Its success ratio is reset giving the agent a clean slate albeit with lesser competency values. Note that agents may choose not to resubmit but then run the risk of holding on to a poor success ratio that may prevent them from being chosen for future tasks.

## 5 Discussion and Future Research

There are many issues that are worthy of discussion here including the need to investigate relaxing some of the assumptions discussed in section 2. However, here we focus on a couple of issues to do with the models presented in sections 3 and 4, because these lead to interesting areas for future research.

The first issue is concerned with how the reputation of an agent is determined. Here, we have assumed that reputation relates to a specific competence of an agent rather than the agent itself. Rather, the reputation of an agent, as perceived by the broker, should be determined by all its encounters with that agent. Thus, to produce an estimate of the reliability of an agent, the broker must take into account a number of episodic encounters with it over time. It is also important to consider issues of deception: the agent may use a strategy of revising its competency adverts in an attempt to hide past defections. This problem may occur in any reputation monitoring system. The sequence of encounters, defections, competency revisions, etc., does, however, provide a rich series of data through which the broker may assess the reliability of an agent. This assessment of reputation is, essentially, a summarisation problem, and it may, therefore, be worthwhile investigating summarisation techniques [3] in producing a solution.

In section 4 the broker gives a service provide agent a choice if it fails to satisfy an advertised competence: it must either agree to a revision of its competence or this failure will be treated as a defection. This is a reasonable stance for the broker to take, but the suggested revision of the competence description originates from the broker. The service provider may have further explanation regarding the cause of the failure that may inform the revision. A further extension to this model may, therefore, be for the recognition of a mismatch between the competence model and the perceived behaviour of the supplier to lead to a more complex dialogue between these two agents focused on determining the reason for failure: an inquiry.

## 6 Conclusion

In this paper two models of adaptive brokering have been presented: first, a broker that refines its model of agents' competencies on the basis of experience; and second, an extension to this model is explored with a view to showing how these techniques complement existing models of trust and reputation. There are two important advantages to incorporating these refinement techniques in agent-mediated electronic commerce. First, the recognition of a mismatch between and advertised competence and perceived behaviour and refining the model of this agent's competence may lead to fewer inappropriate allocations of tasks in future situations. Second, the investigation triggered by such a mismatch may lead to more accurate assessments of the reliability and hence the reputation of agents within multi-agent systems.

## 7 Acknowledgements

The initial implementation and evaluation of the adaptive brokering models discussed in this paper was performed by Nial Chapman while he was supported by a MTP EPSRC grant pursuing an MSc in Electronic Commerce Technology [2] at the University of Aberdeen: http://www.csd.abdn.ac.uk/ecommerce.


## References


[1] C. Castelfranchi, and R. Falcone, Principles of trust for MAS: Cognitive anatomy, social importance, and quantification, **Proceedings of the Third International Conference on Multi-Agent Systems**, pp. 72–79, 1998.

[2] N. Chapman, **Refining Models of Agent Competencies**, MSc Thesis, University of Aberdeen, 2002.

[3] S. G. Sripada, E. Reiter, J. Hunter, J. Yu and I. Davy, Modelling the task of summarising time series data using KA techniques, in A. Macintosh, M. Moulton and A. Preece (eds), **Applications and Innovations in Intelligent Systems IX**, Springer, pp. 183-196, 2001.

[4] T. M. Mitchell. **Machine Learning**, McGraw-Hill, 1997.

[5] P. Resnick, R. Zeckhauser, E. Friedman and K. Kuwabara, Reputation systems: Facilitating trust in Internet interactions, **Communications of the ACM**, 43(12):45–48, 2000.

[6] J. Sabater and C. Sierra, Reputation and social network analysis in multi-agent systems, In **Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems**, pp. 475–482, 2002.

[7] M. Schillo, P. Funk and M. Rovatsos, Using trust for detecting deceitful agents in artificial societies, **International Journal of Applied Artificial Intelligence**, 14(8):825–848, 2000.

[8] M. Winter and D. Sleeman, REFINER+: An efficient system for detecting and removing inconsistencies in example sets, In M. A. Bramer, J. L. Nealon and R. Milne (eds.) **Research & Development in Expert Systems XII**, Information Press Ltd, pp. 115-132, 1995.

[9] B. Yu and M. P. Singh, Distributed reputation management for electronic commerce, **Computational Intelligence**, 18(4):535–549, 2002.