

# Analysing Partner Selection Through Exchange Values

Maira Ribeiro Rodrigues and Michael Luck

School of Electronics and Computer Science, University of Southampton, Southampton, UK  
{mrm03r, mml}@ecs.soton.ac.uk

**Abstract.** Dynamic and resource-constrained environments raise interesting issues for partnership formation and multi-agent systems. In a scenario in which agents interact with each other to exchange services, if computational resources are limited, agents cannot always accept a request, and may take time to find available partners to delegate their needed services. Several approaches are available to solve this problem, which we explore through an experimental evaluation in this paper. In particular, we provide a computational implementation of Piaget's exchange-values theory, and compare its performance against alternatives.

## 1 Introduction

In many disciplines, like medicine and biology, researchers are discovering the advantages of collaborative research, in which different types of information and tools are exchanged in order to improve individual or global results [1, 2]. In an ideal situation, in which computational resources are plenty, individuals can collaborate with each other by performing services or giving access to information to any suitable (authorised) requester. However, when there are limited computational resources in the system, the participants must choose which requests to accept. Such collaborative systems are also dynamic, in the sense that participants change both their needs and provided services over time due, for example, to changes in research interests, and to the incorporation or development of new tools and information.

Dynamic and resource-constrained environments that take a collaborative approach to interaction raise interesting issues for partnership formation. First, because computational resources are limited, participants cannot always accept a request and may take time to find available partners to delegate their needed services. As a consequence, participants need some criteria to guide their decisions over interactions; more precisely, they need to choose which requests to accept when providing a service, and where to send requests that have a higher chance of being accepted when delegating a service. Second, because participants may change their needed and provided services over time, maintaining a collaborative interaction becomes more complicated, since the link between clients and services is not permanent. Third, if we assume that reciprocity is the basic motivation for interactions in collaborative environments, since it gives expectations of future interactions, this relation needs to be taken into account in the decision process over partners.

Guidance on partner selection in this context involves primarily two things: first, some criteria for choosing those participants that are more likely to perform a needed service, to avoid losing time trying to find an available interaction partner and, second,

a strategy for accepting or refusing requests from other participants that favours the chances of future interactions (by considering the existence of relations of reciprocity and collaboration).

One way to select between interaction partners is to restrict the collaboration to situations in which a service dependency is observed. In this case, the concept of dependence [3, 4] can be used to motivate the exchange of services between participants and to influence their decisions about interactions. Although this dependence approach is powerful, we believe it is more suitable for situations in which the link between service provision and service request is static. Since dependence relations are built upon the services each individual needs and provides, if the relations change over time, the dependence network could lose accuracy, yet continual updates are likely to be expensive.

In response, we propose to address the problems of partnership formation and partner selection by using a system of exchange values, based on Piaget's theory [5]. Here, exchange values are viewed as the values that individuals associate with their interactions with other individuals, indicating the effort, cost, satisfaction, and benefit to each individual of the received or performed services and, thus, can influence the behaviour of interacting individuals. Piaget defines two main functions for exchange values in social interactions: as a means for individual decision-making, and as a regulation tool for guaranteeing the continuity of social interactions, since exchange values imply *moral commitments* made by individuals during their interactions (in the sense that they do not concern economic values, but instead what individuals owe to each other as a result of their interactions). Commitments are not used here as formal structures of mental representations of an obligation (as in [6]); rather, they are used to represent a moral obligation of one individual to return a favour to another, or to reciprocate an action received in the past.

The application of exchange values for supporting the exchange of services between agents was first proposed in [7], in which the social reasoning mechanism is based on Piaget's system. We believe this approach is suitable for addressing the problem of partner selection in dynamic and resource-constrained environments for two main reasons: first, exchange values provide a system of credits and debits that are seen as moral commitments between agents, which motivates and favours the chances of future interactions (for example, a credit that is gained by performing a service can be charged in the future) and, second, the set of exchange values of each individual (agent) is not related to the specific services they perform, but to the results of past interactions.

Other notions of values have appeared in the multi-agent systems literature. Miceli and Castelfranchi [8, 9], for instance, deal with the cognitive role of evaluations and values. They use values as a special kind of evaluation that mediates beliefs and goals in the agents' knowledge representation. Antunes and Coelho [10, 11] use a notion of "multiple values" to improve agents' decision-making process by explicitly including these values in their agent architecture, the BVG architecture. Although these approaches relate in one way or another to the concept of values, they neither encompass the idea of exchange, nor have the moral commitment implications (that suggest reciprocity and potentially regulation) of the notion of values in Piaget's approach.

In this paper, we investigate these approaches to interaction, through an implemented experimental testbed with a multi-agent simulation system. In particular, there

has not been an implementation of the system of exchange values, so the results are novel and valuable. The key contribution is thus the computational model of the exchange values approach, and the associated experimental analysis of the performance of this approach in comparison to alternatives.

The paper begins with an introduction to the different social approaches we consider, followed by a description of the scenario we use, as well as the details of the strategies used (dependence-based and value-based) in a computational context. We end by presenting the experimental simulations and results obtained.

## 2 Background

This section describes two theoretical approaches that can be used to address the problem of partnership formation: dependency theory and Piaget's system of exchange values, both as a basis for decision-making about interaction partners.

### 2.1 Dependence Theory

The concepts of *social dependence* and *social power* in multi-agent systems were first proposed in [3, 4], with the basic idea that one of the fundamental notions of social interaction is the *dependence relations* between agents. This represents the situation in which an individual needs a resource (such as an object, a task to be performed, a piece of information) that he does not possess or have access to, but which can be provided by another individual, giving the latter influence over the former. The basic definition of social dependence is that an agent  $x$  depends on an agent  $y$  regarding an action  $a$  needed for achieving a goal  $g$ , if  $x$  is not capable of performing  $a$  but  $y$  is. In this case, agent  $y$ 's action  $a$  is viewed by agent  $x$  as a resource for achieving  $g$ .

Associated with this is the notion of *power of influence* of an agent over another. Dependence relations can be associated with power of influence in the sense that when agent  $x$  depends on agent  $y$  for achieving a goal, it becomes susceptible to the influence of the latter agent. In this view, agents structure their interactions to take advantage of situations in which they can exert influence to ensure the compliance of agents with their requests. Consequently, the formalisation of these relations allows agents to achieve better interactions since an agent is able to control its interactions according to its own interests, by obtaining power through dependence relations.

In this way, some models for cooperation based on dependence relations have been proposed [12, 13], in which agents choose partners by identifying dependencies with other agents in a society. Also, work has been done to integrate the notion of dependence and norms with BDI-like agent architectures [14], and to describe social dependence networks as part of a formal framework [15].

### 2.2 Piaget's Theory of Exchange Values

Piaget's theory of exchange values studies and formalises the dynamics of values in such a way that they can form an exchange system [5], specifically in the context of the exchange of *services* between individuals, i.e., actions that an individual performs *on behalf* of another individual. The values that arise from such exchanges,

which result from the individual's evaluation of the exchanged service, are called *exchange values*. They are seen as *moral values*, concerning *moral debits* (the obligation to perform new services in return for services previously received) and *moral credits* (the right to demand the performance of new services in return for services previously given).

Piaget's theory assumes two conditions for the existence of an exchange value system: the individuals involved in an interaction must share a *common scale of values* to ensure the compatibility of their evaluations of performed and received actions (services); and there must be conservation of the exchange values in time, so credits and debits acquired in the past continue to be valid in the future. If these two conditions hold, a system of exchange values can be seen as a mechanism for regulating (or coordinating) the social exchange of services between agents, guaranteeing their continuity (and thus the continuity of the society).

Four types of exchange values are involved in the exchanges between two individuals:

- a *renouncement value*, representing the effort, or *investment* of a provider in performing a service on behalf of another individual;
- a *satisfaction value*, representing the satisfaction of a requester with a received service;
- an *acknowledgement value*, representing the acknowledgement by the requester of a moral obligation with the provider, which is considered a debit; and
- a *reward value*, representing the recognition the provider thinks it received from the receiver for its work done, which is considered a credit.

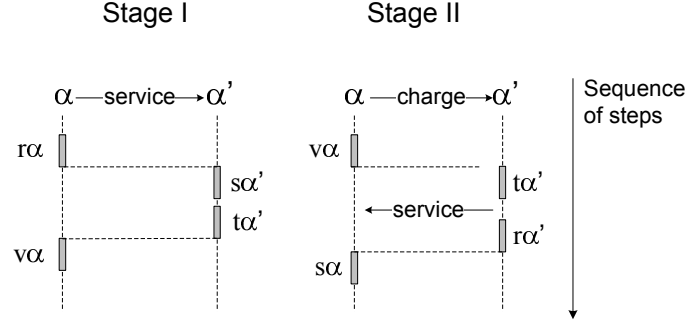
These values are determined by the individuals over time, as a result of their exchanges with others. The renouncement and satisfaction values are viewed as real values in the sense that they are explicitly expressed by individuals. The acknowledgement and reward values are viewed as virtual values, since they are an internalisation of the real values, and provide the notion of acquired debits and credits. (In the remainder of this paper we use debit value instead of acknowledgement value, and credit value instead of reward value.)

Exchanges between two individuals  $\alpha$  and  $\alpha'$  occur in two stages, whose basic forms are as follows: in *Stage I*, an individual, say  $\alpha$ , performs an action on behalf of another, say  $\alpha'$ , acquiring some credit for that action; and, in *Stage II*,  $\alpha$  charges his credit, asking  $\alpha'$  to perform some action for him in return. The event sequence for Stages I and II of a basic social exchange between  $\alpha$  and  $\alpha'$  is shown in Figure 1.

Stage I consists of four steps, as shown on the left of Figure 1:

1.  $\alpha$  performs a service on behalf of  $\alpha'$  and associates with this action a renouncement value ( $r_\alpha$ );
2.  $\alpha'$  receives the performed service, and associates a satisfaction value ( $s_{\alpha'}$ ) with the received service;
3.  $\alpha'$  then acknowledges a debit with  $\alpha$  through a debit value ( $t_{\alpha'}$ );
4.  $\alpha$  feels valued with the recognition of  $\alpha'$ , and associates to this valuation a credit value ( $v_\alpha$ ).

At the end of Stage I,  $\alpha'$  has acquired a debit ( $t_{\alpha'}$ ) with  $\alpha$ , and  $\alpha$  has acquired a credit ( $v_\alpha$ ) with  $\alpha'$ . Note that since these are from the individual perspective of each, they may



**Fig. 1.** The two stages of exchange

not be equal, and may not be known by the other. Later,  $\alpha$  may charge the credit with  $\alpha'$  by requesting performance of some service that benefits  $\alpha$  in return. This begins Stage II of the exchange process, which follows the steps shown on the right of Figure 1, and is explained below:

1.  $\alpha$  requests a service from  $\alpha'$  in exchange for its credit  $v_\alpha$  acquired in Stage I;
2.  $\alpha'$  acknowledges its debit  $t_{\alpha'}$  with  $\alpha$ ;
3.  $\alpha'$  performs the requested service on behalf of  $\alpha$ , and associates with this a renouncement value  $r_{\alpha'}$ ;
4.  $\alpha$  receives the service, and associates a satisfaction value  $s_\alpha$  with this service.

When Stage II finishes, the exchange is said to be complete. In this way, Stage I represents an *accumulation* of exchange values, and Stage II represents their *realisation*, since debits and credits are first generated and then used in the exchange of concrete actions. If the amount assigned to each exchange value in one stage of the exchange is the same, i.e., if they directly correspond, the system is said to be in *equilibrium* with respect to that stage. The equilibrium situation for Stage I can be represented by the following logical implication:

$$\text{Implication} \quad (r_\alpha = s_{\alpha'}) \wedge (s_{\alpha'} = t_{\alpha'}) \wedge (t_{\alpha'} = v_\alpha) \Rightarrow (v_\alpha = r_\alpha) \quad (1)$$

From this, we conclude that if the amounts assigned to the exchange values resulting from the evaluations of the performed or received service are equivalent, then  $\alpha$  is valued by  $\alpha'$  proportionally to the service that was provided. The equilibrium situation for Stage II is represented in a similar way.

It is important to notice that all four values are related, because they all originate from the evaluation of the exchanged service. However, because this evaluation is subjective, and consequently individuals use different, personal criteria in the evaluation (for example, according to their character, personality, personal experience, etc.), these values need not correspond. In summary, perfect or fair judgement results in a direct correspondence between the exchange values and, consequently, in exchanges in equilibrium, while imperfect or unfair judgement results in a divergence of the exchange values and, consequently, there is no equilibrium.

### 3 Scenario

In an effort to experiment with these models, we have adopted the context of a dynamic collaborative research environment (as in Bioinformatics), in which researchers exchange information and tools in order to complete experiments or validate hypotheses. The environment is dynamic because participants are likely to change their needs and capacities over time due, for example, to changes in research interests, the appearance of new theories, or development of new tools. In this scenario, we assume participants are motivated to collaborate based on their expectations of reciprocated services. Also, their computational resources are limited, as every participant has a maximum number of services they can perform at one time. Therefore, when capacity is full, a participant cannot accept more requests to perform a service.

As a consequence, when a participant needs a service from others, she must choose those that are likely to perform the service and, when a participant receives requests from others, she must adopt a *criterion* for accepting or refusing those requests in a way that does not jeopardise her possibilities of future interactions. In this scenario, we assume that each researcher is represented by an agent that takes decisions about partners on her behalf.

The scenario thus consists of a group (or society) of agents, each of which has a set of services they need to execute, and a set of services they can provide (i.e., the agents are service clients and providers at the same time); agents exchange services in order to achieve individual goals. Since participants in a collaborative research environment are likely to change their needs over time, the agents that model them also change their set of needed services at a regular rate.

In addition, we assume that agents are not capable of performing their needed services (since we want to evaluate service exchange), but they always find interaction partners. High level relations between agents like goals or organisations are not considered, just relations between provided and needed services, so relationships like mutual or reciprocal dependence are omitted.

#### 3.1 Strategies

A decision-making process towards the selection of partners, and the continuity of interactions, must establish not only the process of *choosing a partner*, but also the process of *analysing the received requests*, since the acceptance or refusal of a request will influence the chance of future interactions when reciprocal relations are considered. Therefore, an agent's decision over partners has two different contexts: when the agent receives a request and must decide whether to accept it; and when the agent must choose agents to send a service request, giving preference to those more likely to accept the request.

A common way to select between interaction partners is to restrict the collaboration to situations in which dependencies are observed. For example, an agent A is likely to collaborate with agent B by performing a service on its behalf, only if B is capable of performing a service that A needs. In the same way, if B needs to request a service from others, it will give preference to those with which it identifies a dependence (or that are seen as more likely to collaborate with it). In this case, the *criterion* used in the selection process is the *dependence*. The principle here is that the relationship between

the services an agent needs and those services that others can provide determines if it is likely to be influenced by others, and can be used in the decision process. We call this approach the *dependence-based approach* for decision-making regarding interaction partners.

Another way of selecting between partners is to use exchange values as a basis for decision-making. This uses the basic exchange values system, according to which the agent's decision is not based on the services each agent can perform, but on the results of past interactions and expectations of future interactions. It implements a system of credits and debits, derived from the performed and received services, and is used to choose those agents that are more likely to perform tasks, as well as to choose whether to collaborate with other agents. In this case, the *criterion* used is the set of *exchange values*. The principle here is that the moral debits and credits acquired by an agent as a result of its interactions with other agents determine if it is likely to be influenced by others. We call this approach the *value-based approach* for decision-making. In what follows, we refer to agents using the former strategy, employing dependence-based reasoning, as *DAgents*, and the latter value-based agents as *VAgents*.

To analyse requests for a service, an agent considers: information on past interactions and on services provided and by whom, current capacity for service provision, and preferences regarding interactions. To analyse possible partners to send service requests, an agent considers information on past interactions, services provided, and preferences regarding interactions.

Information on past interactions and on services is used according to each approach. For example, the dependence-based approach uses information about service provision to infer the dependence between agents, while the value-based approach uses information on past interactions related to the history and state of exchange values.

In the context of this paper, we assume the requests are analysed as soon as they arrive, and that when the agent's current capacity is less than half full it will always accept requests. However, when the current capacity is equal to, or more than, half the maximum, the agent's decides whether to accept the request based on a set of preferences that are different for each approach. For example, the dependence-based approach has preferences related to dependence relations, while the value-based approach has preferences related to the state and history of exchange values.

In this scenario, agents request needed services according to the initial configuration, one after another, and must deal with any request messages that arrive. To decide to which agent a request is sent, and from which agents a request should be accepted, the agents take into account the limitations imposed by the environment, and also the criteria defined by each approach for partner selection, dependence-based or value-based.

### 3.2 Dependence-Based Agents

DAgents use information on dependencies to choose between interaction partners based on the assumption that partners with which a dependence is identified are considered to be more likely to accept requests. Conversely, requesters in a dependence relation can influence the decision when analysing incoming service requests. In order to derive information on dependencies, DAgents keep a service log with a record of the agents for which they performed a service.

**Table 1.** Dependence-based algorithm for sending service requests

---

```

01 For each  $sn_i$  in  $S_n$  do
02   search the resulting  $L_{pp}$  for  $sn_i$ ;
03   order  $L_{pp}$  giving priority to agents in  $L_{DepOn}$ ;
04   send request for first agent in  $L_{pp}$ ;
05   if request refused then
06     repeat
07       send request to next agent in  $L_{pp}$  ;
08     until request accepted OR  $L_{pp} = \phi$ 
09   if  $L_{pp} = \phi$  AND request not accepted then
10     go to line 4 and repeat the process.

```

---

**Table 2.** Dependence-based algorithm for analysing incoming requests

---

```

01 For each  $sr_i$  do
02   if ( $current\_capacity < maximum\_capacity/2$ ) then
03     accept request;
04   if ( $current\_capacity = maximum\_capacity$ ) then
05     refuse request;
06   if ( $current\_capacity \geq maximum\_capacity/2$ ) then
07     if requesting agent is in  $L_{IsDep}$  then
08       accept request;
09     else
10       refuse request.

```

---

Formally, if we take  $\alpha'$  to be a dependence-based agent,  $L_{pp}$  the set of possible partners for  $\alpha'$  when requesting a service  $sn_i$  from the needed services set  $S_n$ ,  $sr_i$  a service being requested from  $\alpha'$ ,  $L_{DepOn}$  the agents that depend on  $\alpha'$  to perform a service, and  $L_{IsDep}$  the set of agents on which  $\alpha'$  is dependent, we can define two algorithms to represent  $\alpha'$ 's decision-making. The algorithm in Table 1 describes the decision process for sending requests for services, and the algorithm in Table 2 describes the decision process for analysing incoming requests.

In this approach, preference is given to agents with which a dependence relation is observed, since they are more likely to accept a request. In order to determine the current dependence relation, DAgents search the service log for agents for which they have previously performed a service (so that these agents depend on them for that service). When resources are limited (in that the current capacity is almost full), requests cannot always be accepted, and we assume that DAgents prefer to collaborate only with agents they depend on for performing a service, as shown in Table 2.

### 3.3 Value-Based Agents

Agents using the exchange values approach store four types of values [7]: renouncement (r), satisfaction (s), debit (t) and credit (v). For each interaction in which a Vagent  $\alpha$  participates, it stores a current set of exchange values  $V_{\alpha\alpha'}$ , which contains the accumulated exchange values associated with the other agent involved ( $\alpha'$ ). VAgents also maintain an individual *history of interactions* in which they participate, with the



resulting exchange values, and *exchange values state*, which is the set of exchange values (r,s,t,v) accumulated through interactions with other agents. After each interaction, VAgents calculate their exchange values based on the evaluation of the provided or received service, and update both their history of interactions and their exchange values state with the current set of exchange values.

Gains and losses determine how exchange values are accumulated; that is, if each accumulated exchange value should decrease or increase. For example, when an agent  $\alpha$  acquires a debit with another agent  $\alpha'$ , this should increase its accumulated debit with  $\alpha'$ . Similarly, when  $\alpha$  pays its debit, this should decrease its accumulated debit with  $\alpha'$ .

For VAgents, we take a simplified value set for exchange values, namely integers (though other models are possible), and define each increase or decrease during a single step of an exchange process to be by an integral number of units,  $+n$  or  $-n$ . We also simplify the way exchange values are calculated, by assuming that the values are always in equilibrium.

As before, if we take  $\alpha$  to be a VAgent,  $L_{pp}$  to be the set of possible partners for  $\alpha$  when requesting a service  $sn_i$  from the needed services set  $S_n$ ,  $sr_i$  a service being requested from  $\alpha$ ,  $L_{HasCred}$  the set of partner agents with which  $\alpha$  has credits, and

**Table 3.** Value-based algorithm for sending requests for services

---

```

01 For each  $sn_i$  in  $S_n$  do
02   search the resulting  $L_{pp}$  for  $sn_i$ ;
03   order  $L_{pp}$  giving priority to agents in  $L_{HasCred}$ ;
04   send request for first agent in  $L_{pp}$ ;
05   if request refused then
06     repeat
07       if refusing agent is in  $L_{HasCred}$  then
08         devalue refusing agent;
09       send request to next agent in  $L_{pp}$  ;
10     until request accepted OR  $L_{pp} = \phi$ 
11     if  $L_{pp} = \phi$  AND request not accepted then
12       go to line 4 and repeat the process.

```

---

**Table 4.** Value-based algorithm for analysing incoming requests

---

```

01 For each  $sr_i$  do
02   if ( $current\_capacity < maximum\_capacity/2$ ) then
03     accept request;
04   if ( $current\_capacity = maximum\_capacity$ ) then
05     refuse request;
06   if ( $current\_capacity \geq maximum\_capacity/2$ ) then
07     if sender is charging a credit then
08       accept request;
09     else if sender has negative exchange history then
10       refuse request;
11     else
12       accept request.

```

---

$L_{HasDeb}$  the set of agents with which  $\alpha$  has debits, we can specify two algorithms to represent the decision-making of  $\alpha$ . In Table 3, we show how to send service requests and in Table 4, we show how to analyse incoming requests.

In this approach, preference is given to agents with which an agent has credits, since they are more likely to accept a request because they have a *commitment* with the requester, and are motivated to collaborate and pay their debits (a service was received, and a service is provided in return). Since the algorithm was defined in such a way that an agent first selects the possible partners for a needed service (that is, those that have the capability to provide it), and then uses the information about debits to order them, the situation in which an agent is not able to pay a debit because it is not able to perform the requested service is not considered.

If a request charging a credit is refused, it indicates that the other agent did not keep its commitment, so it is devalued by the requester (the refusing agent loses credits with the requesting agent). This indicates that the refusing agent is a poor partner for collaborative interactions. When resources are limited and requests cannot always be accepted, we assume that a VAgent prefers to collaborate with agents with which it has debits, since they will try to keep their commitments by paying their debits and completing the exchange, or with agents with a good collaboration history (and do not have a negative exchange result in the history of exchanges). In summary, collaboration is modelled by seeking complete exchanges, where services are received and performed in return.

## 4 Experiments

In seeking to determine which approach performs better in the changing environment, we have implemented an experimental testbed and undertaken some experiments to compare the success of each approach in exploring collaborative situations and finding interaction partners in dynamic, resource-constrained environments.

To do this, we measure the number of request messages that an agent must send until the service request is accepted (the *effort*), and the number of services an agent can delegate (the *task completion* or accomplishment). It is expected that agents with good performance will have a low value for effort (indicating that they were successful in choosing partners more likely to perform a service), and a high value for task completion (indicating that they were able to delegate more services). In summary, it is desirable that agents find partners quickly (and hence reduce the number of messages in the system), but also that the reasoning process does not consume too much time and consequently reduce the number of delegated services. Since we have restricted the scenario, assuming that there is always a provider for every available service, we guarantee that the task completion measure only represents tasks that could not be completed because partners did not accept them.

### 4.1 Simulation Configuration

A simulation in our experiments consists of a number of agents requesting services from, and performing services for, others at the same time, for a predetermined duration. For the agents' service configuration, each agent has a set of services it needs to request from other agents, and a set of services it can provide. The number of needed

**Table 5.** An example of service configuration

<i>Agent</i>	<b>Needs</b>	<b>Provides</b>
<i>Vagent1</i>	$S1, S2, S2, S1$	$S4, S3$
<i>Vagent2</i>	$S3, S4, S5, S3$	$S1, S2$
<i>Dagent3</i>	$S1, S3, S5, S4$	$S2, S6$
...		

and provided services is the same for all agents, but there is no relationship between the number of needed services and the number of provided services. In addition, all available services are allocated to at least one provider, and no agent can provide and request the same service. This kind of *service configuration* is shown in Table 5 where, for example, *Vagent1* needs services  $S1, S2, S2, S1$  in order, and can provide  $S4$  and  $S3$  (in any order, and possibly concurrently). The service configuration is changed by randomly replacing half the services in the set of needed services of all agents in the system. Each simulation has a basic simulation configuration including, for example, the duration of the simulation, total number of agents, total number of available services, maximum capacity for providing services at the same time, frequency of changes in service configuration, etc. We vary this basic configuration in different experiments in order to analyse the agents' behaviour under different conditions.

Also, to compare agent behaviour in uniform environments and heterogeneous environments, we defined some simulations to run with only one type of agent (single), and others with both types of agents (mixed). As a baseline, we also used a generic agent (GAgent) with random partner selection.

## 4.2 Results

First, we varied the number of changes in the service configuration, to see if there is a difference in the performance of each approach when there is no change at all in the environment, and when the changes are more frequent. For this experiment, we defined simulations, each with single groups of 30 VAgents, DAgents and GAgents.

The results, in Figures 2 and 3, show that VAgents generally performed better, in both effort and task completion, and also had stable behaviour in relation to the number of requests per service (effort) and delegated services (task completion), despite the changes. DAgents had an increase of effort and a decrease in the number of tasks completed in response to the increase in the number of changes in service configuration.

In the second experiment, we varied the number of agents in the system, to observe the behaviour of the agents when we scale up the system. Here, we also simulated single groups of VAgents, DAgents and GAgents and used a frequency of 25 changes per simulation.

The results in Figures 4 and 5 again show that VAgents perform better with a convergence to stable behaviour in terms of effort (in requests) but with a decrease in task completion as the number of agents increases. However, we observe that all approaches showed a decrease in their number of completed tasks in response to the increase in the number of agents. Interestingly, for this experiment we also measured the number of requests sent by the agents, to investigate if the decrease in task completion was due to an

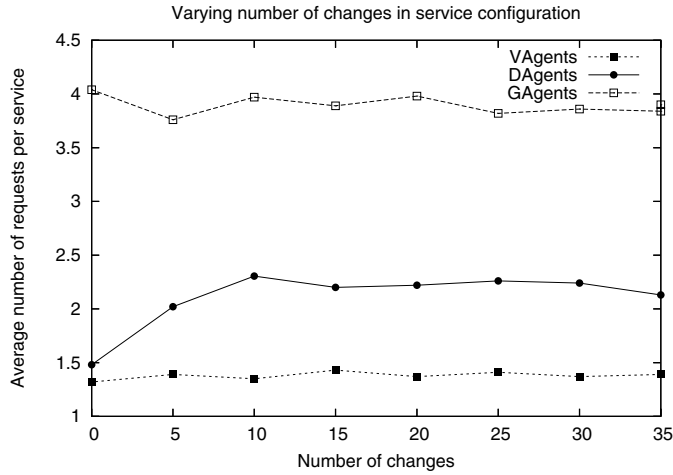


Fig. 2. Results for effort when varying frequency of changes in service configuration

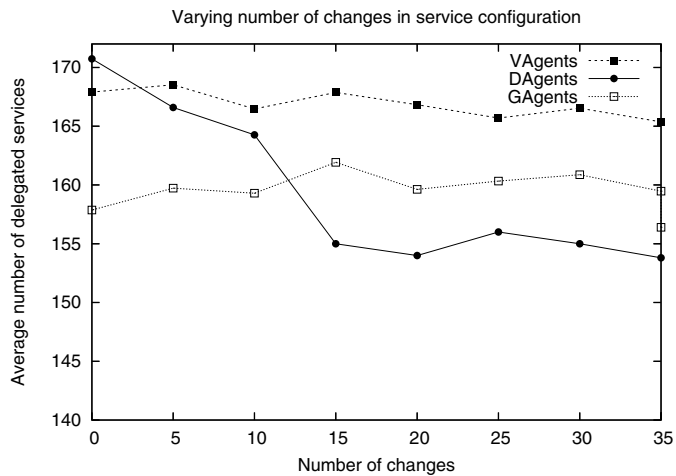


Fig. 3. Results for task completion when varying frequency of changes in service configuration

increase in the number of requests that could be responsible for making agents busier. We found that for GAgents, the number of request messages sent increased, but for the DAgents and VAgents the number decreased. This indicates that the latter spend more time in reasoning than in sending messages but, since the difference between task completion of VAgents and GAgents was relatively close, and the effort for VAgents was much better, we can conclude that as the number of agents increases the lower effort of VAgents compensates for the reasoning time. In comparison with DAgents, VAgents also performed better, stabilising at lower effort level and diverging from DAgents with better task completion as the number of agents increase.

Finally, we mixed the three types of agents in the same simulation, using the same number of agents of each type, keeping the parameters fixed to observe if their effort

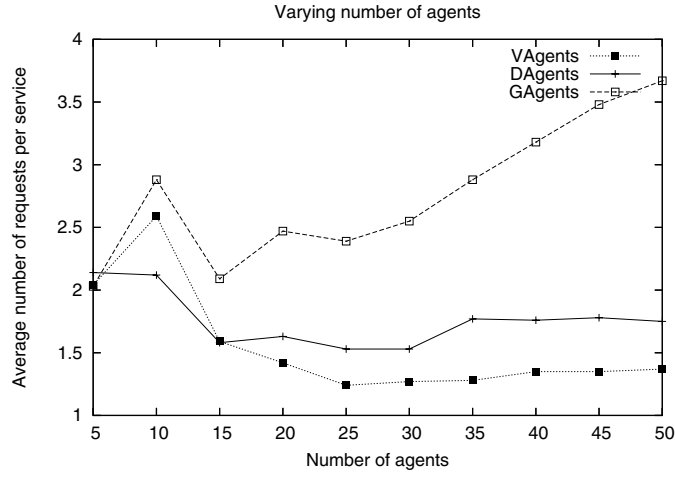


Fig. 4. Results for effort when varying the number of agents in the system

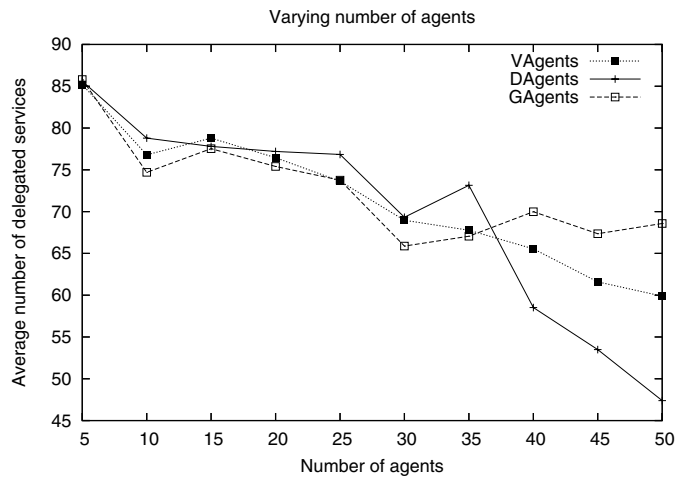
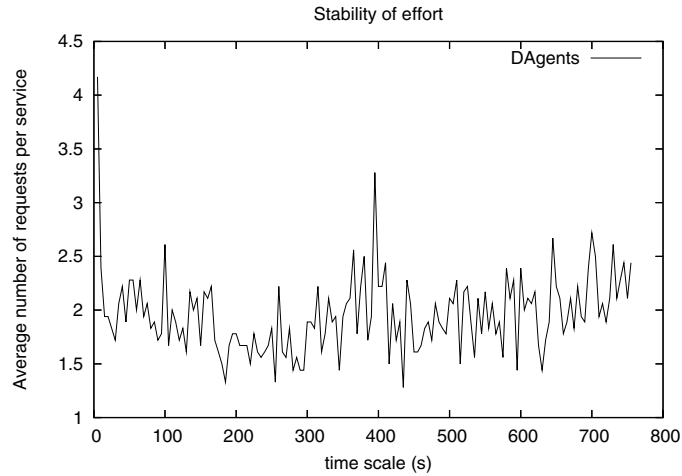
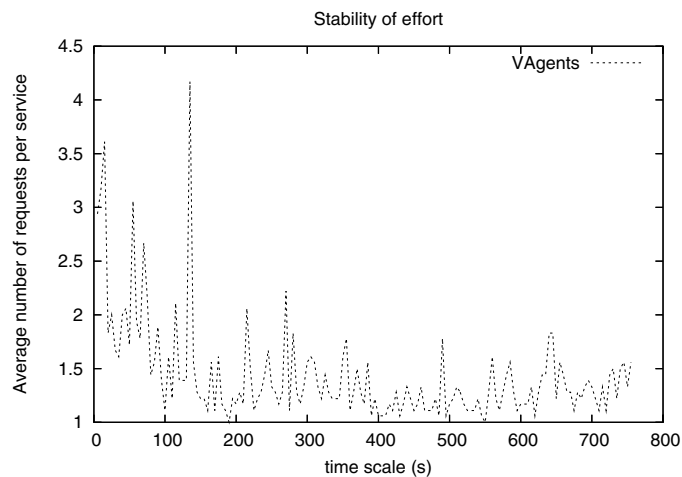


Fig. 5. Results for task completion when varying the number of agents in the system

stabilises over time, indicating that they can adapt to the heterogeneous environment. In this experiment, we mixed VAgents, DAgents and GAgents, 16 of each type, and used a frequency of 25 changes per simulation. The results for DAgents and VAgents in Figure 6 and 7 show that although DAgents do not have very high values for effort, they cannot achieve stable behaviour in effort. By contrast, VAgents have slightly higher values for effort at the start of the simulation, but achieved fairly stable behaviour towards the end, with very small variations in the number of requests per service.



**Fig. 6.** DAgents stability of effort during the simulation



**Fig. 7.** VAgents stability of effort during the simulation

### 4.3 Discussion

We investigated three aspects of the partner selection approaches: the capacity to cope with changes in service configuration (in the first experiment), the capacity to cope with system scalability (in the second experiment), and the capacity to adapt to heterogeneous environments with agents using different strategies (in the third experiment). We also investigated the difference between the effort in finding interaction partners of agents using a strategy for partner selection as opposed to those not using one (by using random selection).

The results in the previous section show that in most cases VAgents perform better in the changing environment. However, we note that initial configurations can distort this as, for example, shown on Figure 3.

In the results for the second experiment (in Figures 4 and 5), the peaks we observe in the graphs are due to the distribution of services. Since each new agent added to the system has a different set of provided and needed services, a biased combination can either cause an agent to be busy if the new agents request services that were already scarce in the environment, or cause a more equal distribution of provided services if the new agents provide services that were scarce in the previous environment. Even though different service distributions could give different graphs, the results are not influenced, since the same configuration is used for all types of agents, and it is the relative comparison between them that identifies which approach performs better.

In summary, the results show that each approach has different advantages, and can therefore complement each other. For example, one possibility of combining both approaches is changing the algorithm for sending requests in a way that it first selects the agents with dependencies as possible partners, and then orders this set according to the exchange values approach, or vice-versa. The dependence approach could also be used in cases in which an agent has no credits or debits with the possible partners (as a result of not having encountered them previously).

## 5 Conclusions and Future Work

In this paper, we have described an experimental testbed with a multi-agent simulation system to simulate the process of interaction partner selection in dynamic and resource-constrained environments. To model agent decision-making in relation to interactions and interaction partners, we used two different approaches: a dependence-based approach and an exchange values-based approach. These address the issue of finding a partner more likely to accept requests, in contrast to alternative approaches that seek the partner with the highest utility (as proposed in [16, 17]).

We described the experiments undertaken to test the behaviour of agents using both approaches in the task of finding interaction partners, and specifically to analyse the behaviour of agents using exchange values in their reasoning, since there is no previous empirical evidence of this approach. The results showed that agents using exchange values-based decision-making were more stable in their behaviour, and needed less effort to find interaction partners than agents using a dependence-based approach, despite the changes in the environment. Also, by implementing a generic agent type that chooses interaction partners randomly, and comparing it to the other two types of agents, we observe that a strategy aimed at choosing interaction partners can significantly reduce the effort to find interaction partners, which can be very important when dealing with systems with resource limitations and dynamic behaviour.

Apart from these advantages, we believe that applying the system of exchange values in multi-agent systems can help more generally in regulating interactions, and motivating continuity of interactions, since exchange values imply moral and legal commitments, made by the agents during their interactions. Future work aims to analyse the combination of both approaches, dependence-based and exchange values-based (as proposed in [7, 18]) for the purpose of partnership formation. We also aim to extend agent reasoning and partner selection to include the notion of evaluation of provided and

received services, so that the best interaction partner for delegating a service in terms of performance satisfaction can be considered.

**Acknowledgements.** The first author is supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) of the Brazilian Ministry of Education. We would also like to thank Antonio Carlos da Rocha Costa (UCPel, Brazil) for his valuable comments regarding the Piaget's theory of exchange values.

## References

1. Stein, L.: Creating a bioinformatics nation. *Nature* **417** (2002) 119–120
2. Overbeek, R., Disz, T., Stevens, R.: The SEED: A peer-to-peer environment for genome annotation. *Communications of the ACM* **47** (2004) 47–50
3. Castelfranchi, C.: Social Power: A point missed in multi-agent, DAI and HCI. In Demazeau, Y., Müller, J.P., eds.: *Decentralized A.I.* Elsevier Science, Amsterdam (1990) 49–62
4. Castelfranchi, C., Miceli, M., Cesta, A.: Dependence Relations Among Autonomous Agents. In Werner, E., Demazeau, Y., eds.: *Decentralized A.I. 3*, Amsterdam, Elsevier Science (1992) 215–227
5. Piaget, J.: *Sociological Studies*. Routledge, London (1973)
6. Jennings, N.R.: Coordination techniques for distributed artificial intelligence. In O'Hare, G.M.P., Jennings, N.R., eds.: *Foundations of Distributed Artificial Intelligence*. (1996) 187–210
7. Rodrigues, M.R., da Rocha Costa, A.C., Bordini, R.H.: A system of exchange values to support social interactions in artificial societies. In: *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, Melbourne (2003) 81–88
8. Miceli, M., Castelfranchi, C.: A Cognitive Approach to Values. *Journal for the theory of social behavior* **2** (1989) 169–193
9. Miceli, M., Castelfranchi, C.: The Role of Evaluation in Cognition and Social Interaction. In Dautenhahn, K., ed.: *Human Cognition and Social Agent Technology*. Volume 19 of *Advances in Consciousness Research*. John Benjamins, Amsterdam (2000) 225–261
10. Antunes, L., Coelho, H.: Decisions based upon Multiple Value: The BVG agent architecture. In Barahona, P., Alferes, J.J., eds.: *Progress in Artificial Intelligence*, Proceedings of the Ninth Portuguese Conference on Artificial Intelligence. Volume 1695 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin (1999) 297–311
11. Antunes, L., Faria, J., Coelho, H.: Improving choice mechanisms within the bvg architecture. In Castelfranchi, C., Lespérance, Y., eds.: *Intelligent Agents VII: Proceedings of the Seventh International Workshop on Agent Theories Architectures and Languages*. Volume 1986 of *Lecture Notes in Computer Science*. Springer-Verlag, Boston (2001) 290–304
12. Sichman, J.S., Conte, R., Castelfranchi, C., Demazeau, Y.: A social reasoning mechanism based on dependence networks. In Cohn, A.G., ed.: *Proceedings of the Eleventh European Conference on Artificial Intelligence*, John Wiley and Sons (1994) 188–192
13. David, N., Sichman, J.S., Coelho, H.: Agent-based simulation with coalitions in social reasoning. In Moss, S., Davidson, P., eds.: *MultiAgent-Based Simulation*. Volume 1979 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin (2001) 245–266
14. Lopez, F.L., Luck, M., d'Inverno, M.: A framework for norm-based inter-agent dependence. In: *Proceedings of the Third Mexican International Conference on Computer Science*. (2001) 31–40
15. d'Inverno, M., Luck, M.: A formal view of social dependence networks. In Zhang, Lukose, eds.: *Distributed Artificial Intelligence Architecture and Modelling: Proceedings of the First Australian Workshop on Distributed Artificial Intelligence*. Volume 1087 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin (1996) 115–129



16. Saha, S., Sen, S., Dutta, P.S.: Helping based on future expectations. In: Second International Joint Conference on Autonomous Agents and Multiagent Systems, Melbourne (2003) 289–296
17. Sarne, D., Kraus, S.: Time-variant distributed agent matching applications. In: Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York New York (2004) 168–175
18. Rodrigues, M.R., da Rocha Costa, A.C.: Using qualitative exchange values to improve the modelling of social interactions. In Hales, D., Edmonds, B., Norling, E., Rouchier, J., eds.: Multi-Agent-Based Simulation III. Volume 2927 of Lecture Notes in Computer Science. Springer-Verlag, Berlin (2003) 57–72