

Measuring the Social Ability of Software Agents

Fernando Alonso José L. Fuertes Loïc Martínez
Fac. de Informática
Universidad Politécnica de Madrid
28660 – Boadilla del Monte, Madrid (Spain)
{falonso, jfuertes, loic}@fi.upm.es

Héctor Soza
Fac. de Ingeniería y Ciencias Geológicas
Universidad Católica del Norte
Av. Angamos 0610, Antofagasta (Chile)
hsoza@ucn.cl

Abstract

To evaluate the global quality of a software agent it is necessary to define appropriate quality characteristics and to determine a set of measures for these features. A comprehensive set of measures has not yet been developed for agent-oriented software. However, some software measures have been adopted from other software paradigms, especially from the object-oriented paradigm because they have features in common. A key characteristic defining a software agent is its social ability, that is, its ability to interact with other agents to achieve its goals. This paper presents a first approximation to a set of measures for evaluating the social ability of software agents for use in the calculation of a global value for this characteristic.

1. Introduction

Software development is an engineering discipline that has embraced different paradigms in its short lifetime: procedural, object-oriented, component-driven and, recently, agent-oriented software. This rapid development has prevented the consolidation of standard quality measurements for all paradigms because each one has distinctive characteristics that call for the application of specific quality measures. There is now a well-established set of software measures for procedural and object-oriented software, but not for agent-oriented software.

Whereas some specific measurements have been developed for the agent paradigm, others have been borrowed from the object-oriented paradigm (e.g. [1], [2], [3]) because agents have characteristics in common with software objects, such as a modular programming approach, encapsulation, information hiding, etc. However, there is no set of measures defining the quality of agent-based software or each individual agent. This is what has motivated us to develop this research with the objective of defining a set of measures to evaluate the quality of a software agent.

As pilot results of this investigation, we present in this paper a preliminary set of measures to evaluate agent social ability. This characteristic indicates the agent's ability to interact with other agents and humans in order

to achieve its design objectives [4].

The paper is structured as follows. The next section presents some related work on the development of measures related to software agents. In section 3 we discuss agent types and characteristics. Section 4 proposes some attributes associated with the characteristics of agent-oriented software. Section 5 suggests measures of agent social ability. Section 6 summarizes the process of calculating social ability and its application to a case study. The last section includes some concluding remarks and future research.

2. Related work

Looking at the research developed in recent years on this subject, Franklin et al.'s work [5] is noteworthy. Franklin et al. developed a methodology to quantitatively and qualitatively evaluate how intelligent software agents are. To do this, they used a set of tests to analyse agent behaviour and performance in different environments and situations. At the same time a set of measures were being developed, considering product, process and resources, to evaluate the performance of software agents and bring an empirical criterion into the evaluation [1].

Later, a method for building a quality taxonomy for software agents was developed using object-oriented development with Java and extreme programming [6]. However, Benedicenti et al. did not report measures related to this study. At a later date, a novel performance analysis approach for quantitatively gauging the performance characteristics of different mobile-agent platforms was developed. This approach was implemented as a hierarchical framework of benchmarks designed to isolate performance properties of interest at different levels of detail [7].

In 2004, a project report presented the results of adapting some product measurements from the procedural and object-oriented paradigms to agent-oriented software [3]. Shin compared objects and agents, and developed a program to evaluate the measures applied to an example. Shin does not evaluate the characteristics associated with a software agent. Nor does he determine associated attributes for determining quality. This research is confined to adapting existing object-oriented software

measures to evaluate the software agent.

Considering that just how good an agent is at attaining its goals is a function of the quality of its features, we find that none of the above studies provides specific quality measures for these attributes that can be used to get a global quality measure of the software agent. This is the focus of this research.

3. Software agent characteristics and types

To examine agent quality, quality is considered in this article to be decomposed into several levels as defined in ISO 9126 [8]: characteristic – subcharacteristic (attribute) – measure.

It is an acknowledged fact in the agent-oriented software field that the characteristics defining a software agent are [4], [1], [3], [9], [10]:

- **Social ability:** The agent is able to interact with other agents, and possibly humans, in order to achieve its design objectives [4].
- **Autonomy:** The agent is able to operate on its own without the need for any human guidance or the intervention of external elements. It has control over its own actions and internal states [1].
- **Proactivity:** The agent is able to exhibit goal-directed behaviour by taking the initiative in order to achieve its design objectives. This capability often requires the agent to anticipate future situations (to take the initiative), to interact with other agents and to perceive its environment [4].
- **Reactivity:** The agent is able to perceive its environment and respond in a timely fashion to any environmental changes in order to achieve its design objectives [1]. Its actions will cause changes to the environment aimed at achieving its goals [10].
- **Adaptability:** The agent is able to adapt, is flexible and has the capability to set up its own goals based on its implicit purpose [1].
- **Intelligence:** The agent is able to reason, plan, solve problems, think abstractly, comprehend ideas and language, and learn [11].
- **Mobility:** The agent is able to move itself in the environment or other environments, preserving its internal state. It must be able to interact in the new environment to gather the necessary information in order to accomplish its goal [9].

Agents do not necessarily have all or the same measure of these characteristics defining software agent behaviour. This will depend on the type of software agent.

For clarity and understanding, Nwana [12] divided software agents into six different categories:

- **Collaborative:** This agent type emphasizes autonomy and social ability in order to perform tasks. It helps to provide solutions where problems, expertise and

information resources are distributed and allows multiple existing legacy systems to interconnect and interoperate to solve a problem that is too large for one centralized system.

- **Interface:** This agent type emphasizes autonomy, learning and communication with users rather than other agents. Interface agents learn to assist their user better by observing and imitating the user, receiving feedback and explicit instructions from the user and asking other agents for advice. The motivation is to reduce the end user's workload over time. Agent behaviour towards the user is mostly repetitive, though it is different for every user. The agent can adapt to its user's preferences and habits, and agents may share the knowledge of different users in the same community.
- **Mobile:** This agent type is able to move from one environment to another autonomously and continue running in the target environment. These agents are a powerful tool for distributed operations.
- **Information:** This agent type performs the role of managing, manipulating or collecting information from many distributed sources.
- **Reactive:** This agent type acts/responds in a stimulus-response manner to the present state of the environments in which it is embedded. These agents do not need to have internal, symbolic models of their environments. They are relatively simple and interaction with other agents is rudimentary.
- **Smart:** This agent type represents the class of agents that contain most agent behaviours: autonomous, sociable, proactive, adaptable, intelligent, reactive and mobile.

Apart from the above agent categories, there are many hybrid agents that combine different agent types [12].

Software agent quality then will be determined by the set of the quality measures for each of the above characteristics, adapted to the agent type. And the quality of a multi-agent system will be a function of the quality of each of the agents participating in the system.

Table 1 shows how important each characteristic is by agent type according to the following values: the characteristic is not necessary (not at all), it is not very necessary (not very), it is averagely necessary (average), it is very necessary (very) for the type of agent.

Characteristics	Agent Types					
	<i>Collaborative</i>	<i>Interface</i>	<i>Mobile</i>	<i>Information</i>	<i>Reactive</i>	<i>Smart</i>
<i>Social ability</i>	Very	Not very	Average	Average	Not very	Very
<i>Autonomy</i>	Very	Very	Very	Not very	Very	Very
<i>Proactivity</i>	Average	Average	Average	Not very	Not very	Very
<i>Reactivity</i>	Average	Very	Average	Average	Very	Very
<i>Adaptability</i>	Average	Average	Very	Not very	Not very	Very
<i>Intelligence</i>	Not very	Very	Average	Average	Not at all	Very
<i>Mobility</i>	Not at all	Not at all	Very	Average	Not at all	Very

Table 1. Importance of characteristic by agent type

4. Attributes related to agent characteristics

Each agent characteristic can be defined by a series of attributes. These attributes represent properties that can be measured and, therefore, can be used to quantify each characteristic. By measuring these attributes we will be able to evaluate the extent to which an agent has this characteristic. This way, the quality of each agent type can be assessed taking into account how important each characteristic is for the agent type (see Table 1).

In the following we give an overview of the attributes associated with each agent characteristic, and then we discuss the attributes associated with agent social ability.

4.1. Agent attributes

Based on the definitions shown in section 3, we can identify attributes that are representative of each characteristic.

1. **Social ability:** This characteristic is defined by the agent's ability to exchange information with other agents and humans (communication), the ability to collaborate with other agents (cooperation), and the ability to agree upon courses of action to reach agreements and accomplish its goals (negotiation) [13].
2. **Autonomy:** This characteristic is defined by the agent's ability to control its own actions (self control), the ability to encapsulate its behaviour or other data types (encapsulation), the ability to gather information about the environment and to generate new capabilities (learning) and the ability to provide the necessary actions to adjust to new goals (evolution) [14].
3. **Proactivity:** This characteristic is defined by the agent's ability to undertake an action with the objective of accomplishing a goal (initiative) and the ability to interact with other agents and their environment (interaction) [15].
4. **Reactivity:** This characteristic is defined by the agent's ability to acquire, interpret, select and organize sensorial information relative to its environment (perception), the ability to perceive its internal state (proprioception) and the ability to respond to a stimulus or event (reaction) [16].
5. **Adaptability:** This characteristic is defined by the agent's ability to change its state to adapt to the environment (structural adaptation), the ability to evolve and adapt its functionality to the environment at execution time (dynamic adaptation) and the ability to correctly deal with the exceptions produced by the environment to rate the agent's ability to adjust and subsist (exception handling) [17].
6. **Intelligence:** This characteristic is defined by the agent's ability to form and operate on concepts in accordance with reason and logic (reasoning), the ability to create plans (planning), the ability to solve problems not specified in its design in the process of

accomplishing the goal (problem solving) and the ability to learn from experience (learning) [18], [19].

7. **Mobility:** This characteristic is defined by the agent's ability to save its own state and transport this saved state to another host and then resume execution from the saved state (save and move), the ability to execute actions in a non-blocking scheme allowing the main program flow to continue processing (asynchronous execution), the ability to adapt to the environmental conditions (adaptation), the ability to operate without an active connection between client and server (network fault tolerance) and the ability to remedy defects in deployed software (flexible maintenance) [20].

4.2. Attributes of social ability

An agent's social ability is represented by the attributes related to communication, cooperation and negotiation.

1. **Communication:** The ability of communication is identified by the reception and delivery of messages by the agent to achieve its goals. Good communication depends on three things. The first is the agent's level of conversation, which can be assessed taking into account the number of messages invoked in response to a message received by the agent. The second is the number of incoming and outgoing messages received and sent by the agent to maintain a meaningful communication link or accomplish some goals. The third is the message size, considered as the size of the messages sent by the agent during execution [3].
2. **Cooperation:** Cooperation indicates the agent's ability to respond to the services requested by other agents and to offer services to other agents. Good cooperation depends on the agent's level of collaboration with other agents that require its services, meaning that an agent's collaboration level is measured by its ability to accept or reject services requested by other agents and by its ability to offer services [4].
3. **Negotiation:** Negotiation is the agent's ability to make commitments, resolve conflicts and reach agreements with other agents to assure the accomplishment of its goals. Good negotiation depends on agent goal accomplishment, the number of messages sent by the agent when another agent requests a service from it, and the number of messages sent by the agent when it requests a service from other agents [3], [4].

5. Measures for the attributes of social ability

Before discussing the measures for each attribute defining agents' social ability, let us make a number of general points related to the measures that we are going to present.

5.1. Considerations on the measures

Given a software product like a software agent, a product quality measure can be applied in the analysis phase, in the design phase or when the product is finished. In this research we aim to measure the characteristics associated with a constructed software agent, considering its key attributes.

The measures can be divided into three categories: syntax-based or static measures (examine source code providing software characteristics and statistics), execution-based or dynamic measures (measure the running software's characteristics) and objective-based measures (compare the software requirements with software operation) [21]. The measures presented here belong to the second group: execution-based or dynamic measures.

Measurements used for comparisons should be valid and accurate enough to allow reliable comparisons to be made. This means that measurements should be objective, empirical, and reproducible. To be reproducible, the measurement procedures should assure that different people applying the same measure to the software product on different occasions will get the same values (within appropriate tolerances) [22].

To gather valid results in a software product quality evaluation, this evaluation should be conducted in a controlled environment, which we will call the benchmark. This assures that the evaluated measures are repeatable and comparable. Additionally, the evaluation should be based on independent evidence of the implementation of the software under evaluation. This benchmark shall precisely specify the conditions in which the system under evaluation should be run for each dynamic measure.

Each measure is stated by means of a formula that expresses this measure as a function of one or more parameters. The results of each measure are normalized in the interval $[0, 1]$ (where 0 is a poor result for the measure and 1 is a good result).

Figure 1 shows the types of formulae used for the social ability measures obtained by an expert analysis of these formulas. All the measures depend on the argument x . The constants k , k_1 and k_2 are parameters that the software engineer can configure to fine tune the formula performance for each particular case.

Curve (a) indicates that the value of the measure is constant at 1 (optimum measure value) until x reaches a value k (k is a parameter associated with a change in the measure expression and indicates the point at which it is considered that the value of the measure should no longer be optimum). As x grows then, the value of the measure gently descends to zero, describing an exponential curve. An example of this curve is the work done by a trained professional, which is optimal for a period of time, up until, let's say, 30 years (k), when it starts to fall due to problems of competency and ageing.

Curve (b) indicates that the measure grows, describing

a parabola, as x increases up to a value defined by the parameter k_1 . At this point, the measure remains unchanged at the maximum value 1 as long as x is between the parameters k_1 and k_2 . Then its value starts to descend gently down to zero, describing an exponential curve. An example of this is how a top-competition sportsman or woman's physical skills develop over time. Sportsmen and women acquire physical skills until they reach the optimum after about 20 years (k_1). Provided they keep up the same level of physical activity, they then retain their skills for another 15 years ($k_2=35$), at which point their skills start to deteriorate due to age-induced effects.

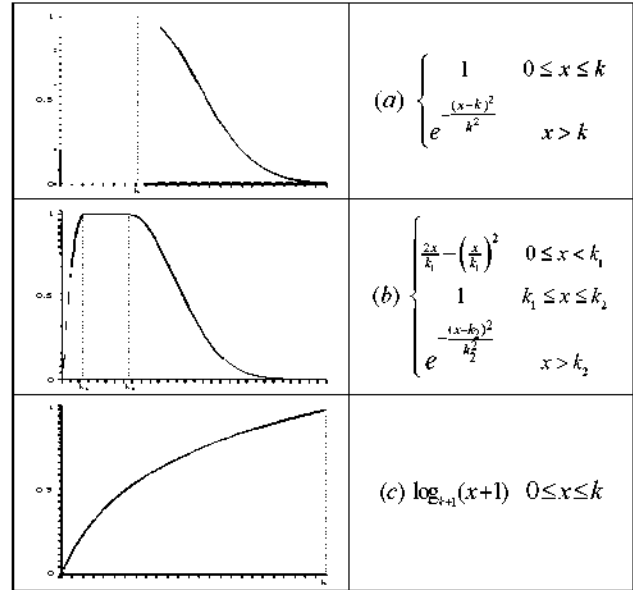


Figure 1. Formula types used in the measures

Curve (c) indicates that the measure grows, rapidly at first, and, as it progresses, its value increases until it reaches the value 1 when x reaches the value k . An example of this curve are mother tongue skills. Mother tongue skills increase rapidly in a child's early years, improving ever more slowly with use until they reach the optimum at about 60 (k) years of age.

5.2. Measures of the attributes of social ability

Next, we present the measures of the attributes of communication, cooperation and negotiation that define an agent's social ability. Some of these measures are based on research on this subject conducted about the agent paradigm, some have been borrowed from other paradigms (procedural, object-oriented, etc.), and adapted to agents technology, and others are new measures proposed in this work. To define the parameters used in the formulas below, we have considered a series of experiments [23] analysed empirically by experts.

5.2.1. Communication. This attribute can be measured using the following measures:

- *Response for message (RFM)*: RFM measures the amount of messages that are invoked in response to a message received by the agent. To process the incoming message, new messages might be sent to another agent requesting new services. If $SM(i)$ is the number of messages sent in response to the i^{th} message received, and n is the total number of messages received by the agent during one execution of the benchmark, then the value of the average number of messages M is (equation 1):

$$M = \frac{\sum_{i=1}^n SM(i)}{n} \quad (1)$$

The RFM measure (equation 2) depends on the value of M and describes curve (b) in Figure 1. In other words, its value increases as the agent gets more communicative until M reaches a particular value k_1 . At this point, it reaches its maximum value and remains unchanged until the value of M is k_2 . As of then the message overload impedes agile communication between the agents.

$$RFM = \begin{cases} \frac{2M}{k_1} - \left(\frac{M}{k_1}\right)^2 & 0 \leq M < k_1 \\ 1 & k_1 \leq M \leq k_2 \\ e^{-\frac{(M-k_2)^2}{k_2^2}} & M > k_2 \end{cases} \quad (2)$$

The value of the parameters k_1 and k_2 depends on the environmental setting in which the agent runs and communicates. The experiments conducted lead us to recommend values of between 4 and 6 for k_1 and between 8 and 10 for k_2 , respectively.

- *Average message size (AMS)*: AMS measures the influence of the data size of the messages sent by the agent on its communication. Let us define MS_{out} as the average data size of the messages sent by the agent during its execution [3], where n is the total number of messages sent by the agent and MB_i is the data size, measured in bytes, of the i^{th} message (equation 3).

$$MS_{out} = \frac{\sum_{i=1}^n MB_i}{n} \quad (3)$$

The AMS measure (equation 4) describes curve (b) in Figure 1 and is a function of the value of MS_{out} . Its value increases up to 1 as the average message size increases and reaches an acceptable value at k_1 . The average message size remains within adequate limits in the interval $[k_1, k_2]$, whereas, as of k_2 , the message size is no longer appropriate and reduces the value of AMS. Too large a message size can result in very poor communication, as a lot of information has to be communicated to other agents.

$$AMS = \begin{cases} \frac{2MS_{out}}{k_1} - \left(\frac{MS_{out}}{k_1}\right)^2 & 0 \leq MS_{out} < k_1 \\ 1 & k_1 \leq MS_{out} \leq k_2 \\ e^{-\frac{(MS_{out}-k_2)^2}{k_2^2}} & MS_{out} > k_2 \end{cases} \quad (4)$$

Considering the FIPA standard [22], the values for the parameters k_1 and k_2 are between 150 and 200 bytes and 250 and 300 bytes, respectively [25].

- *Incoming Message (FIM)*: FIM measures the relation of incoming messages to agent communication during its lifetime. Higher values indicate that the agent has more dependent agents requiring its services [3]. Let us define IM as the total number of incoming messages received by an agent during one execution of the benchmark. The FIM measure (equation 5) describes curve (b) in Figure 1. If the messages sent are less than a quantity k_1 , then this measure grows until it reaches the value 1 for $IM=k_1$, staying at this value in the interval $[k_1, k_2]$. Above k_2 the value of the measure must decrease, as the greater number of messages received affects the agent's communication.

$$FIM = \begin{cases} \frac{2IM}{k_1} - \left(\frac{IM}{k_1}\right)^2 & 0 \leq IM < k_1 \\ 1 & k_1 \leq IM \leq k_2 \\ e^{-\frac{(IM-k_2)^2}{k_2^2}} & IM > k_2 \end{cases} \quad (5)$$

The value of parameters k_1 and k_2 depends on the type of activity performed by the agent and the environment in which it operates. The experiments run lead us to recommend values of between 10 and 12 for k_1 and of between 18 and 20 for k_2 .

- *Outgoing Message (FOM)*: FOM measures the relationship between direct outgoing messages and agent communication during its lifetime. Higher values could indicate that the agent is dependent on other agents [3]. Let us define OM as the total number of outgoing messages sent by an agent during one execution of the benchmark, less the messages that are invoked in response to a message received by the agent. The FOM measure (equation 6) describes curve (b) in Figure 1. If the messages sent are less than a quantity k_1 , then this measure grows until it reaches the value 1 for $OM=k_1$, staying at this value in the interval $[k_1, k_2]$. Above k_2 the value of the measure decreases, as the greater number of messages sent reduces the agent's ability to do its job.

$$FOM = \begin{cases} \frac{2OM}{k_1} - \left(\frac{OM}{k_1}\right)^2 & 0 \leq OM < k_1 \\ 1 & k_1 \leq OM \leq k_2 \\ e^{-\frac{(OM-k_2)^2}{k_2^2}} & OM > k_2 \end{cases} \quad (6)$$

The value of parameters k_1 and k_2 depends on the type of activity performed by the agent and the environment in which it operates. Experiments run lead us to recommend values of between 5 and 8 for k_1 and of between 15 and 18 for k_2 .

5.2.2. Cooperation. This attribute can be measured using the following measures:

- *Services Requests Rejected by the Agent* (SRRA): SRRA measures the influence of the percentage of rejected agent services on cooperation, when other agents require its services. Let us define SA as the total number of messages requiring a service received and accepted by the agent and SR as the total number of messages requiring a service rejected by the agent during its execution. We then define the value of RS (equation 7) as the percentage of requests rejected by the agent, when $SA + SR > 0$.

$$RS = \frac{SR}{SA+SR} * 100 \quad (7)$$

The measure for SRRA (equation 8) describes curve (a) in Figure 1. This measure is considered to be optimum if the percentage of rejected services is low (SRRA=1). However, this value starts to fall in excess of a quantity k of rejected services, because agent cooperation is found to decrease as the percentage of rejected services increases.

$$SRRA = \begin{cases} 1 & RS \leq k \\ e^{-\frac{(RS-k)^2}{k^2}} & RS > k \end{cases} \quad (8)$$

The value of parameter k depends on the type of activity performed by the agent and the environment in which it operates. Experiments conducted lead us to recommend a value of between 20 and 30 for k.

- *Agent Services Advertised* (ASA): ASA measures the number of services that the agent advertises on the yellow pages directory in its environment. Let us define S as the quantity of services that the agent provides. These services are usually advertised by the agent on the yellow pages directory in its environment [3]. The ASA measure (equation 9) follows curve (b) in Figure 1. As S increases, the value of ASA grows, because a low level of offered services is a clear indication of deficient cooperation. ASA reaches the value 1 for k_1 , and remains at this value during the interval $[k_1, k_2]$. As of here, with an excess of services offered, ASA decreases because the agent loses its ability to cooperate, since the number of services it has to deal with increases.

$$ASA = \begin{cases} \frac{2S}{k_1} - \left(\frac{S}{k_1}\right)^2 & 0 \leq S < k_1 \\ 1 & k_1 \leq S \leq k_2 \\ e^{-\frac{(S-k_2)^2}{k_2^2}} & S > k_2 \end{cases} \quad (9)$$

The value of the parameters k_1 and k_2 depends on the environment in which the agent operates and the services that it provides. The experiments run lead us to recommend values of between 3 and 6 for k_1 and of between 8 and 10 for k_2 , respectively.

5.2.3. Negotiation. This attribute can be measured using the following measures:

- *Agent Goals Achievement* (AGA): AGA determines the negotiating efficiency of agents that use negotiation to accomplish their goals. We define G as the number of objectives achieved by the agent in the benchmark execution time. AGA (equation 10) then describes curve (c) in Figure 1.

$$AGA = \log_{e+1}(G+1) \quad (10)$$

The value of k is equivalent to the maximum possible number of objectives to be achieved by the agent. We find that as the quantity of objectives achieved by the agent increases, the value of AGA increases up to the value 1 when $G=k$.

- *Messages by a Requested Service* (MRS): MRS measures the influence of the number of messages exchanged by the agent doing the negotiation when another agent is requesting services from the agent. Let MS be the number of messages exchanged by the agent when it receives a service request. Then the MRS measure (equation 11) is evaluated using curve (a) in Figure 1, if $MS \geq 1$. Let k be the quantity of messages required to accept or reject the request. If MS is less than or equal to k, this measure is 1 (optimum value), but if MS is greater than k, the measure starts to decrease because many exchanged messages affect agent negotiation.

$$MRS = \begin{cases} 1 & 1 \leq MS \leq k \\ e^{-\frac{(MS-k)^2}{k^2}} & MS > k \end{cases} \quad (11)$$

The value of parameter k depends on experimental studies and the opinions of intelligent agent design experts. Experiments conducted lead us to recommend a value of between 3 and 7 for k.

- *Messages Sent to Request a Service* (MSS): MSS measures the influence of the number of messages exchanged by the agent doing the negotiation when the agent is requesting a service from another agent. Let MR be the quantity of messages exchanged by the agent when it requests a service from another agent. The MSS measure (equation 12) then is evaluated by curve (a) in Figure 1. Let k be the quantity of messages required to accept or reject the request. If MR is less than or equal to k, this measure is 1 (optimum value), but if MR is greater than k, the measure starts to decrease because many exchanged messages affect agent negotiation.

$$MSS = \begin{cases} 1 & 0 \leq MR \leq k \\ e^{-\frac{(MR-k)^2}{k^2}} & MR > k \end{cases} \quad (12)$$

The value of parameter k depends on experimental studies and the opinions of intelligent agent design experts. Experiments conducted lead us to recommend a value of between 5 and 8 for k.

6. Case Study

We have selected a multiagent system in which multiple autonomous agents interact [26]. It is an intelligent agent marketplace which includes several kinds of buyer and seller agents that cooperate and compete to process sales transactions for their owners. Additionally, a Facilitator agent is developed to act as a manager of the marketplace. We have used this system to evaluate the system's Buyer and Seller agents' functional quality of social ability.

All agents must register with the Facilitator before they can interact with any other agents in the marketplace. Once the Buyer and Seller agents have been registered by the Facilitator, they continue to communicate indirectly through the Facilitator. There are three types of buyers and sellers in the marketplace. These agents are Basic, Intermediate and Advanced Buyers and Sellers. They have the same negotiation capacities, but they differ as to how sophisticated the techniques used to implement their negotiation strategies are, ranging from simple, hard-coded logic to forward-chaining rule inference. The whole process is carried out using the Contract Net communication protocol, and the Facilitator agent participates as an intermediary.

The Seller agents send messages reporting the articles that they have to sell, and the Buyers respond stating their willingness to buy and what they offer for the article. The Seller agents respond by accepting or rejecting the offer, and, when they receive this message, the Buyer agents return a confirmation message.

The social ability study focused on six agents: the three types of Buyer agents and the three types of Seller agents. Table 2 shows the values of the communication, cooperation and negotiation measures during the assessment.

	Basic Buyer	Inter. Buyer	Adv. Buyer	Basic Seller	Inter. Seller	Adv. Seller	System
RFM	0.93	0.94	0.98	1.00	0.91	1.00	0.97
AMS	0.98	1.00	0.79	0.98	1.00	0.79	0.92
FIM	0.99	0.96	0.91	1.00	0.99	0.96	0.97
FOM	0.98	0.92	0.83	1.00	0.98	0.83	0.92
SRRA	0.89	0.64	0.44	0.89	0.64	0.44	0.66
ASA	0.84	0.84	0.84	0.84	0.84	0.84	0.84
AGA	0.50	0.80	0.80	0.50	0.80	0.80	0.70
MRS	0.85	0.96	1.00	1.00	1.00	0.89	0.97
MSS	1.00	0.89	0.64	1.00	1.00	0.64	0.85

Table 2. Measures of the social ability attributes

The values of the social ability measures are found to be generally high. The ASA measure has the same value for all agents because each agent provides the same number of services. Similarly, there is not much difference between the quantity of messages received or sent by each agent, and some of the SRRA measures are the same. For the Basic Buyer and Seller, the value of the

AGA measure is low, because they accomplish just one objective for every two of the other agent types.

Table 3 shows the values for the measure of each attribute, calculated by aggregating the measures for each attribute. Row 5 is the value of the social ability characteristic for each agent, calculated by aggregating the measures of all the attributes. Finally, the last column shows the value of the system measure, calculated by aggregating the values of the attribute and characteristic measures for all the agents. Thus the bottom right table cell contains the value of the system's overall social ability. The above values are aggregated in each case using the arithmetic mean. Even so, the results could be refined, using a weighted mean with weights elicited from experts using any of the existing weighting techniques.

	Basic Buyer	Inter. Buyer	Adv. Buyer	Basic Seller	Inter. Seller	Adv. Seller	System
communication	0.96	0.95	0.88	1.00	0.99	0.89	0.95
cooperation	0.87	0.74	0.64	0.87	0.74	0.64	0.75
negotiation	0.78	0.89	0.81	0.83	0.90	0.81	0.84
Social Ability	0.86	0.84	0.76	0.89	0.87	0.76	0.85

Table 3. Values of social ability attributes

In addition, we can see that Advanced agents have a lower social ability value, whereas Basic agents have a higher one.

We find that all agents have a very high value for the communication attribute. This suggests that communication is very important in this system. This is followed by negotiation, influenced by the number of messages sent to accept or reject a service and, finally, cooperation with lower scores for some agents.

Analysing the aggregated values for the whole system, the communication measure is the highest scorer (95%). This reflects the influence of the messages, sent, received and message size on this attribute. The value of the system negotiation attribute is 84%. This is influenced by the high scores for the messages sent when services are requested from an agent or when this agent requires services, although the lower value of AGA, caused by the fact each agent has a set number of objectives, brings down the score for this attribute. The lowest scoring attribute for this system is cooperation (75%). This is because the system SRRA measure is not very high. SRRA decreases as the agent gets more specific, and this has an impact on this value.

Finally, the value for system social ability is 85%, that is, the social ability of the agents in this case study is high. The results of this case study were compared with the opinion of two specialists in the agents field. From this analysis we were able to confirm the reliability of the results.

7. Conclusions and future work

Reviewing the literature on agent-oriented software, we presented the different characteristics of an intelligent

agent: social ability, autonomy, proactivity, reactivity, adaptability, intelligence and mobility. Then we proposed a preliminary approximation of a set of measures for agent-oriented software considering the characteristic of social ability. This characteristic was decomposed into different measurable attributes, and we discussed the measures considered for its evaluation.

We developed a typical case study to evaluate the attributes associated with this characteristic in each of the participating agents. The designed agents show a high social ability (greater than 75%). The communication attribute (over 85%) stands out above all of them, whereas cooperation presents lower values (although greater than 60%).

The values were 95 % for the communication attribute, 75% for the cooperation attribute, and 84% for the negotiation attribute. Agent social ability for the whole system then was 85%.

Future work will address a thorough study of agent-based systems, analysing the measures of each characteristic for each system agent type, and the role they play in the system quality measure. To achieve this aim, we will propose a quality evaluation model and evaluate this model across several software agent applications, considering the different types and attributes present in agents.

8. References

- [1] R. Dumke, R. Koeppel, and C. Wille, "Software Agent Measurement and Self-Measuring Agent-Based Systems", *Preprint No. 11*, Fakultät für Informatik, Otto-von-Guericke-Universität, Magdeburg, 2000.
- [2] B. Far, and T. Wanyama, "Metrics For Agent-Based Software Development", *Proc. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2003)*, May, 2003, pp. 1297-1300.
- [3] K. Shin, *Software Agents Metrics. A Preliminary Study & Development of a Metric Analyzer*, Project Report No. H98010, Dept. Computer Science, School of Computing, National University of Singapore, 2003/2004.
- [4] M. Wooldridge, *An Introduction to Multiagent Systems*. John Wiley & Sons, Inc., 2002.
- [5] D. Franklin, and A. Abrao, "Measuring Software Agent's Intelligence", *Proc. International Conference: Advances in Infrastructure for Electronical Business. Science and Education on the Internet*, L'Aquila, Italy, August, 2000.
- [6] L. Benedicenti, V. Wei, P. Lee, and R. Paranjape, "Establishing quality control in software agents". *ACM SIGAPP Applied Computing*, Vol. 9, No. 3, 2001, pp. 31-33.
- [7] M. D. Dikaiakos, and G. Samaras, "Performance Evaluation of Mobile Agents: Issues and Approaches", In Dumke, Rautenstrauch, Schmietendorf and Scholz (eds.), *Performance Engineering. State of the Art and Current Trends*, Lecture Notes in Computer Science Series, State of the Art Survey, Vol. 2047, Springer, May 2001, pp. 148-166.
- [8] ISO, *ISO/IEC 9126-1: Software engineering- Product quality- Part 1: Quality Model*, International Standard ISO/IEC 9126-1:2001, 2001.
- [9] A. Mas. *Agentes Software y Sistemas Multi-Agente: Conceptos, Arquitecturas y Aplicaciones*, Pearson Educación, Madrid, 2005.
- [10] I. Duncan, and T. Storer, "Agent testing in an ambient world", in T. Strang, V. Cahill, and A. Quigley (eds.), *Pervasive 2006 Workshop Proceedings*, Dublin, Eire, May 2006, pp. 757-764.
- [11] L. Gottfredson, "Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography", *Intelligence*, 24(1), 13-23, reprinted in G. J. Boyle & D. H. Saklofske (eds.), *Psychology of Individual Differences*, Vol. 1: Individual Differences. Sage, London, 2003.
- [12] H. S. Nwana, "Software Agents: An Overview", *Knowledge Engineering Review*, Vol. 11, No. 3, September 1996, pp. 1-40.
- [13] C. Wille, R. Dumke, and S. Stojanov, "Performance Engineering in Agent-Based Systems Concepts. Modelling and Examples", *Current Trends in Software Measurement - Proceedings of the 11th International Workshop on Software Measurement*, Montreal, Canada, 2001, pp. 153-180.
- [14] R. Pfeifer, and C. Scheier. *Understanding Intelligence*, MIT Press, September 1999.
- [15] S. Covey, *The Seven Habits of Highly Effective People*, Free Press, 1989, 15th anniversary edition, 2004.
- [16] K. Dautenhahn, and C. L. Nehaniv, "The Agent-Based Perspective on Imitation", in C. L. Nehaniv and K. Dautenhahn (eds.), *Imitation in Animals and Artefacts*, The MIT Press, Cambridge, MA, 2002.
- [17] Z. Guessoum. "Adaptive Agents and Multiagent Systems", *IEEE Distributed Systems Online*, Vol. 5, No. 7, IEEE Computer Society, July 2004, pp. 1541-4922.
- [18] B. Clark, "Characteristic of Extreme Intelligence", *Growing Up Gifted*, 5th ed., Simon & Shuster, 1997
- [19] N. Ádám. "The Characteristics of Informatics Intelligence", *Microcad'99 Konferencia kiadványa*, 1999.
- [20] T. Ledoux, and N. Bouraqadi-Saâdani, "Adaptability in Mobile Agent Systems using Reflection", *Middleware'2000 Workshop on Reflective Middleware*, New York, April 2000.
- [21] G. M. Barnes, and B. R. Swim. "Inheriting Software Metrics". *Journal of Object-Oriented Programming*, Vol. 6, No. 7, November/December 1993, pp. 27-34.
- [22] ISO, *ISO/IEC 9126-1: Software engineering- Product quality- Part 4: Quality in use metrics*, International Standard ISO/IEC 9126-4:2004, 2004.
- [23] F. Alonso, J. L. Fuertes, L. Martínez, H. Soza. *Social Ability Experiments*, Technical Report, CETTICO-TR1-08, <http://www.cettico.fi.upm.es/tr/2008/cettico-tr1-08.pdf>.
- [24] Foundation for Intelligent Physical Agents, *FIPA Communicative Act Library Specification*, Geneva, Switzerland, 2002.
- [25] A. Genco, S. Sorce, G. Reina, and G. Santero, "An Agent-Based Service Network for Personal Mobile Devices", *IEEE Pervasive Computing*, Vol. 5, No. 2, April-June 2006, pp. 54-61.
- [26] J. P. Bigus, and J. Bigus. *Constructing Intelligent Agents using Java*. 2nd ed., John Wiley & Sons, Inc., 2001.