We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Quality Measures for Agent-Oriented Software

Héctor Soza

Additional information is available at the end of the chapter

Abstract

The popularity of software agents and multiagent systems has increased, and this is now one of the most active areas in informatics research and development. Agent-oriented software engineering is a new paradigm. But no paradigm would be complete without a means for measuring its quality. This work is the result of an investigation in order to evaluate characteristics of a software agent, measuring its most relevant characteristics. For this, we consider the three most important characteristics of a software agent: its social ability, autonomy and proactivity. Measurable attributes are defined for each feature and then are defined measures to evaluate each attribute, and from these values obtain the value of the characteristics considered are proposed. To validate these measures are applied to a case study.

Keywords: agent quality, social ability, proactivity, autonomy, software quality

1. Introduction

Software quality has been a major issue in software development history. The ultimate aim is to ensure user satisfaction with the reliability and operation of software products based on well-defined characteristics. Several software quality models have been produced, such as the procedural model [1], the object-oriented model [2] and the international standard quality model established by ISO and IEC [3].

To evaluate the software agent, published works have been investigated in relation to measures to evaluate it, which mainly consider the adoption of measures of other paradigms, such as procedural and object-oriented, since there are programming characteristics in common with software oriented to agents, such as modular programming and encapsulation [4, 5].

We are interested in applying the software quality model for another type of software: agent systems. We take the viewpoint that it is essential to analyze the key software agent characteristics in order to evaluate the quality of a software agent. These characteristics define agent behavior inside a multi-agent system and include social ability, autonomy, proactivity, reactivity, adaptability, intelligence and mobility [6, 7].

In this chapter, we present the study of the characteristics of social ability, autonomy and proactivity, with their attributes and associated measures, considering that they are the most representative of the software agent [8–10]. The objective of this research work is to propose attributes associated with these characteristics of the software agent and a set of measures that allow evaluating these attributes [11].

To evaluate the quality of software, ISO/IEC proposed in 2001 to decompose this quality into three hierarchical levels: characteristics (correspond to properties that the software must verify), sub-characteristics or attributes (they are measurable qualities that influence each characteristic) and measurements (are the metrics that allow evaluating the attributes) [3]. Based on this decomposition, this research work has been proposed to evaluate the characteristics of a software agent.

The rest of the chapter is structured as follows. The characteristic of an agent is defined, and attributes related to social ability, autonomy and proactivity are presented. Then, works related to the development of measures to evaluate characteristics of the software agent are presented. Next, measures to evaluate the attributes associated with these characteristics are described, and their application to a case study is presented. The chapter ends with a series of conclusions and lines of future research.

2. Characteristics of the agent software

According to McCall [1] and ISO/IEC [3], to measure the quality of a software product, three aspects of the product must be defined: its characteristics, its sub-characteristics (called attributes) and the measurements of these attributes.

The characteristics considered in this work as more representative of a software agent are its social ability, autonomy and proactivity [11]. **Table 1** summarizes these characteristics, providing a short definition and a list of relevant attributes that we identified in [11].

The importance of the social ability is that in order to achieve its objectives, the agent must be able to communicate, negotiate and cooperate with the other agents of the system [6, 11].

The importance of autonomy lies in the fact that in order to achieve an objective the agent must have the possibility to decide, not only how to achieve an objective, but also that the objectives must be sought on the basis of an interest generated endogenously [12], must control their internal state, have independence and the ability to adapt to the needs of their environment [10].

Characteristics	Definition	Attributes
Social ability	The agent is able to interact with other agents, and possibly humans, in order to achieve its design objectives.	• Communication • Cooperation • Negotiation
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.	• Self-control • Encapsulation • Learning • Evolution
Proactivity	The agent is able to exhibit goal-directed behavior 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.	• Initiative • Interaction • Reaction

Table 1. Characteristics and attributes of software agents.

The importance of proactivity is that it implies that the agent executes actions on his own initiative to achieve his objectives, interacting with the agents in the environment and reacting to the stimuli that appear in this environment, so he must have the ability to evaluate their environment and decide what action to take [13, 14].

3. Related work

Considering the research developed on the characteristics and quality measures of a software agent, until before 2013, few studies in the literature had focused on proposing measures related to the characteristics of social ability, autonomy and proactivity of a software agent and few specify a set of specific measures to evaluate these characteristics.

In [4], it is proposed to measure the social ability of the agent using the complexity of the agent as a way of knowing the degree of the organizational dimensions of the agent (such as social, relational, environmental and personal).

Cernuzzi and Rossi proposed a framework for an evaluation of the analysis and design of agent-oriented modeling methods. They evaluate the autonomy of an agent considering whether or not the modeling techniques check whether the agents have control over their internal state and behavior [15].

In 2004, Shin presented the results of adapting some measures of procedural and object-oriented paradigms to agent-oriented software, as well as adding specific measures for this paradigm. In his work, he does not define attributes to evaluate these characteristics; he only defines measures to assess some aspects of these characteristics of the agent's software [16].

Huber proposes measuring social dependence to evaluate autonomy. To calculate a global value of the autonomy of a software agent, Huber combines the value of the autonomy of social integrity and the value of the autonomy of the dependency [17].

The state of the art revealed the need to propose measures that would allow evaluating the most relevant characteristics of the software agent, with appropriate measures that consider the attributes related to these characteristics.

4. Attributes and measures

The measures proposed for each attribute are the result of research in software agents. Some were selected from the procedural and object-oriented paradigms, and others are new measures specially designed to evaluate these attributes [11].

4.1. Attributes and measures of social ability

The social ability in the agents is the competence they have at the moment of exchanging information between them (communication), the ability in the joint collaboration (cooperation) and the ability to reach an agreement on the way forward to reach their objectives (negotiation) [18].

The communication attribute is identified with the agent's ability to receive and send messages by the agent to achieve its objectives. It has been determined that a good communication of the agent can be evaluated considering: the number of messages invoked in response to a message received by the agent (RFM), the size of the messages sent by the agent during execution (AMS), and the number of incoming and outgoing messages received (FIM) and sent (FOM) by the agent to maintain a significant communication link or perform some objectives [16, 18].

The cooperation attribute is identified with the agent's ability to respond to the services requested by other agents and offer services to other agents. It has been determined that good cooperation can be evaluated considering: the agent's ability to accept or reject services requested by other agents (SRRA), and the agent's ability to offer services (ASA) [6, 18].

The negotiation attribute is identified with the agent's ability to establish commitments, resolve conflicts and reach agreements with other agents to ensure compliance with its objectives. Therefore, this attribute can be evaluated considering: the number of objectives achieved (AGA), the number of messages sent by the agent when another agent requires a service from it (MRS), and the number of messages sent by the agent when requesting a service from other agents (MSS) [6, 18]. **Table 2** presents the nine proposed measures for evaluating a software agent's social ability and its definitions.

4.2. Attributes and measures of autonomy

Autonomy in agents is understood as the agent's ability to control his own actions (self-control), the ability to manage his behavior and other information related to his actions (functional independence) and the ability to self-adjust and adapt to new requirements (capacity for evolution) [13].

Measures to evaluate the attributes of social ability

Communication	Answers by message	RFM: Measures the number of messages that are sent in response to a message received by the agent
	Average message size	AMS: Measures the data size of the messages sent by the agent in their communication
	Number of messagesreceived	FIM: Measures the number of messages received in the agent's communication during its lifetime
	Number of messages sent	FOM: Measures the number of messages sent by the agent in the agent's communication during its execution time
Cooperation	Service requests rejected by the Agent	SRRA: Measures the percentage of services rejected by the agent when other agents require their services
	Amount of services offered by the agent	ASA: Measures the amount of services that the agent advertises in the yellow pages directory of their environment
Negotiation	Objectives achieved by the agent	AGA: Measures the efficiency of agent negotiation to achieve its objectives, with the support of other agents in the environment
	Messages for a requested service	MRS: Measures the number of messages exchanged by the agent when conducting a negotiation, when another agent is requesting their services
	Messages sent to request a service	MSS: Measures the number of messages exchanged by the agent when performing a negotiation when the agent requests a service from another agent

Table 2. Measures to evaluate the attributes of social ability.

The self-control attribute is identified with the fact that the agent is able to operate on its own, without the need for human support or the intervention of external elements, to achieve its objectives [13], and with the level of control that the agent has about its own state and behavior [4]. It has been determined that a good self-control of the agent can be evaluated considering: the amount and complexity of the pointers or references that the agent uses in its programming in the agent's internal state (SC), the size of the agent's internal state (ISS), and the complexity of the services that the agent offers (BC) [13].

The functional independence attribute is identified with the executive tasks that require an action that the agent must perform on behalf of any of the users it represents or of other agents [6, 16]. A good level of functional independence indicates that the agent does not have to perform many executive tasks, which can be evaluated considering the fraction of messages requiring action by the user or other agents of the system with respect to all messages received [13]. It has been determined that a good functional independence of the agent can be evaluated considering the influence on the agent of the fraction of executive messages (that request an action) received from the user, to whom the agent represents, or from other agents (to whom it is obligated to respond) with respect to all received messages (EMR) [13].

The attribute capacity of evolution is identified with the agent's ability to adapt to the needs of the environment [19] and take the necessary measures to self-adjust to new objectives [20]. A good capacity for evolution can be evaluated considering: the ability to update its status by the agent (SUC), and the frequency of updating its status (FSU) [13]. **Table 3** presents the six measures proposed to evaluate the agent's autonomy and its definitions.

Measures to evaluate the attributes of autonomy		
Self-control	Structural complexity	SC: Measures the amount and complexity of the pointers or references that the agent uses in its programming in the agent's internal state
	Size of the internal state	ISS: Measures the size of the agent's internal state
	Behavior complexity	BC: Measures the complexity of the services that the agent offers (only applies to agents that offer services)
Functional independence	Fraction of executive type messages	EMR: Measures the influence on the agent of the fraction of executive messages (that request an action) received from the user, to whom the agent represents, or from other agents (to whom it is obligated to respond) with respect to all received messages (considering communication actions)
Capacity of evolution	Capacity to update the status	SUC: Measures the agent's capacity to update its status
	Status update frequency	FSU: Measures the update frequency of the agent's state during execution

Table 3. Measures to evaluate the attributes of autonomy.

4.3. Attributes and measures of proactivity

Proactivity refers to the ability of agents to achieve their goals. To do this, it must make decisions dynamically (initiative), have the ability to relate to obtain information that helps solve problems (interaction) and must act actively in response to environmental stimuli (reaction) [10].

The initiative attribute is identified with the agent's capacity to meet the objectives defined in its design, through goal-directed behavior [6], and to undertake an action on its own with the goal of achieving its objectives [10, 21]. A good initiative capacity can be evaluated considering: the number of roles that the agent must develop (NOR), the number of objectives reached by the agent (AGA), and the average number of messages exchanged by the agent to achieve its objectives (MAG) [10].

The interaction attribute is identified with the agent's capacity to interact with other agents and their environment [10, 22]. A good interaction capacity can be evaluated considering: the number of services implemented within the agent that enable it to achieve its goals (SA), the number of different types of agent messages that the agent can process (NMT) and the average number of public services called per agent (NSC) [10].

The reaction attribute is identified with the agent's capacity to react to a stimulus from the environment according to a stimulus/response behavior [23]. It has been determined that a good capacity for reaction can be evaluated considering: the number of requests received and resolved at runtime (NPR), and the agent's operation complexity (AOC) [10]. **Table 4** presents the eight proposed measures to evaluate the proactivity of a software agent and their definitions.

Figure 1 shows a summary of the characteristics, attributes and measures proposed for the software agent.

Measures to evaluate the attributes of proactivity		
Initiative	Number of roles	NOR: Measures the influence on initiative by the number of roles assigned to an agent in the system under evaluation and defined in the design phase
	Number of goals	AGA: Measures the relationship between the number of goals achieved by the agent during execution over the originally assigned goals
	Messages to achieve goals	MAG: Measures the number of executive messages used to communicate with other system agents to achieve its goals
Interaction	Services per agent	SA: Measures the number of services implemented within the agent (number of implemented public methods or offered services, not including internal methods) that enable it to achieve its goals
	Number of message types	NMT: Measures the number of different types of agent messages that the agent can process
	Number of services called per agent	NSC: Measures the average number of public services called per agent
Reaction	Number of processed requests	NPR: Measures the number of requests received and resolved at runtime
	Agent operations complexity	AOC: Measures the mean complexity of the operations to be performed by the agent to achieve its goals

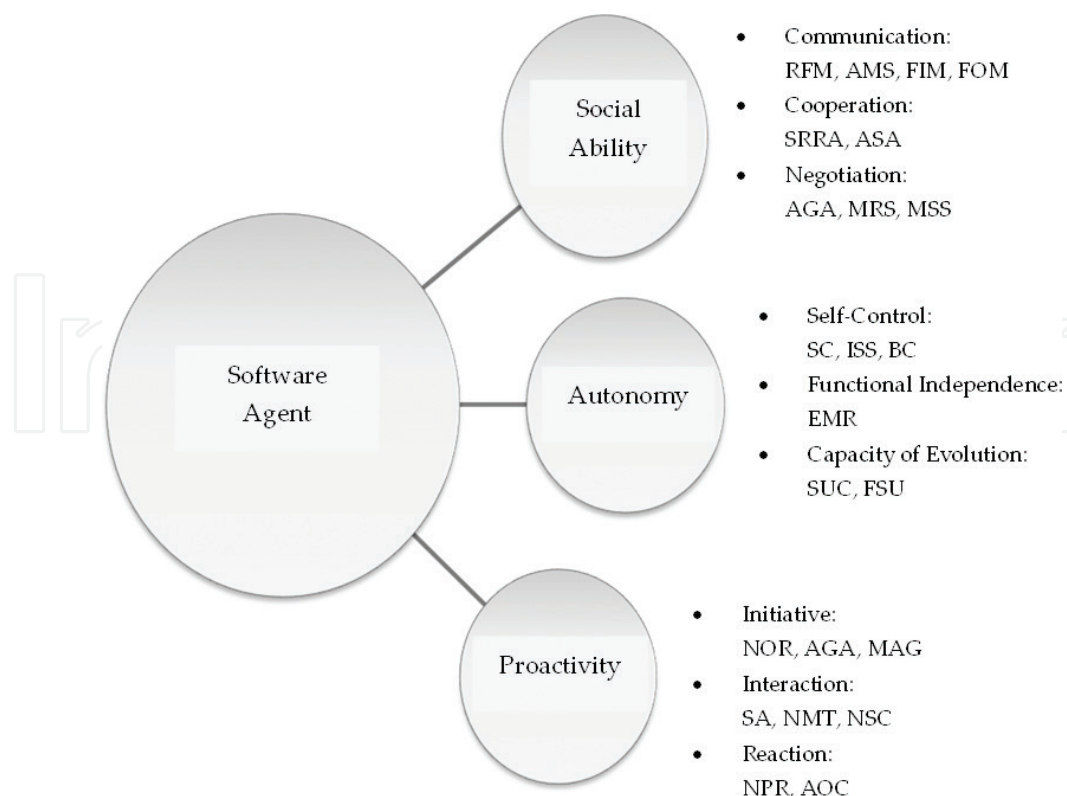**Table 4.** Measures to evaluate the attributes of proactivity.

Figure 1. Characteristics, attributes and measures of a software agent.

To use the quality measures, it is proposed to follow the following steps, which are an adaptation of those proposed in [24].

- Determine the weights of the different characteristics, attributes and measures. To do this, the Hierarchical Analysis Method [25] can be used to obtain the relevant matrices adapted to the problem's environment:

  1. The weights of the characteristics are determined from the relevance matrix obtained through surveys to software engineers.

  2. The weights of the attributes for each characteristic are determined, using surveys to the software engineers to determine the matrices of relevance.

  3. The weights of each measure are determined for each attribute.

- Evaluate the values of each measure:

  1. Calculate the values of each measure according to the scope of application

  2. Normalize the proposed measures if necessary

- Based on the weights and the values of the measurements, the following are determined:

  1. The quality values of the attributes of each characteristic

  2. The quality values of each characteristic

  3. The quality value associated with the software agent system

- Analyze the results

5. Study case

Next, a case study corresponding to the agent management process of a banking environment is presented, which considers agents of the bank type, the client type and the facilitator agent, and which performs an exchange of knowledge and actions among these agents (**Figure 2**).

The agents of the system are related through operations typical of a bank, such as creating an account, turning money or requesting a loan. The system consists of two agents of type clients and three agents of type bank, in addition to a facilitator agent. The first one will initiate the conversations and the banks will try to answer them by carrying out internal processes or questions to other entities of the same nature.

To carry out this study, software was used that applied the evaluations of the measures of this work [26, 27].

The JADE platform was used to design the software [28], and the coding was performed in the Eclipse integrated development environment [29]. Three different experiments were carried out to evaluate the different measures exposed in this work. An aspect of the graphical interface of the systems implemented for the banking system is presented in **Figure 3**.

In all the experiments, the opinion of several experts in the development of software oriented to agents with a score between 0 and 10 of the measures considered in the system was considered; and then the Kolmogorov-Smirnov test was applied [30], in which the arithmetic mean of the opinions of the experts and the arithmetic mean of the measures to make them comparable were considered.

Figure 2. Bank system agents.

Figure 3. Graphical interface of the banking multi-agent system.

In the first experiment, measures were applied to evaluate the social ability of each agent in the system. **Figure 4** shows the average results for each measure for each attribute and for each agent present in the system. **Figure 5** presents the value of the attributes and the social ability, calculated considering the arithmetic mean of the measurements for each attribute. It is observed that the banking agents have a greater value of the cooperation attribute, while the client agents have it in the negotiation. Analyzing the values for the system, the average values of all the attributes are very similar (around 86%). Finally, the value of the social ability of the system is 86%, which is high for this system.

Regarding the evaluation of experts on the measures of social ability applying the Kolmogorov-Smirnov test, the arithmetic mean of the opinions of the experts and the arithmetic mean of the measurements were taken, obtaining a level of significance of 0.50 that allows to reject the hypothesis that both samples come from different distributions. **Figure 6** presents a comparison of the results of the average obtained by the system and the experts, applying the Kolmogorov-Smirnov test.
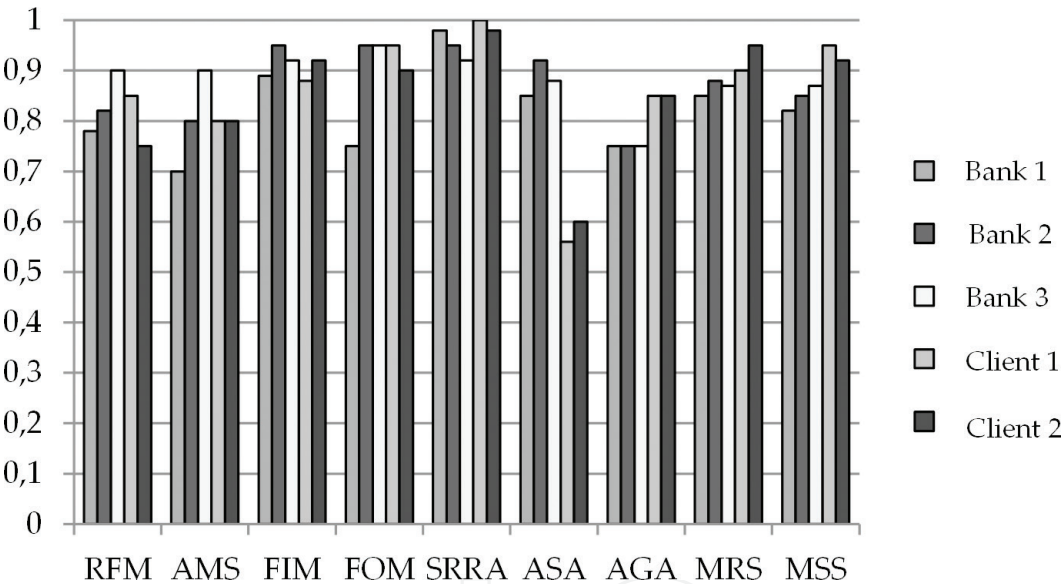

Figure 4. Average values of the measures of the attributes of social ability per agent.
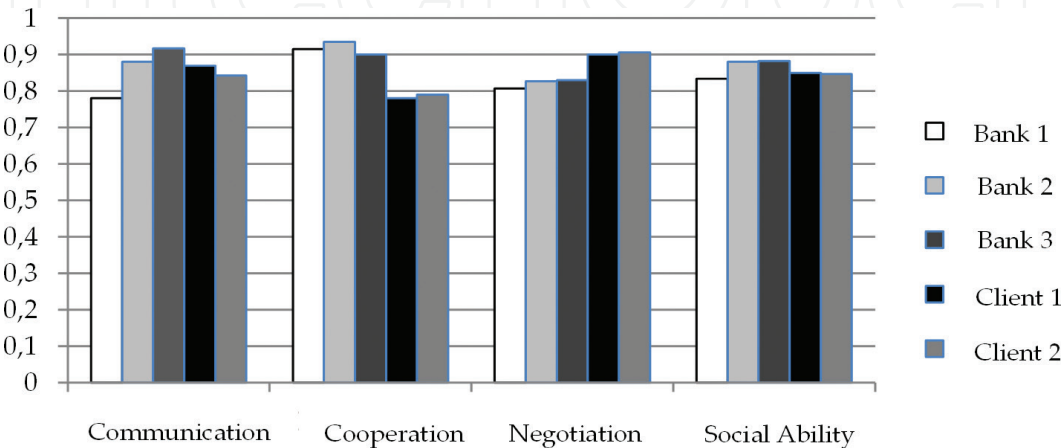

Figure 5. Average values of social ability measures per agent.

In the second experiment, measures were applied to evaluate the autonomy of each agent in the system. **Figure 7** shows the average results for each measure for each attribute and for each agent present in the system. **Figure 8** presents the value of attributes and autonomy, calculated considering the arithmetic mean of the measurements for each attribute. The attributes influence the autonomy value of each agent, being lower than those of the banking agents and a little greater than the one of the client agents because the banking agents have greater capacity of evolution and functional independence. The average autonomy is 80%.

Figure 9 shows the comparison of the results of the average obtained by the system and the experts, applying the Kolmogorov-Smirnov test. Regarding the evaluation of experts on the measures of autonomy of the agents of the system, the arithmetic mean of the opinions of the experts and the arithmetic mean of the measures were considered, obtaining a level of significance of 0.75, which allows to reject the hypothesis that both samples come from different distributions.
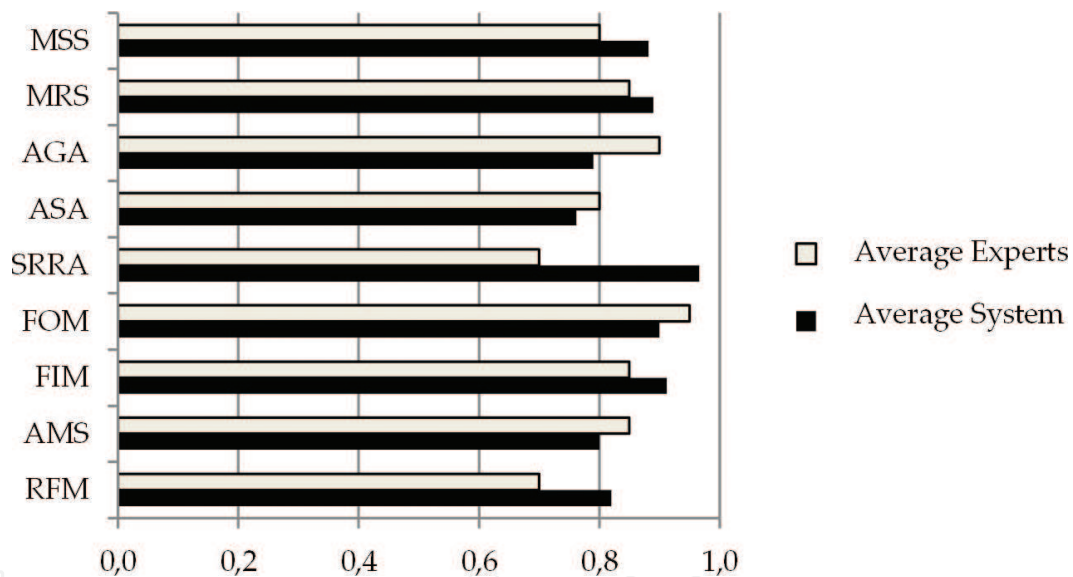

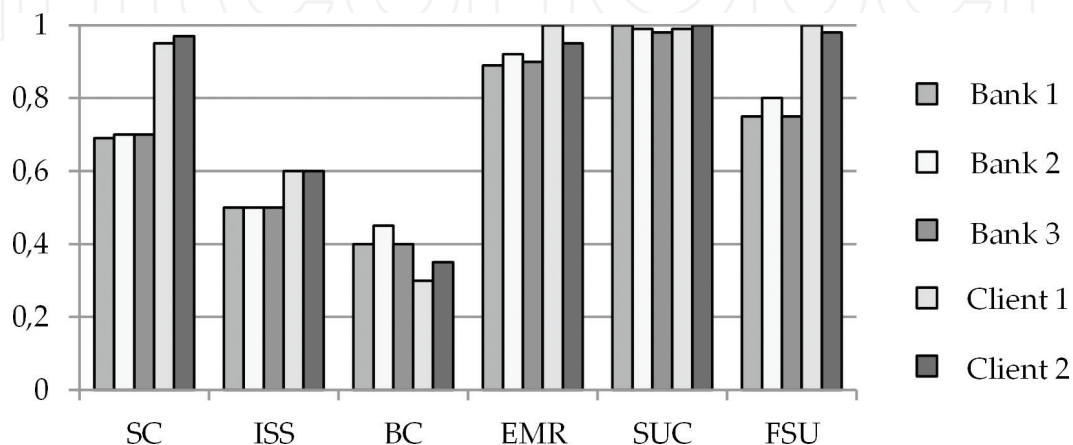**Figure 6.** Average values of the experts and the system.

Figure 7. Average values of the measures of the attributes of the autonomy per agent.
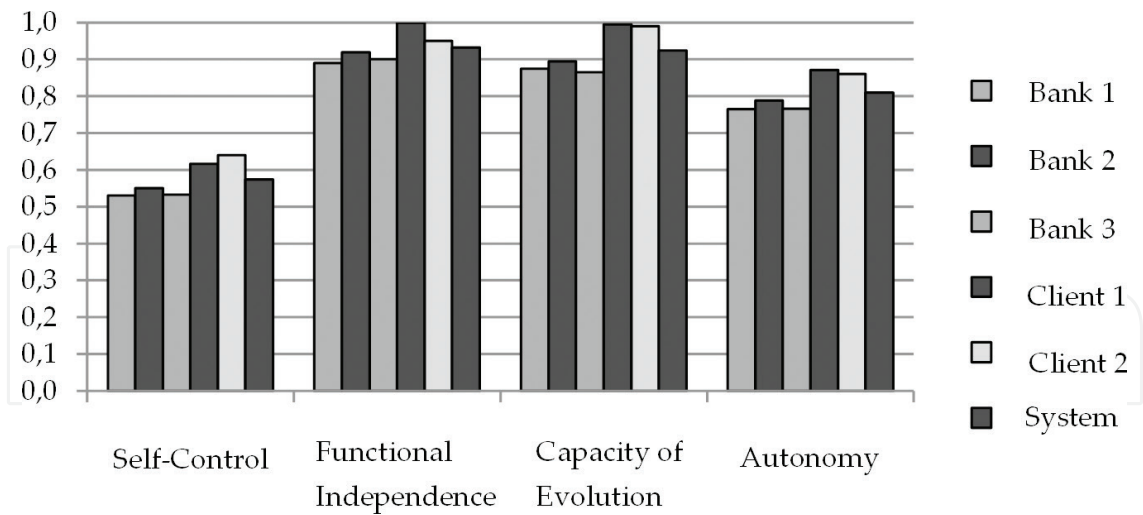
Figure 8. Average values of autonomy measures per agent.
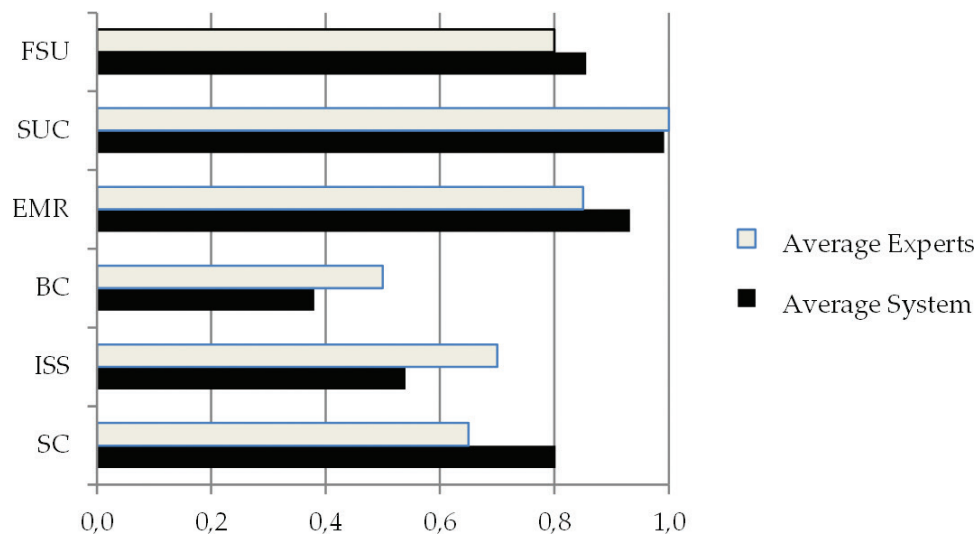

Figure 9. Average values of the experts and the system.

In the third experiment, measures were applied to evaluate the proactivity of each agent in the system. **Figure 10** shows the average results for each measure for each attribute and for each agent present in the system. **Figure 11** shows the value of the attributes and proactivity calculated considering the arithmetic mean of the measurements for each attribute. There are high values of initiative of the banking agents (on average it is 97%) followed by the other two attributes: interaction and reaction (95% on average). Finally, the proactivity of the system of the system is very high, with 96% on average, which is explained by the high interaction of the agents of the system.

Figure 12 shows the comparison of the results of the average obtained by the system and the experts, applying the Kolmogorov-Smirnov test. Regarding the evaluation of experts on the measures of proactivity of the agents of the system, the arithmetic mean of the opinions of the experts and the arithmetic mean of the measures were considered, obtaining a level of significance of 0.45, which allows to reject the hypothesis that both samples come from different distributions.
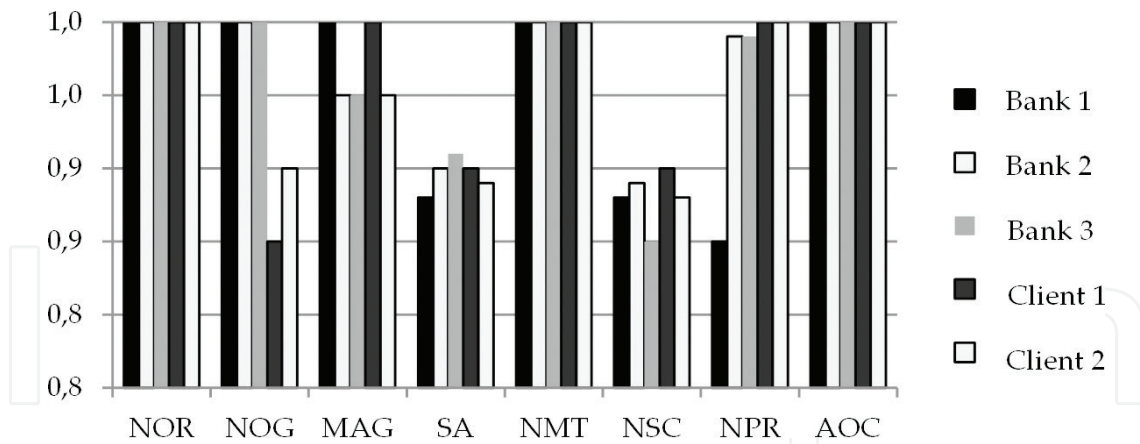
Figure 10. Average values of the measures of the attributes of the proactivity per agent.
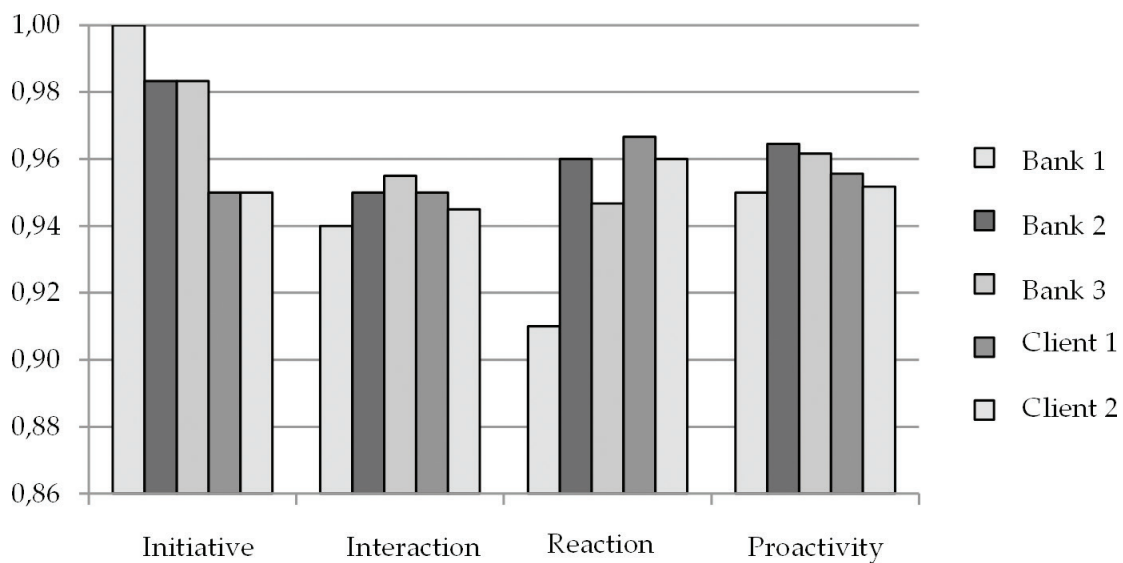

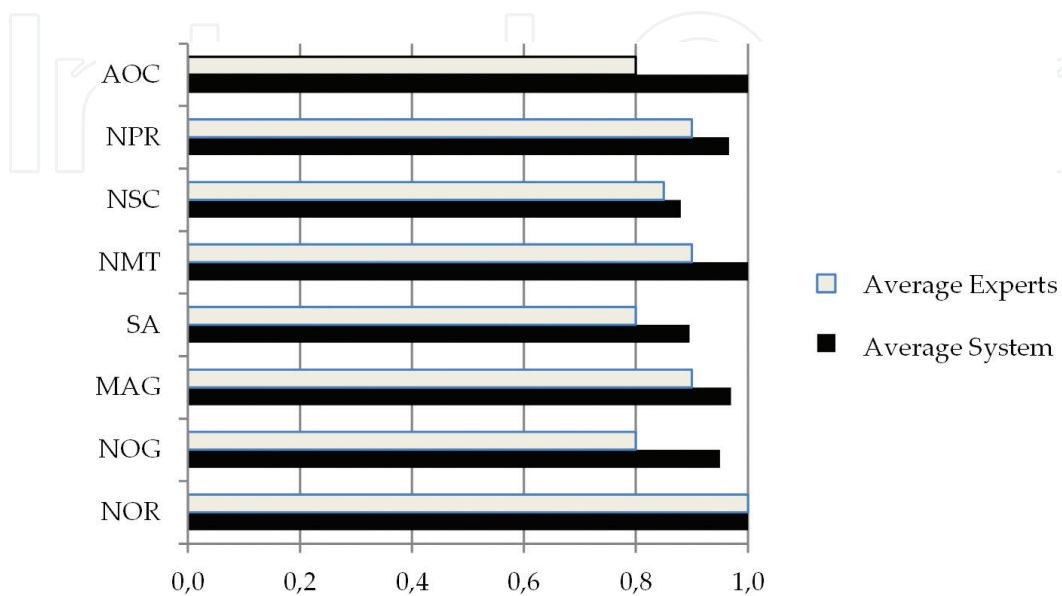**Figure 11.** Average values of proactivity measures per agent.

Figure 12. Average values of the experts and the system.

6. Conclusions

From the study of software agent technology, different characteristics present in this type of software were defined. Considering these characteristics, the work of studying three of them was addressed: social ability, autonomy and proactivity. These, in turn, were divided into nine attributes (three for each characteristic), to consider different factors that represent important aspects of each characteristic, susceptible to being evaluated.

For these attributes, 23 measures were designed for the evaluation of these attributes, and they can be used in an agent-oriented application to obtain a consistent evaluation of the characteristics studied.

An application was also developed [26, 27], making use of the Eclipse platform [29] and object orientation in Java, which allowed to implement a form of automatic evaluation of the proposed measures in this work, on different agent-oriented applications, and that shows that the measures developed are automatable. The use of this tool also provided the possibility of refining the measures to study its behavior in greater detail. In addition, the application allowed to modify the values of the weights associated with the measures to determine the most appropriate for each situation under study.

This work constitutes an advance of the research carried out to reach a quality model in the agent-oriented paradigm, which allows developers to have help with which to satisfactorily evaluate their work, in order to ensure that it is appropriate for the functions for which it was produced, considering the characteristics that each component of the application must fulfill.

Author details

Héctor Soza

Address all correspondence to: hsoza@ucn.cl

School of Engineering, Catholic University of the North, Coquimbo, Chile

References

[1] McCall JA. An introduction to software quality metrics. In: Cooper JD, Fisher MJ, editors. Software Quality Management. New York: Petrocelli Books; 2009

[2] Fuertes JL. Modelo de Calidad para el Software Orientado a Objetos. In: [PhD Thesis]. Facultad de Informática. Madrid: Universidad Politécnica de Madrid; 2003

[3] ISO/IEC 9126-1:2001 Software Engineering-Product Quality–Part 1: Quality Model. International Organization for Standardization; 2001

[4] Dumke R, Koeppe R, Wille C. Software Agent Measurement and Self-Measuring Agent-Based Systems. In: Fakultät für Informatik. Otto-von-Guericke-Universität; 2000

[5] Far B, Wanyama T. Metrics for agent-based software development. Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering. CCECE 2003. pp. 1297-1300

[6] Wooldridge M. An Introduction to Multiagent Systems. Chichester: John Wiley Ltd. Ltd; 2002

[7] Mas A. Agentes Software y Sistemas Multi-Agente: Conceptos, Arquitecturas y Aplicaciones. Madrid: Pearson Educación; 2005

[8] Wooldridge M, Jennings N. Intelligent agents: Theory and practice. The Knowledge Engineering Review. 1995;**10**(2):115-152

[9] Duncan I, Storer T. Agent testing in an ambient world. Proceedings of the Pervasive 2006 Workshop, Dublin, Eire. May, 2006. pp. 757-764

[10] Alonso F, Fuertes JL, Martínez L, Soza H. Measuring the pro-activity of software agents. In: Proceedings of the Fifth International Conference on Software Engineering Advances (ICSEA 2010). France: Nice; August 2010. pp. 319-324

[11] Soza H. Medidas de Calidad para Software Orientado a Agentes. In: Tesis para optar al grado de doctor. Facultad de Informática. Madrid, España: Universidad Politécnica de Madrid; 2013

[12] Bonifacio M, Bouquet P, Ferrario R, Ponte D. Towards a model of goal autonomous agents. Proceedings of the Fourth International Bi-Conference Workshop on Agent-Oriented Information Systems. AOIS '02, 59, pp. 27-28. 2002

[13] Alonso F, Fuertes JL, Martínez L, Soza H. Towards a set of measures for evaluating software agent autonomy. Proceedings of the 8th Mexican International Conference on Artificial Intelligence (MICAI 2009). Guanajuato, México. 2009. pp. 73-78

[14] Chao K-M, James A, Norman P. A framework for intelligent agents within effective concurrent design. Proceedings of the sixth international conference on computer supported cooperative work in design. IEEE. 2001. pp. 338-341

[15] Cernuzzi L, Rossi G. On the evaluation of agent oriented methodologies. Proceedings of the OOPSLA 02 - Workshop on Agent-Oriented Methodologies, Seattle, USA. November, 2002. pp. 21-30

[16] Shin K. Software agents metrics. A preliminary study and development of a metric analyzer. In: Project Report N° H98010. Dept. Computer Science. School of Computing. National University of Singapore; 2003/2004

[17] Huber M. Agent autonomy: Social integrity and social independence. Proceedings of the International Conference on Information Technology ITNG'07, Las Vegas, Nevada. IEEE Computer Society, Los Alamitos, CA, USA. 2007. pp. 282-290

[18] Alonso F, Fuertes JL, Martínez L, Soza H. Measuring the social ability of software agents. Proceedings of the 6th ACIS International Conference on Software Engineering. Research, Management & Applications (SERA 2008). IEEE Computer Society. August, 2008. pp. 3-10

[19] Murdock J. Model-based reflection for agent evolution. In: Technical Report GIT-CC-00-34. Doctoral Thesis. Atlanta: Georgia Institute of Technology; 2000

[20] Friedman B, Nissenbaum H. Software agents and user autonomy. Proceedings of the First International Conference on Autonomous Agents, 1997. pp. 466-469

[21] Rousseau D, Moulin B. Mixed initiative in interactions between software agents. In: Proceedings of the 1997 Spring Symposium on Computer Models for Mixed Initiative Interaction. Portland, Oregon, USA: AAAI Press; March, 1997

[22] Covey S. The Seven Habits of Highly Effective People. London: Simon & Schuster UK Ltd; 1989

[23] Orro A, Saba M, Vargiu E. Using a personalized, adaptive and cooperative multi agent system to predict protein secondary structure. Proceedings of the First International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics, BIOMED'05, Utrecht, The Netherlands, pp. 170-183. 2005

[24] Fuertes JL. Modelo de Calidad para el Software Orientado a Objetos. In: Tesis para optar al grado de doctor. Facultad de Informática. Madrid, España: Universidad Politécnica de Madrid; 2003

[25] Saaty T. How to make a decision: The analytic hierarchy process. European Journal of Operational Research. 1990;**48**:9-26

[26] Villasante O. Evaluación de Calidad de Sistemas Multiagente: una aplicación práctica a un sistema bancario. In: Memoria de Ingeniero Informático, Facultad de Informática. Universidad Politécnica de Madrid; 2011

[27] Villasante O. Evaluación de Calidad de Sistemas Multi-Agente para entornos JADE. In: Tesis de Master en Informática, Facultad de Informática. Universidad Politécnica de Madrid; 2011

[28] Bellifemine F, Poggi A, Rimassa G. JADE: a FIPA2000 compliant agent development environment. In: Proceedings of the Fifth International Conference on Autonomous Agents. ACM; 2001. pp. 216-217

[29] Eclipse Platform Technical Review, Object Technology International Inc. 2003

[30] Chakravart IM, Laha R, Roy J. Handbook of Methods of Applied Statistics, Volume I. John Wiley; 1967