Measures for Evaluating the Software Agent Pro-Activity

Fernando Alonso[1], José L. Fuertes[1], Loïc Martínez[1], and Héctor Soza[2]

[1] Facultad de Informática
Universidad Politécnica de Madrid, Madrid – Spain
{falonso, jfuertes, loic}@fi.upm.es

[2] Escuela de Ingeniería
Universidad Católica del Norte, Coquimbo – Chile
hsoza@ucn.cl

Abstract. This paper is part of research aimed at determining and evaluating software agent quality considering an agent's distinctive characteristics, like social ability, autonomy, pro-activity, etc. We present a study of the pro-activity characteristic, regarded as the software agent's goal-driven behavioral ability to take the initiative and satisfy its design goals. We establish attributes associated with this characteristic and set out measures enabling its global evaluation.

Keywords: Agents quality, pro-activity, software quality.

1 Introduction

Few studies in the literature focus on the development of measures to evaluate the software agent. Those that do exist are generally measures borrowed from the procedural and object-oriented paradigms, and there are few measures created ex professo to evaluate particular characteristics of software agents [1], [2], [3].

Agent pro-activity is one of the more relevant characteristics and defines an agent's ability to exhibit goal-directed behavior by taking the initiative to achieve its goals [4]. It also refers to an agent's ability to take the initiative rather than acting simply in response to its dynamic and unpredictable environment [3] or agents being able to act in anticipation of future goals by taking the initiative [5].

Several studies have analyzed software agent pro-activity, but they are not very related to measures for evaluating this feature. For example, Cernuzzi and Rossi [5] proposed a framework for analyzing and evaluating agent-oriented analysis and design modeling methods, considering qualitative evaluation criteria employing quantitative methods and defining an attribute tree considering internal attributes. Shin [3] presented a project report with the results of the adaptation of some procedural and object-oriented paradigms product measurements to agent-oriented software. None of the above studies provides specific quality measures for evaluating the pro-activity characteristic of agent software. This research aims to advance in the measurement of this feature.

2 Measures for Pro-activity Attributes

From the existing research [4], [6] and based on our experience [7], [8], we propose the following attributes to identify agent pro-activity: **initiative** (an agent's ability to satisfy its design goals through a goal-directed behavior [4], and to take an action with the aim of achieving its goal [9], [10]), **interaction** (an agent's ability to interact with other agents and its environment [9]) and **reaction** (an agent's ability to react to a stimulus from the underlying environment according to stimulus/response behavior, depending on the current state of the software agent [11]).

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 and 1 is a good result for the measure).

Figure 1 shows the types of formula used to normalize the pro-activity measures.

$$\text{(a)}\begin{cases}1 & 0 \le x \le k\\ e^{-\frac{(x-k)^2}{k^2}} & x > k\end{cases} \quad \text{(b)}\begin{cases}\frac{2x}{k}-(\frac{x}{k})^2 & 0 \le x \le k\\ 1 & x > k\end{cases} \quad \text{(c)}\begin{cases}1-\frac{x}{p} & 0 \le x \le p\end{cases} \quad \text{(d)}\ \log_{k+1}(x+1) \quad 0 \le x \le k$$

Fig. 1. Formula types used to normalize the measures

The constant *k* is a parameter that the software engineer can configure to fine tune formula performance for each particular case. The formulae depend on the argument *x*, where *x* is a value defined for each measure. Next, we present the proposed measures for evaluating the attributes defined for the characteristic of pro-activity.

The **initiative** attribute can be measured using the following measures:

- *Number of roles* measures the number of potential roles that agents are to perform. Agent roles are defined in the system design phase [3]. This measure uses curve (a) in Fig. 1, where *x* is the number of agent roles.
- *Number of goals* measures the number of goals achieved by the agent during execution with respect to the number of allocated goals. This measure uses curve (d) in Fig. 1, where *x* is the number of goals achieved by the agent during execution and *k* is the number of goals to be achieved by the agent.
- *Messages to achieve goals* measures agent initiative to achieve its goals by communicating with other system agents. This measure uses curve (b) in Fig. 1, where *x* is the average percentage of executive messages that are sent during agent execution for the goals to be accomplished.

The **interaction** attribute can be measured using the following measures:

- *Services per agent* measures the impact on agent interaction of the number of the services implemented within the agent (not including internal services) enabling it to achieve its goals [3]. This measure uses curve (c) in Fig. 1, where *x* is the number of services implemented within the agent.
- *Number of message types* measures the impact on agent interaction of the number of different types of messages that the agent can process. This measure uses curve (b) in Fig. 1, where *x* is the total of unique incoming and outgoing message types. We comply with FIPA standards for the agent message type [12].

The **reaction** attribute can be measured using the following measures:

- *Number of processed requests* measures the agent's ability to react to the number of received and resolved requests during execution. This measure uses the curve

(a) in Fig. 1, where *x* is the number of received requests (requiring an action to be taken in response) during execution.

- *Agent operations complexity* measures the mean complexity of the operations to be performed by the agent to achieve its goals. The software engineer could use any complexity measure regarded as suitable for achieving a good result (for example, cyclomatic complexity [13]). This measure uses the curve (a) of Fig. 1, where *x* is the mean complexity per goal.

3 Case Study

As an application of this research we have conducted a pro-activity study on the agents of an intelligent agent marketplace. This marketplace includes several kinds of Buyer and Seller agents that cooperate and compete to process sales transactions for their owners. In this system, a Facilitator agent acts as a manager for the marketplace [14]. The names “basic”, “better” and “best” refer to the global evaluation of the software agent strategy to perform its tasks, not to their pro-activeness.

Table 1. Pro-activity attribute values

	Basic Buyer	Better Buyer	Best Buyer	Basic Seller	Better Seller	Best Seller	System
Initiative	0.81	0.88	0.78	0.81	0.79	0.85	0.82
Interaction	0.88	0.92	0.95	0.92	0.88	0.86	0.90
Reaction	0.93	0.97	1.00	1.00	1.00	1.00	0.98
Pro-Activity	0.88	0.93	0.91	0.91	0.90	0.91	**0.90**

Table 1 shows the values of the measure for each attribute calculated from the associated measures. The last row of Table 1 contains the value of the pro-activity characteristic calculated from the measures for all the attributes. Finally, the last column shows the value of the system measures calculated from the values of the attribute measures for all the agents. The bottom, right-hand cell contains the pro-activity value for the entire system. In this study, the above values are aggregated in each case using the arithmetic mean. The results could be refined using a weighted mean, with weights provided by experts using any existing weighting technique.

We find that Basic Buyer agents are less pro-active (88%) than the Best Buyer (91%) and even less than the Better Buyer (93%) agents, because they have different buying strategies. Also, their ability to react is greater than their ability to interact and their initiative. All Seller agents have almost the same pro-activity value because their strategies for achieving their goals are indistinguishable, although, like the Buyer agents, the evaluated measures indicate that their ability to react is greater than their ability to interact and their initiative.

From the above, we conclude, with respect to the attributes, that the system scores highest on Reaction (98%), followed by Interaction (90%), and Initiative (83%). Initiative is influenced by the fact that the agents do not achieve all their goals through cooperation due to the rules that they each apply to achieve their objectives.

Finally, the system’s pro-activity value is 90%, that is, the pro-activity of the system agents as a whole is quite high.

4 Conclusions and Future Work

We have presented a first approximation to a set of measures of agent-oriented software considering the pro-activity characteristic, which has been decomposed into different attributes, and we show the measures considered for its evaluation.

In the future we intend to conduct a comprehensive study of an agent-based system, analyzing the measures of each characteristic of each agent type existing in the system and their contribution to the measure of system quality. To do this, we propose to build a quality evaluation model, and evaluate this model on several software agent applications, considering the different characteristics, and their attributes, present in agents.

References

1. Dumke, R., Koeppe, R., Wille, C.: Software Agent Measurement and Self-Measuring Agent-Based Systems, Preprint No 11. Fakultät für Informatik, Otto-von-Guericke-Universität, Magdeburg (2000)
2. Far, B., Wanyama, T.: Metrics for Agent-Based Software Development. In: Canadian Conf. on Electrical and Computer Engineering, pp. 1297–1300. Montréal, Canada (2003)
3. Shin, K.: 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. Wooldridge, M.: An Introduction to Multiagent Systems. John Wiley, Chichester (2002)
5. Cernuzzi, L., Rossi, G.: On the Evaluation of Agent Oriented Methodologies. In: OOPSLA 02-Workshop on Agent-Oriented Methodologies, pp. 21–30. Seattle (2002)
6. Covey, S.: The Seven Habits of Highly Effective People, 15^{th} anniversary edition. Free Press, Old Tappan, NJ (2004)
7. Alonso, F., Fuertes, J. L., Martínez, L., Soza, H.: Measuring the Social Ability of Software Agents. In: Sixth International Conference on Software Engineering Research, Management and Applications, SERA 2008, pp. 3–10. Prague, Czech Republic, (2008)
8. Alonso, F., Fuertes, J. L., Martínez, L., Soza H.: Towards a Set of Measures for Evaluating Software Agent Autonomy. In: Seventh Joint Meeting of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2009, Amsterdam, Netherlands (2009)
9. Covey, S.: The Seven Habits of Highly Effective People, 15^{th} anniversary edition. Free Press, Old Tappan, NJ (2004)
10. Rousseau, D., Moulin, B.: Mixed initiative in interactions between software agents. In: 1997 Spring Symposium on Computer Models for Mixed Initiative Interaction. AAAI Press, Menlo Park (1997)
11. Orro, A., Saba, M., Vargiu, E.: Using a Personalized, Adaptive and Cooperative Multi Agent System to Predict Protein Secondary Structure. In: First International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics, BIOMED'05, pp. 170-183, Utrecht, The Netherlands (2005)
12. Foundation for Intelligent Physical Agents: FIPA Communicative Act Library Specification. Geneva, Switzerland (2002)
13. McCabe, T. J.: A Complexity Measure. IEEE T. Software Eng. SE-2, 308–320 (1976)
14. Bigus, J., Bigus, J.: Constructing Intelligent Agents Using Java, 2^{nd} edition. John Wiley & Sons, Inc., New York, NY (2001)