

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Establishing Quality Requirements of Multi-Agent Projects

¹Punam Bedi and ²Vibha Gaur

¹Department of Computer Science, University of Delhi, India

²Department of Computer Science, Acharya Narendra Dev College, University of Delhi, India

Abstract: Agent technologies have been revitalized for engineering complex distributed software systems. Nowadays various projects are being revised and restructured using Multi-agent systems. Developing Multi-agent based projects within the constraints of time and resources without the sacrifice of quality is a challenging task. The determination of quality needs of a project requires profound knowledge of distinct characteristics of that project. Identification of quality requirements support developers in achieving explicit quality needs of the system and users. This study first identifies the quality requirements of MAS projects and then categorizes the various MAS applications based on these requirements. Secondly, an algorithm has been proposed to quantify the quality requirements that would assist quality engineer to decide degree of quality assurance functions to achieve quality goals for MAS.

Key words: Multi-agent system (MAS), quality requirement, quality factor, quality assurance

INTRODUCTION

Software engineering community is becoming increasingly dependent on agent technology for engineering complex distributed systems. Software agents are enmeshed in various systems such as consumer shopping, monitoring complex chemical processes, aircraft maintenance, wireless collaboration, financial portfolio management etc. Software agent is an entity that is capable of carrying out goals autonomously across open and distributed environments and can move throughout a network of agent aware computers (Luiqi and Wictor, 2001). These are characterized as autonomous, goal-oriented, situation aware and proactive as well as reactive.

Multi-agent system consists of an environment populated with a set of autonomous entities (software agents) that cooperate to solve a complex problem in decentralized way. MAS models problems in terms of autonomous interacting component-agents, which is proving to be a more natural way of representing task allocation, team planning, user preferences, open environments and so on. MASs have been applied in a variety of domains. Bunch *et al.* (2004) has presented the application of MAS to monitor complex chemical processes. An application of MAS to a service mobile manipulator robot that interacts with a human during an object delivery and hand-over task has been presented (Erden *et al.*, 2004). Guiagoussou and Said (1999) present the implementation of a maintenance application for cellular switching system using the Multi-agent paradigm.

MASs pose challenges for software engineers, as the quality in MAS is not straightforward and there is a pressing need for software-engineering techniques that allow quality of MAS to be managed effectively.

To attain the desired quality of MAS, it is necessary to identify quality specifications that would assist the developers and users for the selection of appropriate measures during the evaluation of system. The determination of quality requirements requires the knowledge of the specific characteristics of the project and this could be achieved by combining the experience and knowledge of various stakeholders involved in the project. This study identifies the quality requirements of MAS projects and categorizes the various MAS projects based on these quality requirements. An algorithm has been presented to quantify the requirements of MAS projects. Quantification of quality requirements would assist the quality engineers to choose the degree of quality assurance function for achieving the quality of MAS.

QUALITY REQUIREMENTS IN MAS PROJECTS

Quality in software system is defined as the degree to which a system, a component or process conforms to specified requirement (David *et al.*, 1990) or fitness for use. ISO Standard 8402 defines quality as; The totality of features and characteristics of a product or service that bears on its ability to satisfy stated or implied

needs. A quality based product results in better user satisfaction. Conventional software quality models such as McCall *et al.* (1977) and ISO 9126 (based on 6 factors: functionality, reliability, usability, maintainability, efficiency and portability) addresses quality of software in terms of factors and criteria. Researchers in software engineering have tried different ways of defining quality. They have adopted product-based view (Boehm *et al.*, 1976; McCall *et al.*, 1977; Kitchenham and Walker, 1986), the manufacturing view (Humphrey, 1988; Paulk *et al.*, 1991; Dowson, 1993), the user based view (Vidgen, 1996; Kitchenham, 1987a) and even combinations of the views (Kitchenham, 1987b).

Far (2002) defines quality in MAS from various viewpoints such as:

- **Conformance:** Conformance to customer's requirements; conformance to standards;
- **Development process quality:** Requirement, design, implementation, test and maintenance quality;
- **End-product quality:** Reliability, usability and availability;
- **Relativity:** Advantage over similar products;

Lee *et al.* (1998) have defined non-functional properties of MASs which include performance, scalability and stability. There have been many quality studies of the views, but none has studied quality requirements from the distinct characteristics of projects.

Quality modeling approaches provide an environment within which stakeholders and analysts can easily cooperate to discover, organize, reconcile and validate the quality requirements. But no single set of quality requirements would be appropriate for all the MAS projects. The set of requirements that would be appropriate for one type of projects may likely be as overkill for another type of MAS projects. Therefore an efficient software process should define a collection of quality requirements; each designed to meet the needs of different types of projects that enforce high quality of MAS.

Quality in MAS can be addressed in terms of non-quantifiable factors that can further broken down into quantifiable criteria. The factors of quality for different types of MAS projects and their respective criteria of evaluation are listed below (David *et al.*, 1990; Lee *et al.*, 1998; Far, 2002; Boehm *et al.*, 1976; McCall *et al.*, 1977; Bashir *et al.*, 2001; Patrick, 2002; Camacho *et al.*, 2001).

Knowledge acquisition: This is the extent to which MAS acquires the knowledge from the user about his habits, preferences so that it could achieve the goal efficiently as per user's preferences and also, when user uses the MAS

next time, it need not to ask the information that has already been provided. Acquisition of knowledge can be assessed by following criteria:

- Effectiveness of interaction between MAS and user;
- Duration of the communication;
- Amount of information provided to the MAS;
- Retention time;

Performance: This is concerned with how much time will be taken by MAS in providing the satisfactory solutions to user within resource constraints. This can be judged by evaluating the MAS for the following criteria:

- Time to generate the first solution;
- Time to generate all the possible solutions;
- Time to generate the relevant solutions;
- Number of successful solutions;
- Effective resource management;
- Consistency in response time;

Fulfillment: A clear and traceable process for accomplishing the goal would allow users to track the status of action they requested. Criteria that are required to review the MAS from task accomplishment viewpoint are as follows:

- Forward and backward traceability of the process-accomplishing task;
- Achievement of goal;
- Amount of information revealed to user while achieving goal;

Security and protection: This is the characteristic of MAS that takes care of authentication of parties involved in system, information access control, information change control procedures and physical security. The following criteria can be used to assess the security and protection of MAS:

- **Authentication:** The process of verifying an identity by or for a system entity;
- **Access control:** Protection of agent's code or protection of system against its malicious behavior;
- **Audit trail:** A chronological record of system activities that is sufficient to enable the recovery of system once it gets crashed or damaged;
- **Confidentiality:** The characteristics of the system that information would not be available to unauthorized persons, processes or entities;
- **Integrity:** The property of the system that information of the system would not be modified or destroyed in an unauthorized manner;

- **Availability:** The characteristics of the system being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system;
- **Nonrepudiation:** A security service that provides protection against false denial of involvement in a communication;

Informative: As value of information is related with quality and accuracy of MAS, it must be assessed for precision and reliability, which in turn depends on the source of information itself. The characteristics of the information brought by software agents can be ascertained for the following criteria:

- Certification and reliability of source of information by looking at icons or text representing seals of approval;
- Conciseness and consistency of information;
- Completeness and relevance of information;

Ease of communication: It perceives the MAS with respect to the utilization and appearance of system by users. The following criteria examine the ease of communication in MAS:

- **Navigation:** The ease of finding things, which result from clear, logical presentation and consistent design.
- **Presentation:** The characteristic of the interface that results from clean, functional layout and a professional presentation. The agent interface should appear professional and official like money or certificates.
- **Ease to use and learn:** Interface that results from comfortable system dialogues is easy to use and learn. Interface must be flexible enough to provide adequate information to users in accomplishing their tasks.

CATEGORIZATION OF MAS PROJECTS

MAS have been designed to accommodate different types of projects. A monolithic approach for dealing with all MAS projects is quite pretentious. Rather, MAS features depend on the specific requirements each application gives more emphasis to (Caridi and Cavalieri, 2004). For this reason, literature reports several proposals of classification of MAS, in the attempt to make a clarification on the different definitions researchers and practitioners have so far provided. The taxonomies of MAS found in literature are based on design

specifications (Decker, 1987) and industrial applications (Jennings and Wooldridge, 1998).

Although it is difficult to develop a comprehensive and clear taxonomy, this study proposes the following category of MAS based projects:

- Simulations-oriented projects are undertaken in which software agents are capable of acting on behalf of its users in order to accomplish tasks. Agents are encapsulated with the expert knowledge in some specific application domain. Once the user delivers ample of information to MAS for achieving the goal, agents act autonomously on behalf of user. These agents simulate user to perform the tasks such as automating matchmaking in a stable two sided B2B market (Ouksel *et al.*, 2004), casting vote (Buhler, 2001), representing buyer or sellers in e-commerce transactions (Maes *et al.*, 1999), automating bid-evaluation process (Mobasher, 1999) etc.
- Service-oriented projects occur to support intelligent techniques for dealing with complex and real world problems. The software agents in service-oriented projects don't simulate human relationships but they assist them in carrying out varying tasks. Mangina (2001) has presented electrical plant condition monitoring that is a Multi-agent based paradigm. Interface system for an intelligent wheelchair has been developed using MAS by Liu and Lin (2000) that provides adjustable interface to the needs of different users with different levels of disabilities in a hospital. The framework of Multi-agent intelligent elevator system has been described by Kung *et al.* (2003).

Along with industrial applications, software agents have also being used as an automation tool in support of software engineering activities. e.g., An MAS in support of software quality assurance for improving the monitoring and controlling of SQA activities has been presented by Leung and Poon (1999). Application maintenance using software agents that work together by automatically informing the appropriate individuals of any changes in open-source Internet software application has been specified by Sharma and Capretz (2001).

- Information-oriented projects occur to exploit Internet search engines such as WebCrawlers, Lycos and Spiders. Essentially, software agents help manage the vast amount of information in wide area network like the Internet and also referred as Information Management System. Junping *et al.* (2005) has described agent based information

management system to meet the demands of different tourism oriented information management systems. Another example of information-oriented system is Recommender systems that attempt to emulate by providing users with knowledgeable suggestions to facilitate their decision making process in some specific domain. An agent based recommender system for supporting communities of people in searching the web by means of a popular search engine using data mining techniques has been presented by Birukov (2005).

EVALUATING QUALITY REQUIREMENTS OF MAS PROJECTS

In literature, although MAS has been categorized on the basis of design specifications (Decker, 1987) and industrial applications (Jennings and Wooldridge, 1998), but none has classified MAS on the basis of quality requirements. E.g. knowledge acquisition about user’s profile is rigorously required in simulation-oriented projects than information-oriented projects while it is least required for service-oriented projects. Similarly informative characteristics are highly demanded in information-oriented projects than service-oriented projects. Achieving quality in MAS projects require extreme knowledge about the specific characteristics of project.

Six adaptive quality factors have been chosen for MAS projects (section 2). Stakeholders on the basis of their concern assign grades to adaptive quality factors, ranging between 1 and 5, where 1 represents a little concern and 5 represents strong interest of stakeholders in that adaptive quality factor. Then average of grades would be used for the computation of quality requirements. Quality experts develop a systematic approach for selecting weights that were assigned to adaptive quality factors depending on the appropriateness of factor to particular projects. The value of the weights for adaptive quality factors range between 0 and 1 and provides an indication of the relative importance of a particular adaptation factor to the types of MAS projects.

To evaluate the quality requirements for a project, following steps would be followed:

- Analyze each of the adaptive quality factors listed in section 2 and assign the grades (1 to 5) based on the stakeholders’ interest.
- Obtain weighing factors assigned to each of factor from quality experts based on relevance of that factor to type of project (Table 1).

- Multiply the grades entered in Table 1 by the weights. The result of the product: grade x weight is placed in the product column of the Table 1 for each adaptive quality factor.
- Compute the average of all the entries in the product column and place the result provided in the space marked Quality Requirement.

The value of quality requirement would range between 0 and 5 for MAS projects, where 0 signifies less rigid requirements and 5 signifies stringent quality requirements.

The value obtained for quality requirement would help designers and developers to design quality assurance function to achieve quality goals of the project. Larger value of quality requirement for MAS projects would require strong emphasis on quality assurance functions than those with small values.

This value would also be used at the time of evaluation by experts and users for ensuring that quality goals have been achieved and if yes, then to what extent.

Case study: Davis (1982) has described four strategies that can be followed to identify and specify the quality requirements namely asking, comparing, analysis and trial and error.

We used questionnaire-based interview to study the quality requirements of Multi-agent based recommender system. The interview was held with 7 project managers, 5 developers and 27 users who were familiar with the operations and the working of recommender system. The interview aimed to focus on the respondents’ perception of quality factors of MAS projects and results of the interview has been presented in Table 2 and has been depicted by bar chart in Fig. 1. The values of grades obtained from participants to quality factors have been substituted in Table 1 to compute the quality requirements for recommender System and Table 2 displays the value of quality requirements for Recommender system.

The value 1.95 signifies that even moderate quality assurance functions would be adequate to achieve quality goal for recommender system.

Table 1: Computing the minimum quality requirement for a MAS project

Adaptive quality factor	Weight multiplier				Product
	Grade	Simulated	Service	Information	
Knowledge acquisition	-	1.0	0.6	0.1	-
Performance	-	0.6	0.9	0.9	-
Security and protection	-	1.0	0.8	0.6	-
Fulfillment	-	1.0	0.7	0.5	-
Informative	-	0.8	0.8	1.0	-
Ease of communication	-	1.0	0.8	0.5	-
Quality Requirement:					-

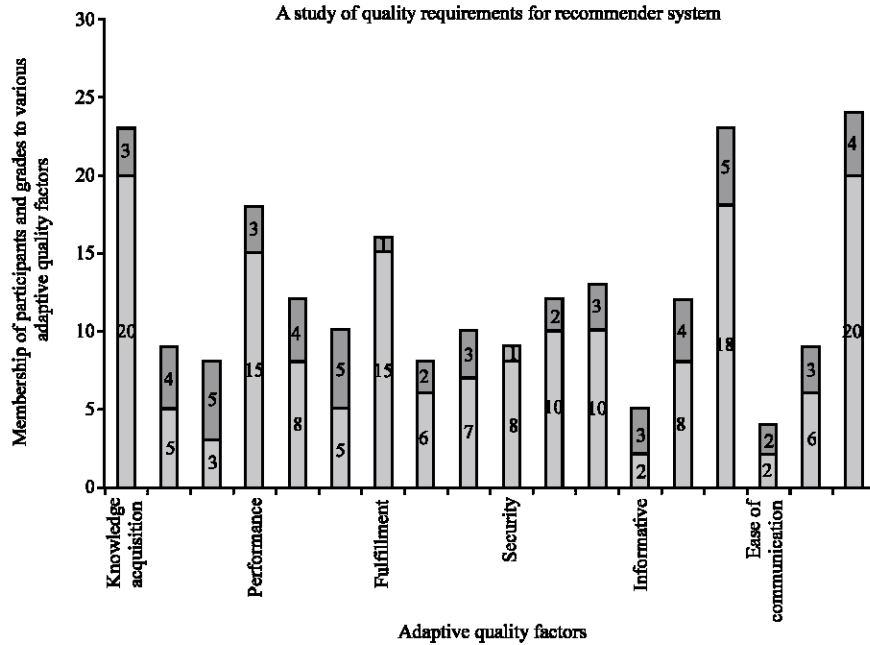


Fig. 1: The grading to adaptive quality factors by participants

Table 2: Grading assigned by subjects to the various quality attributes for travel recommender system

Adaptive quality factor	Grade to quality factors	No. of participants
Knowledge acquisition	3	20
	4	5
	5	3
Performance	3	15
	4	8
	5	5
Fulfillment	1	15
	2	6
	3	7
Security	1	8
	2	10
	3	10
Informative	3	2
	4	8
	5	18
Ease of communication	2	2
	3	6
	4	20

Table 3 displays the value of quality requirements for recommender system.

The value 1.95 signifies that even moderate quality assurance functions would be adequate to achieve quality goal for recommender system.

CONCLUSIONS

Agent applications can be mainly divided into three categories: Simulation-oriented, Service-oriented and Information-oriented. Since quality is the prime thing demanded by all the disciplines involved for all the types of projects. Different perceptions of quality for distinct MAS projects have been presented in this study. The quality requirements of different applications of MASs based on the specific characteristics of the project have been identified and quantified. The quantification of quality requirements would be used to decide the degree of severity of quality assurance functions and hence support the developers and users to achieve explicit quality goals of the projects.

Appendix: Study Questions

Knowledge acquisition

- I believe that gathering of information about user profile would affect the quality of MAS.
- I feel that comfortable system dialogues would assist MAS to acquire effective knowledge from the user.
- I expect that the amount of information provided to MAS would help achieving the quality of MAS.

The values of grades obtained from participants to quality factors have been substituted in Table 1 to compute the quality requirements for recommender system and

Performance

- The promptness of the generating solutions is required for the success of MAS.
- The number of successful solutions provided by MAS would help achieving the quality goal.
- Predictable response time will affect the quality of MAS.

Fulfillment

- MAS should reveal all the intermediate information of task accomplishment to users while achieving the goal.
- The process of goal achievement should be traceable for achieving the quality goal of the MAS.

Security

- An MAS should enforce the mechanisms for verifying the entities using the system.
- An MAS should block the modification of information by unauthorized means.
- Availability of MAS is required for achieving quality goal.

Informative

- The brand of the source of information affects the quality and reliability of MAS.
- Accurate, certified, complete information provided by MAS is essential for producing quality-based system.

Ease of communication

- I believe that ease of finding things in the MAS would help achieving the quality for MAS.
- I feel that professional presentation would assist in building the MAS a success.

REFERENCES

Bashir, I., E. Serafini and K. Wall, 2001. Securing Network Software Applications. *Commun. The ACM*, 44: 28-30.

Birukov, A., E. Blanzieri and P. Giorgini, 2005. An Agent-based Recommendation System for Web Search. In: *AAMAS '05*, ACM Press, pp: 618-624.

Boehm, B., 1989. Software Risk Management, In: *Proceedings ESEC*, Warwick, UK., pp: 1-19.

Boehm, B., J. Brown and M. Lipow, 1976. Quantitative evaluation of software quality. *Proceedings of the 2nd International Conference on Software Engineering*, pp: 592-605.

Buhler, P.A. and M.N. Huhns, 2001. Trust and persistence. *IEEE Internet Comput.*, 5: 90-92.

Bunch, L., M. Breedy, J.M. Bradshaw, M. Carvalho, N. Suri, A. Uszok, J. Hansen, M. Pechoucek and V. Marik, 2004. Agents, interactions, mobility and systems (AIMS): Software agents for process monitoring and notification, March 2004. *Proceedings of the 2004 ACM symposium on Applied computing*, Nicosia, Cyprus, pp: 94-100.

Camacho, D., D. Borrajo and J.M. Molina, 2001. Intelligent Travel Planning: A multi-agent planning system to solve-web problems in the e-tourism domain. *Autonomous Agents and Multi-Agent Systems*, 4: 387-392.

Caridi, M. and S. Cavalieri, 2004. Multi-agent systems in production planning and control: An overview, *Prod. Plan. Control*, 15: 106-118.

David, C., W. Sharma and S. Ravi, 1990. Toward a Diagnostics Instrument for Assessing the Quality of Expert Systems, *ACM SIGBDP Conference on Trends and Directions in Expert Systems*, Orlando, Florida, United States, pp: 72-87.

Davis, G.B., 1982. Strategies for information systems requirement determination, *IBM Syst. J.*, 21: 4-30.

Decker, K.S., 1987. Distributed problem solving techniques: A survey. *IEEE Transactions on Systems, Man and Cybernetics*, 17: 103-124.

Dowson, M., 1993. Software Process Themes and Issues, *2nd International Conference on the Software Process: Continuous Software Process Improvement*, Berlin, Germany, pp: 54-60.

Erden, M.S., K. Leblebicioğlu and U. Halici, 2004. Multi-agent system based fuzzy controller design with genetic tuning for a mobile manipulator robot in the hand over task. *J. Intelligent Robotic Systems*, 39: 287-306.

Far, B.H., 2002. Software agents: Quality, complexity and uncertainty issues. *Proceedings of the 1st IEEE International Conference on Cognitive Informatics ICCT'02*, Canada, pp: 122-132.

Guiagoussou, M. and S. Said, 1999. Implementation of a diagnostic and troubleshooting Multi-agent system for cellular networks. *Int. J. Network Manag.*, 9: 221-237.

Humphrey, W., 1988. Characterizing the software process: A maturity framework. *IEEE Software*, 5: 73-99.

Jennings, N.R. and M. Wooldridge, 1998. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1: 7-38.

- Junping, D., G. Wensheng, T. Xuyan, 2005. A multi-mobile agent based information management system. Proceedings of Network Sensing and Control, IEEE., pp: 71-73.
- Kitchenham, B. and J. Walker, 1986. The meaning of quality, software engineering 86: Proceedings of BCS-IEE Software Engineering 86 Conference, Southampton, England.
- Kitchenham, B., 1987. Toward a constructive quality model Part 1: Software quality modeling, measurement and prediction. *Software Eng. J.*, 2: 105-113.
- Kitchenham, B., 1987. Toward a constructive quality model. Part 2: Statistical techniques for modeling software quality in the ESPRIT REQUEST project. *Software Eng. J.*, 2: 114-126.
- Kung, D.C., H. Bhanbhani and S. Nwokoro, 2003. Lessons learned from software engineering multi-agent systems. *COMPSAC'03, IEEE.*, pp: 50-55.
- Lee, L.C., H.S. Nwana and D.T. Ndumu, 1998. The Stability, scalability and performance of multi-agent systems. *BT Technol. J.*, 16: 94-103.
- Leung, H.K.N. and C. Poon, 1999. Multi-agent environment for software quality assurance, AUTOQ. In: The proceedings of 25th EUROMICRO Conference, Milan, Italy, pp: 294-301.
- Liu, A. and F. Lin, 2000. A study in analysis and design of multi-agent systems. 25th Proceedings of EUROMICRO Conference, Milan, Italy, 2: 294-301.
- Luiqi, B. and W.W. Victor, 2001. Establishing quality control in software agents. *ACM SIGAPP Applied Computing Review*, 9: 31-33.
- Maes, P., R.H. Guttman and A.G. Moukas, 1999. Agents that buy and sell. *Commun. ACM.*, 42: 81-ff.
- Mangina, E.E., S.D.J. McArthur and J.R. McDonald, 2001. The Use of a multi-agent paradigm in electrical plant condition monitoring. Large Engineering Systems Conference on Power Engineering, Halifax, NS, Canada, pp: 31-36.
- McCall, J.A., P.K. Richards and G.F. Walters, 1977. Factors in Software Quality. Vol. I, II and III, RADC reports.
- Mobasher, B., R. Sundareswara, M. Tsvetovat, J. Collins and M. Gini, 1999. Search strategies for bid selection in multi-agent contracting. Proceedings of Workshop on Agent-mediated Electronic Commerce (AmEC-99).
- Ouksel, A.M., Y.M. Babad and T. Tesch, 2004. Matchmaking software agents in B2B markets. Proceedings of the 37th Hawaii International Conference on System sciences. (HICSS'04)-Track 7-7: 70166.1.
- Patrick, A.S., 2002. Building trustworthy software agents. *IEEE Internet Computing*, 6: 46-53.
- Paulk, M., W. Curtis and M. Chrissis, 1991. Capability maturity model for software. Report CMU/SEI-91-TR-24. SEI, Carnegie Mellon University.
- Sharma, A. and M.A.M. Capretz, 2001. Application maintenance using software agents. Proceedings of 1st International Workshop on Source Code Analysis and Manipulation, pp: 55-64.
- Vidgen, R., 1996. A multiple perspective approach to information systems quality, Unpublished Ph.D Thesis. University of Salford, Salford.