An Agent-based Quality Assurance Assessment System

Putsadee Pornphol
Department of Computer Science , Faculty of Science and Technology,
Phuket Rajabhat University , 83000, Thailand.
Suphamit Chittayasothorn
Department of Computer Engineering, Faculty of Engineering,
King Mongkut's Institute of Technology, Landkrabang, Bangkok 10520, Thailand.

Abstract— Agent technology is a software technology with high-level abstraction which provides a convenient way to describe software entities in terms of its behavior rather than only refer to attributes and methods. A software agent is autonomous; the agent is capable of operating as a standalone process and performing actions without user intervention. A software agent is communicative; it communicates with the user, other software agents, or other software processes. A software agent is perceptive; it is able to perceive and respond to changes in its environment. With all these characteristics, it is suitable to employ agents in web-based applications that require intelligent decision making. This paper presents an agent-based quality assurance assessment system for educational institution. Agents are used to check essential requirements that educational institution clients have to meet and prepare reports for assessors. The system proves useful and helps reduce assessment time from education expert assessors.

Keywords—Agent Technology, Agent-Based Assessment, Web-Based Quality Assurance, Database System, Self Assessment Report.

I. Introduction

At present, educational institution in Thailand must be involved in quality assurance to guarantee their quality of services. Quality assurance activities include both internal assessment and external or national level assessment. Both levels of quality assurance requires a lot of manpower both from the education institution client side and from the assess body side. The educational institution must satisfy all the measures imposed by the government and provides evidences that support their key performance indexes.

The government has set up a public organization called The Office for National Education Standards and Quality Assessment (ONESQA) [1] to assess educational institutions. Nowadays experts who work for ONESQA must perform the assessment job almost manually. Besides, each client educational institution may have different internal procedures and may have chosen different key performance indexes. It would be desirable if some assessment steps are automated or at least partially automated.

On the client educational institution side, not only the lack of manpower dedicated to quality assurance is a problem, the lack of quality assurance knowledge and understanding is a big problem as well. There has been an attempt to introduce knowledge management for quality assurance to client educational institution [2]. Knowledge obtained by some educational institutions that have already passed the quality assurance process is transferred to other institution via a web-based knowledge management system. The system proved useful and is now extended to allowed mobile agents from the assessors to perform some pre-assess tasks and report the readiness of the client institution to the assessors. Fig.1 shows the original knowledge management system for educational quality assurance.

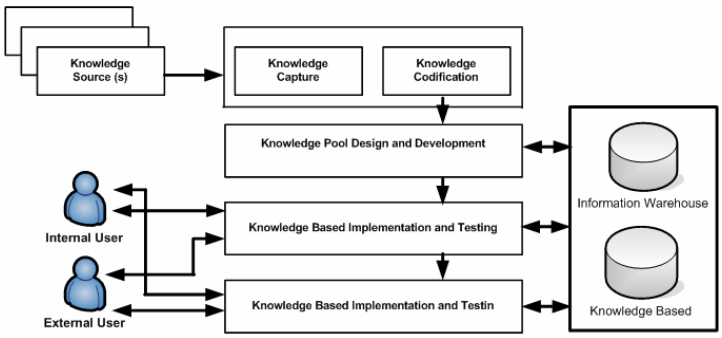

Fig.1 The Knowledge Management System for Higher Education Quality Assurance

II. Agent Technology

The next wave of technological innovation must integrate linked organizations and multiple application platforms. Developers must construct unified information management systems that use the world wide web and advanced software technologies. Software agents, one of the most exciting new developments in computer software technology, can be used to quickly and easily build integrated enterprise systems. The idea of having a software agent that can perform complex tasks on our behalf is intuitively appealing. The natural next step is to use multiple software agents that communicate and cooperate with each other to solve complex problems and

implement complex systems. Software agents provide a powerful new method for implementing these next-generation information systems.

An agent is simply another kind of software abstraction, an abstraction in the same way that methods, functions, and objects are software abstractions. An object is a high-level abstraction that describes methods and attributes of a software component. An agent, however, is an extremely high-level software abstraction which provides a convenient and powerful way to describe a complex software entity. Rather than being defined in terms of methods and attributes, an agent is defined in terms of its behavior. This is important because programming an agent-based system is primarily a matter of specifying agent behavior instead of identifying classes, methods and attributes. It is much easier and more natural to specify behavior than to write code.

There is a minimum set of common features that typify a software agent. A software agent is autonomous; the agent is capable of operating as a standalone process and performing actions without user intervention. A software agent is communicative; it communicates with the user, other software agents, or other software processes. A software agent is perceptive; it is able to perceive and respond to changes in its environment.

Software agents, like people, can possess different levels of competence at performing a particular task. For example, one email software agent might be quite dumb and capable of only forwarding email to a few specified locations. A second, smarter email agent might have the ability to automatically detect and delete spam. While software agents must be autonomous, communicative, and perceptive, they can have different levels of competence (intelligence) as determined by their programs - i.e., their behavioral specifications. [8]

III. The Agent-based Quality Assurance System

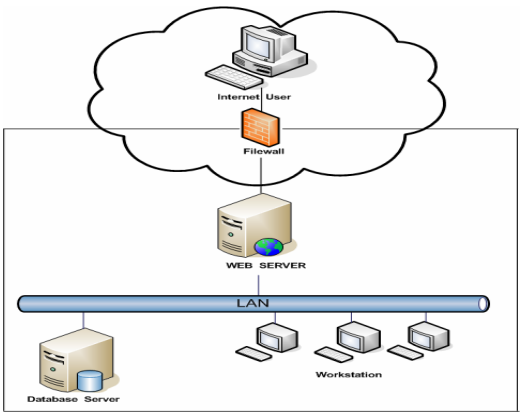

Fig.2 Gross system diagram of quality assurance client Institution

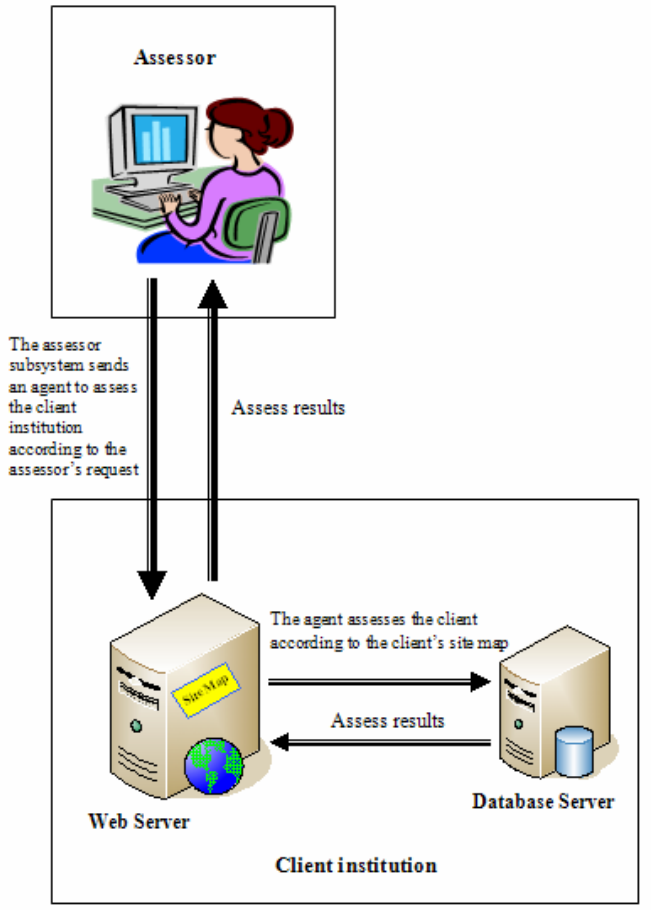

Fig.3 The Agent-based quality assurance assessment system

The system comprises two parts, namely, the client institution subsystem and the assessor subsystem. Fig.2 shows the gross system diagram of the subsystem. The users prepare reports and provide information and documents to the system in the same way as they prepare the Self Assessment Report (SAR). This information is stored in a database server at the client institution. The users at this site must also prepare a site map of the client institution. Since, individual institutions have different ways to organize data, the site map is the essential information required by the assessor agents.

The assessor subsystem employs agents to assess client institution sites. The agent is sent by the assessor to act on her behalf as shown in Fig. 2. It studies the site map of the client site, studies the SAR and other information from the client web site and prepares reports for the assessor.

IV. Framework Architecture

The framework architecture of the two subsystems namely, the client institution subsystem and the assessor subsystem are further discussed in this section.

A. Client Institution Framework Architecture

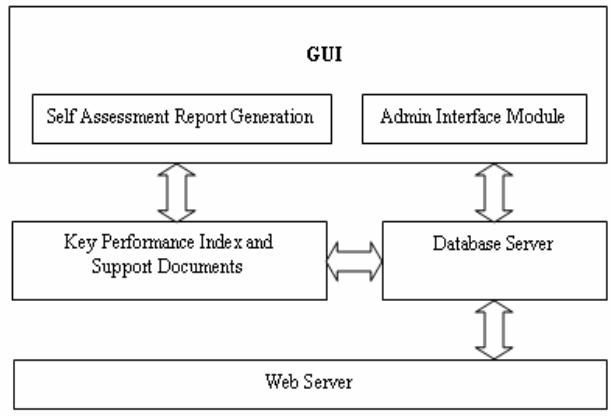

Fig.4 Client Institution System Architecture

The framework architecture of the client institution comprises 4 parts. The first one is the graphical user interface (GUI). The GUI enables the users to prepare SAR reports. The users must specify the KPI for each measure and prepare evidences in the form of documents and data which are collected and stored in the database server. This information will be later studied by the agent from the assessors.

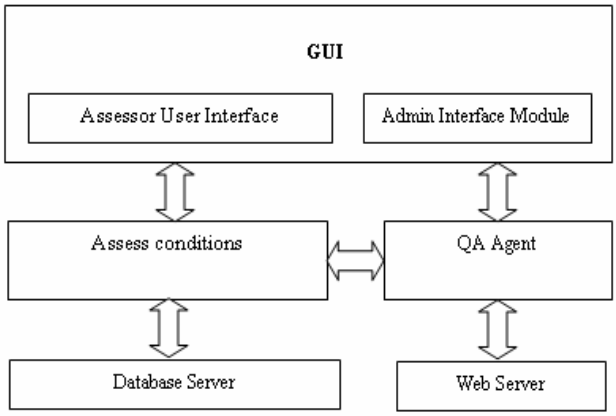

Fig. 5 Assessor System Architecture

The framework architecture of the assessor is shown in Fig. 5. It also has the GUI that allows the assessor to specify assess information and conditions. These assess information and conditions are entered to the agents which perform assessment activities on the human assessor's behalf. The QA agent module is the one which sends the agents to client destinations.

V. CASE STUDY

A case study was conducted at Phuket Rajabhat University in southern Thailand. The university employs the IPOO (Input Process Output Outcome) model [1] for quality assurance. There are seven measures namely, quality of graduates, research and creative works, academic services, cultural activities promotion, human resource and institutional development, curriculum and academic activities and quality assurance methodology. Each measure requires a key performance index (KPI). The KPI is selected by the client institution from a set of ONESQA approved KPIs. These values are used to calculate scores for the institution.

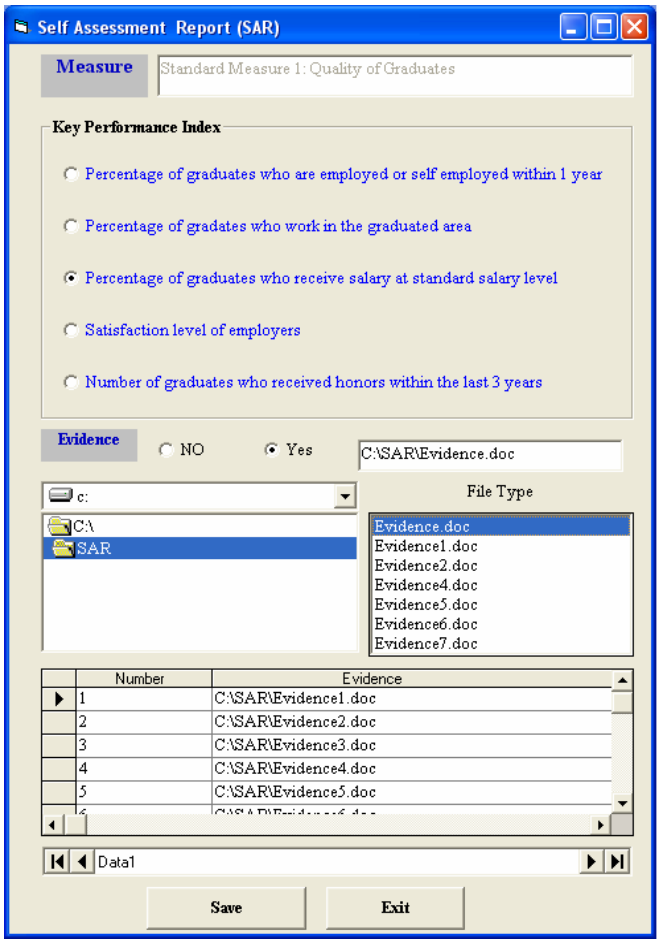

Fig.6 Key Performance Index and Support Document Screen

Self Assessment Report
Phuket Rajabhat University
Year 2006

Measure	KPI	Score level	Evidence
Quality of Graduates	Percentage of graduates who work in the graduate area	3	Y
Research and Creative works	Research grants per full-time lecturer	2	Y
Academic Services	Improve research and teaching quality from academic services	3	Y
Promote Cultural Activities	Percentage of annual budget for cultural and identity promotion	4	Y
Human Resource and Institutional Development	The use of internal and external resources	4	Y
Curriculum and Academic Activities	Number of full time equivalent students per full-time lecturer	2	Y
Quality Assurance	Quality assurance system that leads to continuous quality development	1	Y

Fig.7 An SAR report which is generated by the agent

Fig.6 shows a Self Assessment Screen. The measure is about quality of graduates. There are five KPIs for this measure namely, percentage of graduates who are employed or self employed within 1 year, percentage of gradates who work in the graduated area, percentage of graduates who receive salary at standard salary level, satisfaction level of employers and number of graduates who received honors within the last 3 years. The university must choose one of the KPI and prepare it supported evidences. In this example, the university chooses the third KPI. Fig. 7 shows an SAR report that the agent prepares after reading and studying all the relevant information in the database and the stored documents.

The system is implemented using Microsoft software tools such as Microsoft Visual Studio.net and Microsoft SQL Server 2005 Express [10,11,12,13,14].

MID	EID	Detail
1	1	C:\SAR\Measure1\Evidence1.pdf
1	2	C:\SAR\Measure1\Evidence2.pdf
2	1	C:\SAR\Measure2\Evidence1.pdf
3	1	C:\SAR\Measure3\Evidence1.pdf
3	2	C:\SAR\Measure3\Evidence2.pdf
4	1	C:\SAR\Measure4\Evidence1.pdf
4	2	C:\SAR\Measure4\Evidence2.pdf
5	1	C:\SAR\Measure5\Evidence1.pdf
6	1	C:\SAR\Measure6\Evidence1.pdf
7	1	C:\SAR\Measure7\Evidence1.pdf
7	2	C:\SAR\Measure7\Evidence2.pdf

Fig.10 Sample Instance of the Evidence Relation

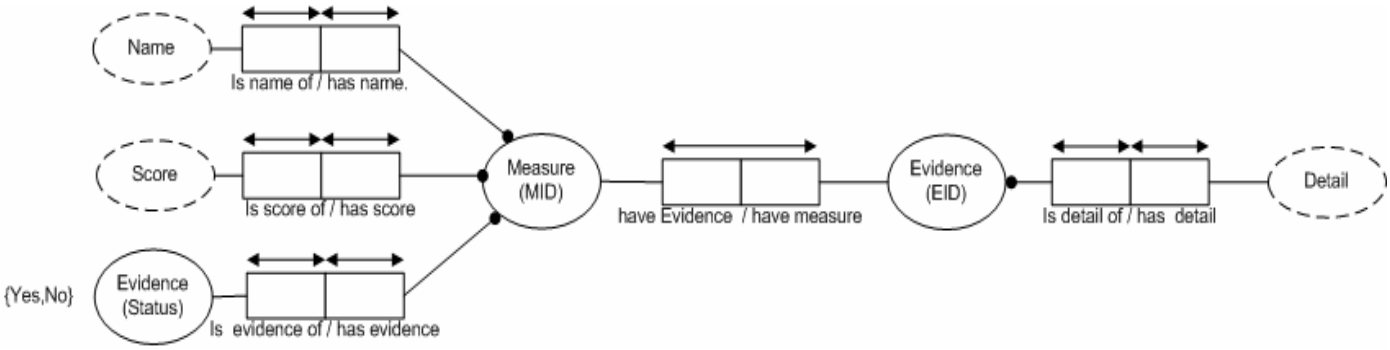

Fig. 8 Database Conceptual Schema of the Client Institution

MID	Name	Evidence (Status)	Score
1	quality of graduates	Y	3
2	research and creative works	Y	3
3	academic services	Y	3
4	cultural activities promotion	Y	3
5	human resource and institutional development	Y	3
6	curriculum and academic activities	Y	3
7	quality assurance methodology	Y	4

Fig. 9 A Sample Instance of Measure Relation

Fig. 8 shows the ORM conceptual schema [15,16] of the client institution side. The conceptual schema is then transformed into relational database schemas. Two relation instances are shown in Fig. 9 and Fig.10. The data is used by the agent which is sent by the assessor side.

Fig. 11 shows an SQL statement which is embedded inside the agent. The assessor simply specifies the web site of the client institution. An agent is sent to the site to gather information using embedded SQL statements and finally produces reports to both the assessor and the client institution.

```
SELECT Measure.MID, Measure.Name,
Measure.Evidance, Measure.Score, Evidence.EID,
Evidence.Detail
FROM Measure INNER JOIN Evidence ON
Measure.MID = Evidence.MID;
```

Fig11. An SQL statement which is embedded in the agent

VI. Conclusion

It can be concluded from the case study that the use of agents to help assess educational institutions that use different key performance indexes significantly reduces the burden of human assessors. The agent also notifies the client institution if some required information and supported documents are missing. This kind of assistance can be applied to other kind of quality insurance activities as well.

References

[1] The Office for National Education Standards and Quality Assessment (Public Organization). Available at http://www.onesqa.or.th/th/home/index.php

[2] Putsadee Pornphol,Suphamit Chittayasothorn, The Knowledge Management System for Higher Education Quality Assurance, Proceedings of Global Information Technology Management Association(GITMA) 2005, Alaska, USA.

[3] Kutluhan Erol, Renato Levy, James Wentworth, Application of Technology to traffic simulation.Availlable at http://www.tfhrc.gov/advanc/agent.htm

[4] W. Pan and I. Hawryszkiewycz (Australia) ,Using Software Agent Technology to Support e-Learning. Available at http://www.actapress.com/PaperInfo.aspx?PaperID=27549

[5] Edmund H. Durfee,Jeffrey S. Rosenschein. Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples.Available at http://www.agent.ai/doc/upload/200405/durf94_2.pdf

[6] Mobile Agents and the Future of the Internet. Available at http://www.cs.dartmouth.edu/~dfk/papers/kotz:future2/

[7] Mihaela Dinsoreanu ,Cristian Gogia,Claudi Anghel,Loan Salomie,Tom Coffey , Mobile Agent Based Solutions For Knowledge Assessment In E-Learning Environments. Available at http://arxiv.org/ftp/cs/papers/0605/0605033.pdf

[8] AGENT BUILDER. Available at http:// www.agentbuilder.com/Documentation/whyAgents.html

[9] Pragnesh Jay Modi and Manuela VelosoPragnesh Jay Modi and Manuela VelosoBumping Strategies for the Multiagent Agreement Problem. Available at http://www.cs.huji.ac.il/course/2005/aisemin/articles2006/docs/pa3d2_390.pdf

[10] Microsoft Agent. Available at http://www.microsoft.com/msagent/default.asp

[11] Microsoft SQL Server 2005 Express Edition. Available at http://www.microsoft.com/downloads/details.aspx

[12] The Microsoft Developer Network (**MSDN**). Available at http://msdn2.microsoft.com

[13] Agent Technology. Available at http://www.capgemini.com/services/technology/agent/

[14] Agent Technology. Available at http://www.semaphorecorp.com/btp/algo.html

[15] Object Role Modeling. Available at http://www.orm.net/

[16] G.M.Nijssen , T.A.Halping, Conceptual Schema and Relational Database Design ,Prentice Hall (1988).