A Generic Integrated Intelligent Agent (GIIA) for Electronic Business

Suraya Hamid
Faculty of Computer Science and Information Technology, University of Malaya
50603 Kuala Lumpur, MALAYSIA
E-mail: suraya_hamid@um.edu.my

ABSTRACT

Intelligent agent is a program that automates our interactions on the Web. It is a program that performs functions such as information gathering, information filtering, or mediation (running in the background) on behalf of a person or entity. This marvelous program can be delegated to do among others, the following tasks: to make airline and hotel reservations, to order new books from an online store, to find out about the latest song from a favorite musician, or to monitor stock portfolios. Some of the more sophisticated intelligent agents can even negotiate the purchase of raw materials for a factory, schedule factory production, negotiate delivery schedules with a customer's intelligent agent, and automate the billing process. As a body of knowledge, intelligent agent is indeed, an area extensively researched. However, most of the existing intelligent agents are not generic in the sense that they are not designed to be compatible with one another and the interfaces including the navigational systems are often cumbersome to use. Thus, a Generic Integrated Intelligent Agent (GIIA) is proposed to be developed in order to overcome the two major limitations of the available intelligent agents mentioned earlier. The GIIA should be able to integrate the existing agents capabilities into one powerful system (by making the system generic enough for a multipurpose usage) and also by providing a single, common and friendly user interface for the whole activities need to be run by the agent's user. The GIIA architecture to be built should fulfill all the following six systems requirements: open, extensible, distributed, parallel, mobile and multimodal. As electronic commerce is now fast becoming trend which simplifies lives, it is expected that electronic commerce applications such as GIIA will enhance the living quality of its users.

1.0 OVERVIEW

1.1 Introduction to Intelligent Agent

Intelligent agent, or software agent as it is sometimes called, is a program that automates our interactions on the Web. It is a program that performs functions such as information gathering, information filtering, or mediation (running in the background) on behalf of a person or entity. In this sense, software agents are programs to which one can delegate a task. To some extend, some people even refer software agent as an electronic butler [BotTechnology.com, 2000].

Intelligent agents differ from 'traditional' software in that they are personalized, continuously running and semi-autonomous. These qualities make agents useful for a wide variety of information and process management tasks as well as they are particularly useful for the information-rich and process-rich environment of electronic commerce that is the intention of this paper.

Meanwhile, from the linguistic perspective, the word 'agent' taken from The Concise Oxford Dictionary of Current English, is defined as: "one who, or that which, exerts power or produces an effect". Meanwhile, Webster's Revised Unabridged Dictionary defines agent as: "one who acts for, or in the place of, another, by authority from him; one entrusted with the business of another; a substitute; a deputy; a factor". Another definition of agent looked from computer network perspective is given by The Free On-line Dictionary of Computing. They define agent as: "<networking> In the client-server model, the part of the system that performs information preparation and exchange on behalf of a client or server. Especially in the phrase "intelligent agent" it implies some kind of automatic process which can communicate with other agents to perform some collective task on behalf of one or more humans" [Online Dictionary, 2004].

1.2 History of Intelligent Agents

The idea of employing agents in the interface to delegate certain computer-based tasks was introduced by visionaries such as the Director of MIT's Media Lab, Nicholas Negroponte in 1970 in his book, "The Architecture Machine: Towards a More Human Environment". This is followed by Carl Hewitt in 1977 when this proponent of Artificial Intelligence wrote in his paper titled "Viewing Control Structures as Patterns of Passing Messages" in the Artificial Intelligence Journal. In his work, Hewitt proposed the concept of concurrent of a self-contained, interactive and concurrently-executing object which he termed an 'actor'. This object had some encapsulated internal state and could respond to messages from other similar objects [Hewitt, 1977]. Alan Kay in 1984 in his article titled "A Computer Software" in Scientific American Journal [Maes, 1994] discussed intelligent agent from the perspective of interface agents.

Currently, the number of firms and universities pursuing agent technology is quite many and the list is ever growing. Massachusetts Institute of Technology's Media Lab is the leading institution in this research area. The number of software agents in numerous applications are also growing day by day

and this will be discussed in more detail in the following part.

1.3 The Use of Intelligent Agent in Various Disciplines

Intelligence agent is a study of various disciplines. The major part comes from the area of Artificial Intelligence, one of the themes in Computer Science (CS) study. Intense flowering of interest also come from other mainstream themes of CS such as data communications, concurrent systems, robotics, and user interface design.

As the potentials of intelligent agent does not stopped in the world of CS only, there are now many software agents developed for use in non-CS sectors. They can be found in industrial applications, commercial applications, medical applications and also in entertainment industry [Jennings and Wooldridge, 1998].

Intelligent agents in industrial applications can be found in areas like process control, manufacturing and in air traffic control. One example of agents in process control is ARCHON, applied in electricity transportation management and also for particle accelerator control purposes. In manufacturing, an agent called YAMS (Yet Another Manufacturing System) is used to efficiently manage the production process at a milling, lathing, grinding and spraying plant. In air traffic control system, a successful agent called OASIS was used at Sydney Airport in Australia. Intelligent agents used here are used to represent both aircraft and the various air traffic control systems in operation. In this system, as an aircraft enters Sydney airspace, an agent is allocated for it and the agent is instantiated with the information and goals corresponding to the real-world aircraft.

In the area of commercial applications, software agents are used in the information management area, electronic commerce and business process management. In the information management field, agents are used to solve the problem of information overload as the richness and diversity of information available to the users for their everyday lives has grown due to the massive information offered on the Internet. Agents like Maxims, Newt and Bonzi Buddy are available to the users. In the area of electronic commerce, agents are poised to take over the almost entirely driven activity by human interactions as what it is practice nowadays. Agents like Kasbah, MAGMA, and Letizia are the good examples of intelligent agents used in this area. More detail of these agents used in electronic commerce will be further discussed in Section 2. Meanwhile, in the area of business process management, agents such as ADEPT is used to enable company managers to make informed decisions based on a combination of judgment and information from many departments. ADEPT handles the complex and time-consuming process activities such as in obtaining pertinent, consistent and up-to-date information across a large company for the managers to make wise decisions.

Intelligent agents are also used for medical applications. One of the examples of agent in patient monitoring is the Guardian system used in the Surgical Intensive Care Unit. In particular, this agent is to help specialists in monitoring the patient condition in a minute-by-minute status of a patient. Another agent is used in the healthcare industry. This prototypical system is a distributed medical care system which integrate the patient management process and contains a knowledge-based system for the patient care process.

In the entertainment scene, agents have an obvious role in computer games, interactive theater and some related virtual reality applications. Agent such as Real Time Able (RTA), the agent is programmed in terms of behavior. These behaviors are simple structures which loosely resemble rules but do not require complex symbolic reasoning. The example of computer game used for such purpose is in Tetris. For the interactive theater and cinema purposes, the agents developed should be able to play the part of human actors in plays or films, interacting with artificial, computer characters that have the behavioral characteristics of real people. These kinds of agents that play the part of humans are known as believable agents and this term was coined by Joe Bates in his article "The role of emotion in believable agents", which appeared in the Communications of ACM Journal in 1994.

1.4 Agent Classifications

The discussions above involve a host of properties of an agent (such as autonomous, continuously running and personalized). Having settled on a much less restrictive definition of an autonomous agent, these properties may help us further classify agents in useful ways. The table below lists several of the properties mentioned above and more.

Table 1: Properties of Agents

Property	Other Names	Meaning
reactive	(sensing and acting)	responds in a timely fashion to changes in the environment
autonomous		exercises control over its own actions
goal-oriented	pro-active purposeful	does not simply act in response to the environment

temporally continuous		is a continuously running process
communicative	socially able	communicates with other agents, perhaps including people
learning	adaptive	changes its behavior based on its previous experience
mobile		able to transport itself from one machine to another
flexible		actions are not scripted
character		believable "personality" and emotional state.

These properties are normally will constitute the types of agent itself. The types or also called taxonomy of agent is shown in the figure below [Franklin and Graesser, 1996].

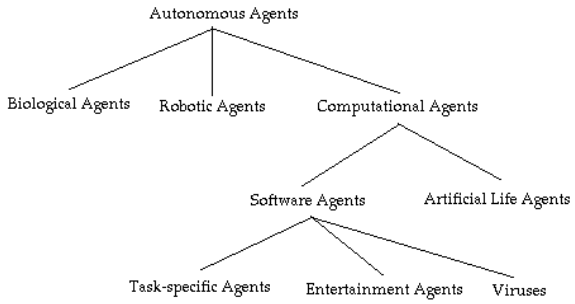

Figure 1: Taxonomy of Agents

2.0 GIIA DEVELOPMENT LIFE CYCLE

2.1 Overview of GIIA Development

In this section, the analysis of GIIA development is discussed.

The GIIA software will work on the Web and hence it must be able to understand the basics of the Web. As such, it must be able to work with HTTP protocol. Since majority of pages are on the Web in the HTML format that uses the HTTP protocol, it becomes a mandatory need for GIIA to understand the HTTP protocol.

The second thing is that the coding of GIIA itself. Software agents can be written in many programming languages. But then, if we want the agent to be extensible, and reusable by other agents, or we want other agents to be in a position to communicate with this agent then we must develop a suitable messaging Application Programming Interface (API). The messaging should also be defined precisely so that the agent does not get lost during its life cycle.

Java is a preferred language. Many of the other languages like C++ are good but then they are not suitable for the Web-based programming. Java is platform independent and hence looks more like an obvious choice of the people who are coding the agent. The second important thing is that agents are not piece of software like application programs that do the same thing again and again. Agents are built in such a way that it keeps “learning” as they are being used. Hence the knowledge base of the agent cannot be hard-coded into the software itself. The knowledge base must be separate from the agent itself. That is a challenging task. This immediately makes almost all software on the Net redundant as must of the software have their logic hard-coded and do not learn as time passes by.

That is why there is no surprise that the agents are written in languages that are AI-capable. AI or artificial intelligence languages work in such a way that the knowledge base keeps getting accumulated. Lisp and Prolog are very old but still important languages. AI that used to be used a lot in analog programming has suddenly got an upcoming and a vast area for people to use. But then it is not easy to code in the AI languages. Though the program constructs are quite similar of selecting and iteration that are used in programming languages, but then they need a more scientific approach and rigorous coding method.

The knowledge domain that is depicted by GIIA too, should be searchable and the search queries should run very fast. This is the third test in writing an agent. The agent to decide whether a piece of information is relevant or not and whether should it be added to its knowledge base or not is a challenging task. The GIIA database (or called GIIA Page Store) should not occupy a large space on one hand and on the other hand should be very fast. The queries would be multidimensional, able to query multimedia elements and they should execute very fast.

The next challenge for writing an agent is designing a data structure. The data needs to be initially defined to some extent. This would depend on the domain of knowledge this agent is going to operate. For example, if an agent is designed to be a buying agent for medicines, then some knowledge of medicines should be fed into the database. As for GIIA, it is an integrated, multifunctional agents that cover a vast applications of electronic commerce. Therefore, its database and data structure is expected to be quite huge. The agent can take these data structure and page store as a starting point of knowledge and it can keep building on this base.

At the same time, another point of concern is that the agent cannot keep itself limited to this pattern of knowledge representation only. Intelligent agents use “heuristics” approach in order to take or to reach a decision [Nwana, Wooldridge, 1996]. Heuristics means doing an analytical or a reasoning based search

to reach a desired destination or goal. The heuristics are stubs that are defined to be the knowledge statements. As defined in one of the agents discussed in the first part, the heuristics would start to build the beliefs of the agent. As the time passes the agents beliefs would either get reinstated or confirmed more solidly or would get negated and the belief would come down on these stubs.

2.2 GIIA development phases

In building GIIA, the methodology followed is a simplified System Development Life Cycle (SDLC) which usually consists of the Analysis, Design, Coding, Testing, and Maintenance phases. The followings are the simplified SDCL for GIIA development:

1. analyzing and defining the domain and behavior of GIIA operations
2. designing and developing interfaces, database, engines and networks of communicating agents
3. coding
4. debugging and testing

As of now, GIIA is still in the Phase 2 of its development life cycle.

2.3 GIIA Development Tools

In any agent development, the developer needs to understand many issues. Among others, the agent communication language, agent development language, its environment, user interface, KQML and its ontology.

2.3.1 Agent language

Agent languages define how we are to program agents. What are the right primitives and performatives for the tasks we want them to operate? How are we to effectively compile and execute agent programs?

For development GIIA itself, Java is used as mentioned earlier. In the future enhancement, consideration is directed at XML (Extensive Markup Language) as part of the way to complement of using Web as the main vehicle.

2.3.2 KQML

The Knowledge Query and Manipulation Language (KQML), a standard agent communication language (ACL) consists of three layers: the content, message and communication layers [Finnin, Fritzon, McKay, McEntire, 1994]. KQML allows more than thirty performatives that defined the allowed 'speech acts' that agents may use, which provide the substrate for constructing more complex coordination, negotiation strategies as well as to support interoperability among intelligent agents in a distributed application. These performatives, which GIIA will use are shown in the following table.

Table 2: KQML and its reserved performatives

Category	Reserved performative names
Basic information performatives	Tell, deny, untell, cancel
Basic query performatives	Evaluate, reply, ask-if, ask-about, ask-one, ask-all, sorry
Multi-response query performatives	Stream-about, stream-all
Basic effector performatives	Achieve, unachieve
Generator performatives	Standby, ready, next, rest, discard, generator
Capability definition performatives	Advertise
Notification performatives	Subscribe, monitor
Networking performatives	Register, unregister, forward, broadcast, pipe, break
Facilitation performatives	Broker-one, broker-all, recommend-one, recommend-all, recruit-one, recruit-all

Another closely related topic to KQML is agent ontology. Ontology is a description of the concepts or bodies of knowledge understood by a particular community and the relationships between those concepts. As such, GIIA need to understand the same ontology to make it versatile in the cyberspace and to be able to communicate in the same language with other agents that it will meet on the Net. GIIA, based on its 'communicative' and 'learning' properties should be able to develop its own ontology during its lifetime.

2.4 GIIA Architecture

In her article 'The agent network architecture', Pattie Maes [1991] defines an agent architecture as 'a particular methodology for building agents. It specifies how the agent can be decomposed into the construction of a set of component modules and how these modules should be made interact. The total set of modules and their interactions has to provide an answer to the question of how the sensor data and the current internal state of the agent determine the actions… and future internal state of the agent. An architecture encompasses techniques and algorithms that support this methodology.'

Another researcher, Kaelbling considers an agent architecture as 'a specific collection of software (or hardware) modules, typically designated by boxes with arrows indicating the data and control flow among the modules. A more abstract view of an architecture is as a general methodology for

designing particular decomposition for particular tasks [Kaelbling, 1991].

However, as for GIIA, the architecture is the definition of how are we to construct agents that satisfy the properties we expect of them, and what software and hardware structures are appropriate in building them. GIIA architecture in its schematic depiction is shown below.

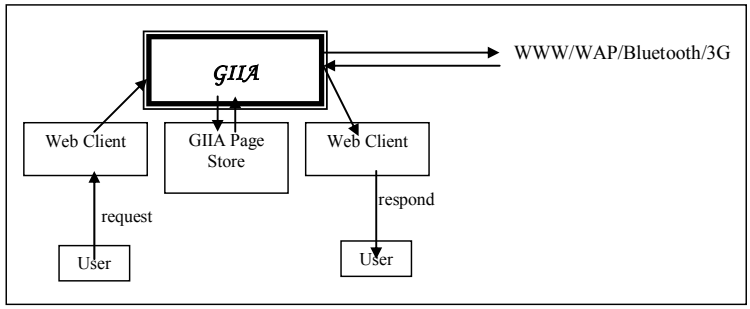

Figure 2: GIIA Architecture

In addition, the GIIA architecture to be built will fulfill all the following six systems requirements: open, extensible, distributed, parallel, mobile and multimodal. However, for the first version of GIIA to be implemented, not all of the six systems requirements below might be implemented.

- *Open* is in terms of the creation of GIIA where it should be created in multiple programming languages (Java and later in XML) and capable of interfacing with existing legacy systems.
- *Extensible* is whereby, the GIIA modules can be added or replaced individually at run-time.
- When the GIIA is spread across any network-enabled computers, it has fulfilled the *distributed* requirement.
- *Parallel* is in the sense of the ability of the GIIA to cooperate or compete on tasks in parallel function.
- When GIIA is run on lightweight user interfaces such as handheld PDA's, mobile phones (using enabled-WAP, 3G, or Bluetooth mobile phones) or in a Web browser using Java or HTML, it is called *mobile*.
- *Multimodal is* where communication between the GIIA and its users is through one or any combinations of the following methods: handwriting, speech, pen gestures, or direct manipulation of graphical user interface (GUI).

2.5 GIIA Interface

In this section, the proposed user interfaces for the GIIA agents are shown below. In terms of navigational issue, it is expected that the users will face minimal problems as the interfaces are based on point-and-click approach, ubiquitous in today's computing environment. This easy-to-use user interface will ensure little disorentation to GIIA owners. Even though the working of GIIA is more in the background/behind the scene, and on the Internet, the user can always see, update and maintain the status of their GIIA quite easily. Figure 3-Figure7 shows a suggestion of user interfaces for GIIA.

Figure 3: The Login Interface of GIIA

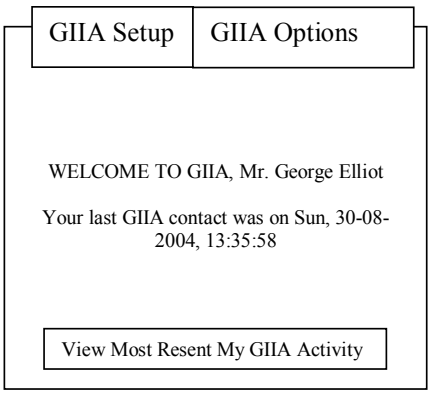

Figure 4: Successful Login

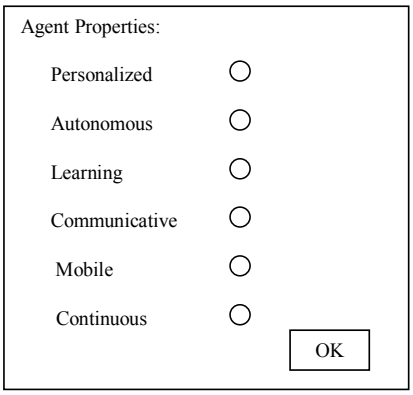

Figure 5: GIIA Agent Properties Setup

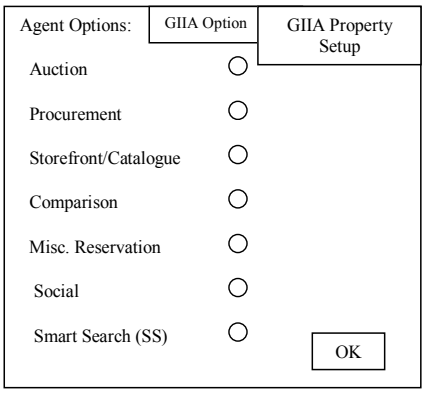

Figure 6: GIIA Agent Options

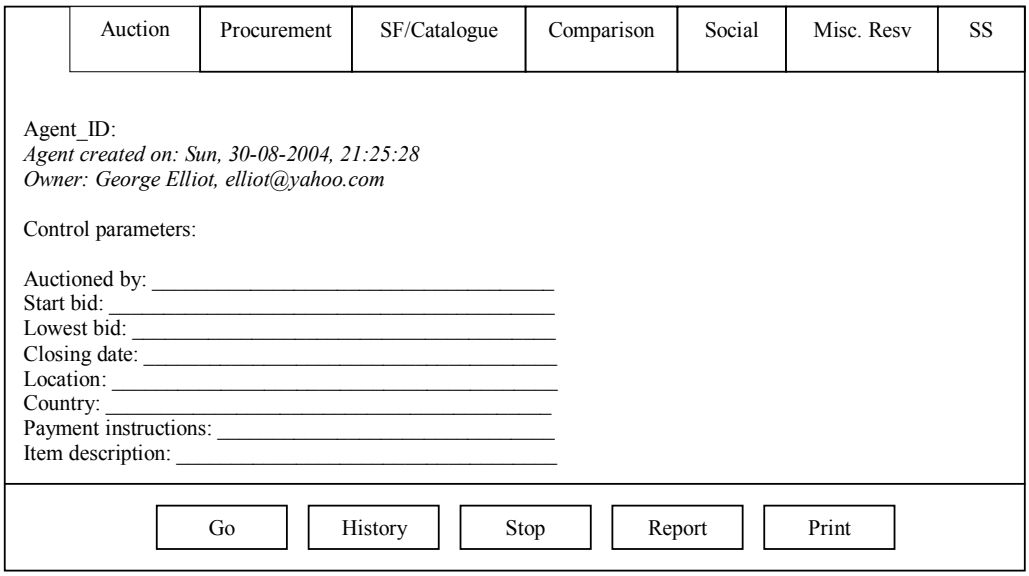

Figure 7: GIIA when used for Auction Module

As from the proposed user interfaces above, it is clear that GIIA really aims to be generic in nature (single and uniform interface across functions) and integrated in functions (auction, procurement, storefront and catalogue, comparison, social, miscellaneous reservation and smart search).

3.0 CONCLUSION

In order to develop a sophisticated software agent such as GIIA from scratch, specialized skills and knowledge in a variety of areas are required. The author is still in the process of learning to get intimate to software agents and intend to make this area as his academic specialization area.

It is realized that in software agent research, areas that one need to master are as follows: agent architecture, communications technology, reasoning systems, knowledge representation, agent communication language languages, protocols and also machine learning. It might take years to master all these but the potentials of intelligent agent, be it in electronic commerce or in entertainment or education, is tremendous.

It is indeed that with the increased capacity and quantity of software agents nowadays, they will become indispensable to its users in the near future. As electronic commerce is soon becoming the way people simplify and live their life, it is expected that electronic commerce applications such as GIIA will enhance the living quality of its users.

REFERENCES

BotTechnology.com, Bots and intelligent agents, 2000. URL: http://www.nottechnology.com/latest_articles/article25.htm

Chavez, A, Maes, P, Kasbah: An Agent Marketplace for Buying and Selling Goods. MIT Media Lab, 1996 URL: http://citeseer.nj.nec.com/chavez96kasbah.html

Doorenbos, R, Etzioni, O, Weld, D, ShopBot: A Scalable Comparison-Shopping Agent for the World-Wide Web. University of Washington, 1997 URL: http://citeseer.nj.nec.com/doorenbos97scalable.html

Feldman, S, Yu, E, Intelligent Agents: A Primer. 1999. URL: http://www.findarticles.com/cf_dls/m0DPC/9_7/56200326/p1/

Finnin T, Fritzon R, McKay D and McEntire R: 'KQML as an agent communication language', in Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM), New York, 1994. URL: http://www.cs.umbc.edu/kqml/papers/

Franklin, S. Graesser, A. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, in Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages, Springer-Verlag, 1996. URL: http://www.msci.memphis.edu/~franklin/AgentProg.html

Guttman, R, Moukas, A and Maes, P, Agents as mediators in Electronic Commerce, MIT, in Electronic Markets, Vol 8, No. 1, 1998, pg 22-23.

Hewitt, C, Viewing Control Structures as Patterns of Passing Messages. Artificial Intelligence, 8, No 3, pp 323-364 (1977).

Jennings, N. R, and Wooldrigde, M, Applications of Intelligent Agents, 1998 URL: http://citeseer.nj.com/jennings98applications.html

Jennings, R.N, Sycara, K, Wooldrige M, A Roadmap of Agent Research and Development, Journal of Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers, Boston, 1998 URL: http://citeseer.nj.nec.com/jennings98roadmap.html

Kaelbling, L.P, A Situated Automata Approach to the Design of Embedded Agents. SIGART Bulletin, 1991, 2(2):85-88

Lau, A, H.M, Arthur, D.Bruza, P, WEBS: Adaptive Profiling Agents for Electronic Commerce. University of Queensland, 2000 URL: qut.edu.au/~raymond/ec.ps

Lieberman, H, Letizia: An Agent That Assists Web Browsing. Massachusetts Institute of Technology, 1995 URL: http://citeseer.nj.nec.com/lieberman95letizia.html

Maes, P, The agent network architecture. SIGART Bulletin, 1991, 2(4):115-120.

Maes, P, Agents that Reduce Work and Information Overload, 1994 URL: http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-94.p2.html

Negroponte, N. The Architecture Machine; Towards a More Human Environment, MIT Press, 1970.

Nwana, H.S, Wooldrige, M, Software Agent Technologies, BT Technology Journal, Vol. 14, No. 4, 1996. URL bt.com/projects/agents/...report5.ps.gz

Online Dictionary, 2004 URL: http://www.dictionary.com

Tsvetovatyy, M, Mobasher, B, Gini, M, Wieckowski, Z MAGMA: An Agent-Based Virtual Market for Electronic Commerce. University of Minnesota, 1997 URL: umn.edu/~gini/papers/magma.ps.gz

List of academic and research projects on software agents: URL: http://www.agentbuilder.com/AgentTools/academic.php