ISSN(online):2250-141X
Vol. 2 Issue 3, June 2012

INTELLIGENT AGENTS IN DECISION SUPPORT SYSTEM

[1]Sachin Kashyap Research Scholar

Singhania University Rajasthan (India)

[2]Dr. Pardeep Goel, Asso. Professor

Dean Of Science Faculties,

M.M. College Fatehabad (Haryana)

ABSTRACT:

Several investigations have been done on agents which can take their own decisions as per requirement or data entered by end user in Decision Support System. This paper explains an intelligent software agent framework designed for DSS. Software Agents have been proclaimed as fast growing area of Information Technology of the last two decades. Many software companies are keenly trying to incorporate these software agents known as autonomous programs into their product. Many researchers in this field try to define the concept of agencies and to develop architectures improving agents' capabilities. DSS researchers have been keen to incorporate agents into their applications.

Keywords: Software Agents, Decision Support Systems, Intelligent Software Agent Framework, Autonomous Program.

Introduction

Software Agents

In this paper we describe framework for Intelligent Software. Now a day's intelligent computers which accept instruction as input and perform the job on behalf of us since the birth of first computer. Now a days as per need computer and software application have become an important part of our daily life. The process which copies human like behavior in computers is very tedious task.

For over last two decades Artificial Intelligence Research has been going on. This study and experience has given a lesson to us that the mechanism which comes under human behavior are truly very typical and not easily produced again. An agent is basically software abstraction of another kind. However an agent is very high level software abstraction which provides an easy and powerful way to describe a complicated software entity.

Instead of defining in term of methods and attributes. An agent is known of its functions which it can perform (behavior). So it is important for researcher to write code for agent based system which focused on agent's behavior (functionality) rather than identifying attributes, method and classed. It is very easy natural process to specify behavior as compared to writing code. There is minimum set of common characteristics that an software agent must have.

A software agent must be autonomous program which is capable of performing a single process and actions without user interference. A software agent must have communicative nature which can communicate with user as well as other software agent application. A software agent must be perceptive which is capable of understanding and responding to changes in environment. Like people, software agent can behave different level of intelligence to perform a specific task. For example one web billing software agent can send bill by email to company's customers whose email addresses are specified. Whereas another web billing software agent may be smart enough to send bill by email as well as messages over cell, this may be the ability of software agent to keep check on the bill amount limit and remind the user before exceeding limit. So software agents must be autonomous, perceptive and competence, having different level of intelligence as per the client need.

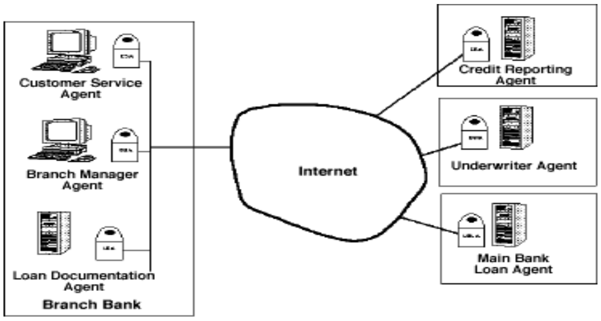
In the simple term we can define a software agent as a computer program that performs job for the user as per their need. As fast growing technology the definition of software agent is not concisely defined where as some companies, on release of software application, claims about their product to be agent enabled. Researches keep on thinking what features must be added into agent so that software agent cam be different from regular computer program and make them better capable to help us. Interested agent researchers thinks how to incorporate human like behavior into software program and what functionality can make these software agent most effective catering to our need and perform action on behalf of us.

Agent theories try to suggest which type of behavior will make a computer program to show agency, where as the architecture of agents describe the artificial intelligence mechanism used for the purpose of implementing these behaviors. Researches on agent theories and architectures are still widely going on. So far no single theory on architecture has got unanimous acceptance. The different explanations of agents and agency with different discipline as per different need prevent the acceptance of universal set of theories and architecture.

Why Are Agent Important

The usage of high level abstraction in developing complicated software is grown by Software developer and designer to manage complications. An abstraction focuses on important and essential properties of a problem. Thus agents provide a new and easy way to manage complications of describing a computer system and process. Using agent oriented process or agent enabled application.

Consider, for example, some issues of system design evolved in developing an application which approves a loan that ties together various branch banks, same bank, loan underwriting companies, and credit reporting companies, and automates much of the loan approval process. Developing this system using emerging and current technology is a tedious task because the system disintegration compels the developer to handle with low-level concepts (e.g. loan applications, account balances, credit ratings) when describing the whole system architecture. In addition, significant design time must be dedicated to describing the communications protocol and interfaces that will allow the bank to exchange data with the credit reporting agencies and loan underwriters. In an agent-oriented system design, the system solution might include a customer service agent, a loan application analysis agent, an underwriter agent, etc. The focus is placed on the behavior of each of these agents and communication between agents. The problem is made much easier because the level of abstraction is much higher and the programming problem becomes one of specifying agent behavior.



Multiple Software Agents for Enterprise Applications.

A solution which is base on agents is captivating and useful because mostly agents being used in solution implicitly know how to process many things. For example, agents know how to have communication with other agents. In such a pass software agents developer don't have to design protocols for communication and formats for messages. These capabilities are the part of basic agent mechanism. Generally this is the implicit capabilities of agent develop model of their environment , keep vigil the different states of the environment in which it is being used, reason and make decisions depending on that state . So software developer's agents have to do in the give condition.

Intelligent Agents.

The concept of intelligent agents is quickly becoming an important area of research (Bhargava & Branley, 1995; Etzioni & Weld, 1994; Khoong, 1995). Informally, intelligent agents are also known as software agents with intelligent behavior, that is, they are a combination of software agents and intelligent systems. Formally, the term agent is used to denote a software-based computer system that enjoys the following [31].

Features of Software Agents

As we know the nature and the potential behavior of software agents are abstract when added into one makes it complex for a single definition of a software agent to be widely accepted[18],[31] .

Software agent must be a program having characteristics differentiating it from a software application. Several key characteristics are common among most various software agents and offer minimum need for software to be classified as agent.

• ***Autonomy*****:** agents should be able to perform the majority of their problem solving tasks without the direct intervention of humans or other agents, and they should have a degree of control over their own actions and their own internal state.

• ***Social ability*****:** agents should be able to interact, when they deem appropriate, with other software agents and humans in order to complete their own problem solving and to help others with their activities where appropriate.

• ***Responsiveness*****:** agents should perceive their environment (which may be the physical world, a user, a collection of agents, the INTERNET, etc.) and respond in a timely fashion to changes which occur in it.

• ***Proactiveness*****:** agents should not simply act in response to their environment. They should be able to exhibit opportunistic, goal-directed behavior and take the initiative where appropriate.

• ***Mobility*****:** Agents are able to travel through computer networks. An agent on one computer may create another agent on another computer for execution.

Agents may also transport from computer to computer during execution and may carry accumulated knowledge and data with them.

Examples of intelligent software agents [31]

Buyer agents (shopping bots)

Buyer agents travel around a network (i.e. the internet) retrieving information about goods and services. These agents, also known as 'shopping bots', work very efficiently for commodity products such as CDs, books, electronic components, and other one-size-fits-all products.

User agents (personal agents)

User agents, or personal agents, are intelligent agents that take action on your behalf. In this category belong those intelligent agents that already perform, or will shortly perform, the following tasks:

- You can Check your e-mail, sort it as per the user's order of preference, and alert you when important emails arrive.
- Play computer games as your opponent or patrol game areas for you.
- Assemble customized news reports for you. There are several versions of these, including CNN.
- Find information for you on the subject of your choice.
- Fill out forms on the Web automatically for you, storing your information for future reference
- Scan Web pages looking for and highlighting text that constitutes the "important" part of the information there
- "Discuss" topics with you ranging from your deepest fears to sports
- Facilitate with online job search duties by scanning known job boards and sending the resume to opportunities who meet the desired criteria
- Profile synchronization across heterogeneous social networks

Monitoring-and-surveillance (predictive) agents

Monitoring and Surveillance Agents are used to observe and report on equipment, usually computer systems. The agents may keep track of company inventory levels, observe competitors' prices and relay them back to the company, watch stock manipulation by insider trading and rumors, etc.

For example, NASA's Jet Propulsion Laboratory has an agent that monitors inventory, planning, and scheduling equipment ordering to keep costs down, as well as food storage facilities. These agents generally keep vigil complicated computer networks that can keep track of the configuration of each computer connected to the network.

Data mining agents

This agent uses information technology to find trends and patterns in an abundance of information from many different sources. The user can sort through this information in order to find whatever information they are seeking.

A data mining agent operates in a data warehouse discovering information. A 'data warehouse' brings together information from lots of different sources. "Data mining" is the process of looking through the data warehouse to find information that you can use to take action, such as ways to increase sales or keep customers who are considering defecting.

Decision Support System

Two decade back the research on DSS had been started [22]. As per Gory and Scott Morton build the phrase ‘DSS’ in 1971 “ A Decision Support System is an interactive computer based system that helps decision makers to utilize data and models to solve unstructured problems”. Again in 1978 as per Keen and Scott Morton , A DSS couples the intellectual resources of individuals with the capabilities of the computer based

ISSN(online):2250-141X
Vol. 2 Issue 3, June 2012

support system for management decision makers who deals with semi structured problems “ . So we can DSS provide support semi structured and unstructured problems and depict a multi disciplinary field for the researchers from Management Information System , Operation research, Artificial Intelligence, Organizational studies , Banking solutions, Airline reservation, Accounting models , Suggestion systems. As per Peter Keen DSS should helps in solving ill structured, unstructured or semi structured problems which only supports Manager , not replace manger, DSS also contribute decisions effectiveness rather than efficiency.

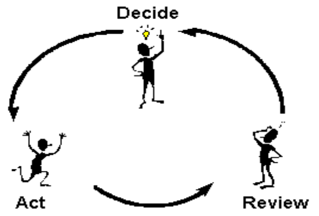
So we can say “Decision Support Systems (DSS) are a specific class of computerized information system that supports business and organizational decision-making activities. A properly designed DSS is an interactive software-based system intended to help decision makers compile useful information from raw data, documents, personal knowledge, and/or business models to identify and solve problems and make decisions. Typical information that a decision support application might gather and present would be:

- Accessing all of your current information assets, including legacy and relational data sources, cubes, data warehouses, and data marts.
- Comparative sales figures between one week and the next.
- Projected revenue figures based on new product sales assumptions.
- The consequences of different decision alternatives, given past experience in a context that is described.

DSS is built on top of a transaction system, a database and a data model, all of which provides the DSS with data and information which is processed and presented to the user in a simple form. The very first important aspect of DSS is that they provide information used in the decision making process. The focus is not here on quantity of information but on quality of data.

Having summarized the most important functions of a DSS, it should be remembered that a DSS is only as good as the individual components that it consist of: DSS is built on top of a transaction system, a database and a data model, all of which provides the DSS with data and information that is processed and presented to the user in a simplified form. The fist important aspect of DSS is that they provide information which are used in the decision making process. The emphasis here is not on the quantity of information, but rather the quality. There are multiple factors that qualify information as having good quality (such as timeliness, relevant, accurateness, consistency, unbiased, etc.) but the important consideration factor is how information is used in order to attain a certain goal.

It is a common notion that information can be misinterpreted due to which wrong conclusions can be given which adversely affect the quality of the decision making process inside an organization (as is humoristically depicted in figure 1…).

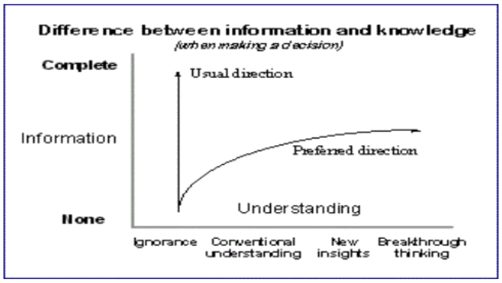


“There is nothing so inefficient as very efficiently doing things that should not be done at all”

Managers often protract complicated decisions and request “more information” without really trying to perceive the information which is already available to them. They falsely depend on the assumption that information should give them guidance, instead of realizing that their understanding of relevant information will help them inculcate their own ideas, which should be based on both a intricate perception of the information, the necessary knowledge available around them and ultimately their intuition that is develop based on experience. This curvilinear relationship between information and comprehending information is depicted in figure 2. The graph suggests that the quantity of information is only useful up to a certain point, after which more information becomes redundant/obsolete.

Thus, the user should try to digest the current information available to him/her first and resist the temptation to seek more information without first understanding the current information. Only as the user gains “new insights” and “breakthrough thinking”, additional information should be sought where relevant. To make a good decision, one needs not only information about the specific instance, but also an understanding of the domain. In other words, one needs a set of principles, models, templates or other abstractions. Better understanding enables the identification of what information is relevant and consequently, less information is required because the irrelevant components can be ignored. This not only decreases the complexity of the decision process, but also decreases the processing load on the manager and leaves him with more time to focus on critical and situation-relevant information segments.

Figure 2



A successful business makes good decisions, implementing them well - and then learns from the past experience and abstractions in order to make things better next time. These abstractions are then re-usable for making new decisions with different information, facilitating the process of knowledge management and ultimately enhancing the overall quality of decision making inside the organization.

Fundamental Components of DSS

User Interface or Dialog Generation Management System Component

The DGMS component provides an user interface using which end user interacts with a Database Management System and Model Base Management System (MBMS) subsystem [28]. DGMS is the one part of DSS but when user interacts with DGMS, the user views the DGMS subsystem as a whole DSS. These user interfaces help in model building but also during interaction with the model such as gaining insight and recommendations from it. This is the main responsibility of DGMS is to extends the ability of end user to utilize and benefit from the DSS.

Database Management System (DBMS)

The main task of DBMS is to capture and store the internal and external data. The DBMS part is kept separate from transactional database. The process of capturing internal data involves other application or remote site from the same organization. Whereas the process of capturing external data involves the extraction of data from either internal collection source or directly from external source. The process of extraction must be automated so that it cannot disrupt changing data needs, heterogeneous data sources, distributed data sources, decision making process of end user.

The database is the "feeder" of the DSS and good database design is thus crucial in improving the overall performance of the DSS. Database design and management is a complex and often ill-conceived topic and extends well beyond the scope of this paper. Consequently our discussion on databases will parsimonious at best. A good DBMS ensures data integrity, reduces data redundancy, follows a logical sequence and are consistent in its performance.

Relational databases are in most cases the system of choice when it comes to designing a DSS. This is primarily because of the flexibility associated with a relational DB but also because it allows normalization (reduction in data duplication), which helps with maintenance of a large database. The ability of these databases to identify relationships between entities makes information analyzes an important feature to the success of the DSS. Some hierarchical and network databases are still being used today, but merely so because of the costs involved with migrating to a totally new DB platform overshadows the maintenance costs of legacy systems – even this is gradually changing as more organizations are putting additional requirements on their DSS, underscoring the need for a more suitable DB design.

There seems to be a general trend in recent years, to either migrate to web-based DSS, using a task specific search-engine, or to build the DSS around a thin client fat server environment, using network and web-based technologies. This is perhaps a further justification of the need for an advanced database system since the Internet is the most comprehensive network of thousands of interconnected databases and web pages.

The Model Base Management System Component

The basic function of the MBMS subsystem is the development, storage, and update of models which enable the problem solving process within a DSS [28]. The models utilize the data stored within the DBMS subsystem in creating alternate solutions for the user. DSS should fit the style and modus operandi of each user, and that models should automatically be decided upon and invoked without burdening the decision-maker with the requirement of expertise in such matters. Two of the four functions of the MBMS identified by Sprague and Carlson deal with the way models are generated or restructured within a DSS [28]. If the models used to propose alternatives are outdated or specified incorrectly, the usefulness of the DSS declines. The complexity and time consuming nature of specifying models, whether the models are stored as subroutines, statements or data, contributes to the ease with which models become outdated.

DSS Applications

So far, most DSSs are applied to structured and semi-structured problems. A list of existing DSS applications, non-exhaustive of course is given below**:**

•Corporate financial planning

- loan amortization
- depreciation
- lease versus buy
- discounted cash flow and net-present value
- break-even analysis

• Marketing analysis

- forecasting
- sales analysis
- promotion analysis
- consumer sales audits

• Real estate investments

- financing alternative
- cash flows
- impact on taxes
- payoff

• Mineralogical exploration
• Transportation routing
• Portfolio analysis

Benefit of DSS

- Improves personal efficiency.
- Speed up the process of decision making.
- Increases organizational control.
- Encourages exploration and discovery on the part of the decision maker.
- Speeds up problem solving in an organization.
- Facilitates interpersonal communication.
- Promotes learning or training.
- Generates new evidence in support of a decision.
- Creates a competitive advantage over competition.
- Reveals new approaches to thinking about the problem space.
- Helps automate managerial processes.

Current DSS Research

Research and Technological advancements in other disciplines have been quickly embraced within the individual components or subsystems of DSS. In the last decade, within the DGMS, user interfaces have improved substantially in appearance and usability through the use of visual programming languages like java from Sun Micro System, Microsoft .Net development environments. Similarly, the interoperability and content of the DBMS component have been enhanced through Open Database Connectivity (ODBC), JDBC (Java Database Connectivity) data warehousing, and web-based data access. The MBMS, however, is considered the least developed component and is the focus of much of the current research in DSS. DSS researchers have noted the emerging focus on model management issues in DSS research [14]. While some of this emphasis has been in response to the lack of standardization within the MBMS, the exploration of intelligent DSS has also fueled the interest in MBMS research [2]. The combination of model management and artificial intelligence is considered essential in providing decision support and is viewed as the cornerstone of DSS development [26]. This current focus on model management and artificial intelligence provides an ideal environment for the integration of DSS and software agents.

Future Aspects of DSS

No acceptable definition has been phrased so far, as per the need of research the researchers has created their own definition of DSS. The concept of DSS is still in the market b because it complements the transaction processing aspect of EDP. In future, However, DSS researching will take some time to reach mature level. As it stands, the following future trends of development of DSS are identifiable:

(1) Group DSS

Group decision making plays a very

important role determining corporate affairs. How to design and develop group DSSs for supporting group meetings is a complicated job because of the different type of people, places, time communication networks, individual preferences, and other technologies. Using teleconferencing techniques a meeting for a group can be arranged at the same place, or at remote locations attended by different groups of people. Group DSS is supposed to support any one of the possible combinations.

(2) Decision Support Centre

Decision support centre is an upcoming concept. Decision support groups, staffed by information systems professionals who understand the business environment, form the core of decision support centre, with advanced information technology. A decision support centre is usually located in close proximity to top management so that instant decision support can be provided.

(3) Strategic DSS

DSS for supporting strategic management is a well recognized area of importance and significance. It is an area where DSS can make a substantial impact on the top management and the corporation.

(4)Intelligent DSS

However, most authors under-estimate the difficulties in representing common-sense knowledge which is an unsolved problem in AI. Some authors notably suggest the adaptation of artificial intelligence (AI) and expert systems techniques to DSS.

Agents in Decision Support System

The researchers of DSS have been expediting to exercise with software agent. In DSS oriented journals, the incorporation in DSS have begun to unearth, Due to the lack of theories and architecture of agent the researcher had to move on this experiment with basic conceptual guidance like , how , when , where to use software agents. There has been no discussion why and how the upcoming technology is apt for DSS . The topic of the development of agent theory and architecture towards DSS research would bridge the current gap in this.

CONCLUSIONS

This study has conferred a definition of a *software agent*, which explains the important and enabling qualities of such agents. This also describes the advantages which may be produced from incorporating agents into DSS. It was noted that fitting agents with mobility, intelligence, and interactivity can enable them, extending the benefits these agents given to DSS users. To achieve real goal of DSS we can incorporate intelligent agents processing into DGMS with DBMS.DSS can be extremely beneficial to any organization’s overall performance.

As a future aspect both software agents and DSS are useful in all enterprises to make work efficiently. As market demand various programming languages and RDBMS’ can be used to achieve the desired goal.

REFERENCES

[1] Agre, P. & Chapman, D. (1987). PENGI: An Implementation of a Theory of Activity. *Proceedings of the Sixth National Conference on Artificial Intelligence*, 6(1), 268-272.

[2] Alter, Steven L. (1980). *Decision Support Systems: Current Practice and Continuing Challenges.* Reading, MA: Addison-Wesley Publishing Company.

[3] Applegate, L.M.; Konsynski, B.R; & Nunamaker, J.F. (1986). Model Management Systems: Design for Decision Support. *Decision Support Systems*, 2(1), 81-91.

[4] Blanning, R.W. (1993). Model Management Systems: An Overview. *Decision Support Systems*,9(1), 9-18.

[5] Bennett, J.L. (1983). Overview. In J. Bennett (Ed.) *Building Decision Support Systems*. Reading, MA: Addison Wesley Publishing Company, 1-14.

[6] Blanning, R.W. (1993). Model Management Systems: An Overview. *Decision Support Systems*,9(1), 9-18.

[7] Bonczek, R.H., Holsapple, C.W. & Whinston, A.B. (1981). A Generalized Decision Support System Using Predicate Calculus and Network Data Base Management. *Operations Research*, 29(2), 263-281.

[8] Bonczek, R.H., Holsapple, C.W., & Whinston, A.B. (1980). Future Directions for Developing Decision Support Systems. *Decision Sciences* 11(1), 616-631.

[9] Boy, G.A. (1997). Software Agents for Cooperative Learning. In J. Bradshaw (Ed.) *Software Agents*. Menlo Park, CA: AAAI Press/MIT Press, 223-245.

[10] Bradshaw, J.M. (1997). An Introduction to Software Agents. In J. Bradshaw (Ed.) *SoftwareAgents*. Menlo Park, CA: AAAI Press/MIT Press, 3-46.

[11] Bradshaw, J.M., Dutfield, S., Benoit, P., & Woolley, J.D. (1997). KAoS: Toward an Industrial- Strength Open Agent Architecture. In J.

[12] Bradshaw (Ed.) *Software Agents*. Menlo Park, CA: AAAI Press/MIT Press, 375-418.

[13] Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, 2(1), 14-23.

[14] Change, A., Holsapple, C.W., & Whinston, A.B. (1993). Model Management Issues and Directions. *Decision Support Systems*, 9(1), 19-37.

[15] Dolk, D. & Konsynski, B. (1984). Knowledge Representations for Model Management Systems. *IEEE Transactions on Software Engineering*, 10(6), 619-628.

[16] Elofson, G., Beranek, P.M., & Thomas, P. (1997). An Intelligent Agent Community Approach to Knowledge Sharing. *Decision Support Systems*, 20(1), 83-98.

[17] Ephrati, E., & Rosenchein, J. S. (1996). Deriving Consensus in Multiagent Systems, *Artificial Intelligence*, 87(1), 21-74.

[18] Foner, L., “What’s An Agent? Crucial Notions,” 23 May 1998, Franklin, S., & Graesser, A. (1996). Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*. Springer-Verlag, 21-36.

[19] Genesereth, M. R. (1997). An Agent-Based Framework for Interoperability. In J. Bradshaw (Ed.) *Software Agents*. Menlo Park, CA: AAAI Press/MIT Press, 317-345.

[20] Holsapple, C. W. & Whinston, Andrew B. (1996). *Decision Support Systems: A Knowledge-Based Approach.* St. Paul, MN: West Publishing Company.

[21] Keen, P.G.W. & Gambino, T.J. (1983). Building a Decision Support System: The Mythical Man-Month Revisited. In J. Bennett (Ed.) *Building Decision Support Systems*. Reading, MA: Addison Wesley

Publishing Company.

[22] Keen, P.G.W., & Scott Morton, M.S. (1978). *Decision Support Systems: An Organizational Perspective*. Reading, MA: Addison Wesley Publishing Company.

[23] Minsky, M. (1985). *The Society of Mind*. New York, NY: Simon and Schuster.

[24] Negroponte, N. (1997). Agents: From Direct Manipulation to Delegation. In J. Bradshaw (Ed.) *Software Agents*. Menlo Park, CA: AAAI Press/MIT Press, 57-66.

[25] Riecken, D. (1997). The M System. In J. Bradshaw (Ed.) *Software Agents*. Menlo Park, CA:AAAI Press/MIT Press, 247-267.

[26] Radermacher, F.J. (1994). Decision Support Systems: Scope and Potential. *Decision Support Systems*, 12(1), 257-265.

[27] Simon, H. (1960). *The New Science of Management Decision*, Harper & Row Publishers, New York.

[28] Sprague, R.H., & Carlson, E.D. (1982). *Building Effective Decision Support Systems.*Englewood Cliffs, NJ: Prentice Hall.Sun Microsystems (1998). Java™ Development Kit.

[29] Whinston, A. (1997). Intelligent Agents as a Basis for Decision Support Systems. *Decision Support Systems*, 20(1), 1.

[30] White, J. E. (1997). Mobile Agents. In J. Bradshaw (Ed.) *Software Agents*. Menlo Park, CA: AAAI Press/MIT Press, 437-472.

[31] Wooldridge, M., & Jennings, N.R. (1995). Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 10(2), 115-152.

[31]Software Agent: http://en.wikipedia.org/wiki/Software_agent