A Software Agent for Report Development

Dulce Magalhães de Sá, and Luis Joyanes Aguilar

***Abstract* — Several areas are increasingly in need of software applications to provide automatic report writing; for instance, to communicate company-customer matters, educative assessments or medical diagnoses. In this paper we present the prototype of an agent that integrates a report writing support system, which now has the capability to generate reports based on a reduced set of information elements provided by the human user. The agent in question has characteristics such as mobility and cloning and has the capability to write a certain amount of text automatically and independently. This system works both locally and on a network and never produces two identical reports, even if the contents provided by the user are exactly the same.**

***Index Terms* — Agent Architecture, Agent Classification, Intelligent Agents, Mobile Agents, Report Writing Systems.**

I. Introduction

The expansion of the Internet and mobile communication systems has contributed to increase the applicability of software agents and to improve their functionalities, especially due to the growing need for autonomy and mobility in applications for supporting business, business processes or decision-making. Software agent applications can vary between trade, industry, health and education or entertainment sectors. They can support e-learning systems, securities portfolio management systems, Internet search systems or, as far as this paper is concerned, a report writing support system, enabling users to employ automatic report writing in Portuguese, based only on a few topics or pieces of information. Of other related works, the one that most resembles the characteristics of the report writing support system, referred to in this article, is the RAFI system [7]. However, this system generates summaries of documents in French by selecting text contained in such documents, written by humans, there being no room for automatic text creation by software agents.

This paper is organized as follows: first we examine the 'agent' concept, its common designations, characteristics, classification of intelligence levels, with special emphasis on the particularities of mobile agents. Then we describe the functionalities of the agent in question in this paper and adapt to this specific case the intelligence classification of agents helping search engines, analyzing the performance of the prototype agent in its current stage of development. Finally, we address aspects to be improved concerning prototype performance and offer some advice for future works.

Manuscript received January 7, 2008. Dulce Magalhães de Sá is with the Faculty of Computer Science of the Universidad Pontificia de Salamanca, Campus de Madrid, Madrid, Spain (corresponding author phone: +34697233729; fax: : +34697233729; e-mail: dulce@ ieee.org). Luis Joyanes Aguilar is with the Faculty of Computer Science of the Universidad Pontificia de Salamanca, Campus de Madrid, Madrid, Spain (e-mail: joyanes@fpablov.org).

II. Conceptual Framework

In this section we provide a few definitions, properties, applications and classifications of agents as well as aspects inherent in their architecture, analysis and benchmarking .

A. Definitions

Agents are autonomous programs that act either locally or on a network within a specific context and towards a given set of goals [4]. Typically, these programs are designated as software agents or intelligent agents, but they are also named after the environment in which they operate or the goals they pursue, for instance information agents access different and remote sources of information in a Web environment.

According to [11], user-oriented programs that automate their interactions with the Web are, among other expressions, designated as software agents, intelligent agents or autonomous agents. A common 'intelligent agent' concept is that a software entity autonomously performs a task or assists the performance of tasks by a human user or another entity, especially by another agent or another software application.

Reference [2] regardan intelligent agent as being a physical or virtual entity with the following characteristics:

- Capability to operate in a specific environment;
- Capability to communicate directly with other agents;
- Aspect-oriented;
- Uses its own resources;
- Capability to understand its environment, though only partially;
- Has a partial view of its environment;
- Has skills and renders services;
- May be apt to multiply;
- Aims to attain the goals for which it was designed.

Agents are also designated according to an important attribute relating to the performance of functions and characterization, for instance the capability to move. Mobile agents are network-aware entities with a capability to change their performance environment by transferring their activity from place to place [1]. The property of mobility occurs when an agent has the capability to migrate through different platforms in order to perform the tasks it was designed for in the first place, interacting with local or remote users.

The mobile agent is programmed to be able to travel autonomously from one site to another through the Internet in

order to perform its tasks or queries in different servers [8]. A mobile agent can generate and distribute clones through different platforms in such a way as to secure performance of simultaneous tasks for distinct users, and this characteristic constitutes an important advantage as an alternative to overloading the application server through simultaneous access by different users.

B. Properties

The properties that distinguish agents from other programs can be classified into two categories: mandatory and orthogonal [9]. Mandatory properties relate to characteristics of autonomy, active or reactive decision-making capability, temporal continuity and orientation towards goals. Orthogonal properties relate to mobility, capability to communicate with other software agents or users, capability to collaborate with other agents and learning capability.

If we regard an intelligent agent as being a computer system capable of performing autonomous and flexible actions to achieve the goals it is set for, [5] regards flexibility as a property of the system with the following characteristics:

- Responsive – the agent understands its environment and responds opportunely to its changes;
- Proactive – the agent has a capability for initiative;
- Social – the agent is capable of interacting with other artificial agents or humans.

Intelligent agents have the following properties: autonomous, collaboration capability, learning capability, goal-oriented, flexibility, self-starting, individual character and mobility through different architectures and platforms [2]. The property of mobility as an agent requirement results from the evolution of the Web as an information warehouse. It is not a mandatory property, though it is becoming increasingly widespread in view of the growing implementation of distributed systems and mobile technologies.

C. Applications

Domains of agent application include industrial, commercial, medical, and entertainment applications. In industrial applications, the agents operate on control of processes or control of air traffic [5]. In commercial applications, agents can facilitate electronic trade, business process management in various sectors or information management, for instance by providing mechanisms to retrieve information. In the entertainment industry, agents may be used in games or in television, movies or interactive theater. Agents can search for information automatically, answer specific questions, inform about a particular event, provide information which is oriented and adjusted to a specific user, help acquire knowledge about a specific subject, support decis ion-making or render a wide range of automatic services.

Some types of agent application are mentioned by [9]. Single agent systems include applications such as personal assistants, information retrieval, e-mail management or search agents. Multi-agent systems can be used in a variety of applications, for instance resulting from the use of natural language, network management, communication systems or real-world simulations.

D. Classifications

Intelligent agent classifications may differ according to aspects inherent in control framework, computational environment, programming languages or application type. Reference [4] present an agent classification that brings together biological, robotic and computational agents at the top level of a tree representation. In this representation, computational agents are further divided into software agents and artificial life agents. Software agents are in turn subdivided into specific task agents, entertainment agents and viruses. Agents can act independently and be classed as single agent or else function as an agency system and be classed as multi-agent system, with their individual character losing importance. According to [9], the compulsory attributes of a single agent are: autonomy, decision-making capability, temporal continuity and orientation towards goals. As for multi-agent systems, in addition to the properties of single agents, there is a capability to communicate and collaborate with other agents of the system. Agents can also be classified based on intelligence levels.

Reference [12] refer to four levels of intelligence which are numbered from zero to three and applied to agents that collaborate with search engines. At the lowest intelligence level (level 0) the agent only responds to concise orders, for example accepting a specific URL to extract documents from the Web. At level 1 the agents facilitate search on the relevant Web pages, starting from a search initiative by a human user. At level 2 the agents manage to keep the user's profile, monitor the Internet and notify the user whenever they find results that match that profile. At level 3 the agents rely on components for learning and deducting the user's profile in order to help formalize queries or redirect the user to specific results.

E. Architecture

The main agent architecture models relate to peer-to-peer structure, group with facilitator structure and tree structure. In the peer-to-peer structure there is no centralized control, each agent stores its own metadata required to carry out activities and each agent has one connection. In the group with facilitator structure the facilitator controls the agents and the metadata is stored in a local database; facilitator and local server are connected. The agents interconnect through the facilitator. In the tree structure, the connection is hierarchic, there being local facilitators and a generic facilitator that coordinates the local facilitators. These in turn control the agents ascribed to them. The storage of metadata is secured by a local database. These architectures tend to generalize as they fail to address some aspects specific to mobile agents. According to [10] a mobile agent should stipulate life-cycle model, computational model, security model, communication model and navigation model. These models apply to characteristics specific to mobile agents and are intended to facilitate their design as well as improve performance of upcoming results.

According to [1], some important topics when approaching mobile agents include:

- code and data portability– the agent's architecture should have an independent format;

- secure execution models– the agent should secure privacy and safety in the execution environment;
- efficient execution– the agent should not require too many resources from its execution environments;
- suitable coordination models – coordination between mobile entities should be based on models that are adequate, efficient and safe.

The architecture of an agent server should also observe some functionalities inherent in agent management, their interaction, behavior, stability and safety. Reference [9] maintain that a generic mobile agent server architecture consists of the following elements:

- agent manager – to send and receive agents;
- agent execution environment– to facilitate code execution by agents;
- security manager – to protect the server from unauthorized agents;
- reliability manager – to ensure the agents operate in case of system failure;
- inter-agent communication manager – to facilitate communication between agents;
- applications gateway – to ensure safety during interaction between agent and server applications;
- directory manager – to enable the agents to identify their location and migrate.

F. Analysis and Benchmarking

One of the ways to analyze an agent's performance is to measure some aspects of behavior. Below is a benchmark model by [3] to unify autonomous agents.

intelligent agent = mental state + behavioral repertoire + control process

where

mental state = awareness + knowledge

behavioral repertoire = task structure + logics + communications

control process = deliberative control + reactive control

This model addresses different topics inherent in agent performance at environment level, at representations or 'mental states' level, at cognitive functions level and at control level. At the environment level characteristics such as perception, action and communication are taken into account. At the representation level beliefs, intentions and knowledge are taken into account. At the cognitive functions level reasoning, decision-making, planning, scheduling and learning are taken into account. And finally, at control level, deliberative control and reactive control are taken into account as well as autonomy.

As regards mobile agents, mobility aspects should be particularly born in mind, it being possible to use the attributes that give them advantages over other agents in order to analyze and measure the performance of this agent type.

As highlighted by [1], the advantages of mobile agents are:

- save bandwidth– agents can move freely to the resources they need, thus avoiding transfer of large amounts of data;
- service flexibility – the agent can move according the needs of the management of the service;
- mobile interaction – the agent can interact in different mobile computing system environments, according to its current location within a mobile system;
- connection performance – where the Internet is concerned, applications based on non-mobile communications often lead to errors or time-outs, so mobile agents can move to other available connections in order to continue with their tasks.

Through analysis of agent-based system architecture models it is possible to assess a system's performance. Reference [6] consider two analysis approaches: one relates to static characteristics and the other relates to dynamic characteristics.The first refers to organizational characteristics of the agent-based system, for instance system complexity factors. The second refers to behavioral aspects, for instance the run time of a specific task in a specific environment.

III. TEXT ORIENTED AGENT

The agent described here is a software application that integrates a report writing support system. In its current stage of development, it is a prototype that automatically generates a small amount of text (about 20 lines) in Portuguese based on a few information topics supplied by the human user. A characteristic of this application is that it never generates two identical texts, even when the information supplied by the user (or users) is identical. In order to obtain different results, the agent administers adjective expressions and noun expressions, playing with random effects applied to the relevant variables.

The primary purpose of the agent is to provide remote users with examples as to how they should write their reports or, ultimately, the finished report. Whereas in the current stage the result obtained is closer to resembling an introduction to the intended report, at a later stage the agent is expected to be capable of generating a larger amount of text as well as administering information topics. The field of application may vary, for example technical, business or medical reports or preparation of simple literary texts. In its current stage the agent prototype is capable of working in a Web environment, though it is expected to expand its scope to mobile devices such as cellular phones, as they do not require that the users introduce large amounts of information.

A. Agent Classification

The prototype addressed in this paper is a text-oriented single-agent software that asks the user to supply a text with: identification of report owner; organization represented; general subject of report; and specific area of report. The classification of Web search agents by [12] can stretch to other types of agents, generalizing their base fundamentals.

As far as this paper is concerned, at intelligence level 0 the agent is unable to add value to the information from the user, all it does is to execute a concise order from the user, for example sending information to another user based on a name and accessing an electronic name and address database. At level 1 the agent is activated on the initiative of the user and has the capability to improve the information received by

generating support forms for acquisition of knowledge or decision-making by the user. It manages to add value to the original piece of information. At level 2 the agent would be able to recognize the user's profile in a second interaction and would adapt its behavior accordingly, by storing profile update information at the end of the session. At level 3 the agent has the capability to learn, deduce and automatically make decisions in such way as to guide the users through the formalization of their needs and present them with appropriate solutions. Only levels 2 and 3 are thought to truly concern intelligent agents. At level 0 the agent is thought to be devoid of intelligence and at level 1 the agent is thought to be quasi-intelligent.

Given this interpretation of agent intelligence, the agent presented in this paper is, in its current stage of development, a quasi-intelligent agent with some characteristics of level 3 agents, especially the fact that it is able to make decisions autonomously. In reality, based on the information supplied by the user and the information base held by the agent, it has the capability to decide or select adjective and substantive expressions with which to produce the report text that will be presented to the user.

B. *Agent Properties*

In its current stage of development, the agent has the following properties:

- Autonomy – the agent is autonomous in regard to the portion of text it generates and is also autonomous in that it selects and combines different textual expressions from its information base. Thus, for every report the agent creates different sentences. As for the remaining text contained in the report, the agent resorts to its own information base.
- Decision capability – the agent decides on the composition of the sentences, based on random matching of textual expressions from its information base as well as information topics from the user. For every expression 'match', the agent presents different results, thus writing unique reports in each utilization session.
- Temporal continuity – for every expression 'match', the agent comes up with successive 'matches' until it produces the report to be presented to the user. At the end of this process, the agent waits for information updates or current session closing.
- Goal-oriented – the agent is focused on a primary goal, which is to produce a report in Portuguese, based on some pieces of information originally provided by the user. In order to achieve this, the agent has to meet other specific goals at each stage to finally fulfill the primary goal. In section III. *D. Agent Analysis* we will address the agent's task cycle and specific goals in more detail.
- Mobility – the agent has the capability to move between different architectures and platforms. It also has the ability to function in different environments. The agent relies on its own resources as it operates independently, with no strings attached. It can operate both locally and on a network.
- Cloning – the agent does not need resources to assist different simultaneous users, due to its cloning capability. The agent server can send different clones to different users or to different sessions of the same user. If the agent is served by a network, for each user/session, the server sends one agent clone. If the agent is in local mode, it can clone itself to different simultaneous sessions of the same user or machine or device.
- Collaboration – in its current stage of development, the agent lacks this property, having only potential to have this property implemented with no drastic changes to the agent's body.

C. *Agent Architecture*

The report writing support system where the agent operates consists of five components: user, formulation, agent, resources and results, which interact so that a report can be produced by the agent, as illustrated in Fig.1 below. The various system components are responsible for gathering, handling, storing and spreading the results for which the system was designed, the agent being responsible for the bulk of the work.

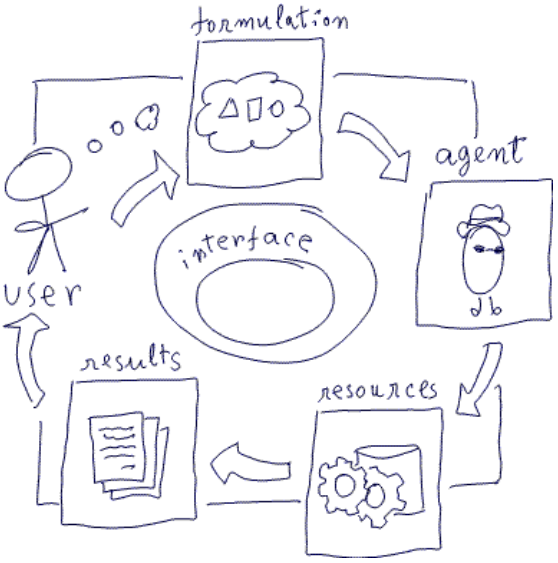

FIGURE 1
COMPONENTS OF THE REPORT WRITING SUPPORT SYSTEM

User. Users can be of two types: humans or other agents with specific functions differing from the agent referred to in this paper. In the current stage of development, the only type of user is human. It is the human user that takes the initiative to 'activate' the agent.

Formulation. The formulation of requests to the system is a mental formulation from the user according to what he/she intends to obtain from the system, the user being responsible for supplying the agent with the following information: identification of report owner, organization represented, general subject of report, and specific area of report.

Agent. Once 'activated' by the user, it is the job of the agent to ask for pieces of information that will assign a meaning to the report. The agent will process the information obtained as well as other textual objects and tools available in order to execute consecutive 'matches' until a unique report is produced for the user. In order to work out the 'matches' the

agent relies on algorithms that will help it process consistently the adjective and noun expressions stored in its information base and combine such expressions with the information provided by the user.

Resources. Two types of resources can be considered: agent resources and system resources. The agent's resources include the information base with adjective and noun expressions, the processing models (algorithms) from various stages developed by the agent, some form of random memory information storage and meta-information. As for system resources, the usual information system resources apply.

Results. This is the system's output, which in this particular case are text reports written in Portuguese, consistent with the items of information supplied by the users and generated by the agent or its clones. In the current development stage, the agent receives one or two text lines and produces about 20 lines with phrases adapted to the information originally supplied.

As mentioned earlier on, the agent can be located in a local server or on a network. If located on a network, its architecture is the peer-to-peer type, as there is no centralized control over the agent's activities and the agent is connected and relies on its own resources. Fig. 2 illustrates agent behavior on a network. In this figure, future agent functionalities are considered, for instance telephone interaction.

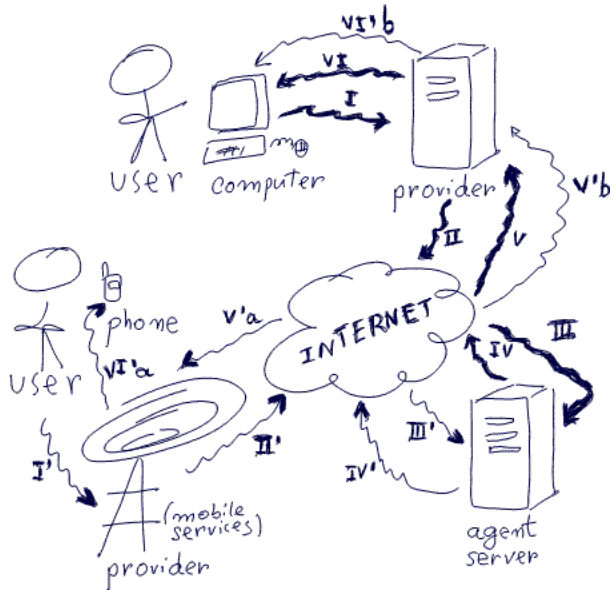

FIGURE. 2
THE AGENT ACTIVITY ON THE NETWORK

The user contacts the agent through a computer or mobile phone, accessing the relevant service provider (activities I or I') and being redirected to the Internet (II for computer and II' for telephone) to reach the agent server (activities III or III' respectively). The agent server then sends agent clones, and activities IVi, Vi and Vii are executed and subdivided into various interactions between the users and relevant clones until the latter obtains all the information needed to prepare the reports and present them to the users.

The life-cycle model includes the processes listed below, where Ag represents the agent, U represents the user, INFO represents the information contents and DB represents the agent's information base consisting of expressions in Portuguese.

1. Ag *queries* U *until it obtains all* INFO.
2. Ag *memorizes* INFO.
3. Ag *reads* DB.
4. Ag *activates processing algorithm.*
5. Ag *processes* INFO *with* DB *data.*
6. Ag *memorizes result.*
7. Ag *formats result.*
8. Ag *presents report to* U.
9. Ag *awaits acceptance by* U *or new request.*

(If *new request is executed*, Ag *resumes process in* 1).

The main characteristic of the agent on a network environment is its capability to clone itself, saving server resources and bandwidth. The agent can operate on different platforms and architectures, in a synchronized way, due to data and code portability. It is the job of the server to manage the agent, sending its clones to any users requesting them.

D. Agent Analysis

The algorithm supporting the agent is object-oriented. In its current stage of development, this algorithm is oriented to a main object, that is, a text. At a later stage, it could be expanded to include other object types such as images or graphs generated from user's information. Currently the agent algorithm has three defined objects, namely adjective expressions, noun expressions and information topics supplied by the user. Next, we present a portion of the agent algorithm in pseudo-code (Java like), which generates textual expressions at random based on adjectives and nouns.

```
// Get a random textual expression from the "texp"
array.
function getTexp (texp)
{ var s = ""
    var i = 0
    while ( s == "" )
    { i = randnum (texp.count )
        if (texp[i] == "" )
        { i = ( i + 1 ) % texp.count }
      else
        {   s = texp [i]
            texp [i] = "" }
    }
    return s
}
//  Return a random textual expression.
function textadj () { return getTexp ( textadj )
}

function textnoun() { return getTexp ( textnoun
) }
```

The result obtained from applying the above mentioned portion of the agent algorithm is illustrated by the two following paragraphs, each of which was obtained at a different interactive session with the agent. Each paragraph was extracted from around 20 lines of text produced by the agent in each of the sessions. In both sessions, the user supplied the same information contents to the agent. The contents are:

- [Identification of report owner(s)] = John Doe; Mary Doe
- [Organization of report owner] = J&M Doe Organization
- [General subject of report] = Gestão de Informação

(Information Management)

- [Specific area of report] = Análise de Sistemas baseados na Web (Web-based System Analysis)

The resulting textual differences are shown underlined.

Extract of Session 1:

O tópico de *Análise de Sistemas baseados na Web* tem vindo a gerar alguma discussão na disciplina de *Gestão de Informação*. Os estudos desenvolvidos focalizam-se essencialmente em aspectos funcionais. No entanto, o trabalho desenvolvido por *J&M Doe Organization* de autoria de *John Doe* e *Mary Doe* demonstra que pela análise de aspectos organizacionais, mais avanços podem ser conseguidos.

[The topic of *Web-based System Analysis* has been stirring some discussion in the discipline of *Information Management*. Studies developed essentially point to functional aspects. However, work developed by *J&M Doe Organization* and signed by *John Doe and Mary Doe* demonstrates that through analysis of organizational aspects, further advancement can be achieved.]

Extract of Session 2:

O tema de *Análise de Sistemas baseados na Web* tem contribuído para o debate na área de *Gestão de Informação*. Os estudos realizados tiveram o seu fulcro em conceitos de funcionalidade, mas o trabalho desenvolvido por *J&M Doe Organization* de responsabilidade de *John Doe* e *Mary Doe* demonstra de que modo resultados diferentes podem ser alcançados.

[The subject of *Web-based System Analysis* has helped promote debate in the field of *Information Management*. Studies carried out so far have been grounded on functionality concepts, but the work developed by *J&M Doe Organization* under the responsibility of *John Doe and Mary Doe* demonstrates how different results can be attained.]

The run time to execute reports from the moment information is supplied by the user until a final report is produced is approximately 30 seconds for about 20 lines written and formatted by the agent. Run time includes the creation of a HTML page for each report the delays are ascribable to network traffic and the processing speed of the machine supporting the user's activities. The machine used in the tests is a Pentium II processor computer with 64 Mb of RAM and MSWindows 2000 operating system, connected through a 56Kb modem.

IV. Conclusion

Agents are goal-specific programs that execute tasks autonomously or assist users in executing them, providing solutions or guidance. In this paper we presented a report-oriented agent prototype with the following properties: autonomy to generate and format text; decision capability to select textual expressions; temporal continuity, mobility to shift to different architectures and platforms, cloning capability and capability to collaborate with other agents.

The components of the report writing support system are the user, the request formulation process, the agent and its clones, the resources and the results or output. The agent operates locally or on a network with a peer-to-peer architecture.

In its current stage of development, the agent prototype manages to generate a page of text in Portuguese. The page is formatted and produced in HTML. The run time to execute tasks from the moment relevant information is supplied until the final report page is produced is around half a minute.

This prototype still has some flaws, especially when distinguishing between singular and plural textual expressions, as well as distinguishing between masculine and feminine nouns. In future work, we intend to improve on these aspects, providing the agent with a capability to generate a larger amount of text, developing its potential to collaborate with other agents and enabling it to be capable of interacting with mobile equipment, for instance cellular phones.

References

[1] Cabri, G., Leonardi, L. and Zambonelli, F., 2000. Agents for Information Retrieval: issues of mobility and coordination. *In Journal of Systems Architecture*, Vol. 46, pp. 1419-1433.

[2] Ferber, J. and Gutknecht, O., 1998. A meta-model for the Analysis and Design of Organizations in Multi-Agent Systems. *In Proceedings of the 3rd International Conference on Multi Agent Systems*, IEEE Computer Society, pp. 128-137.

[3] Fox, J., Beveridge, M. and Glasspool, D., 2003. Understanding Intelligent Agents: analysis and synthesis. *In AI communications*, Vol.16, No. 3, pp. 139-152.

[4] Franklin, S. and Graesser, A., 1996. Is it an Agent or just a Program. *In Proceedings of the International Workshop on Agent Theories, Architectures and Languages*.

[5] Jennings, N. and Wooldridge, M., 1998. *Applications of Intelligent Agents. In Agent Technology: foundations, applications and markets*, N. Jennings and M. Wooldridge (Eds.), Springer-Verlarg, pp. 3-28.

[6] Lee, S. and Hwang, C., 2004. Architecture Modeling and Evaluation for Design of agent-based System. *In Journal of Systems and Software*, Vol. 72, pp. 195-208.

[7] Lehmam, A., 1999. Text Structuration Leading to an Automatic Summary System: RAFI. *In Information Processing & Management*, Vol. 35, pp. 181-191.

[8] Klush, M., 2001. Information Agent Technology for the Internet: a survey. *In Data & Knowledge Engineering*, Vol. 36, pp. 337-372.

[9] Manvi, S. and Venkataram, P., 2004. Applications of Agent Technology in Communications: a Review. *In Computer Communications*, Vol. 27, pp. 1493-1508.

[10] Oham, V. and Karmouch, A., 1998. Mobile Software Agents: an overview. *In IEEE Communications*. Special Issue Mobile Agent Telecommunications, pp. 25-37.

[11] Special Libraries Association, 2000. Just what is an Intelligent Agent? *In Information Outlook*. Available on http://www.findarticles.com/p/articles/mi_m0FWE/is_7_4/ai_64694222 [Accessed 5 January 2008]

[12] Turban, E. and Aronson, J., 2001. *Decision Support Systems and Intelligent Systems*. Prentice Hall.