International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 09 Issue 10 ǁ 2021 ǁ PP. 77-85

An Overview of Multi-Agent System for Academic Advising (MASACAD)

Omankwu, ObinnayaChinecherem, Ubah Valentine and Ngene, Chigozie Chidinma

[1]*Computer Science Department,Michael Okpara University of Agriculture, Umudike Umuahia, Abia State, Nigeria*
[2]*Computer Science Department,Michael Okpara University of Agriculture, Umudike Umuahia, Abia State, Nigeria*
[3]*Computer Science Department, Nnamdi Azikiwe University, Awka Anambra State, Nigeria,*

Abstract
In this paper we present Multi Agent System Academic Adviser (MASACAD), a multi-agent system that learns to advise students in selecting their courses. The main idea is to approach information customization using a multi-agent paradigm in combination with a number of aspects. The growth and advancement in the Internet and the World Wide Web has led to an explosion in the amount of available information. This staggering amount of information has made it extremely difficult for users to locate and retrieve information that is actually relevant to their task at hand. Dealing with this problem of ''information overload'' will need tools to customize the information space.

Keywords: *Information customization; Multi-agents; E-Learning;*

Date of Submission: 06-10-2021 Date of acceptance: 20-10-2021

I. Introduction

The recent proliferation of personal computers and communication networks has a strong scientific, intellectual and social impact on the society. Rapidly evolving network and computer technology, coupled with the exponential growth of the services and information available on the Internet, has already brought us to the point where hundreds of millions of people should have fast, pervasive access to a phenomenal amount of information, through desktop machines at work, school and home, through televisions, phones, pagers, and car dashboards, from anywhere and everywhere. The challenge of complex environments is therefore obvious: software is expected to do more in more situations, there are a variety of users, there are a variety of systems, there are a variety of interactions, and there are a variety of resources and goals.

To cope with such environments, the promise of information customization systems is becoming highly attractive.

The recent popularity of the World Wide Web (Web) has provided a tremendous opportunity to expedite the dispersement of various information creation/diffusion infrastructures. The mass of content available on the Web raises important questions over its effective use. With largely unstructured pages authored by a massive range of people on a diverse range of topics, simple browsing has given way to filtering as the practical way to manage Web-based information. Today's online resources are therefore mainly accessible via a panoply of primitive but popular information services such as search engines.

In this paper, following the same long-term objective of providing a complete E-Learning environment for students and striking for the more general goal of information customization, we describe Multi-Agent System for ACademic ADvising; (MASACAD), a multi-agent system that advises students by adopting a machine learning paradigm. Machine learning methods can be used to deal with many different aspects of the problem of advising.

Intelligent agents

Because of the information overload problem created by the unstructured nature of the Web, the trend to use intelligent agents as a promising solution for assisting and facilitating the processes of information extraction and information customization was increasing in recent years. [46–49]are just some examples.

The intelligent agent's roles normally revolve around satisfying user demands for information. In general, the agents involved take on the task of gathering information to meet a variety of user needs. In [46], for example, it is suggested that a more comprehensive framework would include both supply and demand agents. Supply agents provide information to demand agents. Specifically, supply agents effectively configure information for information consumers. Demand agents search for needed information.

These agents are typically designed to search the Web for information to meet user goals. There are several types of these agents being developed in recent times. One example is Internet Softbot, which was developed at the University of Washington [47].Softbot is a prototype implementation of a high-levelassistant. In contrast to systems for assisted browsing or information retrieval, the Softbot can accept high-level user goals and dynamically synthesize the appropriate sequence of Internet commands to satisfy those goals [47]. A third type of agent may exist, called a broker agent, which match supply agent capabilities with demand agent needs [46].

Agents could also play various roles in an agent-enabled system architecture. Some could act in the role of intelligent user interface managers, drawing on the resources of other agents working behind the scenes [50–54]. Such agents would work in concert to help coordinate the selection of the appropriate display modes and representations for the relevant data [55], incorporating semantic representations of the knowledge in the documents to enhance navigation and information retrieval [56–60,129]. Because the layout and content of the views would be driven by context and configuration models rather than by hand-crafted user-interface code, significant economies could be realized as the data and software components are reused and semi-automatically reconfigured for different settings and purposes. Some agents might be represented explicitly to the user as various types of personal assistants [61]. Ideally, each software component would be ‘‘agent-enabled’’, however for practical reasons components may at times still rely on traditional inter-application communication mechanisms rather than agent-to-agent protocols.

Both of the views discussed earlier (supply/demand agents, intelligent user interface manager/agents working behind the scenes) can be easily identified in the multi-agent system described in this article. The Bee-gent framework is used to ‘‘agentify’’ the system components and make them rely on agent-to-agent protocols.

Though research in the domain of intelligent agents had been ongoing for more than 20 years, agents really became a buzzword in the popular computing press (and also within the artificial intelligence and computing communities) around 1994. During that year several key agent-related publications appeared. A report titled ‘‘Intelligent agents: the new revolution in software’’ [62]wildly speculated on market sector totals for agent software and products by the year 2000. The first of now several special issues of the Communications of the ACM on agents appeared in 1994 and it included articles like ‘‘Agents that reduce work and information overload’’ [61]and ‘‘How might people interact with software agents’’ [63]. Then there was a Byte article on ‘‘The network with smarts’’ [64]. Indeed, during late 1994 and throughout 1995 and 1996, there was an explosion of agent-related articles in the popular computing press. It is no coincidence that this explosion coincided with that of the Web. The field has clearly matured since with the publication of certain key papers and books including [65–69]amongst others. Several annual and biennial conferences now grace the area including the International Conference on Multi-Agent Systems (ICMAS), the International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM), and the International Conference on Autonomous Agents (AA). ICMAS, PAAM and AA held their first conferences in 1995, 1996 and 1997, respectively. These conferences and numerous other agent-oriented national and international workshops, many agent special issues of journals, agent books, agent standardization initiatives such as FIPA, the Journal of Autonomous Agents and Multi-Agent Systems (JAMAS), the book ‘‘Readings in Agents’’ [70], initiatives such as the AgentLink Network of Excellence, etc. all bear testimony to a quickly maturing area.

Despite the narrow scope of early agent research, a fair number of useful agent systems have been developed. In contrast to former days, information has long lost its automatic association with text. In addition to text there are graphics, image, speech, video, and mixtures of all these information types. These ‘new’ information types gain significance not only in everyday life but also within the professional realm of enterprises that are faced with today’s information flood. As a consequence, the methods and algorithms devised in the early days of agents have to be reconsidered, adapted, or even re-developed. New agent methods and algorithms are now emerging because of the efforts of both IT professionals in companies and agent researches in academia. The agent systems emerging must have the ability to cope with hypermedia information, e.g. hypertexts and a multitude of different media. The problem in enterprises is often that it is unknown at the time of purchasing or devising new systems what kind of information must be considered and – above all – in what form this information will be available when the agent system is in operation. An important problem is therefore: how agent systems can be used to find and deal with unknown information.

Some current agent research work is therefore focusing on issues such as:

Adaptive choice of and balanced access to sources: the emphasis here is on collaborative learning of how to select information sources/search engines based on the principle of maximum expected utility and on improving resource utilization and global performance (load balancing) [71].

Cooperative and adaptive distributed data mining: the emphasis here is on collaborative learning of how and when to apply methods for knowledge representation and discovery in distributed information sources [72].

There are also many other problems that arise when using agents to implement IC systems. The major problem of Internet and Web, namely the huge amount of heterogeneous information sources consisting of large volumes of (non-, semi-) structured, volatile (dangling links, relocated), redundant (mirrored, copied) data, causes an ‘‘information overload’’ of the user. This complicates the process of searching for relevant information (‘‘needle-in-the-hay-stack’’) and the process of coping with system, structural and semantic heterogeneity [73].

The impact of heterogeneity and globalization on information overload is sketched by the following problems and steps towards solving them:

How to reconcile differences in data structure? Reconcile different data models, content representation (schematic).

How to deal with semi-structured or unstructured data? define appropriate wrapper and translator.

What about media data? Extract and correlate various heterogeneous audio, video and image data.

How to cope with differences in underlying terminologies? choose and map different domain ontologies used to describe content of information sources (semantics).

How to reconcile contextual heterogeneity? Differentiate between implicitly assumed contextual interpretations.

How to cope with differences in query languages? use specific languages and operations for retrieving information.

How to properly model content of sources? Different levels of abstraction of information modeling at sources.

How to query and fuse content from sources? Identify/focus and keep track on which source has what subset of relevant information and combine partially relevant information from sources.

How to discover relevant information resources? Determine and keep track of relevant information sources, formulate appropriate requests to (potentially millions of) sources.

For these reasons, a framework that offers not only basic agent functionality but also the thread of control and the structure of the yet unknown agent application to be devised should be developed. This framework should allow a great deal of flexibility to a system developer and provide, at the same time, maximum support.

E-Learning,

E-Learning is a valuable extension of the distance education paraphernalia, enabled by the new information and communication technologies. Distance education normally occurs in a different place from teaching and as a result requires special techniques of course design, special instructional techniques, special methods of communication, as well as special organizational and administrative arrangements (Moore, 1996). E-Learning is often described as the use of network technology, namely the Internet, to design, deliver, select, administer and extend learning.

One key issue in E-Learning is communication between participants, for which there are two basic types of technological solutions: asynchronous and synchronous. In the asynchronous approach, the interaction between parties does not require them to be engaged at the same point in time. In synchronous communications the interaction between participants requires simultaneous engagement of the partici-pants.

Online education is today a reality in many sectors of the society, especially in educational centers such as colleges and universities, increasingly high schools, and also professional groups demanding continuous access to education. Several years ago this new educational method was considered as an experimental approach with more disadvantages than advantages. However, today it should be considered not only a complementary educational resource but also a serious alternative that competes to conventional and now classical methods. Both methods will coexist and the logical initial inertia to ignore the new opportunities provided by the new media should be reduced and be faced sooner better than later in the same manner in which many other areas were modified throughout history.

Obviously the adaptation to the new features and services of the E-Learning environment is not immediate and requires experience, time, investment, pedagogical and technical resources, and government or campus administration support.

Artificial intelligence in education

Adding interactivity and intelligence to Web educational applications is considered to be an important direction of research. There are quite a few number of interesting developments and tools that are being developed in this field. De Rossi [98]provides a collection of sources on the topic. McArthur et al. [99], for example, view the application of AI to education, as a continuum, with omniscient, tutor-controlled ITS (intelligent tutoring systems) on one end, and completely student-controlled ILEs (interactive learning environments) on the other. The continuum represents a variety of different ways of dealing with the problems and weaknesses of the two opposing approaches. Ideas from AI neither support ‘‘instructionist’’ (ITS) or ‘‘constructionist’’ (ILEs) views of teaching and learning wholeheartedly; rather, they can and will be used to implement a diverse set of methods of learning and teaching, perhaps aiming at different kinds of learning outcomes. Jafari argues that commercial learning management systems would benefit from the development of agent-based capabilities and conceptualizes three types of agents to assist teachers and students: Digital Teaching Assistant, Digital Tutor, and Digital Secretary. Reimann[101]looks at the current state of the art in supporting learning groups. Managing online collaboration by means of intelligent support can take a number of forms. In a taxonomy for collaboration management tools three types of tools can be used to support collaboration management: mirroring, meta-cognitive and advise tools. He raises the question on how to come to a rational decision which kind of tool to employ, and concludes that tracing and mirroring is effective when compared to the ‘‘no further support’’ condition. Smith and Blandford[102]try to address the problem that commonly faces hypertext users, particularly in educational situations, namely, the difficulty of identifying pages of information most relevant to their current goals or interests. They present a prototype system to test the technical feasibility of using machine learning techniques to analyze browsing patterns within hypertext, and to use this analysis to provide adaptive navigational support without the need for pre-defined stereotypical profiles or user relevance feedback. They report that the quantitative results from the evaluation study are indicative of improved performance for the adaptive version and were helpful in identifying a number of issues about the design that should be addressed. Some other recent applications have pointed out the potential of computational intelligence in the area of intelligent Web-based education. A special issue of the Journal of Interactive Learning Research (JILR) [103] discusses recent developments in this rapidly progressing and technologically attractive area that has an important social impact. In the present contribution, we demonstrate, on the example of a system for the academic advising of students, the capability of exploiting the digital infrastructure enabled by the online mode of teaching and learning to extract and infer useful knowledge.

The problem domain: academic advising

Having decided on a research direction, the following question emerges. What constitutes a good domain and problem? The key characteristic of an interesting domain is that there is a variety of resources in differing formats but there is some common overall structure. Too much structure reduces the problem to known methods. Too little structure makes the problem very difficult. Having structure is useful to guide the search and identification of relevant information.

We will illustrate our ideas using MASACAD, an example consisting of an E-Learning application. In this application the focus is on academic advising for students.

Academic advising

Advising Services are committed to supporting students throughout the development and achievement of their educational goals. The foundation of this support is the establishment of an educational partnership between the student and his or her academic advisor. Advisors often are thought of as the people who help students identify appropriate courses for their degree program. However, the course registration process is only a piece of the partnership that helps to guide students along other paths of their educational development. Advising Services adhere to a developmental approach to academic advising, where the advisor is a facilitator, and partner in the learning process. It is understoodthat the advisor, while a valuable resource and educator, is not the holder of all truths, but rather, will assist the student to discover and develop his/her educational plan. By working collaboratively with students, advisors help to ensure that they have a successful college experience.

An effective advising relationship is a partnership between the student and the advisor. Both the advisor and the student have responsibilities in this relationship. All students are therefore assigned an academic advisor whose interests are compatible with those of the major. The advisor is a very helpful companion who provides insight during the college experience. The advisor also provides some assistance with:

Course selection, major requirements, and scheduling.Applications to graduate or professional programs. Student activities, including student/faculty research.Employment opportunities.

But the student, as the advisee, must take the initiative and request to meet with his/her advisor. For most majors, regular meetings with the advisor are of benefit to the student during his/her college experience.

Why a software assistant is needed for academic advising?

Academic advising is designed to provide the student with the information and the support that he/she needs to make informed plans and decisions about his/her program and to achieve his/her educational goals. The student is always in charge of his/her education – he/she makes the decisions. Academic advisors can help the student learn what he/she needs to know, sort out his/her questions, think through his/her options, identify shortcuts, avoid pitfalls, and make his/her program right for him/her. Advisors help the student find opportunities, avoid problems, and work through any difficulties he/she might encounter. Just as one might look to a lawyer for advice on a legal problem, or to a financial advisor for help with investment plans, one may think about an academic advisor as the consultant of the student on his/her educational planning.

Academic advising is an important key to academic success. The relationship between quality advising and student success is significant. That is why many people on campus are involved in the process. Faculty, staff, students and professional advisors contribute to the advising relationship by:

Encouraging ongoing, supportive, and informed contact with students.

Explaining policies, procedures, and academic requirements.Exploring skill levels, such as writing, mathematics, and study skills.

Making referrals to campus resources.

Assisting with degree planning as well as career exploration and preparation.

Ultimately, however, the responsibility for seeking adequate academic advising belongs to the student. Students must:

Why a software assistant is needed for academic advising?

Academic advising is designed to provide the student with the information and the support that he/she needs to make informed plans and decisions about his/her program and to achieve his/her educational goals. The student is always in charge of his/her education – he/she makes the decisions. Academic advisors can help the student learn what he/she needs to know, sort out his/her questions, think through his/her options, identify shortcuts, avoid pitfalls, and make his/her program right for him/her. Advisors help the student find opportunities, avoid problems, and work through any difficulties he/she might encounter. Just as one might look to a lawyer for advice on a legal problem, or to a financial advisor for help with investment plans, one may think about an academic advisor as the consultant of the student on his/her educational planning.

Academic advising is an important key to academic success. The relationship between quality advising and student success is significant. That is why many people on campus are involved in the process. Faculty, staff, students and professional advisors contribute to the advising relationship by:

Encouraging ongoing, supportive, and informed contact with students.

Explaining policies, procedures, and academic requirements. Exploring skill levels, such as writing, mathematics, and study skills.

Making referrals to campus resources.

Assisting with degree planning as well as career exploration and preparation.

Ultimately, however, the responsibility for seeking adequate academic advising belongs to the student. Students must:Know and meet degree requirements.

Ensure timely progress toward a degree through appropriate course selection.

Be aware of current academic and departmental information.

In order to help the student, improve the advising process and make it easier, and overcome the many problems that may occur such as:

The limited number of advisors that are available in contrast to the huge number of students.

The advisors are not available all the time.

Some advisors are new to the university and do not have enough knowledge/experience with advising.

Serious consequences may occur if mistakes are made, for example, during course selection.

An intelligent assistant in form of a computer program will be of great interest. Such an intelligent assistant will automate the advising process in the university, and hence, help the university in its efforts towards introducing new technologies. It will also simplify the task of faculty, staff, students, and professional advisors and make it possible to save time and effort and prevent mistakes. All these advantages are added to the many advantages of any assistant software that is used to solve problems that ordinarily require human expertise such as:

Permanence: expertise does not leave with personnel.

Multiple experts: knowledge from multiple experts can be combined.

Increased reliability: increased confidence that a decision is correct.

Fast response: may respond faster than a human.

Steady and unemotional: stress and fatigue are known to affect human performance.

Increased process quality: provide consistent advice, there-fore reducing the size and rate of errors.

Restriction of the general goal of academic advising

The general goal of academic advising is to assist students in developing educational plans which are consistent with academic, career, and life goals and to provide students with information and skills needed to pursue those goals. More specifically, advisors will assist students in the following ways:

Guide students through the university's educational require-ments.

Assist in scheduling the most appropriate courses. Introduce them to pertinent resources.

Promote leadership and campus involvement. Assist in career development.

Assist students with the timely completion of their degree. Help students find ways to make their educational experience

personally relevant.

The goal of academic advising, as stated above, is too general because many experts are involved and because a huge amount of expertise is needed. Hence, realizing an intelligent software assistant that is able to deal with all the details shown above will be too difficult, if not impossible. In the following implementation we will therefore restrict academic advising and understand it as just being intended to provide the student with an opportunity to plan programs of study, select appropriate required and elective classes, and schedule classes in a way that provides the greatest potential for academic success.

Agent approach

A convenient metaphor for building software to interact with the range and diversity of online resources is that of an agent. An agent is a person that performs some task on your behalf. We would like to have a program that navigates the online resources to find the specific information that is strongly suspected to be there. You care about the result, and are happy to delegate the process to an assistant. You expect an agent to act even if all the details are not specified, or the situation changes. You expect an agent to communicate effectively with other agents.

Agents can be viewed as a new model for developing software to interact over a network. This view has emerged because of the predominance of networks in the world. Information, knowledge, and electronic resources in general, are distributed across a network and programs and methods are needed to access them and present them in a customized manner. Using agents adds a layer of abstraction that localizes decisions about dealing with local peculiarities of format, knowledge conventions, etc. and thus helps to understand and manage complexity. Agents should therefore be seen as an abstraction that appears to provide a powerful way of conceptualizing, designing, and implementing a particularly complex class of software systems.

Multi-agent systems are systems composed of multiple interacting agents, where each agent is a coarse-grained computational system in its own right. The hypothesis/goal of multi-agent systems is creating a system that interconnectsseparately developed agents, thus enabling the ensemble to function beyond the capabilities of any singular agent in the setup. To arrive at a multi-agent solution, concepts such as those found in object-oriented computing, distributed computing, expert systems, etc. are necessary but do not suffice because distributed computing modules are usually passive and dumb. Also, their communications are usually low-level while multi-agent systems require high-level messages. Lastly, and importantly, multi-agent systems applications require a cooperation-knowledge level while these systems (object-oriented computing, expert systems, etc.) typically operate at the symbol and knowledge levels [13].

Benefits and limitations of MASACAD

Using MASACAD, the student is able to obtain up to date advice at any time without needing to consult human academic advisers. He needs only to initiate a query and the system will do the rest. The student needs only to state what he wants. In this case he wants advice about which courses he should enroll himself in for the coming semester. The system is responsible for deciding which resources to invoke. In this case the system will consult the course announcement Web site and the database containing the student details. The system is also responsible for the way how to invoke the resources once found. In this case, each of the resources (Web site or database) is invoked (queried or searched) by the agent wrapper, which is created manually (by the programmer). The mediation agent, also created manually, migrates between the agent wrappers transporting queries and collecting answers to the queries.

MASACAD, in addition to providing a solution for the problem of academic advising, was also used to demonstrate, on an example, the suitability of the multi-agent paradigm for dealing with information customization and personalization. Information customization and personalization provides users with relevant content based on their needs and interests and this is exactly what MASACAD does. It provides the user (student) with an interactive, personal Web experience and customizes the information to the user's desire and needs. The preferences (profile) of the user determine the behavior of the system. Using the multi-agent approach combined with other techniques, we were able to devise an elegant solution to a problem that depends on distributed information existing in differing formats. Conversely, as mentioned earlier, the distributed and dynamic nature of today's information resources makes the adoption of the multi-agent approach necessary.

However, there are some other important characteristics of information customization systems that were not demonstrated on this example. In general, in an information customization system there should be some strategy for discovering the location of potential useful information for example among the whole Web, among a reasonable part of the Web, among a set of databases, or among any other heterogeneous set of resources. Also it should be possible to automate the creation of the mediation agent and the agent wrappers, which will facilitate the (still manual) creation of the whole multi-agent system. The multi-agent system to be created depends on the resources to be invoked and on their locations. The automation task may therefore be very difficult: the resources can be numerous and may have very different formats (they may be unstructured Web pages, databases, agents) and this complicates the creation of the agent wrappers (how to wrap an unknown application?), as well as the creation of the mediation agent (which route to take to navigate to the different agent wrappers, how to communicate with the different agent wrappers).

Another limitation of MASACAD in its current version is that is does not exploit any kind of feedback (except that it asks the student about the courses he is interested in before providing the advice) to refine its functionality. Feedback is a useful mechanism for providing the system with information about how the advice

it produces is assessed by the user. Having this information can lead to significant improvements in the performance of the application.
These and other limitations of MASACAD as a system for academic advising as well as possible improvements are worth to focus on in future work.

The Bee-gent system

As a solution to the problems of network communication, we use Bee-gent (Bonding and Encapsulation Enhancement Agent) [29], a communication framework based on the multi-agent model. The Bee-gent framework is comprised of two types of agent. ‘‘Agent wrappers’’ are used to agentify (i.e. providing an agent interface) existing applications, while ‘‘Mediation Agents’’ support inter-application co-ordination by handling all communications. The mediation agents move from the site of an application to another where they interact with the agent wrappers. The agent wrappers themselves manage the states of the applications they are wrapped around, invoking them when necessary.

The Bee-gent system has many desirable features. The mediation agent manages the coordinating interactions of the various applications in a unified manner. It is easy to maintain consistency because it is not necessary to divide and distribute the coordinating interactions on the basis of each individual application, and therefore development is simplified. It is also easy to modify the system configuration and the coordinating interactions because these interactions are encapsulated.

When the mediation agent migrates it carries its own program, data and current state. Frequency of communication is reduced compared to a purely message-based system and network loads are decreased largely because communication links can be disconnected after launch of the mediation agent. Processing efficiency is improved because the mediation agent communicates with the applications locally.

II. Conclusions

In this paper we were able to demonstrate on the E-Learning example of MASACAD how the multi-agent paradigm, combined with ideas from machine learning, user modeling, and Web mining, can be used to approach a solution for the problem of information customization.

MASACAD is a decision support system and at the same time an information customization system. Without MASA-CAD, the student, in order to take the decision about course enrolment, has to invest significant effort to find the right information which is distributed, dynamic, and available in different formats.

Many of my students have had the opportunity to run the system and test it. It was interesting to observe that the most fascinating point for them was to see the system presenting the information found in a somehow customized manner for each individual student. The results are encouraging and future work will be focused on improving the system and studying how such simple examples built with insight should lead to identification of key difficulties, useful abstractions and a general method for solving the problem and revelation of the issues.

References

[1]. M.S. Hamdi, An agent-based approach for intelligent user interfaces, in: Proceedings of the Fourth International Conference on Intelligent Pro-cessing and Manufacturing of Materials (IPMM-03), Sendai, Japan, 2003.
[2]. M.S. Hamdi, Information extraction using multi-agents, in: Proceedings of the International Conference on Internet Computing (IC ’03), Las Vegas, Nevada, USA, 2003.
[3]. M.S. Hamdi, Automation of information extraction, in: Proceedings of the Seventh International Conference on Automation Technology (Auto-mation 2003), National Chung Cheng University, Chia-yi, Taiwan, 2003.
[4]. M.S. Hamdi, A multi-agent based approach for retrieving information, in: Proceedings of the Second IASTED International Conference on Communications, Internet and Information Technology (CIIT 2003), Scottsdale, Arizona, USA, 2003.
[5]. M.S. Hamdi, Extracting and customizing information using multi-agents, in: A. Scime (Ed.), Web Mining: Applications and Techniques, Idea Group Inc., 2005, ISBN: 1-59140-414-2, pp. 228–252.
[6]. M. Moore, Distance Education: A Systems View, Wadsworth Publishing Company, 1996. http://www.cde.psu.edu/de/what_is_de.html#definition.
[7]. J. Mostafa, Guest editor’s introduction: information customization, IEEE Intell. Syst. 17 (6) (2002) 8–11.
[8]. G.R. Andrews, Foundations of Multithreaded, Parallel and Distributed Programming, Addison-Wesley, 2000.
[9]. A. Kobsa, User modeling in dialog systems: potentials and hazards, AI
[10]. Soc.: J. Human Machine Intell. 4 (3) (1990) 214–231.
[11]. E. Rich, User modeling via stereotypes, Cognitive Sci. 3 (1979) 329– 354.
[12]. F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. (CSUR) 34 (1) (2002) 1–47.
[13]. T.M. Mitchell, Machine Learning, McGraw-Hill, 1997.
[14]. A. Newell, The knowledge level, Artif. Intell. 18 (1982) 87–127.
[15]. J.M. Bradshaw, An introduction to software agents, in: J.M. Bradshaw (Ed.), Software Agents, AAAI Press, Menlo Park, California, 1997, pp. 3–46.
[16]. S. Franklin, A. Graesser, 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, published as Intelli-gent Agents III, Springer-Verlag, 1997, pp. 21–35.
[17]. E.M. Stuart, Interface agents: a review of the field, Technical Report
[18]. Number: ECSTR-IAM01-001, University of Southampton, August 2000, ISBN 0854327320.

[19]. P. Jackson, Introduction to Expert Systems, 3rd ed., Addison-Wesley, Harlow, England, 1999.

[20]. E.A. Feigenbaum, The art of artificial intelligence: themes and case studies of knowledge engineering, in: Proceedings of the Fifth Interna-tional Joint Conference on Artificial Intelligence, 1977, pp. 1014–1029.

[21]. H.A. Simon, Why should machines learn? in: R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning, Tioga, Palo Alto, CA, 1983 (Chapter 2).

[22]. J.G. Carbonell, R. Michalski, T. Mitchell, An overview of machine learning, in: R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning, Tioga, Palo Alto, CA, 1983 (Chapter 1).

[23]. T.G. Dietterich, Learning at the knowledge level, Mach. Learn. 1 (1986) 287–316.

[24]. J.R. Quinlan, C4.5: Programs for Empirical Learning, Morgan Kauf-mann, San Francisco, CA, 1993.

[25]. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal repre-sentations by error propagation, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing, vol. 1, MIT Press, Cambridge, MA, 1986, pp. 318–362.

[26]. J.W. Shavlik, R.J. Mooney, G.G. Towell, Symbolic and neural net learning algorithms: an empirical comparison, Mach. Learn. 6 (1991) 111–143.

[27]. L. Atlas, R. Cole, Y. Muthusamy, A. Lippman, J. Connor, D. Park, M. El-Sharkawi, R.J. Marks, A peerformance comparison of trained multi-layer perceptrons and trained classification trees, Proceedings of IEEE 78 (1990) 1614–1619.

[28]. R. Cooley, B. Mobasher, J. Srivastava, Web mining: information and pattern discovery on the World Wide Web, in: Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence (ICTAI '97), Newport Beach, CA, November, 1997.

[29]. M. Hearst, Distinguishing between web data mining and information access, in: Proceedings of the Knowledge Discovery (KDD-97), Newport Beach, CA, 1997.

[30]. R. Krishnapuram, A. Joshi, O. Nasraoui, L. Yi, Low complexity fuzzy relational clustering algorithms for web mining, IEEE Trans. Fuzzy Syst. 9 (4) (2001) 596–607.

[31]. T. Kawamura, T. Hasegawa, A. Ohsuga, S. Honiden, Bee-gent: bonding and encapsulation enhancement agent framework for development of distributed systems, Syst. Comput. Jpn. 31 (13) (2000) 42–56.

[32]. T.G. Dietterich, Machine learning research: four current directions, AI Mag. 18 (4) (1997) 97–136.

[33]. D.H. Wolpert, The lack of a priori distinctions between learning algo-rithms, Neural Comput. 8 (7) (1996) 1341–1390.

[34]. C. Schaffer, A conservation law for generalization performance, in: W. Cohen, H. Hirsh (Eds.), Proceedings of the 11th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 1994, pp. 259–265.

[35]. J.L. McClelland, D.E. Rumelhart, G.E. Hinton, The appeal of parallel distributed processing, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, MIT Press, 1986, pp. 3–44.

[36]. J.V. Haxby, E.A. Hoffman, M.I. Gobbini, The distributed human neural system for face perception, Trends Cogn. Sci. 4 (6) (2000) 223–233.

[37]. R.R. Poznanski (Ed.), Biophysical Neural Networks: Foundations of Integrative Neuroscience, Mary Ann Liebert Inc. Publishers, New York, 2001, ISBN: 0-913113-90-5.

[38]. D.O. Hebb, The Organization of Behaviour, John Wiley, New York, 1949.

[39]. P. Lewicki, T. Hill, M. Czyzewska, Nonconscious acquisition of infor-mation, Am. Psychol. 47 (6) (1992) 796–801.

[40]. D. Patterson, Artificial Neural Networks, Prentice Hall, Singapore, 1996.

[41]. L. Fausett, Fundamentals of Neural Networks, Prentice Hall, New York, 1994.

[42]. S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan Publishing, New York, 1994.

[43]. P.J. Werbos, Beyond regression: new tools for prediction and analysis in the behavioral sciences, Ph.D. Dissertation, Committee on Mathematics, Harvard University, Cambridge, MA, 1974.

[44]. C. Bishop, Neural Networks for Pattern Recognition, University Press, Oxford, 1995.

[45]. M. Watson, Practical Artificial Intelligence Programming in Java, 2002, http://www.markwatson.com/opencontent/.

[46]. G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, Fuzzy ARTMAP: a neural network architecture for incremental super-vised learning of analog multidimensional maps, IEEE Trans. Neural Networks 3 (1992) 698–713.

[47]. A.H. Tan, Adaptive resonance associative map, Neural Networks 8 (3) (1995) 437–446.

[48]. C. Brown, L. Gasser, D.E. O'Leary, A. Sangster, AI on the WWW supply and demand agents, IEEE Expert 4 (1995) 50–55.

[49]. O. Etzioni, D.S. Weld, Intelligent agents on the internet – fact, fiction, and forecast, IEEE Expert 4 (1995) 44–49.

[50]. B. Hermans, Intelligent Software Agents on the Internet: an inventory of currently offered functionality in the information society and a prediction of (near-)future developments, Ph.D. Dissertation, Tilbury University, Tilbury, The Netherlands, 1996.

[51]. J.R. Oliver, On artificial agents for negotiation in electronic commerce, Ph.D. Dissertation, University of Pennsylvania, 1996.

[52]. Y. Arens, S. Feiner, J. Foley, E. Hovy, B. John, R. Neches, R. Pausch, H. Schorr, W. Swartout, Intelligent User Interfaces, Report ISI/RR-91-288, USC/Information Sciences Institute, Marina del Rey, California, 1991.