From a Generic MultiAgent Architecture to MultiAgent Information Retrieval Systems

Andrea Addis
University of Cagliari
Piazza d'Armi,
I-09123, Cagliari, Italy
addis@diee.unica.it

Giuliano Armano
University of Cagliari
Piazza d'Armi,
I-09123, Cagliari, Italy
armano@diee.unica.it

Eloisa Vargiu
University of Cagliari
Piazza d'Armi,
I-09123, Cagliari, Italy
vargiu@diee.unica.it

ABSTRACT

The continuous growth of documents in digital form, together with the corresponding volume of daily updated contents, makes the problem of retrieving information a challenging task. In this paper we present X.MAS, a generic multiagent architecture explicitly devoted to implement information retrieval tasks. The proposed architecture has been adopted in several applications. To put into evidence how to bridge the gap from theory to practice, we illustrate and discuss three relevant applications of X.MAS.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval, Systems and Software

Keywords

Information Retrieval, MultiAgent Systems, Applications

1. INTRODUCTION

In the last fifteen years, AI researchers have concentrated their efforts in the field of intelligent autonomous agents, i.e., systems capable of autonomous sensing, reasoning and acting in complex environments. Suitable single-agent architectures have been devised to overcome the complexity problems that arise while trying to give agents a flexible behavior [14], [13], [27], [1]. Let us briefly recall that an agent architecture is essentially a map of the internals of an agent, i.e., its data structures, the operations that may be performed on them, and the corresponding control flows [36]. Furthermore, to allow cooperation and to implement suitable multiagent systems, several multiagent architectures have been devised. From a general perspective, we identify two kinds of these architectures: general-purpose [18], [26], [8] and application-oriented [11], [34], [35].

Although generic guidelines to build general-purpose architectures are required, in our view it is important to concentrate the efforts in a specific application field in order to bridge the gap between theoretical and pragmatical issues. To this end, in this paper we present a generic multiagent architecture explicitly devised to implement information retrieval tasks.

Jung, Michel, Ricci & Petta (eds.): *AT2AI-6 Working Notes, From Agent Theory to Agent Implementation, 6th Int. Workshop,* May 13, 2008, AAMAS 2008, Estoril, Portugal, EU.

Not for citation

The remainder of the paper is organized as follows: Section 2 gives a brief introduction about agents in information retrieval (*theoretical foundations*). Section 3 presents the architecture from a conceptual (*standard architectures*) and a technological point of view (*off-the-shelf platforms*). Then, Section 4 illustrates three actual systems built upon the proposed architecture (*applications*). Finally, Section 5 draws conclusions and points to future work.

2. FUNDAMENTALS ON AGENTS AND INFORMATION RETRIEVAL

Due to the increased availability of documents in digital form and the consequential need to access them in a flexible way, automated content-based document management tasks have gained a main task in the information systems field [30]. In particular, web information retrieval is highly popular and presents a technical challenge due to the heterogeneity and size of the web, which is continuously growing (see [17], for a survey).

In our opinion, it is very difficult for users to select contents according to their personal interests, especially when contents are frequently updated (e.g., news, newspaper articles, reuters, rss feeds, wikis, and blogs). Supporting users in handling the enormous and widespread amount of information (especially the one provided by the web) is becoming a primary issue. To this aim, several online services have been proposed that provide a personalization mechanism based on keywords, which is often inadequate to express what the user is really searching for. Moreover, users must often refine by hand the results provided by the service.

Agents have been widely proposed as a solution to these problems. An information agent is an agent that has access to one or more information sources, and is able to store and process information obtained from these sources in order to answer queries posed by users and other information agents [37]. The information sources may be of many types, including web services, web sites, RSS-feeds, and traditional databases.

In the literature, several centralized agent-based architectures aimed to perform information retrieval tasks have been proposed. Among others, let us recall NewT [32], Letizia [21], WebWatcher [3], and SoftBots [12].

NewT [32] has been designed as a society of information-filtering interface agents, which learn user preferences and act on her/his behalf. These information agents use a keyword-based filtering algorithm, whereas adaptive techniques are relevance feedback and genetic algorithms. Letizia [21] is an intelligent user interface agent able to assist a user while browsing the Web. The search for information results as a cooperative venture between the user and the software agent: both browse the same search space of linked web documents, looking for interesting ones. WebWatcher [3] is an information search agent that follows web hyperlinks according to user interests, returning a list of links deemed interesting. In contrast to systems for assisted browsing or information retrieval, SoftBots [12] accept high-level user goals and dynamically synthesize the appropriate sequence of Internet commands according to a suitable ad-hoc language.

Despite the fact that a centralized approach could have some advantages, in information retrieval tasks it may encompass several problems, in particular how to scale up the architectures to large numbers of users, how to provide high availability in case of constant demand of the involved services, and how to provide high trustability in case of sensitive information, such as personal data. To this end suitable multiagent systems devoted to perform information retrieval tasks have been proposed. In particular, Sycara [33] proposed Retsina, a multiagent system infrastructure applied in many domains, has been presented. Retsina is an open MAS infrastructure that supports communities of heterogeneous agents. Three types of agents have been defined: interface agents, able to display the information to the users; task agents, able to assist the user in the management of her/his information; and information agents, able to gather relevant information from the selected sources.

Apart from Retsina, in the literature, several multiagent systems have been proposed and implemented. Among others, let us recall IR agents [19], CEMAS [6], and the cooperative multiagent system for web information retrieval proposed in [31].

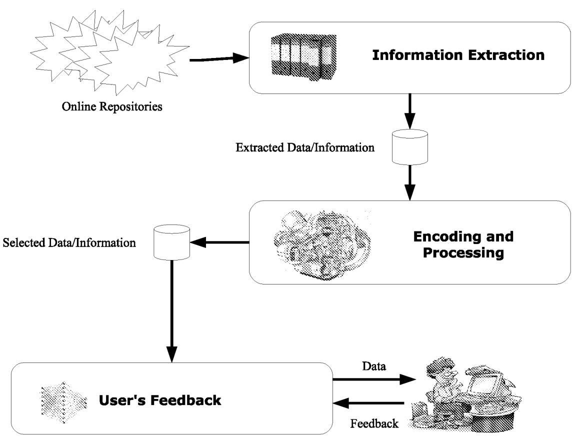

Figure 1: The abstract architecture.

IR agents [19] implement an XML-based multiagents model for information retrieval. The corresponding framework is composed of three kinds of agents: (i) managing agents, aimed to extract the semantics of information and to perform the actual tasks imposed by coordinator agents, (ii) interface agents, devised to interact with the users, and (iii) search agents, aimed to discover relevant information on the web. IR agents do not take into account personalization, while providing information in a structured form without the adoption of specific classification mechanisms. In CEMAS (Concept Exchanging Multi-Agent System) [6] the basic idea is to provide specialized agents for each main task, the main tasks being: (i) exchanging concepts and links, (ii) representing the user, (iii) searching for new relevant documents matching existing concepts, and (iv) supporting agent coordination. Although CEMAS provides personalization and classification mechanisms based on a semantic approach, the main drawback is that it is not generic, being mainly aimed to support scientists while looking for comprehensive information about their topic area. Finally, in [31] the underlying idea is to adopt intelligent agents that mimic everyday-life activities of information seekers. To this end, agents are also able to profile the user in order to anticipate and achieve her/his preferred goals. Although the approach is quite interesting, it is mainly focused on cooperation among agents rather then on information retrieval issues.

3. X.MAS: THE PROPOSED GENERIC ARCHITECTURE

Focusing on the role of software agents, the following categories can be identified in a context of information retrieval: (i) *information agents*, tailored to extract and handle information while accessing information sources[24], (ii) *filter agents*, able to transform information according to user preferences [23], (iii) *task agents*, able to help users to perform tasks typically in cooperation with other agents [15], (iv) *interface agents*, in charge of interacting with the user such that s/he interacts with other agents throughout them [22], and (v) *middle agents*, devised to establish communication among requesters and providers [10].

3.1 The Abstract Architecture

From a theoretical point of view, an information retrieval task involves three main activities: (i) extracting the required information, (ii) encoding and processing it according to the specific application, and (iii) providing suitable feedback mechanisms to improve the overall performances. Figure 1 shows a generic architecture able to perform these activities.

The information extraction module is aimed to extract data from information sources through specialized wrappers. In general, given an information source S, a specific wrapper W_S must be implemented, able to map each data D_S, designed according to the constraints imposed by S, to a suitable description O, which contains relevant information in a structured form –such as title, author(s), description, and images.

The encoding-and-processing module is aimed to encode information that flows from external sources (i.e., the selected information sources) and to progressively filter it to the end user by retaining only relevant data. The actual encoding strictly depends on the specific application (preprocessing activities, such as feature selection, could be performed to prepare the data to be processed). Data are processed according to high-level procedures, which are independent from the specific user. If needed, a personalized process can be performed according to user needs and preferences.

The user feedback module is devoted to deal with any feedback optionally provided by the end-user. In general, trivial –though effective– solutions can been implemented, e.g. based on artificial neural networks (ANNs) or a k-NN

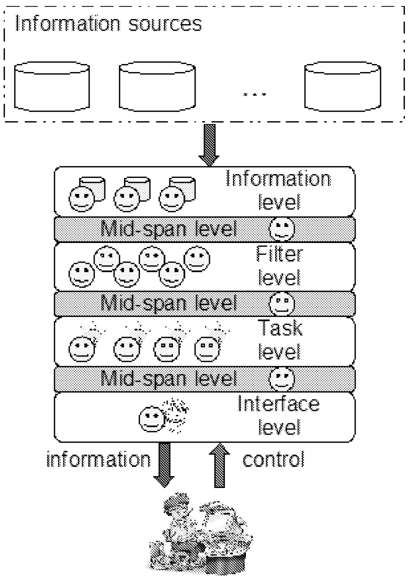

Figure 2: The concrete architecture.

classifiers.

3.2 The Concrete Architecture

An information retrieval system must take into account several issues, the most relevant being: (i) how to deal with different information sources and to integrate new information sources without re-writing significant parts of it, (ii) how to suitably encode data in order to put into evidence the informative content useful to discriminate among categories, (iii) how to control the unbalance between relevant and non relevant articles (the latter being usually much more numerous than the former), (iv) how to allow the user to specify her/his preferences, and (v) how to exploit the user feedback to improve the overall performance of the system.

The above problems are typically strongly interdependent in state-of-the-art systems. To better concentrate on these aspects separately, we adopted a layered multiagent architecture, able to promote the decoupling among all aspects deemed relevant. In particular, the functionalities of the abstract architecture, described in the previous section, have been implemented exploiting the X.MAS architecture. X.MAS is a generic multiagent architecture, aimed to retrieve, filter and reorganize information according to user interests. The X in X.MAS points out the generic nature of the architecture, playing the role of a wildcard to be substituted with an identifier specific of the corresponding application. The X.MAS generic architecture has been implemented on top of the well known JADE [5] agent-based infrastructure.

3.2.1 Macro-Architecture

The X.MAS architecture (depicted in Figure 2) encompasses four main levels: information, filter, task, and interface. The communication between adjacent levels is achieved through suitable middle agents, which form a corresponding mid-span level.

At the *information level*, agents are entrusted with extracting data from the information sources. Each information agent is associated to one information source, playing the role of wrapper.

At the *filter level*, agents are aimed to select information deemed relevant to the users, and to cooperate to prevent information from being overloaded and/or redundant. In general, two filtering strategies can be adopted: generic and personalized. The former applies the same rules to all users; whereas the latter is applied when a customized behavior is required for a specific user.

At the *task level*, agents arrange data according to users personal needs and preferences. In a sense, they can be considered as the core of the architecture. In fact, they are devoted to achieve user goals by cooperating together and adapting themselves to the changes of the underlying environment.

At the *interface level*, a suitable interface agent is associated with each different user interface. In fact, a user can generally interact with an application through several interfaces and devices (e.g., pc, pda, mobile phones, etc.).

At *mid-span levels*, agents are aimed to establish communication among requesters and providers. Agents at these architectural levels can be implemented as matchmakers or brokers, depending on the specific application [10].

3.2.2 Micro-Architecture

X.MAS agents can implement several capabilities, depending on the actual application and on their specific role. In particular, X.MAS agents, other than autonomy and flexibility, can provide any subset of the following capabilities: (i) *personalization*, to fulfill user interests and preferences, (ii) *adaptation*, to adapt to the underlying environment, (iii) *cooperation*, to interact with other agents in order to achieve a common goal, (iv) *deliberative capability*, to reason about the world model and to engage planning and negotiation, possibly in coordination with other agents; and (v) *mobility*, to migrate from node to node in a local- or wide-area network.

X.MAS agents are JADE agents capable of (i) interacting exchanging FIPA-ACL messages, (ii) sharing a common ontology in accordance with the actual application, and (iii) exhibiting a specific behavior according to their role. As for agent internals, Figure 3 shows the micro-architecture for agents belonging to each architectural level. Let us note that each agent encompasses a scheduler aimed to control the information flow between adjacent levels. Information and interface agents embody information sources and a specific devices, respectively. Filter and task agents encompass an actuator that depends on the actual application. Middle agents contain a dispatcher aimed to handle interactions among requesters and providers.

4. BUILDING INFORMATION RETRIEVAL SYSTEMS BY USING X.MAS

In order to highlight how to bridge the gap from theory to practice by adopting X.MAS, three relevant systems are presented. The first is concerned with the problem of classifying Wikipedia contents according to a predefined set of classes, the second is focused on giving a support to professors and students while interacting with a media asset management system, and the third is devoted to address the problem of monitoring boats in a marine reserve.

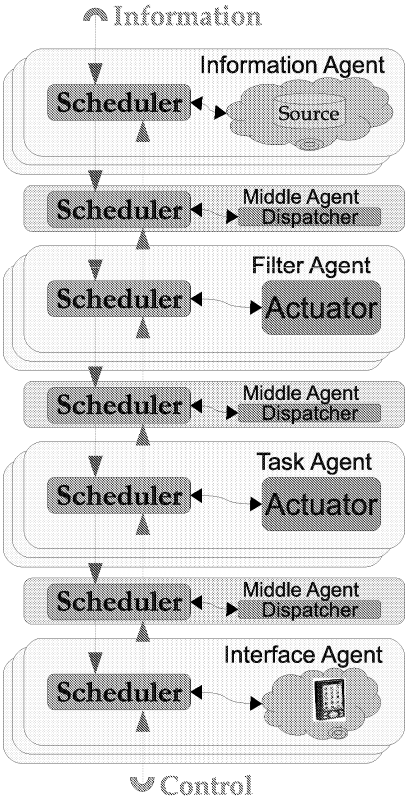

Figure 3: Agent internals.

4.1 WIKI.MAS: X.MAS for Classifying Wikipedia Contents

4.1.1 The Scenario.

As already pointed out in Section 2, supporting users in handling the enormous and widespread amount of web information is becoming a primary issue. Currently, the most overshadowing and noteworthy web information sources are developed according to the collaborative-web paradigm [7], also known as Web 2.0. It represents a paradigm shift in the way users approach the web. Users (also called prosumers) are no longer passive consumers of published content, but become involved, implicitly and explicitly, as they cooperate by providing their own content in an "architecture of participation" [28].

Among others, Wikipedia [1], an online encyclopedia based on the notion that an entry can be added/edited by any web user, has became an important benchmark for all Internet users interested in searching for definitions and/or references. Unfortunately, Wikipedia search engine allows users to choose their interests among macro-areas (e.g. *Arts*, *History*, and *Science*), which is often inadequate to express what the user is really interested in. Moreover, such search engine does not provide a feedback mechanism able to allow the user to specify non-relevant items –with the goal of progressively adapting the system to her/his actual interests.

Using X.MAS, we developed a system explicitly aimed to retrieve and classify Wikipedia contents according to a predefined set of classes, i.e., those belonging to WordNet Domains [25].

4.1.2 The Implementation.

To implement this specific application, X.MAS has been customized as follows:

- *Information level.* Agents at this level are aimed to deal with the huge information source provided by Wikipedia. To this end a suitable wrapper has been implemented, able to handle the structure of a document by saving the informations about the corresponding metadata (e.g. title, abstract, keywords, section headers) and by surfing across the whole reference links through the cooperation with other information agents.

- *Filter level.* Filter agents are aimed to select information deemed relevant to the users, and to cooperate to prevent information from being overloaded and redundant. A suitable encoding of the text content has been enforced at this level to facilitate the work of agents belonging to the task level. In particular, all non-informative words such as prepositions, conjunctions, pronouns and very common verbs are removed using a stop-word list. After that, a standard stemming algorithm [29] removes the most common morphological and inflexional suffixes. Then, for each category, feature selection, based on the information-gain heuristics [38], has been adopted to reduce the dimensionality of the feature space.

- *Task level.* Task agents are devoted to identify relevant Wikipedia documents, depending on user interests. Agents belonging to this architectural level are aimed to perform two kinds of actions: classify any given input in accordance with the selected set of classes, and decide whether it may be of interest to the user or not. Each task agent has been trained by resorting to state-of-the-art algorithms that implement the k-NN technique, in its *"weighted"* variant [9]. Furthermore, to express what the user is really interested in, we implemented suitable composition strategies by using extended boolean models [20]. In fact, typically, the user is not directly concerned with "generic" topics that coincide with the selected classes (such as *Arts*, *History*, or *Science*). Rather, a set of arguments of interest can be obtained by composing these generic topics with suitable logical operators (i.e., *and*, *or*, and *not*). In the proposed system, we adopted a quite general soft boolean perspective, in which the combination is evaluated using P-norms [16].

- *Interface level.* Information agents are aimed to perform the feedback from the user –which can be exploited to improve the overall ability of discriminating relevant from non relevant inputs. So far, a simple solution based on the k-NN technology has been implemented and experimented to deal with the problem of supporting the user feedback. When a non-relevant article is evidenced by the user, it is immediately embedded in the training set of the k-NN classifier that implements the feedback. A check performed on this training set after inserting the negative example allows to trigger a procedure entrusted with keeping the

[1] http://www.wikipedia.org/

number of negative and positive examples balanced. In particular, when the ratio between negative and positive examples exceeds a given threshold (by default set to 1.1), some examples are randomly extracted from the set of "true" positive examples and embedded in the above training set.

4.2 MAM.MAS: X.MAS for a Media Asset Management System

4.2.1 The Scenario.

E-learning differentiates from the traditional learning in its ability to train anyone, anytime, and anywhere, thanks to the openness of the Internet. Without the temporal and spatial limitation, one can have an independent and individual learning space. Currently, several Digital Asset Management (DAM) systems, also called Media Asset Management (MAM), have been proposed and devised [4]. As for e-learning, such systems are aimed to store, manage, and organize course materials, bibliography, and teacher notes.

Among other provided services, MAM systems must supply suitable support during the insertion phase. In particular, classification techniques might be devised to improve such systems to suitably organize contents and to help users in managing such data.

Using X.MAS, we developed a system aimed to support users in inserting multimedia contents in a MAM system. Being interested in handling university courses, typically organized in a hierarchy of classes, suitable hierarchical classification techniques has been studied. In particular, the current version of the system implements a hierarchical text categorization approach and is able to deal with text documents. Let us note that any multimedia document could be processed if a suitable textual description is given. In our experiments the system classifies data according to the taxonomy related to university courses, i.e., the one concerned with the bachelor's degree in electronic engineering. Figure 4 illustrates a portion of the adopted taxonomy.

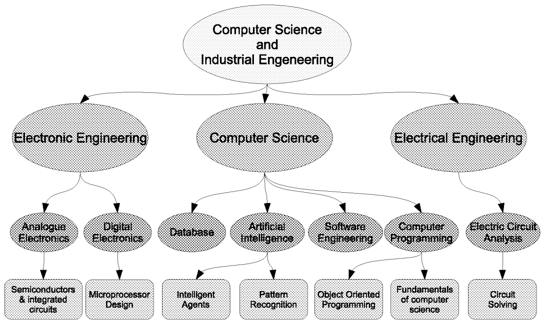

Figure 4: A portion of the adopted taxonomy.

4.2.2 The Implementation.

As for the implementation, X.MAS has been customized as follows:

- *Information level.* Input documents are the files provided by teachers during the insertion phase. Such documents can be simple-text-, formatted-, pdf-files, or multimedia contents together with their textual description. Information agents are able to handle all these kinds of documents by extracting the corresponding information.
- *Filter level.* As in WIKI.MAS, filter agents are aimed to select information deemed relevant to the users, and to cooperate to prevent information from being overloaded and redundant. Suitable encoding techniques have been enforced: all non-informative words are removed using a stop-word list; a standard stemming algorithm removes the most common morphological and inflexional suffixes; and, for each category, feature selection, based on the information-gain heuristics, has been adopted to reduce the dimensionality of the feature space.
- *Task level.* Task agents perform the hierarchical text categorization, resorting to a progressive filtering technique as described in [2]. In particular, each task agent has been trained by resorting to state-of-the-art algorithms that implement the *wk*-NN technique, whereas the progressive filtering is provided by the cooperation of the corresponding agents.
- *Interface level.* Interface agents are devoted to interact with the user within the MAM in order to support her/him while inserting documents. Further agents are aimed to perform user feedback. So far, a simple solution based on an ANN has been implemented. This solution consists of training an ANN with a set of examples classified as "of interest to the user". When the amount of feedback provided by the user has trespassed a given threshold, the ANN is trained again –after updating the previous training set with the information provided by the user.

4.3 SEA.MAS: X.MAS for Monitoring Boats in Marine Reserves

4.3.1 The Scenario.

Setting up a marine reserve involves issues concerning how to enforce access monitoring, with the goal of avoiding intrusions by not authorized boats, also considering that typically marine reserves are located in areas not easily accessible.

Currently, intrusion detection in marine reserves is carried out by adopting radar systems or by resorting to suitable cameras activated by movement sensors.

Ee are currently developing with X.MAS a system aimed to monitor boats in marine reserves. The corresponding scenario involves authorized boats, equipped with GPS+GSM devices, as well as not authorized boats. Boats are tracked by a digital radar that detects their positions. In this way, not authorized boats are easily identified comparing radar signals with those received from GPS+GSM devices.

4.3.2 The Implementation.

As for the implementation, X.MAS is being customized as follows:

- *Information level.* Each information agent encapsulates a specific information source. Two kinds of wrappers will be implemented to retrieve data from the devices and the radar, respectively.
- *Filter level.* Filter agents are devoted to collect the data retrieved by the information agents. In particular, they disregard redundant information, as signals

detected more than once by the same device (caching) or from different devices (information overloading).

- *Task level.* Each task agent corresponds to a boat. Information provided by filter agents allows to know exactly boat positions. In so doing, authorized and not authorized boats can be easily detected. The main tasks of the agents belonging to this architectural level are: (i) to follow a boat position during its navigation, also dealing with any temporary lack of signal; (ii) to promptly identify not authorized boats alerting the interface agents; and (iii) to handle messages coming from the interface level in order to notify the involved devices.

- *Interface level.* A suitable interface agent allows users to interact with the system. Final users are the system administrator and staff operators.

5. CONCLUSIONS AND FUTURE WORK

In this paper X.MAS, a generic architecture designed to support the implementation of applications g manage information among different and heterogeneous sources, has been presented. To put into evidence how to bridge the gap from theory to practice by adopting X.MAS, three relevant applications have been briefly described: the first concerned with the problem of classifying Wikipedia contents according to a predefined set of classes, the second focused on giving a support to professors and students while interacting with a media asset management system, and the third devoted to monitor boats in a marine reserve.

As for the future work, we are investigating how to improve the intelligent capabilities of agents with more complex forms of proactive and deliberative capabilities. Moreover, the possibility to implement further applications using X.MAS is currently under study.

Acknowledgments

The case study related to the implementation of a Wikipedia information retrieval system has been supported by the Italian Ministry of Education, under the project "DART - Distributed Architecture for Semantic Search and Personalized Content Retrieval".

The case study related to the implementation of a support for a multimedia asset management system has been supported by Regione Autonoma della Sardegna, under the project "SOFIA - Online Services of University Courses and Education".

The case study related to the implementation of a system for monitoring boats in marine reserves is supported by Regione Autonoma della Sardegna, under the project "A Multiagent System for Monitoring Intrusions in Marine Reserves".

6. REFERENCES

[1] G. Armano, G. Cherchi, and E. Vargiu. An agent architecture for planning in a dynamic environment. *Lecture Notes in Computer Science*, 2175:388–394, 2001.

[2] G. Armano, F. Mascia, and E. Vargiu. Using taxonomic domain knowledge in text categorization tasks. *International Journal of Intelligent Control and Systems*, 12(2):150–157, 2007. Special Issue on Distributed Intelligent Systems.

[3] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering*, pages 6–12, 1995.

[4] D. Austerberry. *Digital Asset Management.* Focal Press, 2006.

[5] F. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology).* John Wiley and Sons, 2007.

[6] M. Bleyer. *Multi-Agent Systems for Information Retrieval on the World Wide Web.* PhD thesis, University of Ulm, Germany, 1998.

[7] J. Burdman. *Collaborative Web Development: Strategies and Best Practices for Web Teams.* Addison-Wesley Longman Ltd., 1999.

[8] D. Camacho, R. Aler, D. Borrajo, and J. Molina. A multi-agent architecture for intelligent gathering systems. *AI Communications, The European Journal on Artificial Intelligence*, 18(1):1–Ű19, 2005.

[9] W. Cost and S. Salzberg. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10:57–78, 1993.

[10] K. Decker, K. Sycara, and M. Williamson. Middle-agents for the internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI 97)*, pages 578–583, 1997.

[11] F. Dorca, C. Lopes, and M. Fernandes. A multiagent architecture for distance education systems. In *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies*, pages 368–369, 2003.

[12] O. Etzioni and D. Weld. Intelligent agents on the internet: fact, fiction and forecast. *IEEE Expert*, 10(4):44–49, 1995.

[13] I. Ferguson. *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents.* PhD thesis, lare Hall, University of Cambridge, UK, 1992.

[14] M. Georgeff and A. Lansky. Reactive reasoning and planning. In *Proceedings of the Sixth National on Artificial Intelligence (AAAI-87)*, pages 677–682, 1987.

[15] J. Giampapa, K. Sycara, A. Fath, A. Steinfeld, and D. Siewiorek. A multi-agent system for automatically resolving network interoperability problems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1462–1463, 2004.

[16] G. Golub and C. V. Loan. *Matrix Computations.* Baltimore: The Johns Hopkins University Press, 1996.

[17] L. Huang. A survey on web information retrieval technologies. In *Technical report, ECSL*, 2000.

[18] C. A. Iglesias, J. C. Gonzalez, and J. R. Velasco. Mix: A general purpose multiagent architecture. In *In M. Wooldridge, J. P. Muller, and M. Tambe, editors,* Intelligent Agents II*, Springer-Verlag*, pages 251–266, 1995.

[19] W. Jirapanthong and T. Sunetnanta. An xml-based multi-agents model for information retrieval on www. In *Proceedings of the 4th National Computer Science*

and Engineering Conference (NCSEC2000), 2000.

[20] J. Lee. Properties of extended boolean models in information retrieval. In *Proceedings of the 17th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, pages 182–190, 1994.

[21] H. Lieberman. Letizia: An agent that assists web browsing. In C. S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[22] H. Lieberman. Autonomous interface agents. In *Proceedings of the ACM Conference on Computers and Human Interface (CHI-97)*, pages 67–74, 1997.

[23] E. Lutz, H. Kleist-Retzow, and K. Hoernig. Mafia an active mail-filter-agent for an intelligent document processing support. *ACM SIGOIS Bulletin*, 11(4):16–32, 1990.

[24] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):31–40, 1994.

[25] B. Magnini and G. Cavaglià. Integrating subject field codes into wordnet. In *Gavrilidou M., Crayannis G., Markantonatu S., Piperidis S. and Stainhaouer G. (Eds.)* Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, pages 1413–1418, 2000.

[26] P. Marcenac and S. Giroux. Geamas: A generic architecture for agent-oriented simulations of complex processes. *International Journal of Applied Intelligence*, 1998.

[27] J. Mueller. A cooperation model for autonomous agents. In *In J.P Mueller, M. Wooldridge and N.R. Jennings (eds)* Intelligent Agents III*, LNAI, Vol. 1193. Springer-Verlag, Berlin Heidelberg New York*, pages 245–26, 1997.

[28] T. O'Reilly. *What is Web 2.0, Design Patterns and Business Models for the Next Generation of Software.* O' Reilly, 2005.

[29] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[30] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–55, 2002.

[31] K. Shaban, O. Basir, and M. Kamel. Team consensus in web multi-agents information retrieval system. In *Team Consensus in Web Multi-agents Information Retrieval System*, pages 68–73, 2004.

[32] B. Sheth and P. Maes. Evolving agents for personalized information filtering. In *Proceedings of the 9th Conference on Artificial Intelligence for Applications (CAIA-93)*, pages 345–352, 1993.

[33] K. Sycara, M. Paolucci, M. van Velsen, and J. Giampapa. The RETSINA MAS infrastructure. Technical Report CMU-RI-TR-01-05, Robotics Institute Technical Report, Carnegie Mellon, 2001.

[34] S. Walczak. A multiagent architecture for developing medical information retrieval agents. *Journal of Medical Systems*, 27(5):479–498, 2003.

[35] D. Wei and A. B. Abrahams. A multiagent architecture for semantic web resources. In *Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 289–292, 2005.

[36] M. Wooldridge. Intelligent agents. In *G.ăWeiss (ed)* Multiagent Systems*, MIT Press, Berlin Heidelberg New York*, 1999.

[37] M. Woolridge. *Introduction to Multiagent Systems.* John Wiley and Sons, Inc., 2001.

[38] Y. Yang and J. O. Pedersen. Feature selection in statistical learning of text categorization. In *Proceedings of the 14th International Conference of Machine Learning ICML9*, pages 412–420, 1997.