POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

TECHNOLOGIE INFORMATYCZNE W ZARZĄDZANIU SYSTEMY WSPOMAGANIA DECYZJI

pod redakcją:

Jana Studzińskiego,

Ludosława Drelichowskiego,

Olgierda Hryniewicza,

Janusza Kacprzyka

TECHNOLOGIE INFORMATYCZNE W ZARZĄDZANIU
SYSTEMY WSPOMAGANIA DECYZJI

Polska Akademia Nauk • Instytut Badań Systemowych

Seria: BADANIA SYSTEMOWE
tom 26

Redaktor naukowy:

Prof. dr hab. Jakub Gutenbaum

Warszawa 2000

TECHNOLOGIE INFORMATYCZNE W ZARZĄDZANIU SYSTEMY WSPOMAGANIA DECYZJI

pod redakcją
Jana Studzińskiego, Ludosława Drelichowskiego
Olgierda Hryniewicza i Janusza Kacprzyka

Książka zawiera wybór referatów przedstawionych na konferencji "Komputerowe systemy wielodostepne KSW'2000" w Ciechocinku w 2000 r. Konferencja pod patronatem Komitetu Badań Naukowych została zorganizowana przez Akademię Techniczno-Rolniczą w Bydgoszczy, Instytut Badań Systemowych PAN, Komisję Informatyki PAN - Oddział w Gdańsku oraz Bydgoskie Zakłady Elektromechaniczne "BELAM" S.A. w Bydgoszczy.

Komitet Naukowo-Programowy konferencji:

Witold Abramowicz, Ryszard Budziński, Ryszard Choraś, Ludosław Drelichowski (przewodniczący), Grzegorz Głownia, Adam Grzech, Jakub Gutenbaum, Olgierd Hryniewicz, Janusz Kacprzyk, Zbigniew Kierzkowski, Jerzy Kisielnicki, Adam Kopiński, Maciej Krawczak, Henryk Krawczyk, Bernard F. Kubiak, Roman Kulikowski, Marian Kuraś, Ludwik Maciejec, Marek Miłosz, Janusz Stokłosa, Jan Studziński, Zdzisław Szyjewski.

© Instytut Badań Systemowych PAN, Warszawa 2000

ISBN 83-85847-53-7
ISSN 0208-8028

Rozdział 4

Metody i algorytmy obliczeniowe w systemach komputerowych

INTELLIGENT AGENTS TO SUPPLY THE HYPERSDI SYSTEM

Witold Abramowicz, Paweł Jan Kalczyński
Katedra Informatyki Ekonomicznej
Akademia Ekonomiczna w Poznaniu
{W.Abramowicz, P.Kalczyński}kie.ae.poznan.pl

The idea of the HyperSDI system was developed in the Department of Computer Science of the Poznań University of Economics and is based on three main concepts: Hypertext, Information Retrieval and Selective Distribution of Information (SDI). Currently, the HyperSDI system undergoes extensive development in numerous projects. The development heads for using the system to supply Management Information Systems with relevant information filtered from the Web. This concept required system's reengineering according to the Internet paradigm. This paper presents the new model of the HyperSDI system based on the Intelligent Software Agents technology.

1. Introduction

As opposed to traditional Information Retrieval (IR) systems (Salton, McGill, 1983, van Rijsbergen, 1979) and Internet search engines[1], where a single query is performed on numerous documents, in Information Filtering (IF) systems a set of queries is performed on a single document currently analyzed by the system. Moreover, in traditional IR systems the number of documents in the queried collection is fixed and the documents are usually indexed and queried with keywords found in the collection. On the contrary, the number of documents analysed by an IF system can be practically infinite what makes the traditional IR measures such as *precision* or *call* (van Rijsbergen, 1979, Salton, McGill, 1983) inapplicable to the estimation of efficiency. Modern information filters retrieve documents from the Internet, which is a huge, immediate and dynamic but still heterogeneous and uncer-

[1] Infoseek search engine: http://www.infoseek.com
Hotbot search engine: http://www.hotbot.com
Yahoo search engine: http://www.yahoo.com

tain information source. The traditional mechanisms applied in document filtering and retrieval, proved to be insufficient for filtering information from the Web (Amati i inni, 1997).

2. The HyperSDI System

The HyperSDI system was developed in the Department of Computer Science of the Poznań University of Economics (Abramowicz, 1985, 1990, Abramowicz, Grabowski, 1990, Abramowicz, Ceglarek, 1998). The system's concept is based on Hypertext, Information Retrieval and Selective Distribution of Information filtered from the Web sources.

2.1 The idea of the HyperSDI

The HyperSDI system is an information filter that provides users with relevant information stored as digital documents. These documents are filtered from active and passive sources on the Web. Active sources, such as e-mail, newsgroups or distribution lists, supply the system with documents. On the contrary, passive information sources, such as Web pages, on-line databases or various services, must be repeatedly visited by the systems. Thus, the HyperSDI system sieves documents sent by or retrieved from registered sources. These sources are mostly commercial online services, such as benchmarking services, financial services or general business services. Thus, the system avoids the threat of being flooded with irrelevant external information.

HyperSDI users represent their interest in *user profiles*. In traditional information filters, a user profile contains a list of terms (keywords and key-phrases) with corresponding weights (Ceglarek, 1997). In the HyperSDI system, a user profile also contains a weighted list of sources, similarity measures, similarity thresholds and also some additional parameters for each term. The system then filters documents incoming from active information sources or retrieved from passive sources. Filtering is executed by indexing the analyzed document and comparing it against each of user profiles. The comparison results in assigning a value for a chosen similarity measure such as cosine, Dice or Jaccard (Ceglarek, 1997). Documents, which are similar to a particular profile, are then sent to the profile's user for evaluation. The user then estimates the relevance of the new document. Relevant documents are included in the user collection, which is a subset of the HyperSDI document collection.

User profiles are continuously refined according to documents accepted and rejected by users. Weights of terms that frequently appear in accepted documents are increased, whereas weights of terms that are common

in rejected documents are diminished. The process of user profile refinement is commonly referred to as *the relevance feedback.*

Documents included in the HyperSDI document collection are hypertext-way reorganised. Reorganisation can be performed automatically (i.e. by means of the Cluster Analysis) (Abramowicz, Ceglarek, 1998), semi-automatically or manually. Additionally, users may access all relevant documents, by performing traditional IR queries on the HyperSDI document collection.

The new model of the HyperSDI system should be able to supply users and organizations with relevant information filtered from the Web sources. This information should significantly increase their knowledge on processes that take place inside and outside the particular organization, thus it will be further referred to as *benchmarking information.*

2.2 The HyperSDI System Architecture

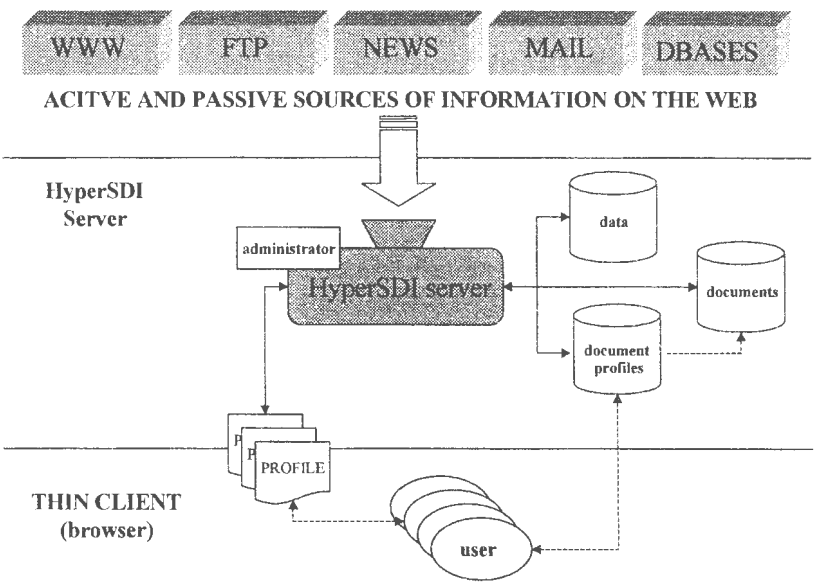

Figure 1. The HyperSDI system architecture

The HyperSDI system architecture consists of the following elements:

- information sources
- system users

- the HyperSDI server
- the document collection
- the document profile collection
- the system's database.

Information sources for the HyperSDI systems are mostly online business and benchmarking, e-mail and newsgroup services[2]. The set of HyperSDI users may include people, roles in organisational workflow but also a group of users that have common interests (*the collective user*). The system's administrator registers users and sources and supervises system's performance. The HyperSDI document collection is the set of hyperlinked documents. Documents in the collection are made available to users via *document profiles*. Document profiles make up users' private collections, thus a single document may be shared by numerous users. All document profiles in the HyperSDI system are referred to as *the document profile collection*. The HyperSDI database stores facts on information sources, users and their information interests (profiles). Finally, the HyperSDI server co-ordinates the whole process of information filtering.

3. Intelligent Software Agents in the HyperSDI System

3.1 Software Agents

The term "Software Agent" can be used either in the weak or in the strong notion (Wooldridge, Jennings, 1995). In the weak notion, an agent is a machine or a computer program with the following characteristics:

- autonomy – the ability to perform independently of the user and other agents
- social ability – agents communicate with users and other agents in the specific language
- reactivity – the ability to spot changes that happen in the environment and the ability to react
- pro-activity – extends the previously mentioned reactivity by the ability of agents to react according to general objectives (on their own initiative) rather than with simple algorithms (Wooldridge, Jennings, 1995).

---

[2] ACNielsen's homepage: http://acnielsen.com,
Dom Maklerski Banku Ochrony Środowiska, homepage: http://www.dmbos.com.pl,
Financial Services and Banking Benchmarking Association: http://fsbba.org,
The Benchmarking Network : http: //www.well.com /user /benchmar /tbnhome.html.

In the strong notion, an agent is a being that shows some mental capabilities such as knowledge, faith, sense of duty or intention, or even emotional capabilities (Wooldridge, Jennings, 1995). The detailed classification of software agents is presented in Stan Franklin's article (Franklin, Graesser, 1996) and will not be discussed further in this paper. Instead, we would consider the use of software agents, grasped in the weak notion, for filtering documents from the Web sources.

3.2 Retrieving Documents from the Web Sources

The new HyperSDI system retrieves information from Web sources of information. In comparison to the traditional information filtering and retrieval environments, the Web is a constantly changing and practically unlimited storehouse of information. Information on the Web is stored as hyperlinked digital documents in various formats and heterogeneous sources accessed by numerous protocols. Thus, a new HyperSDI system must be able to execute in such a complex environment.

3.2.1 Navigation in Hypertext

While searching for relevant information in a Web source, a human-user usually browses its contents and/or navigates among hyperlinked documents. As distinct from browsing, navigation is a systematic visiting consecutive nodes in order to find the required information. While navigating, users utilize their knowledge on hypertext collections, and the knowledge on the source of information and their information needs (Tomaszewski, 1998).

3.2.2 Filtering Documents from Heterogeneous Sources

The heterogeneity of information sources on the Web makes traditional information filters practically incapable of performing in this environment. Information on the Web is made available through various services such as WWW, FTP, telnet, e-mail, news, gopher, etc. These services operate on numerous protocols, such as HTTP, SHTTP, WAP, FTP, telnet, PPP, SMTP, etc. Moreover, documents are stored in various formats, such as HTML, XML, PDF, RTF, PS, etc. All these characteristics require different access, retrieval and navigation techniques (Balabanovic, Shoham, 1995).

Thus, efficient information filtering from the Web requires providing the HyperSDI filter with certain knowledge on Internet services, protocols, file formats and navigation techniques. Hence, we believe that introducing the Software Agents technology in the HyperSDI system seems to be justified.

3.3 HyperSDI Agents

Building agents that navigate rather than browse[3] and filter the contents of the Internet sources (Balabanovic, Shoham, 1995, Edwards i inni, 1996, Jaccard i inni 1999, Pazzani, 1995, Petrie, 1996) requires supplying them with the previously mentioned capabilities and the knowledge on users' information needs.

The knowledge on navigation techniques in hypertext collections could be embedded in the rule-based Expert System (ES), which would conclude whether hyperlinks in the analyzed document should be followed. The knowledge base of such ES consists of facts, values, rules that create values and conditions that make up rules. Some facts must be instantiated before concluding process starts. These facts are referred to as "observations". Observations make up a *knowledge vector*. The Expert System then instantiates successive facts according to the predefined rules. The process finishes when conclusions are drawn (Mulawka, 1996). The HyperSDI Expert System consists of over 50 rules that take into consideration document's size, relevance, "depth" and "distance" from the source URL, etc. Figure 2 presents the knowledge base of the HyperSDI Expert System.

Below we present sample rules that make up the HyperSDI Expert System:

R1: **if** word_count >1000 **then** should_be_indexed

R2: **if** source **is-a** WWW_service **&** document **contains** hyperlinks **then** document **is-a** HT_document

R3: **if** document **was** indexed **&** document **is** relevant **then** analize_hyperlinks

R4: **if** document **was** indexed **&** document **is-not** relevant **&** parent document **is-not** relevant **then** ~analize_hyperlinks

Artificial Intelligence added Information Filtering is commonly referred to as "Intelligent Information Filtering" (Palme, 1998).

3.4 Intelligent Information Filtering in the HyperSDI System

The process of intelligent information filtering in the new HyperSDI system can be divided into three phases: preparation of agents, harvest and relevance evaluation.

---

[3] Internet search engines use *softbots* (Software Robots) that visit and index one Web node after another as far as they are able to reach.

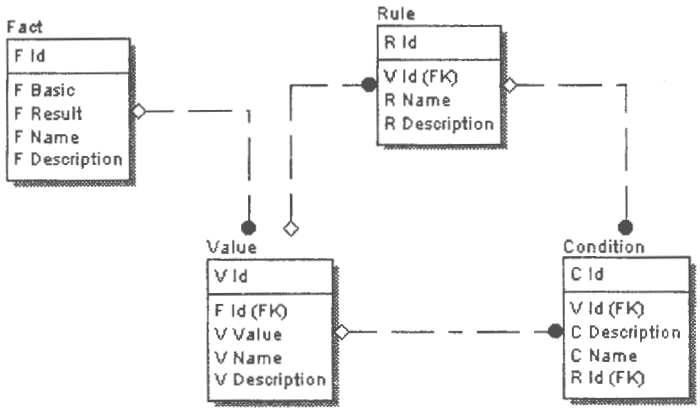

Figure 2. The HyperSDI Expert System Knowledge Base

3.4.1 Preparation of Agents

This phase is initiated by the notification of a new document incoming from an active source or by the necessity of searching a passive source by the system. Hence, the whole filtering process concerns one particular source that will be further referred to as *the current source*. Regardless the current source type, the HyperSDI system performs the following activities.

1. Facts concerning information sources and user needs (profiles) are loaded from the system's database.
2. Only the profiles that contain the current source are taken into consideration; they will be further referred to as *current profiles*.
3. A *compound profile*, the specific structure that contains a dictionary of terms, is constructed out of current profiles. Each term in the compound profile is accompanied by the set of pointers to the profiles, which contain this term.
4. An Aho-Corasick (AC) finite automaton (Aho, Corasick, 1975) is made up out of the compound profile elements. The AC machine will be further applied for multiple key matching in text documents.
5. The starting URL address for the particular source is added to the *URL queue*. This queue contains URLs that ought to be checked by the system.

3.4.2 Harvest

This phase is initiated by the appearance of the starting URL in the URL queue. Then the HyperSDI system performs the following activities.

1. A software agent is created as a new application thread.
2. The agent checks the URL queue for the first address to visit. Before accessing the source, it also checks whether other agent visited this address before. If the address is new, the agent starts penetrating document at the current URL.
3. The agent then parses the document by extracting the contents from the structure. Then, it analyzes the contents (counts the words) by means of previously constructed AC machine.
4. The agent then consults the compound profile and finds the users profiles that include terms found in the analyzed document.
5. Then the agent creates document indexes for each of the previously mentioned profiles and counts the similarity between the profile and the index (Ceglarek, 1997).
6. The agents consults the HyperSDI Expert System to find out whether the links found in the analyzed document should be followed. If so, the URLs are added to the URL queue.
7. If the similarity measure value for the checked document is greater than the threshold defined in the user profile, the document is distributed to the user for relevance evaluation.
8. Afterwards the agent is killed by the system.

The HyperSDI system continuously generates agents for each URL in the queue. Hence, numerous agents (in practice about 200) may perform at a time. The whole process suspends when the URL queue is empty and no more agents are alive.

3.4.3 Relevance evaluation

The final relevance evaluation is performed by users who decide which of the harvested documents should be included in their collections. Relevant documents are made available to the particular user via a new document profile. The documents considered irrelevant by all users are ultimately rejected by the system. Relevance evaluation results with the refinement of the user profile.

3.5 Synchronizing HyperSDI Agents

Concurrent performance of a few hundred software agents require synchronization of the shared resources. HyperSDI agents share the following resources:

- the URL queue that contains the addresses that ought to be visited by an agent
- the Aho-Corasick automaton
- the Compound profile
- the dictionary of visited hyperlinks that contains URLs penetrated by the HyperSDI agents.

Synchronization does not imply the lost of autonomy. Agents consult shared structures in order to prevent multiple checking of documents accessed by a particular URL address, yet they are still capable of performing independently of users and other agents (Wooldridge, Jennings, 1995).

4. Implementation notes

The new HyperSDI system was implemented in SUN's Java programming language[4]. The system's database was created according to the relational methodology and the document collection is based on the HTML language.

In the implementation model, each HyperSDI agent is a separate Java thread running on the Java Virtual Machine (JVM)[4] with the access to the Web. The HyperSDI server is a Java application that coordinates running agents (threads), which concurrently filter documents from the Web sources. The coordination is executed by synchronizing access to the shared resources. Dependently on the hardware platform for the JVM, the number of concurrent threads may vary from a dozen or so to a few hundred. Relevant documents are stored in the HyperSDI document collection. Users access the whole system's functionality by an Internet browser. After successful authorization they may view their private collections, create and change their profiles and query the whole document collection by means of Java applets[5].

5. Summary

The new HyperSDI system filters information from active and passive sources on the Web. Heterogeneity of these sources, variety of protocols, formats and navigation techniques require the use of Intelligent Software

---

[4] SUN's Java homepage: http://www.java.sun.com.

[5] SUN's Java homepage: http://www.java.sun.com

Agent technology. HyperSDI agents perform on behalf of system users, supplying them with possibly relevant documents. These agents are furnished with knowledge on users' information needs (profiles), are capable of learning (profile refinement), are capable of reacting and pro-acting for the events in their environment (Internet) and – finally – are capable of choosing the proper navigation technique. Hence, they achieve better results than traditional mechanisms in filtering information from Web sources (Amati i inni, 1997).

The new HyperSDI system is based on the fat server/thin client model. Thus, not only it was adapted for filtering the Web, but also was made available on the Web through the Web browser. The latter was achieved thanks to the use of Internet standards such as HTML, XML and Java.

Bibliography

Abramowicz, W. (1985) Computer Added Dissemination of Information on Software in Networks, *Proceedings of Compas '85 - The European Software Congress*, December 10-13, Berlin West, 491-505.

Abramowicz, W. (1990) Hypertexte und ihre IR- basierte Verbreitung, *Humboldt Universität Berlin*, 292 +VII.

Abramowicz, W., Grabowski J. (ed.) (1990) Information Dissemination to Users with Heterogenous Interests. *Computers in Science and Higher Education, Mathematical Research,* Vol. 57, Akademie-Verlag, Berlin, s. 62-71.

Abramowicz, W., Ceglarek D. (1998) Applying Cluster-Based Connection Structure in the Document Base of the SDI System. *WebNet'98 World Conference of the WWW, Internet & Intranet,* Nov. 7-12, Orlando, Florida, USA.

Aho, A. V., Corasick M. J. (1975) Efficient String Matching: An Aid to Bibliographic Search; *Communications of the ACM*, vol. 18 no. 6.

Amati, G., D'Alosi D., Giannini V., Ubaldini F. (1997) A Framework for Filtering News and Managing Distributed Data; *Journal of Universal Computer Science,* vol. 3 no. 8, p. 1007-1021.

Balabanovic, M., Shoham Y. (1995) Learning Information Retrieval Agents: Experiments with Automated Web Browsing; *Department of Computer Science*, Stanford University.

Ceglarek, D. (1997) Applying Taxonomous Methods in Selective Distribution of Information (SDI) Systems Supplying Users with Business Information, Ph.D. *Thesis, The Poznan University of Economics*, Faculty of Economics, Poznań.

Edwards, P., Bayer D., Green C. L., Payne T. R. (1996) Experience with Learning Agents which Manage Internet-Based Information; *AAAI Spring Symposium on Machine Learning in Information Access*, Stanford.

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

Jaccard, J., Gautero M., Tomassini M. (1999) WebSailor: Smart Agent for Information overLOad Reduction; *Proceedings of The Fourth international Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM99)*, London, April 1999, pp 463-464; url: http: //www-iis.unil.ch /~wsailor /publications /paam99 /paam99.html.

Mulawka, J. J. (1996) Systemy ekspertowe, *Wydawnictwo Naukowo-Techniczne*, Warszawa.

Palme, J. (1998) Information Filtering; *Proceedings of the ITS'98 Conference.*

Pazzani, M., Nguyen L., Mantik S. (1995) Learning from hotlists and coldlists: Towards a WWW information filtering and seeking agent; *Department of Information and Computer Science*, University of California, Irvine; url: http://www.ics.uci.edu/~pazzani/Coldlist.html.

Petrie, C. J (1996) Agent-Based Engineering, the Web, and Intelligence, *IEEE Expert December 1996*; url: http: //cdr.stanford.edu /NextLink /Expert.html.

van Rijsbergen, C.J. (1979) Information Retrieval, *Butterworths*, London; url: http://www.dcs.gla.ac.uk/Keith/Preface.html.

Salton, G., McGill M. (1983) Introduction to Modern Information Retrieval, *McGraw-Hill Book Company.*

Tomaszewski, T. (1998) Filtering Legal Information from Hypertext Systems, *Ph.D. Thesis, The Poznan University of Economics, Faculty of Economics*, Poznań.

Wooldridge, M., Jennings N. (1995) Intelligent Agents: *Theory and Practice; Knowledge Engineering Review*, Volume 10 No 2, June 1995; url: http: //www.elec.qmw.ac.uk/dai/pubs/KER95.

ISSN 0208-8029
ISBN 83-85847-53-7

W celu uzyskania bliższych informacji i zakupu dodatkowych egzemplarzy prosimy o kontakt z Instytutem Badań Systemowych PAN
ul. Newelska 6, 01-447 Warszawa
tel. 837-35-78 w. 241 e-mail: bibliote@ibspan.waw.pl