Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 35 (2008) 524–530
Expert Systems with Applications
www.elsevier.com/locate/eswa

BioCrawler: An intelligent crawler for the semantic web

Alexandros Batzios *, Christos Dimou, Andreas L. Symeonidis, Pericles A. Mitkas

Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece

Abstract

Web crawling has become an important aspect of web search, as the WWW keeps getting bigger and search engines strive to index the most important and up to date content. Many experimental approaches exist, but few actually try to model the current behaviour of search engines, which is to crawl and refresh the sites they deem as important, much more frequently than others. BioCrawler mirrors this behaviour on the semantic web, by applying the learning strategies adopted in previous work on ecosystem simulation, called BioTope. BioCrawler employs the principles of BioTope's intelligent agents on the semantic web, learns which sites are rich in semantic content and which sites link to them and adjusts its crawling habits accordingly. In the end, it learns to behave much like the state of the art search engine crawlers do. However, BioCrawler reaches that behavior solely by exploiting on-page factors, rather than off-page factors, such as the currently used link popularity.

Keywords: Web crawling; Focused crawling; Multi-agent system; Semantic web

1. Introduction

Today's search engines use bots, commonly referred to as crawlers or spiders, in order to traverse and index the Web. Search engines keep the exact algorithms of these spiders as "closely guarded secrets" in order to prevent webmasters from manipulating the engines and skew search results to their favour.

From the Webmaster guidelines, as well as various announcements, of Google, Yahoo, and MSN, it seems that the entire search engine's index is used to determine the quality and importance of a site and thus, the frequency this site should be crawled. Google for example, still relies on the PageRank (Brin & Page, 1998) algorithm, which is effectively a measure of a page's popularity, to determine the importance of a page. Yahoo and MSN also consider link popularity as a factor when ranking and crawling a site, although no details have been officially announced.

This extensive use of off-page factors to determine page attributes has given birth to the phenomenon of "link-buying", where webmasters purchase links on other webmaster's pages for a monthly fee. In Google's case, for example, it is common knowledge that a link from a page with a Google PageRank of 8 can get a site crawled in about 24–48 hours, and such a link can cost anywhere from $100 to $500 per month, depending on the site's traffic and the number of other links on the page. On the other hand, there are spam sites that try to trick search engine crawlers into visiting them often and indexing their content. These sites are artificially linked together so as to maximize their link popularity.

However, as the web grows, it is becoming increasingly impractical to use the whole index of a search engine to determine site crawling. Furthermore, it is not certain that the current assumptions about link popularity will be correct in the coming semantic web, or during the early stages of its realization, where most pages will not contain semantic data but the ones that do will form their own *semantic net*.

Based on these facts, the authors feel that intelligence and learning will become increasingly crucial for web crawling in the years to come. Within the context of this

* Corresponding author.
E-mail addresses: alex@issel.ee.auth.gr (A. Batzios), cdimou@issel.ee.auth.gr (C. Dimou), asymeon@issel.ee.auth.gr (A.L. Symeonidis), mitkas@auth.gr (P.A. Mitkas).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.07.054

work we propose an agent-based framework for developing and testing intelligent crawlers for a semantic web search engine. We attempt to address some of the current issues web crawlers face, such as determining important sites, and creating a foundation for crawling the semantic web. The framework developed is envisioned as a testbed for a future semantic search engine to crawl and index semantic content, as well as semantic descriptions of web services. Since it is highly unlikely that the current linking scheme of the WWW will make sense in the semantic web, especially in the case of semantic web services, on-page factors will be used to determine a site's importance and crawling frequency. The proposed framework adopts ecosystem rules that apply to a set of autonomous cooperating entities that crawl the simulated web content, first introduced in the concept of BioTope (Symeonidis, Valtos, Seroglou, & Mitkas, 2005). This is the reason for naming our framework "BioCrawler".

BioCrawler is an extension to focused crawling in the sense that spiders need to focus on the distributed exploration of web sites and pages that contain semantically enhanced content, which will later be processed by the search engine. BioCrawler attempts to provide a configurable and controlled simulation framework in which crawlers will be tested in an artificial and controlled web environment so as to compare their throughput and overall effectiveness.

The rest of this paper is structured as follows: Section 2 reviews the background theory and work that is related to ecosystem simulation; in Section 3 we outline the proposed framework and its components; Section 4 delineates a set of experiments that underline a set of key concepts related to spider and environment efficiency. Finally, Section 5 summarizes the work done and concludes the paper.

2. Background work

2.1. Web crawling

Web spidering or web crawling is the automated and methodological traversing and indexing of web pages for subsequent search purposes (Kobayashi & Takeda, 2000). Since RBSE (Eichmann, 1994), WebCrawler (Pinkerton, 1994) and WWW worm (McBryan, 1994), the first published elementary crawlers, many advances have been made, including scheduling and indexing integration into crawlers (e.g., GoogleCrawler (Brin & Page, 1998), CobWeb (Da Silva et al., 1999) and Mercator (Heydon & Najork, 1999)). Advancing to the latest generation of crawlers, focus is given on the distribution of load to communities of cooperating spiders.

WebFountain (Edwards, 2001), for example, is a completely distributed and decentralized crawler, where scheduling, fetching, parsing and storing pages is shared among clusters of machines. Usually, a site is assigned into each cluster, and very large sites may be split among multiple clusters. FAST Crawler (Risvik & Michelsen, 2002) also follows a similarly decentralized approach where each machine in a cluster is assigned a different portion of the web to crawl. UbiCrawler (Boldi, Codenotti, Santini, & Vigna, 2004) is another distributed crawler that uses a series of cooperating software agents that autonomously coordinate their behaviour in such a way that each of them scans its share of the web. Finally, WebRACE (Zeinalipour-Yazti & Dikaiakos, 2002) is a crawler that follows a more centralized approach, but uses independent components (a MiniCrawler and an Annotation Engine) to optimize fetching, storing and indexing.

In addition to the above, intense research has been conducted towards 'focused crawling', a term referring to the collection of pages related to given keywords, topics or other web pages. Prior efforts on focused crawling, including Menczer, Pant, Srinivasan, and Ruiz (2001), Hersovici, Jacovi, Maarek, Pelleg, and Shtalhaim (1998), Diligenti, Coetzee, Lawrence, Giles, and Gori (2000), Chakrabarti (1999), address issues of content similarity while disregarding efficient distributed architectural design. Furthermore, while many of the above papers as well as previous works (e.g., Boldi, Santini, & Vigna, 2004; Cho, Garcia-Molina, & Page, 1998) refer to simulation of web crawling none offers a general, configurable simulation framework for testing and monitoring intelligent focused crawlers.

2.2. Ecosystem simulation

Recently, biology-inspired simulation techniques have drawn the attention of both biology and computer science researchers. The most prominent of such systems include *Biomolecular* (Bates & Maxwell, 1993; Branden & Tooze, 1999), *Metabolic* (Kleinstein, 2000; Seiden & Celada, 1998), *Ecosystem* (Haile & Weidhaas, 1997; Iyengar, 1998) as well as *Artificial life* (Minar, Burkhart, Langton, & Askenazi, 1996). The latter two applications in particular have drawn the attention of researchers in distributed artificial intelligence (DAI) for autonomous software entities deployment and testing purposes.

Existing efforts in the literature vary from building individual based models that are focused on the environmental aspects of the system (Haefner & Crist, 1994), to multi-species communities that consider both ecological and evolutionary dynamics (Pecala, 1986). Focus has also been set on the societal aspects of a model (Bousquet, Cambier, & Morand, 1994) or the learning processes employed in society evolution (Epstein & Axtell, 1996; Holland, 1995; Hraber, Jones, & Forrest, 1997; Pollack & Ringuette, 1990; Ray, 1992). An interesting approach is that of Biotope (Symeonidis et al., 2005), a configurable tool for designing distributed computing simulation environments. According to Biotope's architecture, the environment and its participating entities can be easily used as abstract models.

BioCrawler employs BioTope's modelling infrastructure for designing a web crawler-specific simulation environment. It is our strong belief that this work contributes in the following aspects:

(1) Deployment of a configurable simulation framework for crawlers traversing the Semantic Web.
(2) Introduction of semantically enhanced simulated web content.
(3) Study of evolutionary learning on cooperating web crawlers in uncertain environments.
(4) Provision of a communication framework that helps crawlers minimize work effort and maximize their effectiveness in the simulation space.
(5) Allows exhaustive testing of different learning and communication schemes.

3. Overview of BioCrawler

In this section, we provide a detailed overview of BioCrawler's architecture and components, namely (a) the simulation environment,(b) the agent model, (c) the communication framework and (d) the knowledge model of the agents.

3.1. Overall architecture

BioCrawler's overall architecture consists of a hypothetical semantic search engine that uses a certain number of autonomous crawlers to search the web for semantic content, either in content pages or within the semantic description of web services. These crawlers traverse the web by starting from a random page and continue by following links to other pages. At each hop, the crawler evaluates the visited content and sends the acquired content back to the search engine. Crawlers may also exchange information with other crawlers in order to notify their peers about sites with "rich" semantic content.

According to the BioCrawler approach, a crawler is considered to be an autonomous living entity that is equipped with a certain amount of energy, vision, moving and communication abilities, as well as an intrinsic knowledge model that is refined after any interaction of the crawler with its environment. Crawlers are motivated to find useful content in their search so as to maintain a higher energy level. Therefore, they are rewarded with energy increase for indexing semantic information; conversely, they are penalized with energy reduction for wasting bandwidth in invalid links or sites where a *robots.txt* file forbids crawling.

3.2. Simulation environment

The actual space that crawlers move is a constrained environment of randomly interlinked websites. A percentage of these sites have OWL files with semantic content, another percentage of pages contain invalid links and some sites use a robots.txt file to forbid crawling. A page on a site has a random number of outgoing links and these links lead to combinations of all the above. This environment is easy to create through the BioCrawler interface. The parameters used to define the total number of sites as well as the percentages of each type of content are discussed in Section 4.2.

Following the terminology introduced in Symeonidis et al. (2005), each page may be classified into one of the following states: (a) vacant, (b) resource enrichment, (c) resource reduction and (d) obstacle. In BioCrawler, however, vacant and obstacle also lead to resource reduction by different amounts of energy. The above terms are mapped to corresponding BioCrawler terms, as summarized in Table 1. The BioCrawler-specific terms are based on the following architectural concepts:

(1) *No semantic content*. Denotes a plain HTML web page that contains no additional semantically enhanced information. Crawlers do not need to index its content. Crawlers lose some of their energy by visiting such pages. No semantic content web pages are considered to be vacant cells.
(2) *Semantic content*. Denotes an OWL file that contains semantic data. Crawlers may thereafter index the underlying content. Spiders increase their energy by a fixed amount when discovering OWL files. Semantic content pages are considered to be "food cells".
(3) *Invalid, non-HTML*. A link to a page that either does not exist or is not an HTML file. Examples of such pages are PDF, sound or video files. These pages are BioCrawler's equivalent of the "trap cells" seen in BioTope.
(4) *Robots.txt*. A file contained in the visited web site to denote that the web site owner forbids any indexing of the underlying content. Robots.txt sites are considered to be "obstacle cells".

3.3. Agent model

As already stated in the previous paragraph, a crawler has an initial amount of energy which it aims to increase by finding more semantic content (food) per unit of bandwidth (site visit). For this reason, crawlers are equipped with (a) vision, (b) movement and (c) communication capabilities.

3.3.1. Vision

We consider a vision area of a fixed number of domains for any crawler. This number is defined through the appli-

Table 1
Page content for BioCrawler

Cell content	BioCrawler term
Vacant space	No semantic content
Resource enrichment	Semantic content
Resource reduction	Invalid link, non-HTML file
Obstacle	Robots.txt

cation's user interface. BioCrawlers learn based on domains and not individual pages. The vision vector is populated by the outgoing links of the current page the crawler is at. If the page links to less distinct domains than the actual size of the vision vector, the links of the previous page or pages the crawler has been at are used to fill in the remaining spaces in the vector. Crawlers can choose to visit any of the domains in their vision vector.

3.3.2. Movement

A crawler is generally free to move to any site in the virtual web environment, by following links on the various pages. The only constraint is that the crawler must be able to "see" that domain, which means that the domain should be in its vision vector. While domains are used by crawlers to select their next move, the actual move itself is page-specific. If a crawler comes across a page with links to 2 pages in Domain1 and 3 pages in Domain2, it decides between the domains first and then randomly selects one of the page links to that domain.

3.3.3. Communication

Crawlers need to communicate with each other in order to exploit the already recorded information about the content of the web. When communicating, spiders exchange their best rules (the ones with the highest strength) that determine the optimal routes in their vision field. Communication takes place through a Rule Manager agent that accepts rules from all crawlers and, upon request from a crawler, sends its best rules to that crawler.

Every crawler can initiate communication in order to either propose new rules to the Rule Manager or request new rules at any given time. In the current implementation, the communication period is a user-defined interval of visited sites.

3.4. Knowledge model

The most important feature of the BioCrawler framework is the ability of crawlers to augment their intelligence. Crawlers need to be able to evaluate their decisions and adapt their knowledge model in real time. The learning mechanism that crawlers use comprises two parts. First there is a set of classifiers, that is essentially the knowledge model itself. Each classifier is actually an *IF–THEN* rule, with the IF part containing the links the crawler currently "sees" and the THEN part containing the link the crawler should decide to follow. These classifiers need to be evaluated and adjusted as each crawler gains more experience and knowledge about its environment. Hence, a classifier evaluation mechanism is also adopted, which is based on the amount of energy (semantic content) the crawler is able to gather by following each rule. This amount is added to the rule's *strength*. If multiple rules have the same classifier, the strongest rule is applied. Every application of a rule also includes a strength tax, so that the crawler will not constantly follow the same rule. Both of these mechanisms are described in detail in the BioTope paper (Symeonidis et al., 2005).

4. The BioCrawler framework

4.1. Implemented system

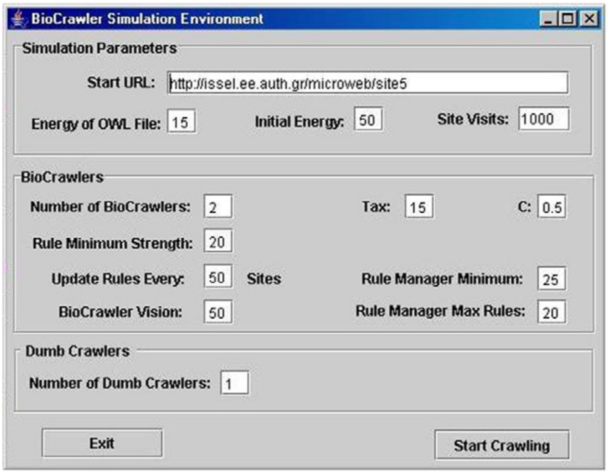
The simulation environment for BioCrawler has been implemented from scratch and is completely independent from that of BioTope. This was necessary since the grid-based approach of BioTope was not deemed suitable for modeling the WWW. All software components were written in Java (v1.5). The Java Agent Development Framework (JADE) (Bellifemine, Poggi, & Rimassa, 2001) has been incorporated in order to construct all the participating agents and communication protocols, according to the FIPA specifications. BioCrawler's user interface, shown in Fig. 1, assisted in easily setting up and conducting the sets of experiments further delineated in the following paragraphs. At this point, it should be noted that the parameters of BioCrawlers, as shown in the interface, are not absolute, pre-defined performance indicators outside the context of BioCrawler. BioCrawler parameters, such as vision and rule tax, work in tandem with the environmental parameters, such as links per page and energy per OWL file. Hence, performance comparisons are only valid within the context of the BioCrawler framework.

4.2. Assessment indicators

We embark our experimental testing of the proposed system by defining a series of *environmental* and *spider performance* assessment indicators that allow users to monitor the effect of changes in spider behavior.

4.2.1. Environmental indicators

The environmental indicators for BioCrawler have been borrowed from Symeonidis et al. (2005) (see Table 2).



Simulation Parameters

Start URL: http://issel.ee.auth.gr/microweb/site5

Energy of OWL File: 15 Initial Energy: 50 Site Visits: 1000

BioCrawlers

Number of BioCrawlers: 2 Tax: 15 C: 0.5

Rule Minimum Strength: 20

Update Rules Every: 50 Sites Rule Manager Minimum: 25

BioCrawler Vision: 50 Rule Manager Max Rules: 20

Dumb Crawlers

Number of Dumb Crawlers: 1

Fig. 1. A snapshot of the BioCrawler User Interface.

Table 2
Environmental indicators used in the experiments

Environment variable	Value
Number of Sites	1000
Semantic %	10
Forbidden %	1
Invalid %	2
Energy reward	15 points

(1) *Semantic* % is defined as the % ratio of the total semantically enhanced web pages *wp* to the total pages in the virtual web environment.
(2) *Forbidden* % is defined as the % of sites in the virtual web environment with a robots.txt file which forbids crawling and indexing of their content.
(3) *Invalid* % is defined as the % of invalid links in the environment.
(4) *Energy reward* is the amount of energy each crawler gains upon finding an OWL file. This amount is the same for all OWL files across the virtual web environment.

4.2.2. Crawler performance indicators

The proposed spider performance indicators in Symeonidis et al. (2005) are not used individually in BioCrawler as the purpose of the environment is not to study the actual behaviour of the crawlers. Instead, the focus has been placed in the crawler's *s-Throughput*, which is defined as the amount of energy gained per unit of bandwidth.

The energy of each crawler increases by a fixed amount when the crawler consumes semantic content and decreases every time the crawler moves. Sites with robots.txt files and invalid links consume more energy than simple moves. A unit of bandwidth is defined as the opening of an HTTP connection by a crawler.

5. Experiments

Three different series of experiments were conducted. Due to bandwidth constraints, only two BioCrawlers and one Dumb Crawler were able to run concurrently. Larger-scale experiments would have produced the same results for Dumb Crawlers, and slightly better results for BioCrawlers since there would be more knowledgeable BioCrawlers exchanging information with each other. In our sets of experiments, the two BioCrawlers were able to communicate with each other and alter their behaviour according to the information they received. A single Dumb Crawler was selected since it does not need to communicate with any of its peers. Crawlers were made to start from the same site, and experiments also tested the individual learning capacity of BioCrawlers (without communication). In all experiments the throughput of each BioCrawler greatly surpassed that of the Dumb Crawler as can be seen in the following paragraphs.

The first series of experiments attempts to determine the near-optimal values for several environmental, communication and learning parameters with respect to the BioCrawler throughput indicator. The more often BioCrawlers are set to visit known semantic sources to refresh their index, the less the rate of discovery of new sites will be and the longer it will take for a crawler to complete a pass. Since the goal is to maximize the semantic content received per unit of bandwidth (site visit) and speed is secondary, BioCrawlers were set to re-visit semantic sources at a rate that will not increase the time required to make a complete pass to more than twice the time a Dumb Crawler would need to make that pass. Furthermore, in terms of scalability, it is unlikely that making a pass of the whole WWW will be significant for a search engine since only a relatively small part of the WWW contains quality sites and an even smaller number of them contain semantic data (see Table 3).

The second series of experiments compares the overall throughput of semantically enhanced content that BioCrawlers discover, to the throughput of Dumb Crawlers, with the Dumb Crawlers using the simple algorithm described in Thom Blum, Doug Keislar, Jim Wheaton, Zeinalipour-Yazti, andDikaiakos,Dika. Crawlers were left to perform 30 000 site visits and every time a crawler finished a pass of the virtual web (had crawled all 1000 different sites) it would start from a random domain, while preserving the rules it had obtained in all previous passes. As illus-

Table 3
BioCrawler parameters used in the experiments

BioCrawler parameter	Value
Initial energy	50
Rule min. strength	20
Sites before rule update	50
Rule mgr max rules	20
Rule mgr min. rule strength	25

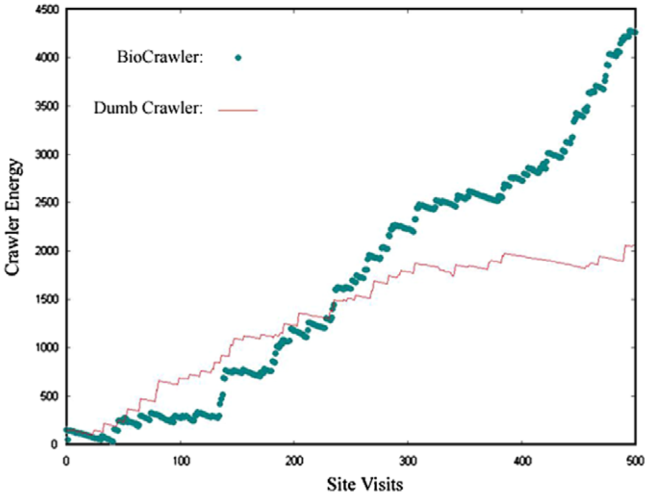


Fig. 2. Crawler energy during the first 500 site visits.

trated in Fig. 2, in the beginning the performance of BioCrawlers fluctuates as they have yet to learn more about their environment. At some point, Dumb Crawlers even surpass BioCrawlers. However, Fig. 3 shows that as BioCrawlers perform more site visits and become more knowledgeable about their web environment, their throughput greatly surpasses that of the Dumb Crawlers.

In the third series of experiments, BioCrawlers and Dumb Crawlers were re-started from a random site (which was the same for all crawlers), after completing each pass, while also deleting all rules from the memory of BioCrawlers. This was done to verify that BioCrawlers do not consistently out-perform Dumb Crawlers due to a lucky start. BioCrawlers and Dumb Crawlers were re-started 100 times and Fig. 4 depicts the average energy level at each site visit from these tests. Again, the difference in semantic throughput is significant.

In all experiments, BioCrawlers periodically re-visited known sites with semantic content while at the same time ensured that they continued to discover new sites. This was due to the tax imposed on each rule, which effectively determines the rate with which crawlers re-visit sites versus the rate with which crawlers discover new sites. Finally, crawler log files were used to verify that crawlers did, in fact, visit all available sites in the virtual web environment and that although crawlers with the same parameters exhibited a similar behaviour, they were always visiting different sites at the same time.

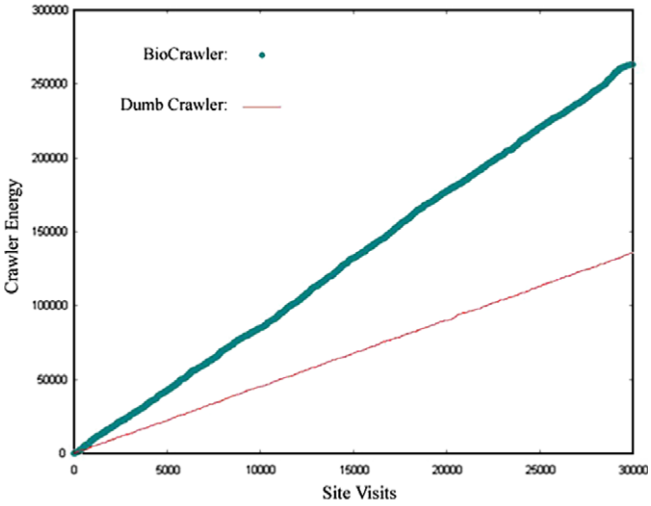


Fig. 3. Crawler energy during 30,000 site visits.

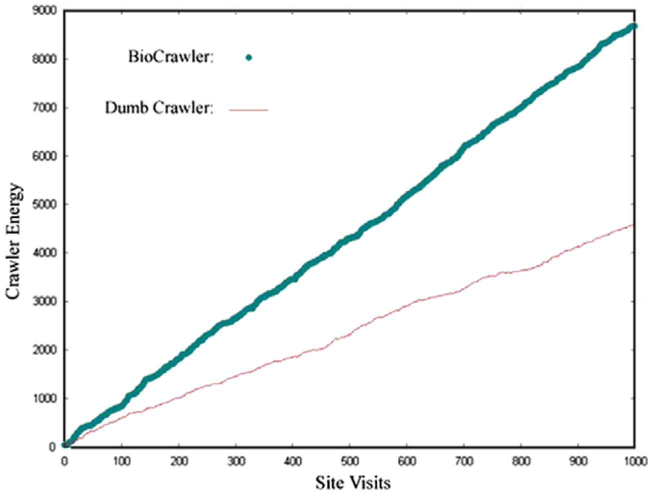


Fig. 4. Average Crawler energy during 100 random re-starts.

To conclude, all experiments showed that BioCrawlers exhibited a behaviour along the lines of the current search engines, but focused only on sites with semantic content. Since no distinction in link popularity is likely to occur for sites with semantic content in particular, the behaviour of re-visiting sites to refresh the search engines index was based solely on on-page factors (content), rather than off-page ones.

6. Conclusions

Exploring the vast WWW has always been a challenge for search engines and human searchers alike. The coming semantic web will definitely introduce new ways of searching, whether it is for content or web services and this might create a niche for semantic search engines. Current search infrastructures heavily depend on link popularity of content pages which may not be a viable approach for the semantic web, especially when it comes to web services. Furthermore, new start-up search engines may emerge in this niche which will not have the vast WWW index of today's major players in search. A focused crawler in this area focusing more on content rather than link popularity such as BioCrawler represents an interesting alternative.

Although BioCrawler's behavior mirrors that of current crawling practices, it is understood that adopting ecosystem rules and knowledge models is not today's alternative for link popularity, and other off-page factors. The authors do feel, however, that more intelligence should be built into crawlers and search engines alike so that more weight can be placed on semantic content and on-page factors in general. BioCrawler provides a good starting point for more intelligent crawlers, perhaps one day conscious of the actual content that they harvest.

References

Bates, A. D., & Maxwell, A. (1993). DNA Topology in Focus series. New York, NY: Springer.

Bellifemine, F., Poggi, A., & Rimassa, G. (2001). Developing multi-agent systems with a FIPA-compliant agent framework. *Software, Practice and Experience, 31*, 103–128.

Boldi, P., Codenotti, B., Santini, M., & Vigna, S. (2004). UbiCrawler: A scalable fully distributed Web crawler. *Software, Practice and Experience, 34*(8), 711–726.

Boldi, P., Santini, M., & Vigna, S., (2004). Do your worst to make the best: Paradoxical effects in pagerank incremental computations. In *Proceedings of the third Workshop on Web Graphs (WAW)*, volume 3243 of Lecture Notes in Computer Science (pp. 168–180). Rome, Italy: Springer.

Bousquet, F., Cambier, C., & Morand, P. (1994). Distributed Artificial Intelligence and Object-Oriented Modelling of a Fishery Mathematical. *Computation Modelling, 20*(8), 97–107.

Branden, C., & Tooze, J. (1999). Introduction to Protein Structure (Second ed.). New York, NY: Garland Publishing Inc.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems, 30*(1-7), 107–117.

Chakrabarti, S., van der Berg, M., & Dom, B., (1999). Focused crawling: a new approach to topic-specific Web resource discovery. In *Proceedings of eighth International World Wide Web Conference* (pp. 545–562).

Cho, J., Garcia-Molina, H., & Page, L., (1998). Efficient crawling through URL ordering. In *Proceedings of the seventh conference on World Wide Web*.

Da Silva, A.S., Veloso, E.A., Golgher, P.B., Ribeiro-Neto, B.A., Laender, A.H.F., & Ziviani, N., (1999). Cobweb a crawler for the Brazilian web. In *Proceedings of String Processing and Information Retrieval (SPIRE)* (pp. 184–191). Cancun, Mexico: IEEE CS Press.

Diligenti, M., Coetzee, F., Lawrence, S., Giles, C.L. & Gori, M., (2000). Focused crawling using context graphs. In *Proceedings of 26th International Conference on Very Large Databases* (pp. 527–534) VLDB 2000.

Edwards, J., McCurley, K.S., & Tomlin, J.A., (2001). An adaptive model for optimizing performance of an incremental web crawler. In *Proceedings of the Tenth Conference on World Wide Web* (pp. 106–113).

Eichmann, D., (1994). The RBSE spider: balancing effective search against Web load. In *Proceedings of the First World Wide Web Conference*, Geneva, Switzerland (pp. 113–120).

Epstein, J.M., & Axtell, R.L., 1996. *Growing Artificial Societies: Social Science from the Bottom Up*. Washington: The MIT Press.

Haefner, J. W., & Crist, T. O. (1994). Spatial model of movement and foraging in harvester ants (Pogonomyrmex) (I): The roles of memory and communication. *Journal of Theoretical Biology, 166*, 299–313.

Haile, D.G., & Weidhaas, D.W., (1997). Computer Simulation of Mosquito Populations (*Anopheles albimanus*) for comparing the effectiveness of control techniques. *Journal of Medicine*, 553–567.

Hersovici, M., Jacovi, M., Maarek, Y.D., Pelleg, D., Shtalhaim, M. & Sigalit Ur., (1998). The shark-search algorithm – An application: Tailored Web site mapping. In *Proceedings of the 7th International World-Wide Web Conference*.

Heydon, A., & Najork, M. (1999). Mercator: A scalable, extensible Web crawler. *World Wide Web Conference, 2*(4), 219–229.

Holland, J. H. (1995). Hidden order: How adaptation builds complexity. Reading, MA: Addison-Wesley.

Hraber, P. T., Jones, T., & Forrest, S. (1997). The Ecology of Echo Artificial Life III. In C. G. Langton (Ed.) (pp. 165–190). Addison Wesley, Longman.

Iyengar, S. S. (1998). Computer Modeling and Simulations of Complex Biological System. CRC Press.

Kleinstein, S.H., & Seiden, P.E., (2000). Simulating the Immune System. *Computing in Science and Engineering*, 69–77.

Kobayashi, M., & Takeda, K. (2000). Information retrieval on the web. *ACM Computer Survey, 32*(2), 144–173.

McBryan, O.A., (1994). GENVL and WWWW: Tools for taming the web. In *Proceedings of the First World Wide Web Conference*, Geneva, Switzerland (pp. 1–13).

Menczer, F., Pant, G., Srinivasan, P., & Ruiz, M.E., (2001). Evaluating topic-driven web crawlers. In *Proceedings of the 24th conference of research and development in information retrieval (SIGIR)* (pp. 241–249). ACM Press.

Minar, N., Burkhart, R., Langton, C., & Askenazi, M. (1996). The Swarm simulation system: A toolkit for building multi-agent systems. Available: http://wiki.swarm.org.

Pecala, S. W. (1986). Neighborhood models of plant population dynamics. *Multispecies models of annuals, 29*, 262–292.

Pinkerton, B., (1994). Finding what people want: Experiences with the WebCrawler. In *Proceedings of the First World Wide Web Conference*, Geneva, Switzerland.

Pollack, M.E., Ringuette, M., 1990. Introducing the Tileworld: Experimentally Evaluating Agent Architectures. In *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 183–189).

Ray, T. S. (1992). An approach to the synthesis of life. In C. G. Langton, C. Taylor, J. D. Farmer, & S. Rasmussen (Eds.), *Artificial Life II* (pp. 371–408). Redwood City, CA: Addison-Wesley.

Risvik, K.M., & Michelsen, R. (2002). Search Engines and Web Dynamics. *Computer Networks, 39*, 289–302.

Seiden, P. E., & Celada, F. (1998). *A Simulation of the Immune System. Experiments in machine*. Singapore: World Scientific Press.

Symeonidis, A. L., Valtos, V., Seroglou, S., & Mitkas, P. A. (2005). Biotope: an integrated simulation tool for Augmenting the intelligence of multi-agent communities residing in hostile environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, Special Issue on Self-organization in Distributed Systems Engineering, 35*(3), 420–432.

Thom Blum, Doug Keislar, Jim Wheaton, & Erling Wold: "Writing a Web Crawler in the Java Programming Language. Muscle Fish LLC, Article on Sun Microsystems Website: http://java.sun.com/developer/technicalArticles/ThirdParty/WebCrawler.

Zeinalipour-Yazti, D., & Dikaiakos, M.D., (2002). Design and implementation of a distributed crawler and filtering processor. In *Proceedings of the Fifth Next Generation Information Technologies and Systems (NGITS)*, volume 2382 of Lecture Notes in Computer Science (pp. 58–74). Caesarea, Israel: Springer.