State of the Art in Metadata Abstraction Crawlers

Hai Dong, Farookh Khadeer Hussain, Elizabeth Chang
Digital Ecosystem and Business Intelligence Institute
Curtin University of Technology
Perth, Australia
E-mail: {hai.dong, farookh.hussain, elizabeth.chang}@cbs.curtin.edu.au

Abstract—**Nowadays, the research of crawlers moves closer to the semantic web, along with the appearance of increasing XML/RDF/OWL files and the rapid development of ontology mark-up languages. As an emerging concept, metadata abstraction crawlers are a series of crawlers that aim to abstract metadata from normal HTML documents, based on various semantic web technologies. In this paper, we make a general survey of the current situation of metadata abstraction crawlers. Fourteen cases in this field are chosen as typical examples, and classified in five clusters. From seven perspectives we horizontally compare and contrast the semantic web crawlers in each cluster, and draw our conclusion in the final section.**

Keywords-focused crawlers, metadata abstraction, OAI-PMH, RDF crawlers, semantic web crawlers

I. Introduction

Semantic web is a vision of future web, in which information is categorized and made comprehensible by various automated tools [12]. The major mission of semantic web is to "*express meaning*", this demands agents to execute more intelligent operations on behalf of users [21]. A crawler is an agent which can automatically search and download webpages [2]. Nowadays, the research of crawlers moves closer to the semantic web, along with the appearance of increasing XML/RDF/OWL files and the rapid development of ontology mark-up languages [10, 11]. As an emerging concept, metadata abstraction crawlers are a series of crawlers that aim to abstract metadata from normal HTML documents, based on various semantic web technologies.

In this paper we make a general survey towards the situation of metadata abstraction crawlers. Fourteen cases, in this field, are chosen as typical examples, and classified in five clusters according to their features, utilized technologies and service objects. From seven perspectives, we horizontally compare and contrast the semantic web crawlers, based on a simple statistical approach, and draw our conclusion in the final section.

II. Metadata Abstraction Crawlers

A. Normal metadata abstraction crawlers

This group of crawlers does not have distinct functions, apart from the function of metadata abstraction.

Davulcu et al. propose an OntoMiner system, with the purpose of organizing the overlapped websites provided by users, based on automatically generated ontologies (Fig. 1). First of all, a web crawler fetches all webpages from a given website. A Semantic Partitioner then analyses the labels in the webpages and builds a hierarchical tree of labels. Next, a Taxonomy Miner clusters the frequent labels into several concepts as the concepts' attributes, by means of the Frequent Tree Mining algorithm, in order to build a conceptual hierarchy. For each concept in the hierarchy, an Instance Miner associates the concept with the potential webpage instances, and computes the labeled and unlabeled attribute values for the instances [3].

Panayiotou and Samaras propose a personalized knowledge portal – mPERSONA – for the collaboration between wireless users and content providers. When the content providers join in this system, they need to submit their URLs and characteristic keywords. To semanticize the content providers' websites, a specialized crawler is designed to convert each page's content to metadata, in order to build a semantic tree. Each node of the tree is represented by the characteristic keywords. A thesaurus is used to find the synonyms for each keyword, in order to enrich each node's semantic meaning. Thereafter the metadata fetched by the crawler are linked to the nodes, which are groups of topics, thus it can clarify the semantic meaning of each node [18].

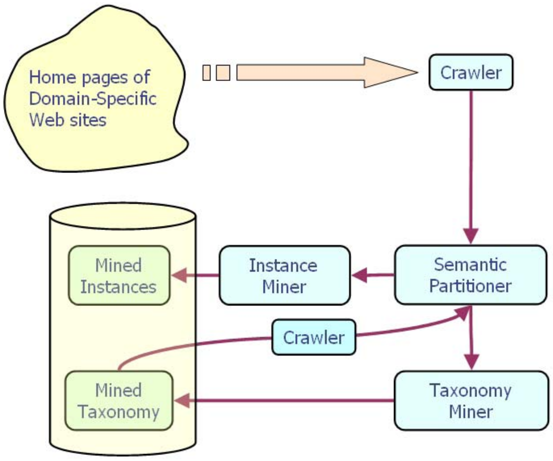

Figure 1. Architecture of OntoMiner

978-1-4244-1706-3/08/$25.00 ©2008 IEEE.

Topic maps are a semantic technology which classifies knowledge by topics. Roberson and Dicheva propose an approach to create the drafts of topic maps for websites. A crawler is used to download all webpages in a website, and then extract the semantic information regarding topic, by means of a set of heuristics [17].

Shimazu and Arisawa propose a content management system for interdisciplinary metadata exchange. A crawler is used to collect source files from a local network. A Natural Language Analyzer is used to parse, identify and annotate the name entities from the source files. Thereafter metadata in the format of Dublin Core are abstracted from the annotated source files, based on the method of 5W1H (when, who, what, where, why and how). Finally, these metadata are indexed and stored in index files for further content search [19].

B. RDF crawlers

The term of RDF crawler originates from Ontobroker, which is a series of crawler in the objective of generating RDF metadata [4]. All of the following systems use the RDF crawler as their technical backbone, to achieve different goals.

Decker et al. propose an Ontobroker system, with the purpose of extracting, reasoning and generating RDF-annotated metadata. Five major components are contained in the system (Fig. 2). First of all, the domain-specific ontologies are stored in a knowledge base. Subsequently, an Ontocrawler is designed to extract the formal knowledge from HTML web pages. Two different approaches are implemented here. For the similarly structured HTML files, a wrapper is used to generate their formal descriptions, by means of referring to an ontology in the knowledge base; for the specially structured HTML files, an annotation language is used. The descriptions are reasoned by an inference engine. Next, a RDF-Maker converts the reasoned descriptions to the metadata in the form of RDF. Finally, a query interface is designed to allow users to browse the ontologies and metadata [4].

Handschuh and Staab design a framework of metadata creator – CREAM. A RDF crawler is utilized to find references for created metadata, with the purpose of avoiding duplication [7]. In the CREAM, when the metadata creator wants to find whether an instance already exists or not, the RDF crawler retrieves the instance from the local knowledge base, which stores the RDF files harvested from the semantic web [8]. If a URI with regards to the instance is returned by the RDF crawler, the creator will then be aware that the relational metadata is created [9].

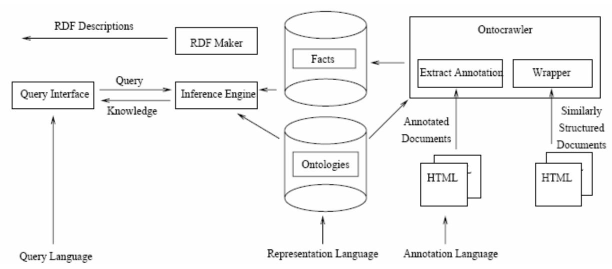

Figure 2. Architecture of Ontobroker

Stojanovic et al. propose a platform – SEAL, for semantic protal development. Ontobroker is the backbone of the platform, which works as a middleware between web server and knowledge warehouse for the service of RDF generation, knowledge portal template generation, ontology query and ranking and so forth. A RDF crawler is used to build the knowledge warehouse by generating RDF documents from the internet [22].

C. OAI-PMH-based crawlers

OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) is a HTTP-based protocol which can be used to retrieve XML metadata [16]. The following crawlers utilize this protocol in order to harvest XML metadata.

Nelson et al. utilize the OAI-PMH to enhance the competence of normal web crawlers, with the purpose of crawling metadata from webpages. OAI-PMH is based on a data model composed of three layers – resource, item and records (Fig. 3). OAI-PMH identifier can identify items which are "*entry point to all records (metadata) pertaining to the resource (web documents)*". To enhance the accuracy of metadata searching, the crawler uses an XML-based complex object format – MPEG-21 Digital Item Declaration Language (MPEG-21 DIDL) for formatting the retrieved digital objects (items), which consists of the concept of item (a group of items/components), component (a group of resource), resource (an individual data stream), container (a group of containers/items), and descriptor/statement (information pertaining a item, a component or a container) [16].

In addition, Smith and Nelson propose to use OAI-PMH to convert web information into the format of CRATE – a self-contained preservation-ready version of the resource, in which an entity is composed of resource and its associated metadata in the format of XML [20].

D. Focused crawlers

Focused (topical) crawlers are a group of distributed crawlers that specialize in certain specific topics [1]. Each crawler can analyze its topical boundary when fetching webpages. With semantic web technologies, focused crawlers are able to semanticize the fetched domain-specific information, in order to make it human-understandable.

Yang proposes a semantic web crawler program working in an ontology-based web environment (Fig. 4). First of all, a knowledge base is designed, which stores ontologies. A web

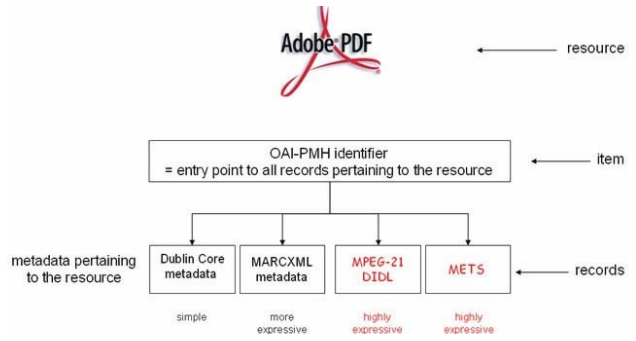

Figure 3. OAI-PMH data model

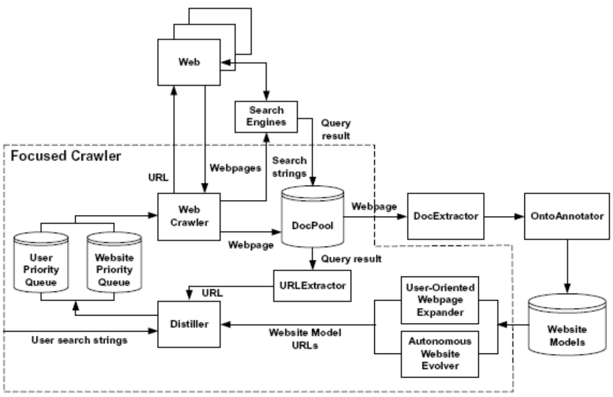

Figure 4. Architcture of focused crawler

crawler then obtains all data from a given website. Next, the web information is modeled, which contains a website profile and all associated webpage profiles. Each profile includes the basic description, static information, and ontological information regarding a corresponding webpage. To realize this objective, a DocExtractor program is designed to extract the basic information from a webpage for the first section, calculating statistical data for the second section and remove all HTML tags. Subsequently, an OntoAnnotator is used to annotate the web metadata for the third section. Within the DocExtractor, a HTML Analyzer is utilized to analyze the webpages from a DocPool which contains the webpages from the retrieved website, then extracts the information regarding URLs, titles, anchors and headings, and calculates the statistical data regarding tags. Thereafter a HTML Tag Filter is used to remove all tags from the analyzed webpages, and a Document Parser converts the tag-free webpages into a list of keywords. These keywords are passed to an OntoAnnotator. In the OntoAnnotator, an OntoClassifier is used to describe each webpage with the mostly matched classes of domain ontology based on the tf-idf algorithm. Following an Annotator is used to annotate the webpage with the classes and their frequencies, and a Domain Marker is used to determine the belonged domain, based on the class frequencies for the webpage [23].

Francesconi and Peruginelli propose a so-called Vertical Portal, with the purpose of providing both resources and available solutions and services to satisfy users' requirements, within a legal domain. A focused crawler is adopted in the system, to crawl the domain-specific web documents. Thereafter a metadata generator automatically transforms the web documents into metadata, by means of extracting. The focused crawler is implemented by computing the possibility of URLs regards to the predefined topics. The metadata format is in accordance with the Dublin Core (DC) scheme in its XML version. Subsequently the tf-idf model is used in order to extract the terms which can represent the documents. Next, two algorithms – Naive Bayes (NB) and Multiclass Support Vector Machines (MSVM) are adopted respectively for the documents classification [5].

Giles et al. propose a niche search engine for retrieving e-business information, with the integration of CiteSeer technique. A set of crawling strategies, including Brute Force, Inquirus-based and focused crawlers are used to fetch web documents. The CiteSeer technique is used to parse citations from the downloaded documents, and subsequently creates metadata based on the documents. To enhance the quality of metadata, the Support Vector Machine (SVM) algorithm is chosen to extract metadata, by the comparison with the Hidden Markov Model (HMM) algorithm [6].

E. Non-text crawlers

Some crawlers are not designed to only retrieve plain text documents. The following crawlers take advantage of semantic web technologies to enhance their ability of searching non-text documents.

Liu et al. propose a media agent for managing personal multimedia files. An online crawler and an offline crawler are introduced in the system, in order to collect the metadata regarding multimedia files. The difference between the two crawlers is that the former can work when a user is operating an online multimedia file; the latter only works according to a user's predefined preference data, when the user is offline. Both of the crawlers contain three subcomponents – semantic collection, features extraction and multimedia file description indexing. The semantic collection is to collect the URLs and semantic descriptions of multimedia files. The latter are extracted from the titles, surrounding texts of the multimedia objects. The feature extraction is to extract feature keywords from the semantic descriptions. Subsequently each multimedia file description is seen as a collection of keywords, and thus can be ranked by means of the tf-idf algorithm from the Vector Space Model (VSM) [13].

Liu et al. propose a specific table search engine – TableSeer [14]. A table crawler is used in the system to crawl PDF documents with tables in a digital library. Then a Doc Classifier classifies the documents into six categories and discards the documents without tables. For each identified table, a table metadata is created. And a specific table rank algorithm – Table Term Frequency – Inverse Table Term Frequency (TTF-ITTF) is used to index the metadata [15].

III. Compare and Contrast of the Metadata Abstraction Crawlers

In the following sections, we make a comprehensive compare and contrast to the introduced metadata abstraction crawlers, and conclude their general features based on each cluster, by means of a simple statistical approach. According to their typical features, we choose seven aspects to analyze, including domain, working environment, special functions, technologies utilized, evaluation methods, evaluation results, and finally we present our comments or suggestions to each crawler. Our conclusion to the comparison result is drawn in the final section.

A. Compare and contrast of the normal metadata abstraction cralwers

Table 1 reveals the result of compare and contrast among the four normal metadata abstraction crawlers. It is not difficult to find that most of these crawlers are designed to apply in general domains (4/4). The same as normal web crawlers, most of the crawlers are used as the component of a larger system (3/4). In addition to the function of metadata abstraction (4/4), many crawlers are able to create a draft of a semantic tree based on the abstracted metadata (3/4). A few crawlers are designed to index the metadata according to the characteristics of metadata (1/4). Each crawler in this group has its own technologies for realizing metadata abstraction and other special functions. Only one case provides the experimental details, and it adopts the evaluation method – recall and precision, from traditional information retrieval research. On account of the different realization methodologies, the experimental results vary significantly between crawlers. In general, our suggestion is to provide more evaluation details (3/4).

TABLE I. COMPARE AND CONTRAST OF THE NORMAL METADATA ABSTRACTION CRALWERS

Name	**OntoMiner Crawler**	**Topic Map Crawler**
Domain	General	General
Working Environment	OntoMiner	General
Special Functions	Automatically generating an ontology according to the fetched webpages in a website.	Creating a draft topic map; abstracting metadata regarding topics from webpages.
Technologies Utilized	Semantic Partitioner for extracting a tree of labels from the fetched documents; Taxonomy Miner and Frequent Tree Mining algorithm for building a ontology whose attributes are labels; Instance Miner for associating instance web pages with ontological concepts and computing their attribute values.	A set of heuristics for abstracting metadata regarding topics, from web pages.
Evaluation Methods	Not provided.	Calculating the crawler's precision and recall values in two example websites.
Evaluation Results	Not provided.	Recall rate varies from 0.77 to 1 for different objects; precision rate varies from 0.38 to 0.57 for the different websites.
Comments/ Suggestions	Provide evaluation details, and more technical details regarding how the crawler model is realized.	Overall precision rate is a bit low.
Name	**mPERSONA crawler**	**Metadata Abstraction Crawler**
Domain	General	General
Working Environment	mPERSONA	A content management system.
Special Functions	Converting the webpages provided by content providers into metadata, in order to build a semantic tree.	Abstracting, indexing and storing metadata from fetched source files.
Technologies Utilized	Thesaurus for finding synonyms for the keywords that users enter.	Natural Language Parser for parsing, identifying and annotating name entities from fetched source files; 5W1H for metadata abstraction method; Dublin Core as metadata format.
Evaluation Methods	Not provided.	Not provided.
Evaluation Results	Not provided.	Not provided.
Comments/ Suggestions	Provide evaluation details.	Provide evaluation details.

B. *Compare and contrast of the RDF crawlers*

Table 2 displays the compare and contrast result among the three RDF crawlers. According to the statistical data, the RDF crawlers work in general domains (3/3), and they are usually encapsulated into a comprehensive system as the backbone (2/3). Due to the application of identical technology (3/3), their proposed functions are similar, which are metadata abstraction (3/3) and reasoning (2/3). None of the crawlers are empirically evaluated, which is the major betterment plan we suggest.

C. *Compare and contrast of the OAI-PMH-based crawlers*

Similar to the RDF crawlers, the OAI-PMH-based crawlers adopt identical technologies to improve their semantic abilities, which can be seen in Table 3. In light of the features of OAI-PMH, both of these crawlers are implemented in general domain, and general working environment. Both of them are used for XML-based metadata abstraction, and the only difference is the format of XML metadata. One of them utilizes the MPEG-21 and the other utilizes the CRATE. Similarly, there are no evaluation details provided by the authors, which can be considered as a weakness.

TABLE II. COMPARE AND CONTRAST OF THE RDF CRAWLERS

Name	**Ontocrawler (RDF Crawler)**	**RDF Crawler 2**	**RDF Crawler 3**
Domain	General	General	General
Working Environment	Ontobroker	CREAM	SEAL
Special Functions	Extracting, reasoning and generating RDF-annotated metadata.	Seeking references for the metadata created by CREAM.	Building the knowledge warehouse by generating RDF documents from the internet.
Technologies Utilized	A wrapper for abstracting metadata from the similarly structured HTML files; an annotation language for annotating specially structured HTML files; an inference engine for reasoning metadata.	Ontocrawler for seeking metadata references.	Ontocrawler for generating RDF documents from the internet.
Evaluation Methods	Not provided.	Not provided.	Not provided.
Evaluation Results	Not provided.	Not provided.	Not provided.
Comments/ Suggestions	Provide evaluation details.	Provide evaluation details.	Provide evaluation details.

TABLE III. COMPARE AND CONTRAST OF THE OAI-PMH-BASED CRAWLERS

Name	**OAI-PMH-based Crawler 1**	**OAI-PMH-based Crawler 2**
Domain	General	General
Working Environment	General	General
Special Functions	Abstracting XML-based metadata from fetched documents.	Abstracting XML-based metadata from fetched documents.
Technologies Utilized	OAI-PMH for abstracting XML-based metadata from fetched documents; MPEG-21 for formatting metadata.	OAI-PMH for abstracting XML-based metadata from fetched documents; CRATE for formatting metadata.
Evaluation Methods	Not provided.	Not provided.
Evaluation Results	Not provided.	Not provided.
Comments/ Suggestions	Provide evaluation details.	Provide evaluation details.

D. Compare and contrast of the focused crawlers

From Table 4, it is discovered that the focused crawlers are mostly in the specific domains (2/3), and encapsulated in more comprehensive systems. Due to the specialty of documents fetched, they need to convert the domain-specific document into more meaningful metadata. Various technologies are utilized for document classification and metadata abstraction. Whilst one crawler does not provide their evaluation details, from the existing evidence we still can observe its prime performance. It is suggested that the authors should reveal their evaluation details and compare with other crawlers to prove their crawler models.

TABLE IV. COMPARE AND CONTRAST OF THE FOCUSED CRAWLERS

Name	**Focused Crawler 1**	**Focused Crawler 2**	**Focused Crawler 3**
Domain	General	Legal	E-business
Working Environment	General	Vertical portal	A niche search engine.
Special Functions	Abstracting metadata from fetched webpages and linking them to domain-specific ontologies.	Collecting legal documents; abstracting metadata.	Parsing citations and abstracting metadata from downloaded documents.
Technologies Utilized	DocExtractor for information extraction; DocAnnotator for metadata generation; OntoClassifier and tf-dif algorithm for linking webpages to the mostly matched ontological classes.	NB and MSVM for document classification; Dublin Core schema and tf-idf for metadata abstraction.	CiteSeer for parsing citations and abstracting metadata from downloaded documents; HMM for similarity estimation; SVM for metadata abstraction.
Evaluation Methods	Comparing the accuracy of OntoClassifier with other three classification models.	Evaluating the classification accuracy values for NB and MSVM respectively.	Not provided.
Evaluation Results	Overall 80% in accuracy, which is superior than the other three models.	82.5% for NB, 85.1% for MSVM	Not provided.
Comments/ Suggestions	Provide experiment to test recall rate.	Compare with other crawlers.	Provide evaluation details.

E. Compare and contrast of the non-text crawlers

The last group includes two irrelevant crawlers. One is used for abstracting the metadata regarding media files in a knowledge portal for a media agent, and another is used for abstracting table metadata in digital libraries for a table search engine. Owing to the different service objects, their mechanisms are quite distinct. For the multimedia crawler, an online and an offline crawler are both adopted to collect media metadata according to user's preference; for the table crawler, a specific table object classifier and a table metadata indexing algorithm are utilized. In contrast to the fact that no evaluation details are provided by the authors of multimedia crawler, table crawler achieve a fine score in the precision and recall test, while the only defect is the number of experiment document is limited.

TABLE V. COMPARE AND CONTRAST OF THE NON-TEXT CRAWLERS

Name	**Multimedia Crawler**	**Table Crawler**
Domain	Multimedia	Digital library
Working Environment	A media agent	TableSeer
Special Functions	Abstracting and indexing multimedia metadata from collected media description documents.	Abstracting and indexing table metadata from fetched documents.
Technologies Utilized	An online crawler for collecting media information operated by users; an offline crawler for collecting media information according to users' predefined preference.	Doc Classifier for table objects filtering and classification; TTF-ITTF for indexing table metadata.
Evaluation Methods	Not provided.	Measuring precision and recall values for abstracting various table objects.
Evaluation Results	Not provided.	The overall precision and recall values all both more than 80%.
Comments/ Suggestions	Provide evaluation details.	The number of experimental documents is limited (200).

F. Conclusion of the compare and contrast of the crawlers

To draw an overall conclusion to this compare and contrast we simply integrate the comparison result of each cluster, as per our survey result, to the current situation of the metadata abstraction crawlers as follows:

First of all, most of the crawlers may be applicable to general domains as shown in this survey (10/14), and usually they are designed as a component of a larger system and located in the beginning stage in a system's workflow (11/14). There is no doubt that abstracting metadata in some form, such as XML (2/14), RDF (3/14), Dublin Core (2/14), MPEG-21 (1/14), CRATE (1/14), is the primary function and objective for the design of these crawlers (14/14). In addition, various technologies are utilized in the abstraction process, such as RDF crawler (3/14), OAI-PMH (2/14), and so forth. Moreover, some special functions are proposed in some crawlers, such as ontology generation (4/14), metadata reasoning (2/14), in order to enhance the capability of the crawlers. In the perspective of empirical experiment, only limited authors reveal their evaluation methods and results (4/14), which need to be improved in the future. However, the existing evidence indicates the crawlers' persuasive performance in general.

IV. CONCLUSION

In this paper, towards an emerging category of crawlers – metadata abstraction crawler, we make a comprehensive survey by means of selecting fourteen typical crawlers from this field. According to their features, we classify them into the five clusters – normal metadata abstraction crawlers, RDF crawlers, OAI-PMH-based crawlers, focused crawlers, and non-text crawlers. Then a compare and contrast is made for each crawler from the seven perspectives - domain, working environment, special functions, technologies utilized, evaluation methods and evaluation results, followed by our comments or suggestions. By analyzing the comparison results, the conclusion with regard to the current situation, features and future of the metadata abstraction crawlers is drawn.

ACKNOWLEDGMENT

We would like to express our gratitude to the assistance of DEBII.

REFERENCES

[1] A. A. Barfourosh, M. L. Anderson, H. R. M. Nezhad, and D. Perlis, "Information Retrieval on the World Wide Web and Active Logic: A Survey and Problem Definition," Department of Computer Science, University of Maryland, Maryland, 2002.

[2] J. Cho and H. Garcia-Molina, "Parallel Crawlers," in *WWW2002*, Honolulu, 2002, pp. 124-135.

[3] H. Davulcu, S. Vadrevu, and S. Nagarajan, "OntoMiner: bootstrapping ontologies from overlapping domain specific web sites," in *WWW2004*, New York, 2004, pp. 500-501.

[4] S. Decker, M. Erdmann, D. Fensel, and R. Studer, "Ontobroker: Ontology based access to distributed and semi-structured Information," in *Database Semantics: Semantic Issues in Multimedia Systems*, R. Meersman, Ed.: Kluwer Academic Publisher, 1999, pp. 351–369.

[5] E. Francesconi and G. Peruginelli, "Searching and retrieving legal literature through automated semantic indexing," in *ICAIL '07*, Standford, 2007, pp. 131-138.

[6] C. L. Giles, Y. Petinot, P. B. Teregowda, H. Han, S. Lawrence, A. Rangaswamy, and N. Pal, "eBizSearch: A niche search engine for e-business," in *SIGIR'03*, Toronto, 2003, pp. 213-214.

[7] S. Handschuh and S. Staab, "Authoring and annotation of web pages in CREAM," in *WWW2002*, Honolulu, 2002, pp. 462-473.

[8] S. Handschuh and S. Staab, "CREAM: CREAting Metadata for the Semantic Web," *Computer Networks*, vol. 42, pp. 579-598, 2003.

[9] S. Handschuh, S. Staab, and A. Maedche, "CREAM — Creating relational metadata with a component-based, ontology-driven annotation framework," in *K-CAP'01*, Victoria, 2001, pp. 76-83.

[10] J. Hendler, "Agents and the semantic web," *IEEE Intelligent System*, vol. 16, pp. 30-37, 2001.

[11] B. J. Jansen, T. Mullen, A. Spink, and J. Pedersen, "Automated fathering of web information: an in-depth examination of agents interacting with search engines," *ACM Transactions on Internet Technology*, vol. 6, pp. 442-464, 2006.

[12] D. Konopnicki and O. Shmueli, "Database-inspired search," in *the 31st VLDB Conference*, Trondheim, 2005, pp. 2-12.

[13] W. Liu, Z. Chen, F. Lin, R. Yang, M. Li, and H. Zhang, "Ubiquitous media agents for managing personal multimedia files," in *MM'01*, Ottawa, 2001, pp. 519-521.

[14] Y. Liu, K. Bai, P. Mitra, and C. L. Giles, "Automatic searching of tables in digital libraries," in *WWW 2007*, Banff, 2007, pp. 1135-6.

[15] Y. Liu, K. Bai, P. Mitra, and C. L. Giles, "TableSeer: automatic table metadata extraction and searching in digital libraries," in *JCDL'07*, Vancouver, 2007, pp. 91-100.

[16] M. L. Nelson, J. A. Smith, I. G. d. Campo, H. V. d. Sompel, and X. Liu, "Efficient, Automatic Web Resource Harvesting," in *WIDM'06*, Arlington, 2006, pp. 43-50.

[17] C. Panayiotou and G. Samaras, "mPERSONA: personalized portals for the wireless user: an agent approach," *Mobile Networks and Applications*, vol. 9, pp. 663–677, 2004.

[18] S. Roberson and D. Dicheva, "Semi-automatic ontology extraction to create draft topic maps," in *ACMSE'07*, Winston-Salem, 2007, pp. 100-105.

[19] K. Shimazu, T. Arisawa, and I. Saito, "Interdisciplinary contents management using 5W1H interface for metadata," in *the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI'06)*, 2006.

[20] J. A. Smith and M. L. Nelson, "Generating best-effort preservation metadata for web resources at time of dissemination," in *JCDL'07*, Vancouver, 2007, pp. 51-52.

[21] L. Stojanovic, N. Stojanovic, and R. Volz, "Migrating data-intensive web sites into the semantic web," in *SAC 2002*, Madrid, 2002, pp. 1100-1107.

[22] N. Stojanovic, A. Maedche, S. Staab, R. Studer, and Y. Sure, "SEAL — a framework for developing SEmantic PortALs," in *K-CAP'01*, Victoria, 2001, pp. 155-162.

[23] S.-Y. Yang, "An ontological website models-supported search agent for web services," *Expert Systems with Applications*, vol. In Press, Corrected Proof.