International Journal of Computing Academic Research (IJCAR)
ISSN 2305-9184 Volume 2, Number 4 (August 2013), pp. 147-154

http://www.meacse.org/ijcar

INTELLIGENT SURFACING MECHANISM FOR SEMANTIC DEEP WEB

Aarti Singh[1] and Shaily Sachdeva[2]

[1]Associate Prof., MMICT & BM, MMU, Mullana, Haryana, India
[2]Research Scholar MMICT & BM, MMU, Mullana, Haryana, India

Abstract
Semantic web is an effort to enhance current web with intelligence so that computer can process the information present on World Wide Web, interpret and connect it. Deep web consists of information in formats that are not understandable by the common search engines thus deep web is not indexed by the search engines. Two approaches of virtual integration and surfacing are presently been employed to extract information from deep web. Focus of semantic deep web is to address the problems associated with accessing the rich, structured back-end data through ontologies. Semantic deep web is a sub domain of present day semantic web, focusing on associating semantics, extracting and facilitating meaningful contents from this hidden treasure. Present approaches of deep web information extraction suffer from scalability problem, which becomes major bottleneck considering web based nature of contents. Present work aims to propose an agent based mechanism for surfacing deep web contents. Being web based this mechanism ensures automation of content extraction, thereby eliminating problem of scalability.

Keywords: Deep Web, Semantic Deep Web, Agent Technology, Surfacing Approach, Deep Web Information Extraction

I. Introduction

The term *Semantic Web* was coined by Tim Berners Lee in 2001. He defined Semantic Web as "a web of data that can be processed directly and indirectly by machines [1]." It extends the network of hyperlinked human-readable web pages by inserting machine-readable metadata about pages and how they are related to each other, enabling automated agents to access the Web more intelligently and perform tasks on behalf of users. Semantic web is an effort to enhance current web with intelligence so that computer can process the information presented on World Wide Web, interpret and connect it [2]. Parallely the term *Deep Web* (DW) was used. The term "deep Web" was coined by Mike Bergman in 2001. DW consists of files in formats that are not understandable by the common search engines thus deep web is not indexed by the search engines. Its contents include information in private databases that are not accessible over the internet. Two techniques are primarily used for accessing the DW contents i.e. virtual integration and surfacing [8] [9].

- Virtual Integration: In this technique users define their interests using keywords that can be as complex as needed, from simple terms to high-level structured queries, and as a response, the system retrieves information related to these keywords. This information is retrieved from many different sources, but it is presented uniformly to the users in a transparent way. This process is online, and therefore response time is an important issue. Information retrieval and information extraction are two complementary steps in the virtual integration process: the former retrieves all the relevant pages, and the latter extracts required information from these pages.

- Surfacing: It is an off-line process that intends to collect all pages behind a web form by submitting pre-computed queries, and not taking into account the user's specific requirements. In surfacing web forms are automatically searched with guessed input field values, and the resulting pages are indexed in search engine index like a normal web page. Surfacing [14] has two important drawbacks -

  (i) The information is duplicated locally at the search engine.
  (ii) The deep sites can be of any domain.

In both above techniques of exploring deep web, manual intervention is required for information extraction, which is tedious and time consuming job, especially considering the size of web and rate of content addition it seems practically infeasible to extract and index whole deep web.
DW contents are important since it contains lot of valuable information such as e-commerce data and accessibility of this part of the web may uncover useful user behavior and trends on the web. Presently it has been accessed by web crawlers through web services or web-form based interfaces.
Considering the importance of DW contents in web based scenario and scalability problem associated with present extraction techniques, present work aims to propose an agent based mechanism for extracting deep web contents. This mechanism improves traditional surfacing strategy and automates the process of surfacing.
Rest of the paper is organized as follows: Section 2 presents literature survey in domain of interest. Section 3 elaborates the proposed mechanism. Section 4 concludes the work and provides future research directions.
Next section provides the review of relevant literature.

II. Literature Survey

Berners T. et. al. in [1] proposed concept of semantic web to weave a web that not only links documents to each other but also recognizes the meaning of information in those documents. However such web would require both human readable & machine readable information. Fensel D. et. al. [2] described applications of semantic web such as knowledge representation and highlighted that SW requires various processes to provide efficient interaction. Singh A. et. al. [3] proposed a design structure for development of ontological database. However this paper provides the flexible model for ontological database but did not describe the performance of model. Benjamins V. et. al. in [4] highlighted the challenges concerning the availability of content, ontology availability, development and evolution, scalability, multilinguality, visualization to reduce information overload, and stability of SW languages. Singh A. et. al. [5] introduced the positive role of ontology in crawling strategies and advantages of ontology based crawling algorithms. Barbosa L. et. al. in [6] proposed a new crawling

strategy that combines the page and link classifiers, however it lacked automated techniques for evaluating quality of forms harvested by the form crawler. Chun S. et. al. [7] presented the shortcomings of current search engines and highlighted the requirements of a Deep Web Service (DWS) search engine. They proposed the semantic metadata and annotation of DWS. The limitation of this work is that DWS required semantic specifications and parameters or non functional properties need to be extracted. Madhavan et. al. [8] discussed the DW is biggest source of structured data on the Web and hence accessing its contents has been a long standing challenge in the data management community and described the *Virtual Integration* and *Surfacing* approaches of Deep Web extraction. Ntoulas A. et. al. [11] described a hidden web crawler that can automatically query a hidden website and download pages from it. But the limitation is that how can the crawler discover the query interfaces. Lerman K. et. al. in [12] proposed the automated data extraction from list and tables by using algorithms. One limitation of this approach is that it requires several pages to be analyzed before data can be extracted from a single list. Madhavan J. et. al. in [14] presented fusion tables for facilitating a much larger class of users to manage their data and integrating their online activities. Lu J. et. al. [15] proposed a crawling method for the queries with low overlapping rate, along with a method for query submission and data extraction. Alba A. et. al. in [16] highlighted the surfacing as query and loading problem. Geller J. et. al. in [17] proposed the term semantic deep web as extension of semantic web focusing towards extraction of deep web, by fusing aspects of the semantic web with the use of ontology aware browsers to extract information from the deep web. Nwana H. [18] presented review of software agents and their various aspects such as types, goals, benefits and related challenges.

A critical look at the above literature highlights the fact that very few researchers have made an attempt towards the automatic extraction of information from deep web, especially automation of surfacing technique. This provided us the motivation to focus on this dimension. This research makes an attempt to propose general design structure for an agent based mechanism for automated surfacing of deep web.
Next section provides the details of the proposed framework

III. Proposed Framework

Agent based Surfacing Mechanism for Semantic Deep Web (ASMSDW) primarily comprises of six agents namely User Interface Agent (UIA), Searcher Agent (SA), Indexer Agent (IA), Query Formulation Agent (QFA), Surfacing Agent (Surf_A) and Crawler Agent (CA).The high level view of the framework is depicted in Figure 1.

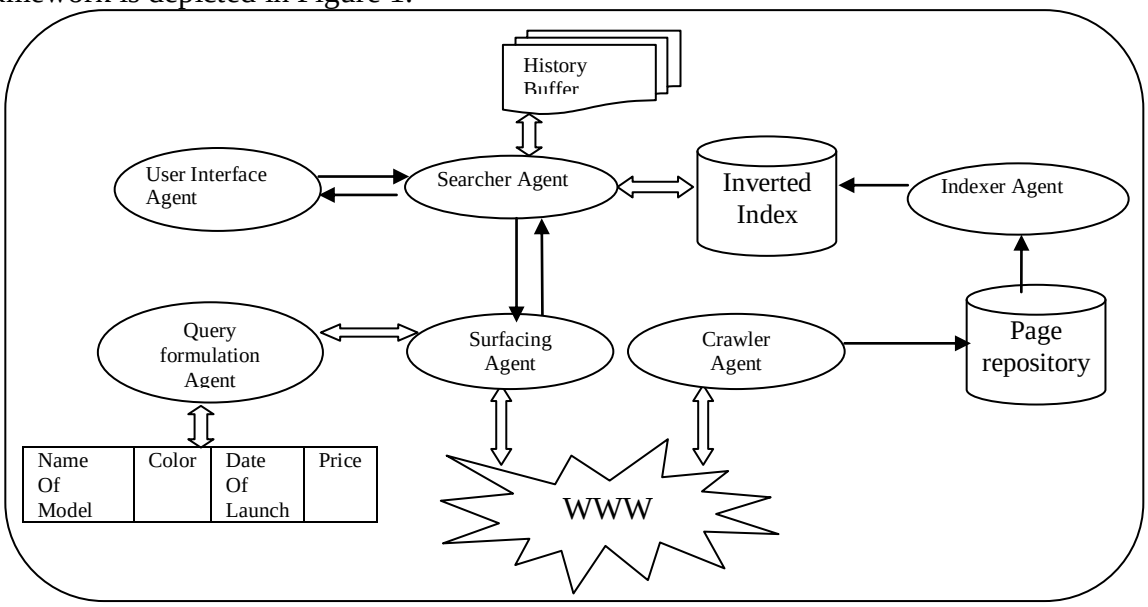

Fig. 1 High Level View of ASMSDW

Role of each agent is illustrated below:

- **User Interface Agent (UIA):** It accepts the initial domain for which surfacing is desired. It passes the domain name to searcher agent for searching relevant forms from the inverted index. If relevant forms are found by searcher agent it returns them to UIA for user approval.
- **Searcher Agent (SA):** Searcher Agent is responsible for searching forms related to user request in the inverted index. On getting relevant forms from there it provides them to user for approval. It is also supported with a history buffer, which stores the recently returned forms for future usage. Considering the large size of inverted index, HB can significantly help reducing the search time, in case of frequent usage. Whenever SA receives a new request, it initially explores HB first, in case results are found in buffer it returns them to UIA else it explores inverted index. In case no relevant form is found in inverted index, SA extracts URL containing the desired domain name for further surfacing.

- **Indexer Agent (IA):** Indexer Agent is responsible for creating inverted index of the data fetched by the crawler. Inverted index contain listing of URLs on the basis of frequency of word occurrences in them. Inverted Indexes make search process efficient.

- **Surfacing Agent (Surf_A):** Searcher Agent passes the URLs containing domain name to Surf_A. Surf_A then sends database access request to administrators of URLs and waits for permission from the other end. Since our point of focus are hidden contents, specifically databases containing information of common interest, but such information can't be accessed from hosting servers unless permission is granted by the administrator. Once permission is granted and access rights are received, Surf_A invokes QFA for formulating queries on that database.

- **Query Formulation Agent (QFA):** QFA is responsible for formulating the queries over the hidden information contained at URLs passed by Surf_A. The result of queries are received in the form of separate forms and passed back to Surf_A for further processing.

- **Crawler Agent (CA):** CA is responsible for fetching the web pages from WWW and places it in page repository. This repository is the basis of inverted index being created by IA.

Next section discusses the working of proposed model of ASMSDW.

3.1 Working of ASMSDW

Whenever user enters a domain of interest in ASMSDW it is received by UIA. UIA acts as an interface between user and the rest of framework. UIA passes this domain name to SA, which initially looks into HB to find relevant forms, in case a match is found resulting forms are being returned to the user. If match is not found in the HB, SA explores inverted index to find relevant forms, in case forms are

available they are returned to UIA otherwise URLs containing domain name are extracted for surfacing process. These URLs are sent to Surf_A in a list. Surf_A sends request for surfacing permission to administrator of the URL server. Since here the concern is to list the hidden information in the search engine indexes to make it visible, but some information is of commercial importance and is deliberately kept hidden, thus unauthorized listing of hidden information is not ethical. The list of URLs containing domain of interest will be quite large most of the times however since permission is required for surfacing which would not be granted by all administrators , thus list of final URLs for surfacing will get reduced. Surf_A will start surfacing only after permission for the same is received. Then Surf_A will call QFA to generate queries on the databases for extracting forms of information. For this purpose, QFA will analyze table_defintion and generate queries by combining different attributes being stored in the database. Different outputs received in response to queries will be treated as different forms to be listed in inverted index. QFA will pass resulting forms to Surf_A and will start processing next URL. This process will be repeated till the list of URLs is exhausted. Surf_A on receiving forms from QFA will pass them to SA. SA will store the resulting forms in HB for future usage as well as index them in inverted index. SA will pass resulting forms to user also as and when required.

Next subsection illustrates flow of instructions in the proposed framework.

3.2 Flow Diagram of ASMSDW

Flow of instructions in ASMSDW is being illustrated in Fig. 2 given below. Step by step details are as follows:

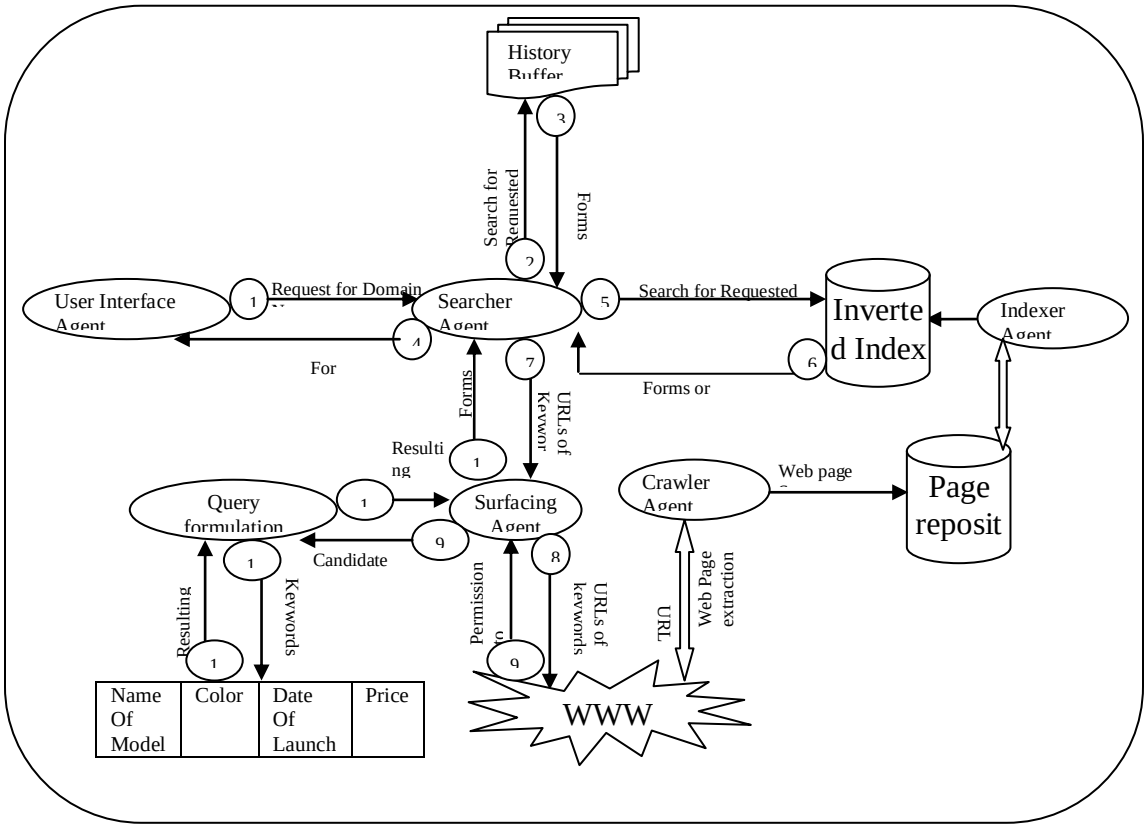

Fig. 2 Flow Diagram of ASMSDW

1. UIA accepts domain name from user and passes it to SA.
2. SA searches the requested domain name in HB initially.
3. In case if the same domain had been surfaced recently then related forms exist in HB, SA extracts those forms from HB.
4. SA passes relevant forms to UIA.
5. If no matching form is found in HB, SA searches the requested domain in inverted index. key terms information in inverted index that contains the pre-computed forms which are listed by IA.
6. SA explores inverted index, either it finds relevant forms listed there or it extracts the URLs where requested keywords appear. Those URLs are candidates for further surfacing procedure.
7. If relevant forms are found from inverted index, they are returned to UIA through step 4, else the candidate URLs are passed to Surf_A for surfacing of the domain.
8. Surf_A sends request for surfacing hidden contents of candidate URLs to their administrators and waits for permission. Since hidden contents may include commercially valuable contents like databases and can't be accessed without permission of the owner.
9. As and when permission is obtained from a candidate URL, Surf_A invokes QFA and passes candidate URL to it.
10. QFA access the databases from the URLs, analyzes the attributes recorded in it. It stores attributes in an array. It picks first attribute, makes its combinations with remaining n-1 attributes in one to one manner and fires SQL queries on the database, the resulting output is saved as separate form to be listed in inverted index. QFA repeats the process by picking up attribute one and two together and makes its combination with rest n-2 attributes. This process is repeated till the list of attributes is exhausted.
11. All forms generated in step 10 are returned to Surf_A for their listing.
12. Surf_A passes the retrieved forms to SA for their listing in inverted index. SA also keeps a copy of recent forms in HB for future usage.SA also provides the desired results to user through UIA.

Algorithms for various agents which are used in ASMSDW are shown in Figures 3(a), 3(b), 3(c) and 3(d) respectively:

```
User_Interface_Agent ()
Input: domain name;
Output: Relevant forms;
{
 Read input domain name from the user;
Pass domain name to SA();
Accept relevant forms from SA and pass it
to user.
}
```

Figure 3(a) User_Inteface_Agent( )

```
Searcher_Agent ()
Input: domain name;
Output: Relevant forms;URLs;
{
 Search HB for domain name;
if relevant result forms found, return();
else
{search inverted index;
if relevant forms found return();
else extract relevant URLs and pass URLs
to Surf_A();
}
}
```

Figure 3(b) Searcher_Agent( )

```
Surfacing_Agent ()
Input: Candidate URLs, forms;
Output: Relevant forms;
{
 If (candidate URLs)
{Store candidate URLs in a list;
    Send surfacing request surf_req to URL server;
   if (surf_req(URL_i==true)
    { Get permission information from server;
       send <URLi,Permission_info>→QFA();
    }
   else
      sleep();
}
else
{receive forms from Query_Formulation_Agent();
   return(forms)→Surfacing_Agent();
 }
```

Figure 3(c) Surfacing_Agent( )

```
Query_Formulation_Agent ()
Input: Candidate URL;
Output: Relevant forms;
{

 Read database from URLs;
copy attributes from database in an array;
for i=1 to n-1
{
read attribute_i;
  use <attribute_i, attribute_i+1> to select data
from database;
save output as a separate form;
return (form)→Surf_A();
}
}
```

Figure 3(d) Query_Formulation_Agent( )

IV. Conclusion

Deep web contains lots of precious information, which might be of interest to many users. However manual extraction of deep web contents is practically infeasible. This work has presented a mechanism for automating surfacing of semantic deep web using intelligent agents. Intelligent agents being autonomous components can be a promising solution in surfacing of deep web. However this mechanism has focused more on surfacing of hidden databases behind web pages, although deep web comprises of more types of data also, which might require different strategy for surfacing. This point is left as part of future work. Presently implementation of proposed mechanism is under progress.

V. References

[1] Berners T., Handler J. and Lassila O. "The Semantic Web", Published in South African Journal of Information Management, Vol.6 (1) March 2004.

[2] Fensel D., Bussler C., Ding Y., Klein M., Korotkiy M. and Siebes R. "Semantic Web Application Areas", Published in Semantic Web Technology, MIT Press, Boston, 2002.

[3] Singh A., Juneja D. and Sharma A.K. "General Design Structure of Ontological Databases in Semnatic Web", Published in International Journal of Engineering Science and Technology Vol. 2(5), 2010, 1227- 1232.

[4] Benjamins V., Contreras J., Corcho O. and Perez A. "Six Challenges for the Semantic Web", Published in Proceedings of International Semantic Web Conference (ISWC2002), Sardinia, Italia, 2002.

[5] Singh A., Juneja D. and Sharma A.K. "Design of Ontology-driven Agent Based Focused Crawlers", Published in 3rd international conference on intelligent systems & networks (IISN-2009), organized by Institute of Science and Technology, Klawad, 14-16 Feb 2009,pp. 178-181, Volume 2, No.8: Jan25,2010.

[6] Barbosa L. and Freire J. "Searching for Hidden-Web Databases", Published in Eighth International Workshop on the Web and Databases (WebDB 2005), June 16-17, 2005, Baltimore, Maryland.

[7] Chun S. and Warner J. "Semantic Annotation and Search for Deep Web Services", Published in 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, pp. 389-395, 2008.

[8] Madhavan J., Afanasiev L., Antova L. and Halevy A. "Deep Web Present and Future", Published in CIDR perspective, 2009.

[9] Khare R., An Y. and Song Y. "Understanding Deep Web Search Interfaces", Published in SIGMOD Record, March 2010 (Vol. 39, No. 1).

[10] Barbosa L. and Freire J. "Siphoning Hidden-Web Data Through Keyword-Based Interfaces", Published in SBBD, 2004.

[11] Ntoulas A., Zerfos P. and Cho J. "Downloading Textual Hidden Web Content Through Keyword Queries", Published in JCDL'05, June 7-11, 2005, Denver, Colorado, USA.

[12] Lerman K., Knoblock C. and Minton S. "Automatic Data Extraction from Lists and Tables in Web Sources", Published in IJCAI-2001 Workshop on Adaptive Text Extraction and Mining, August 2001.

[13] Hernandez I. "Intelligent Web Navigation", Published in Taller de Trabajo Zo-co'09/JISBD, 2009.

[14] Madhavan J., Ko D., Kot L., Ganapathy V., Rasmussen A. and Halevy A. "Google's Deep Web Crawl", Published in PVLDB, 1(2):1241–1252, August 23-28, 2008, Auckland, New Zealand.

[15] Lu J., Wang Y., Liang J. and Chen J. "An Approach to Deep Web Crawling by Sampling", Published in Proc. of Web Intelligence. (2008),pp. 718–724.

[16] Alba A., Bhagwan V. and Grandison T. "Accessing Deep Web", Published in OOPSLA Companion '08: Companion to the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications, ACM, New York, NY, USA, pp 815-818, DOI.

[17] Geller J., Jung Y. and Chun S. "Toward the Semantic Deep Web", Published in Computer, 41(9), pp. 95–97, 2008.

[18] Nwana H. "Software Agents : An Overview", Published in Intelligent Systems Research Advanced Applications & Technology Department BT Laboratories, Martlesham Heath Ipswich, Suffolk, IP5 7RE, U.K. Vol. 11, No 3, pp. 205-244, October/November 1996.

[19] Griss M. "Software Agents as Next Generation Software Components", Published in Component-Based Software Engineering: Putting the Pieces together, eds. G.T. Heineman and W.T. Councill (Addison-Wesley, Boston, 2001), pp. 641–657.

[20] Blazely M., Coltheart M. and Casey B. "Semantic Impairment With And Without Surface Dyslexia", Publishd in cognitive neuropsychological Australia, 2005, 22(6),pp. 695-717.