Automatic construction of vertical search tools for the Deep Web using Web Query Interfaces (ICW)

Dhanshree.D.Thorbole[1], Prof. Subhash.V.Pingale[2]
Dept of CSE , SKN Sinhgad College of Engineering Korti,Pandharpur,Maharashtra
[1]thorboledhanshree1@gmail.com , [2]subhash.pingale@sknscoe.ac.in

Abstract — The most coveted commodity of the information age is indeed information. Information has become a basic need after food, shelter, and clothing. Due to technological advancements, a large amount of information is available on the Web, which has become a complex entity containing information from a variety of sources. Information is found using search engines. A searcher has access to a large amount of information, but it still far from the huge treasury of information lying beneath the Web, a vast store of information beyond the reach of conventional search engines: the "Deep Web" or "Invisible Web. With the constant increase in the volume of information available on the Web, it is more difficult to find the specific information related to a given domain. Users are facing the problem of information overload, in which a query about a specialized subject (local information, e-commerce: hotels, airlines, car rental; science: biology, mathematics, medicine, etc.) on a web search engine, it returns a lot of web pages or results that in most of the cases are outside the domain of interest. This is one reason why the vertical search tools have become a necessity for users that seek specific-domain information from different databases available in the Web through input sources called Web Query Interfaces (ICWs). This paper describes an approach for automatic integration of ICWs, a crucial task to construct vertical search tools. The proposed methodology is validated by realizing a vertical search prototype called VSearch that allows users to transparently query multiple web databases in a specific-domain through a unified ICW. The proposed approach for automatic ICWs integration is based on: i) a hierarchical model called AEV for modelling the visual content of ICW; ii) semantic clustering for the identification of relationships between fields in ICWs; and iii) a field homogenization and unification process of AEV schemes for the construction of a unified ICW. The VSearch prototype was implemented and evaluated. The experimental results demonstrate the high precision in the integration phase and an effective methodology to create a functional vertical search tool.

Keywords— Vertical Search Tool, Web Databases, Web Query Interfaces, Automatic Integration, VSearch.

I.INTRODUCTION

A Deep Web refers to the information that lies in the databases available on the Web [2]. The information available in these web databases is obtained by users through forms inquiries Special HTML called Web Query Interfaces (ICWs). An ICW is an HTML form that allows users send structured queries to a database web and receive a dynamically generated web page as response to the query issued [3]. An ICW consists of multiple fields, text labels associated with each field, Digital images and multimedia components. Fields most common are: text boxes (text-input box), lists of selection (selection-list), buttons (radio-button) and box selection (check-box) [3] .

Currently, there are multiple web databases Specialized available (related to the sale of houses, medical information, electronic commerce, science, etc.). However, the process of consulting several web databases One by one it becomes unfeasible for a user. In a situation like this, it is necessary to use a tool vertical search that allows integrating different ICWs of a specific domain in order to facilitate the process of User search. The integration process Automatic ICWs consist of forming a unified ICW that serves to receive web queries provided by the user distributing said query to each ICW so transparent to the user, integrating each individual result and presenting to the user the results obtained in a way Useful and understandable.

For a domain of interest, there are several ICWs in the Web with variable coverage and query capabilities. Without However, there is no automatic mechanism that integrates ICWs from the same domain and produces a unified ICW that facilitates consultation and increases search coverage.

A user with a specific need, for example, Buying a book can access a popular website like amazon.com to check book information and Possibly make the purchase. However, it would be more useful. for the user to transparently consult several bases of web data and use the information collected from different sources, for example, the price and delivery time, for Make a better purchase decision. Considering that There are several ICWs available on the Web for a domain given (for example for the

domain of hotel rental or purchase of airline tickets), would be tedious and would require more time for the user to find all those ICWs, fill out and send the query in each one, and integrate the results to Obtain the information of your interest manually.

Under this context, in this work, they are proposed as main contributions:

1. A new approach to the integration of ICWs based in a hierarchical representation model that uses the visual information of the ICWs given by the engine HTML rendering (render engine). This information It is presented in a render tree, when which apply various processing algorithms to discard irrelevant information (format or style), identify fields and their labels, group and delimit groups of fields. The approach of proposed integration uses the hierarchical model of each ICW to automatically produce an ICW unified, using field clustering, homogenization and unification of hierarchical schemes of ICWs.
2. A methodology for the automatic construction of vertical search tools that in addition to ICWs integration process, includes the processes of identification and classification of ICWs as previous components to build a unified ICW.
3. The experimental results of the prototype of a vertical search tool called VSearch that implements and validates the proposed methodology for the Construction of this type of tools.

II. RELATED WORK

Building a unified ICW for a domain specific, through identification, classification and integration of individual ICWs. DeepPeep [1] is a search engine specialized in ICWs, developed by the University of Utah to help discover entry points to the Deep Web, including databases and web services. However, DeepPeep does not consider the process of integrating ICWs, only covers the identification and classification processes of ICWs according to the domains registered by the engine of search. Also, not all web pages retrieved contain ICWs, or the recovered ICWs do not belong to the selected domain.

The project called MetaQuerier is a job developed at the University which focuses on the discovery and consultation of web databases. MetaQuerier is divided into two sub-projects: MetaExplorer and MetaIntegrator.

On the one hand, MetaExplorer focuses on discovery, modelling, and structuring of web databases for Build a search repository. On the other hand, MetaIntegrator is used to integrate corresponding ICWs to a specific domain. The internal content of the ICWs is modelled using a 2P grammar (pattern-precedence), the which is inherently ambiguous and can generate false mapping between ICW fields, which makes MetaIntegrator an incomplete project.

The work developed by Kabish [4] considers the ICWs classification and integration processes for the construction of a single ICW. However, said work it does not solve the mapping problems between the fields, which they can cause false mappings.

The integration process proposed work addresses the problem of semantic relations between fields. However it is necessary to address the problem to improve the process of ICW integration achieving better correspondence semantically between fields of different domain ICWs specific.

One of the objectives of the work presented in this article is provide these relationships and reflect them in the ICW unified.

III. METHODOLOGY FOR THE CONSTRUCTION OF VERTICAL SEARCH TOOL

The methodology proposed in this work for the automatic construction of vertical search tools It is shown graphically in Fig. 1. It consists of three stages Main: LOAD, EXECUTION and CONSULTATION. The first stage includes the construction of a repository of pages web using a web crawler .Initially the web crawler is initialize with a set of seed URLs. From this one small set, the web crawler retrieves new pages web and build a local repository stored as a XML file containing only those page URLs web with HTML forms (possible ICWs) and discards those URLs related to static information, such as documents (files with extension .PDF, .DOC, etc.) or images (files with extensions .JPG, .GIF, etc.) The second stage of EXECUTION consists of the three modules necessary for the construction of the tool vertical search:

- an automatic detector of ICWs,
- an automatic classifier of ICWs
- an automatic integrator of ICWs.

From the URL repository built in the stage of LOAD, the EXECUTION stage is responsible for retrieve, analyze and manipulate web pages to discover and extract ICWs through the automatic ICWs detector. A Once ICWs are identified independently of their domain, the automatic classifier of ICWs determines and select only those ICWs that belong to the domain selected of interest. In this work the visual content of each ICW is represented by a hierarchical modeller that exploits the information provided by the tree rendering of each ICW. Each ICW is modelled as a visual tree scheme (AEV). The set of AEVs is sent to the automatic integrator for the construction of a scheme unified that allows to create the HTML form of the ICW unified. Under the proposed hierarchical model they are preserved the father-son relationships present in the schemes individual hierarchical of each ICW.

Finally in the third stage of CONSULTATION, an interface User graph is responsible for invoking and deploying the Unified ICW built from the automatic integrator of ICWs This unified ICW allows users to send transparently queries to different ICWs, as well as retrieve and integrate the information returned from the Different web databases. Once a user sends a query through the unified ICW, internally this communicates with a mediator (software module that Supports integrated views across multiple sources and creates an application layer), which has the function of generating a individual connection with each ICW and collect the results obtained by them.

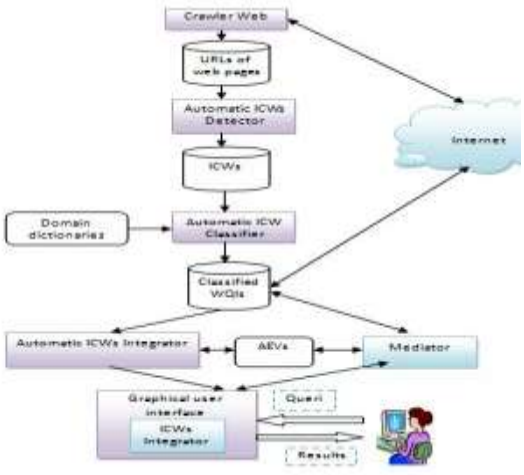

Figure 1. Graphical representation of the proposed methodology to build vertical search tools.

The results obtained by the mediator are encapsulated and sent as a final response to the user interface. He mediator has a direct interaction with each ICW individual. This provides access to data from different sources using a common data model and a language of common query The mediator receives the query from the Unified ICW transforms and transfers the query into one that be understandable by the source of information (original ICW) in an appropriate format, for example <source, action, attributes = {a1 = v1, a2 = v2,…, an = vn}>, where the source corresponds to the URL of the individual ICW, the action specific where to send the formatted data when a query is sent and the attributes contain the values of domain captured in the query.

The results obtained of the web queries sent to the web databases are sorted and presented according to the metric of Google's Page Rank reputation of the web pages that contain the individual ICWs consulted. The consultation sent by the user is a process that occurs online. Although, VSearch can be implemented on a personal PC, VSearch design allows its implementation in several PCs connected by a network in a cluster.

IV. DETECTOR AND AUTOMATIC CLASSIFIER OF ICWS

This work uses the pre-consultation approach proposed in [20] for the detection of ICWs on web pages during the EXECUTION stage. However, the proposed strategy in [20] it was adapted to provide the interfaces of enters and output needed to be integrated into the processing flow in the EXECUTION stage (Fig. 1). The entry to the identifier Automatic ICWs is an L list of n URLs with possible ICWs to be identified and the output is a LICW list of HTML forms classified as ICWs.

On the other hand, the automatic classifier of ICWs (Fig. 1) It has the function of determining the domain of each of the ICWs given. This work implements as a classifier ICWs automatic strategy reported in [19], incorporating the corresponding input interfaces and output so that the module can be included properly in the EXECUTION stage (Fig. 1). The input to the classifier is the resulting list of the detector ICWs.

V. ICWS AUTOMATIC INTEGRATOR

The integrator aims, from a set of ICWs of the same domain, with a variety of coverage and query capabilities, build a unified ICW that preserve the ancestor-successor relationships of the components individual ICWs. At the same time, the integrator of ICWs maintain grouping restrictions between fields as much as possible. The construction of the ICW unified for a given domain, it involves the following tasks:

1. Extraction and modelling of ICWs
2. Calculation of semantic relations between fields of different ICWs in the same domain
3. Construction of the unified ICW
4. Sending a global query through the ICW unified to each individual ICW

In this work the process of integrating ICWs is divided into four phases: a)hierarchical modelling of ICWs; b) clustering of ICW fields; c) homogenization of fields and d) unification of homogeneous schemes.

A. ICWS HIERARCHICAL MODELING

Definition 1. Modelling of ICWs: is the representation logic of the internal structure of an ICW with the objective of find queries involved in components logically related, known as segments.

Definition 2. Rendering engine: component typically embedded in web browsers, email clients, e-book readers or other applications that require viewing (and editing) of web content.

Definition 3. Rendering tree: hierarchical structure created by a rendering engine and confirmed by the description of marked content (HTML, XML, etc.) e format information (CSS, XSL, etc.) [12].

The hierarchical model of an ICW allows to have a view simplified, abstract and easily understandable of your content, retrieving and displaying the components of the ICW in an orderly and independent way. This model represents an ICW as a hierarchically structure organized, where its components maintain an order space. That is, the fields that are related semantically they are usually grouped together in a ICW.

Although the rendering tree of an ICW used by the hierarchical modeller is unlabeled, complex and disorganized, it provides the necessary information that allows the construction of the understandable visual model of an ICW. To achieve this, initially a stage of pre-processing to the rendering tree, removing nodes with unnecessary metadata, such as information from style (font, font size, font color, etc.), format information (td, div, br), and other information additional or multimedia information that is not of interest to the modelling process. What is of interest in the tree Rendering is structural and geometric information (width, height and position of the components) from the view of the ICW. The geometric information allows to exploit the spatial relationships (up, down, left, right, contain, overlap and disjoint) that exist between tree components. These spatial

relationships are used to determine the relationship between each pair of nodes in the tree rendering, for example, the membership relationship intersection (two brother nodes with the same father and the same depth); disjoint (if two nodes in the tree do not share the same common father), etc.

B. ICWS FIELD CLUSTERING

Once AEV schemes are built, the second phase in the process of automatic integration of ICWs it consists of identifying semantic relationships between fields or leaf nodes in AEV schemes and groups of fields or nodes internal according to their similarity. For the clustering of fields a field extraction algorithm was designed that recursively go through AEV schemes and identify and stores information of the visited nodes (leaf nodes e internal) within a vector of nodes V. This vector stores for each node the following information: node identifier, node number, node label, node name (if any), node type (leaf node, node compound, internal node), parent node (if any) and the list of child nodes T = {n1, n2,… nm} (if they exist). So, a pre-processing to each text tag associated with the nodes stored in V. That processing consists of the elimination of empty words (stop-words), normalization of labels applying a stemming process (stemming) and getting synonyms concepts from tags using Word Net

The next step in field clustering is the calculation of simple and complex semantic relationships between nodes of the vector V. In the case of simple semantic relations 1: 1 between the fields or leaf nodes, these are grouped according to its semantic similarity using a clustering algorithm hierarchical agglomerative. The semantic similarity between two fields are calculated as the sum of the linguistic similarity plus domain similarity [20]. Linguistic similarity measures the similarity between field labels and their names, as well as a combination between labels and Names. On the other hand, the domain similarity determines whether The fields have the same type and domain values. He Algorithm 1 calculates the semantic similarity between two fields. The algorithm receives two fields as input and calculates its linguistic similarity as the sum of the similarity between field labels, field names and the combination of They using the cosine function.

Once the linguistic similarity between two is calculated fields, the domain similarity is determined as the sum of the similarity of the type TypeSim and the similarity of the values domain valueSim. The types considered are: time, money, calendar month, numerical and string.

The Dice function [4] was used to measure the percentage of the overlapping range of the values of domain. This function is based on the coefficient Say [3], which considers the number of characteristics of two values of domains that have affinity, multiplied by two, over the Total number of features.

When the clustering process ends, each cluster is tagging with the tag that has the highest frequency of occurrence of all field labels in cluster ci. The result of the field clustering stage is a partition of V, such that only similar fields are in the same cluster and have a representative tag.

C. HOMOGENIZATION OF FIELDS

The homogenization of fields in the automatic integrator of ICWs assigns the same identifier to the nodes or fields that appear in different AEV schemes that have the same meaning. The homogenization of fields makes the process of unification of AEV schemes is easier.

D. UNIFICATION OF HOMOGENEAN AEVS

The last phase of the ICWs automatic integrator corresponding to the unification of homogeneous AEVs . This phase involves several operations on the set of AEVs in its matrix representation, preserving as much as possible, ancestor-successor relationships in each AEV individual within the unified AEV. The unification phase Take homogeneous AEVs as input and transform these schemes in their respective matrix representation for easily manipulate the nodes in each scheme and apply operations on them. As in [17], AEVs are transform into your constraint matrix representation and a set of transformations are applied for construction of a unified average AEV, which represents the center of the data distribution However, unlike [17], the hierarchical schemes AEVs to be integrated are constructed of automatically using the hierarchical modeller proposed in this work.

VI EXPERIMENTAL EVALUATION

A prototype of a search tool was built vertical called VSearch for the Purchase domain of Books, applying the methodology described in the sections previous. Experiments were conducted with the objective main to demonstrate that the proposed approach allows build, from a set of seed URLs and a specified domain of interest, a unified high ICW quality without human intervention, with high accuracy in inter-field mapping for the integration process of ICWs, which allows and facilitates the user to consult various Web databases related to the domain of interest.

From a set of 1025 web pages (URLs) built by a web crawler, six subsets were formed of different sizes (50, 100, 200, 350, 700, and 1025 URLs), selecting URLs randomly. Each one of these subsets was used to validate and evaluate the flow complete for the creation of the corresponding ICW unified in VSearch. The VSearch evaluation consists of measure your performance (runtime) and accuracy (to build the unified ICW in the integration process).

A. IMPLEMENTATION OF VSEARCH

VSearch was written in the Java programming language and tested on a computer with an Intel Core i3 processor 2.27 GHz, 4 G RAM under Windows. VSearch uses a crawler generic web to build the repository of web pages in the LOAD stage, with possible ICWs in different domains Hierarchical modeling of ICWs uses the Wolf Cobra Toolkit to build the rendering trees of the ICWs to be considered in the unified ICW.

B. VSEARCH EVALUATION CRITERIA

The quality of a global conceptual scheme can be measured by three qualitative criteria proposed in [2]: accuracy, integrity and efficiency These three criteria were

taken as basis for the evaluation of the AEV schemes of the ICWs Unified created by VSearch.

Accuracy: For a given ICW, its accuracy is defined as the percentage of mapped local fields correctly between all local fields.

Integrity: Measures how well the query capabilities on each ICW through the ICW
unified.

Efficiency: Evaluate execution time, effort or cost in the construction of the unified ICW deployed on the VSearch GUI. The cost of the consultation It is considered as the time for interpretation and execution of the query.

C. VSEARCH PERFORMANCE

VSearch performance was evaluated by the time of execution and response of its components. Just like him ICWs detector, ICWs classifier, e modeller ICW integrator. The runtime includes the time elapsed since the URL repository of web pages are given to the ICWs detector until the integrator of ICWs builds the unified ICW. Response time includes the time elapsed since the user provides a specific query through the ICW unified until VSearch presents the user with the data resulting from the execution of the query in the different web databases.

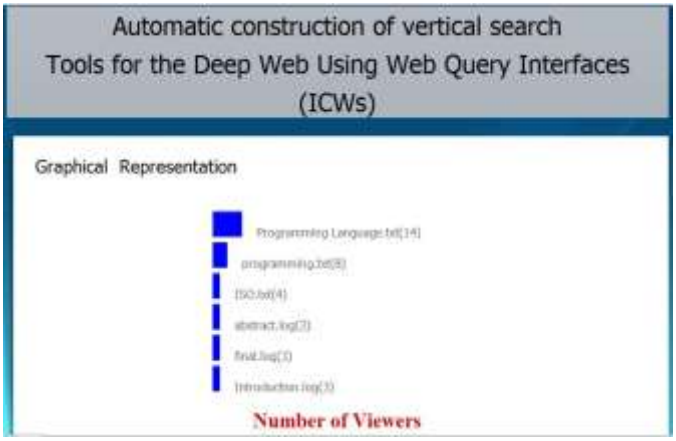

Figure 2:Graphical Representation of VSearch Performance that is number of viewers

Thus, from the results described, the feasibility, efficiency and precision of a prototype tool vertical search that is built using the methodology proposed in this work. This methodology shows the number of viewers.

VII CONCLUSIONS

Serious information seekers can no longer avoid the importance or quality of deep Web information. But deep Web information is only a component of total information available. Searching must evolve to encompass the complete Web.

Directed query technology is the only means to integrate deep and surface Web information. The information retrieval answer has to involve both "mega" searching of appropriate deep Web sites and "meta" searching the surface Web search engines to overcome their coverage problem. The Client-side tools are not universally acceptable because of the need to download the tool and issue effective queries to it. Pre-assembled storehouses for selected content are also possible, but will not be satisfactory for all information requests and needs. Specific vertical market services are already evolving to partially address these challenges. The explosive growth of the information contained in databases demand having search tools vertical that from a unified ICW allows users check information in a specific domain of interest, more simply and integrate results of Different sources.

However, the automatic construction of an ICW unified for a given domain involves several challenges, such as those mentioned in this work. Although there is a reduced number of proposals for construction Automatic of a unified ICW.

The experiments conducted showed that the ICW unified built by VSearch without human intervention It has a high accuracy of field mapping in the process of integration.

REFERENCES

[1]. Barbosa, L., Nguyen, H., Nguyen, T., Pinnamaneni, R., Freire, J.: *Creating and exploring web form repositories.* In: Proceedings of the 2010 international conference on Management of data, SIGMOD '10, pp. 1175–1178. ACM, New York, NY, USA (2010).

[2]. He, B., Patel, M., Zhang, Z., and Chang, K., Accessing the Deep Web, Commun. ACM, vol. 50, pp. 94--101, New York, NY, USA, (2007).

[3]. Marin-Castro, H.M., Sosa-Sosa, V.J., Lopez-Arevalo, I., Escalante- Baldera, H.J.: *Automatic classification of web databases using domaindictionaries*. In: Proceedings of the 9th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM'13, pp. 340–351.Springer-Verlag,Berlin,Heidelberg(2013).

[4]. Kabisch, T.: *Extraction and integration of web query interfaces*. Ph.D. thesis (2011).

[5]. Dragut, E., Wu, W., Sistla, P., Yu, C., Meng,W.: *Merging source query interfaces on web databases*. In: Proceedings of the 22Nd International Conference on Data Engineering,

[6]. ICDE '06, pp. 46–. IEEE Computer Society, Washington, DC, USA (2006).

[7]. Dragut, E.C., Fang, F., Yu, C., Meng, W.: *Deriving customized integrated web query interfaces.* In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, WI-IAT '09, pp. 685– 688. IEEE Computer Society, Washington, DC, USA (2009).

[8]. Furche, T., Gottlob, G., Grasso, G., Guo, X., Orsi, G., Schallhart, C.: *The ontological key: automatically understanding and integrating forms to access the deep web*. The VLDB Journal 22(5), 615–640 (2013).

[9]. Dragut, E.C., Meng,W., Yu, C.T.: *Deep Web Query Interface Understanding and Integration*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers (2012).

[10]. He, B., Chang, K.C.C.: Automatic complex schema matching across web query interfaces a correlation mining approach. ACM Trans. Database Syst. 31(1), 346–395 (2006).

[11]. He, B., Patel, M., Zhang, Z., and Chang, K., Accessing the Deep Web, Commun. ACM, vol. 50, pp. 94--101, New York, NY, USA, (2007).

[12]. He, H., Meng, W., Yu, C., Wu, Z.: *Automatic integration of web search interfaces with wise-integrator*. The VLDB Journal 13(3), 256–273 (2004).

[13]. Kabisch, T., Dragut, E.C., Yu, C.T., Leser, U.: *A hierarchical approach to model web query interfaces for web source integration*. PVLDB 2(1), 325–336 (2009).

[14]. He, H., Meng, W., Yu, C.T., Wu, Z.: *Wise-integrator: An automatic integrator of web search interfaces for e-commerce*. In: VLDB, pp. 357–368 (2003).

[15]. Kabisch, T.: *Extraction and integration of web query interfaces*. Ph.D. thesis (2011).

[16]. Kabisch, T., Dragut, E.C., Yu, C., Leser, U.: *Deep web integration with visqi*. Proc. VLDB Endow. 3(1-2), 1613–1616 (2010).

[17]. Kabisch, T., Dragut, E.C., Yu, C.T., Leser, U.: *A hierarchical approach to model web query interfaces for web source integration*. PVLDB 2(1), 325–336 (2009).

[18]. Li, Y., Wang, Y., Jiang, P., Zhang, Z.: *Multi-objective optimization integration of query interfaces for the deep web based on attribute constraints*. Data and Knowledge Engineering 86, 38–60 (2013).

[19]. Marin-Castro, H.M., Sosa-Sosa, V., Lopez-Arevalo, I.: *A tree-based wqi modelling approach for integrating web databases*. In: 17th International Conference on Information Fusion, FUSION 2014, Salamanca, Spain, July 7-10, 2014, pp. 1–8 (2014).

[20]. Marin-Castro, H.M., Sosa-Sosa, V.J., Lopez-Arevalo, I., Escalante- Baldera, H.J.: *Automatic classification of web databases using domaindictionaries*. In: Proceedings of the 9th International Conference on Machine Learning and Data Mining in Pattern Recognition, MLDM'13, pp. 340–351. Springer-Verlag, Berlin, Heidelberg (2013).

[21]. Marin-Castro, H.M., Sosa-Sosa, V.J., Martinez-Trinidad, J.F., Lopez- Arevalo, I.: *Automatic discovery of web query interfaces using machine learning techniques*. Journal of Intelligent Information Systems 40(1), 85–108 (2013).