WISE-Integrator: A System for Extracting and Integrating Complex Web Search Interfaces of the Deep Web

Hai He, Weiyi Meng
Dept. of Computer Science
SUNY at Binghamton
Binghamton, NY 13902
{haihe,meng}@cs.binghamton.edu

Clement Yu
Dept. of Computer Science
Univ. of Illinois at Chicago
Chicago, IL 60607
yu@cs.uic.edu

Zonghuan Wu
Center for Adv. Compu. Studies
Univ. of Louisiana at Lafayette
Lafayette, LA 70504
zwu@cacs.louisiana.edu

Abstract

We demonstrate WISE-Integrator – an automatic search interface extraction and integration tool. The basic research issues behind this tool will also be explained.

1. Introduction to WISE-Integrator

Many online store websites, such as amazon (www.amazon.com) and barnesandnoble (www.bn.com), have convenient interfaces (like the one in Figure 1) for users to search their merchandise databases. By feeding the HTML page of each such search interface of a group of search engines in the same domain, WISE-Integrator [4, 5] can automatically create a unified interface. Users can then submit queries against this interface and the search mediator will send the translated sub-queries to each site and then return the combined search results of these sites to the users. In other words, users can meta-search these Web databases.

Providing such an integrated search system over Web databases has become a Web application in high demand, as crawling-based general search engines like Google are not fully capable of searching them as their contents, known as *deep Web* or *invisible Web*, are *hidden* behind their Web search interfaces and not effectively crawlable.

As most Web databases can be effectively accessed only through their Web search interfaces, Web search interface integration has become the most popular way to integrate Web databases.

Although a few business Web sites, such as shopping.com and pricegrabber.com, allow users to do comparison-shopping from multiple e-commerce Web databases, their techniques are not publicly reported. A number of tasks are involved in building such deep Web integration systems, such as source discovery and clustering, interface extraction and integration, query translation, result extraction, etc. Due to the semi-structured nature of HTML data and the heterogeneities of the sources, significant laborious human efforts are involved in the building process, especially when the number of sources is large. As a result, building such a system is time-consuming and needs lots of expertise.

WISE-Integrator aims at *maximally automating* the process of building large-scale *deep* Web integration systems, so as to significantly reduce the cost of building and maintaining them. In WISE-Integrator, we focus on integrating structured Web sources that are supported by structured databases with complex Web search interfaces. These interfaces often contain several attributes that are formed by logically related *labels* (description texts) and HTML *control elements* (e.g., textbox, selection list, radio buttons and checkbox). For example, the search interface in Figure 1 has 10 *attributes* and the attribute "Author" consists of a label and a textbox.

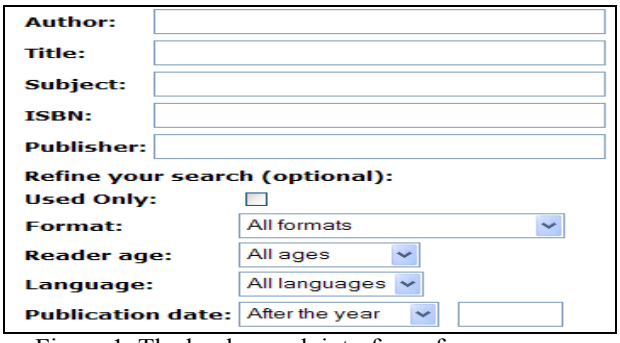

Figure 1. The book search interface of amazon.com

As shown in Figure 2, there are two major sub-systems in WISE-Integrator: (1) interface *schema extractor* and (2) interface *schema integrator*. Given a set of *raw* HTML pages containing the search interfaces of multiple sources in the same domain, the interface extractor identifies logic attributes by grouping related

Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005

labels and elements on each interface, and derives meta-information (e.g., domain type and data type) of these attributes. The constructed interface schemas are output in certain format (e.g., XML). Then, the interface integrator accepts these schemas as inputs. It first identifies matching attributes across different schemas and then merges the discovered matching attributes to generate global attributes. A unified search interface is produced based on the global attributes. The entire process is automatically performed without human interactions.

Although some recent researches such as MetaQuerier [3] have been conducted towards the similar goal, WISE-Integrator is a more comprehensive system for integrating Web databases and its underlying techniques are fundamentally different, as will be seen later.

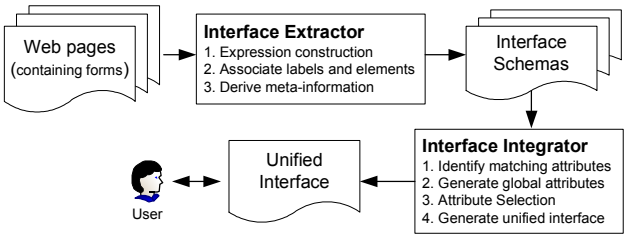

Figure 2. The architecture of WISE-Integrator

2. Interface extractor

Web search interfaces are designed autonomously for human users to understand and use. The semantically related labels and elements of a search interface are viewed as logical attributes, though they are scattered in the HTML text without formal definitions. Therefore, attributes have to be identified by grouping associated labels and elements. Moreover, beyond the labels and elements, a significant amount of semantic/meta information for attributes exists on Web search interfaces. For example, in Figure 1, "Publication date" implies the attribute is semantically a *date* data type, and its two elements are used to specify a *range* query condition with different roles in specifying the condition. Unlike the conventional database schemas, such semantic/meta information is "*hidden*" from computers and not formally defined on Web search interfaces. As such, the "*hidden*" information about each attribute needs to be revealed and defined to enrich the interface schema.

Extracting attributes:
To help attribute extraction, an ***i****nterface* ***exp****ression* (IEXP) is used to capture the *visual layout* of elements and labels on a search interface. As an example, the IEXP of the search interface in Figure 1 can be represented as "te|te|te|te|te|t|te|te|te|te|tee", where the first 't' denotes the label "Author", the first 'e' denotes the corresponding textbox, the first '|' is the first row delimiter. IEXP organizes labels and elements into multiple rows. IEXP provides a high-level description of the *visual layout* of different labels and elements on the interface while ignoring the details of organizing labels and elements.

First, the IEXP of a given search interface is constructed when parsing the search interface to get individual labels and elements. Second, based on the IEXP, related labels and elements are grouped such that each group corresponds to an attribute. This is achieved by our layout-expression-based extraction technique (LEX) [5, 6]. For each element *e* in a row, LEX finds the label either in the same row or in adjacent rows above the current row that is most likely to be the attribute label for *e* based on an association weight of the label with *e*. The association weight is computed using a number of heuristics (e.g., the similarity of label texts and element names, and the adjacency of labels and elements).

The average attribute extraction accuracy of the LEX approach is about 94% based on our experiments. We also compared our approach with the MetaQuerier approach [11] experimentally using a dataset collected by the authors of MetaQuerier; the accuracy of our approach is about 7% better than that of the MetaQuerier approach.

Deriving semantic/meta information:
The following six types of semantic/meta information for each attribute are currently defined in our interface schema model [5, 6]: *layout order*, *domain type*, *data type*, *default value, unit* (e.g., *kilogram* is a unit for *weight*) and *element relationship*. Specifically, there are four domain types (*range*, *finite*, *infinite,* and *Boolean*), seven data types (*date*, *time*, *datetime*, *currency*, *id*, *number* and *char*) and four types of relationships between the elements of an attribute (*range*, *part*, *group* and *constraint*). The Interface Extractor of WISE-Integrator can automatically derive the above semantic/meta information with the help of some domain independent knowledge such as common patterns for date and time. To our knowledge, no work in the literature has addressed the representation and extraction of the above semantic/meta information on Web search interfaces.

3. Interface integrator

Integrating interfaces into a unified interface has two challenging problems: identifying matching attributes and producing a unified search interface.

Identifying matching attributes:
As Web search interfaces are designed autonomously, the semantically similar or same attributes from different Web search interfaces may have different label names, element formats and compositions. To identify matching attributes, WISE-Integrator applies three levels of schema information: attribute names (the dictionary level), field specification (the schema level, e.g., data type), and attribute values and patterns (data content level) [4, 5]. A

two-step clustering approach is employed by WISE-Integrator to identify matching attributes. In the first step, all search interfaces in the same domain are considered and attributes are clustered based on exact matches of attribute names/values. In the second step, further clustering is performed based on approximate matches (e.g., approximate name/value matches) and meta-information matches (e.g., data type match). Utilizing the rich meta-information as well as the two-step clustering technique is a novel feature of WISE-Integrator. Our experimental results with WISE-Integrator show that our approach to attribute matching is highly effective, with accuracy about 95% (see [4, 5] for more details). In addition, we believe that the approach can be applied to other schema matching problems.

Producing a unified search interface:
Producing a unified search interface [4, 5] is primarily a problem of reconciling differences among the discovered matching attributes to generate suitable global attributes that are compatible with the matching attributes. This problem has been rarely addressed in other work, especially in the context of Web interface integration. In WISE-Integrator, the following issues are addressed:

1) **Determine the label for each global attribute**. Since matching attributes may have different label names, a label name needs to be determined for their corresponding global attribute. In WISE-Integrator, a combination of a *majority strategy* (labels that appear in more interfaces are preferred) and a *generality strategy* (more generic names are preferred) are used to determine the label for each global attribute. Then *mappings* from a global attribute to each of its corresponding matching attributes are established.

2) **Determine the domain and values of each global attribute**. For each global attribute, a domain and a value set that are compatible with different domains and values of the matching attributes need to be produced. A set of rules is used to integrate domains, for example, a *finite* domain and a *range* domain are integrated into a *range* domain. Two cases are considered for value integration: *alphabetic values* and *numeric values*. Alphabetic values from different matching attributes are organized into hypernymy hierarchies for ease of use. For numeric values, special attention is paid on integrating range values. For both cases, the impact of the integration on the *cost* of evaluating user queries is taken into consideration.

3) **Determine the layout of the unified Web search interface**. In WISE-Integrator, the layout positions of the matching attributes are aggregated to determine the layout positions of the global attribute. Basically, global attributes whose corresponding local attributes appear frequently in more search interfaces and appear near the top of more search interfaces are positioned near the top of the unified search interface.

A special feature of WISE-Integrator is that users can remove an existing interface from or add a new interface to an existing unified interface at any time on the fly; WISE-Integrator will generate the new unified interface without starting from scratch, which is important for incremental maintenance/integration.

4. Significance of contribution

WISE-Integrator's contribution was discussed in detail in [4] and [5]. Due to the space limitation, we only discuss them briefly here.

1) Because of the scalability resulted from the high degree of automation, this system can be used to efficiently build and maintain large-scale deep Web search systems, which metasearch large numbers of search engines that have heterogeneous, complex interfaces, with little laborious human efforts.
2) Because of the flexibility and robustness, also resulted from the high degree of automation, this system can help ordinary individuals build portable, customized and personal deep-Web search tools, without requiring comprehensive expertise on Web development and programming skills.
3) Compared to the existing state-of-the-art work [1, 3, 8, 11], the Interface Extractor can achieve deeper understanding of Web search interfaces in the sense that more semantic/meta information on search interfaces can be extracted. The enriched interface schema with such semantic/meta information can be used in many applications such as schema matching, query translation, Web database clustering, deep Web crawling, and search result extraction and annotation.
4) We extend the traditional metadata-based database schema matching techniques (e.g., [7]) to the scenario of Web search interface integration, and propose the use of rich *meta-information* and *clustering* techniques to improve the accuracy of attribute matching.
5) We present the idea of automatically merging matching attributes and producing a unified integrated Web search interface over multiple heterogeneous Web search interfaces, which has rarely been discussed in the literature.

We now briefly review other closely related work in the context of deep Web integration.

Interface extraction: The LITE method [8] uses a layout engine to obtain candidate labels that are physically closest to an element. The LITE method is not attribute-oriented, in other words, it extracts labels only for elements rather than attributes. Z. Zhang et al [11] in MetaQuerier view search interfaces as a visual language, therefore, use a number of manually pre-defined grammar rules to extract semantically related labels and elements. However, both LITE method and MetaQuerier approach

do not address the problem of extracting *exclusive attributes* (attribute names occur as values of elements instead of as labels), which in fact exist in many real Web search interfaces. Furthermore, they only focus on grouping related labels and elements, but other semantic/meta information on search interfaces is not considered. In Ontobuilder [1], the extracted ontologies from Web search interfaces are only limited to the properties (existing in HTML text) of labels and elements themselves. Moreover, its extraction method is not detailed in publicly available reports.

Interface integration: Some recent works [2, 9, 10] address the attribute-matching problem in the context of Web search interface integration. B. He et al [2] in MetaQuerier argue that a unified *hidden schema model* exists for each domain. They conduct schema matching by using a statistical/probabilistic approach to obtain the hidden schema model. However, their approach considers only attribute labels but not other useful information on search interfaces such as attribute domain type and attribute values. We find that such schema information is very effective in interface integration. Moreover, it is not clear how semantic relationships between names (such as synonymy and hypernymy) are obtained in this work. The two works [9, 10] were reported after our WISE-Integrator approach. J. Wang et al [9] use query probing to discover matches in interface schemas as well as result schemas. W. Wu et al [10] integrate human interaction into the schema matching process. However, the three works [2, 9, 10] discuss only schema matching, but not attribute merging and global interface generation that are important issues and must be addressed in interface integration. Even in the MetaQuerier demonstration system [3], which is a combination of the works in [2,11], the two issues are not yet fully addressed.

5. Demo plan

In this demo, we will demonstrate the features of WISE-Integrator and the effectiveness of its underlying techniques. To facilitate the demo, a large number of raw HTML pages containing real search interfaces from several domains (e.g., books and music) are pre-collected for WISE-Integrator to use. Through the demo, depending on the different interests, a visitor could 1) instantly create a unified interface by feeding WISE-Integrator with the HTML pages of some pre-collected search interfaces, 2) provide the URLs of the interfaces of multiple new search engines and let WISE-Integrator build a unified interface, 3) add a new search interface into or remove an existing search interface from an existing unified interface. In each case, the unified search interface can be viewed on a Web browser based on an automatically generated HTML file.

For visitors who are interested in more technical details, intermediate results, such as extracted attributes, derived semantic/meta information, will be reported as properties of associated search engines. Others such as identified matching attributes, generated global attributes, etc. will be highlighted in an automatically generated flowchart, as the property of the integrated interface, to visualize the process of interface integration.

To compare WISE-Integrator with other work, during the part of the demonstration that will be led by authors, we will compare WISE-Integrator with other systems such as MetaQuerier (to the best of our knowledge, MetaQuerier is the most similar work to ours), which are available in the public domain, as long as audience is interested. Especially, through this demonstration, visitors could understand the unique features of WISE-Integrator, including the two clustering steps of attribute matching, the integration of attribute domains and value sets, the derived semantic/meta information, and the actual unified search interfaces.

Acknowledgement: This work is supported by the following grants from NSF: IIS-0208574, IIS-0208434, IIS-0414981 and IIS-0414939.

References

[1] A. Gal, G. Modica, and H. Jamil. OntoBuilder: Fully Automatic Extraction and Consolidation of Ontologies from Web Sources. Demo, ICDE 2004.

[2] B. He and K. Chang. Statistical Schema Matching across Web Query Interfaces. SIGMOD Conf., 2003.

[3] B. He, Z. Zhang, K. Chang. Knocking the Door to the Deep Web: Integration of Web Query Interfaces. SIGMOD Conference, Demo, 2004.

[4] H. He, W. Meng, C. Yu, and Z. Wu. WISE-Integrator: An Automatic Integrator of Web Search Interfaces for E-commerce. VLDB Conference, 2003.

[5] H. He, W. Meng, C. Yu, and Z. Wu. Automatic Integration of Web Search Interfaces with WISE-Integrator. The VLDB Journal, 13(3), Sept. 2004.

[6] H. He, W. Meng, C. Yu, and Z. Wu. Construction of Search Interface Schemas of Web Databases. Technical report, Binghamton University, 2004.

[7] W. Li, and C. Clifton. SEMINT: A Tool for Identifying Attribute Correspondences in Heterogeneous Databases Using Neural Networks. Data & Knowledge Engineering, 33: 49-84, 2000.

[8] S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. VLDB Conference, 2001.

[9] J. Wang, J. Wen, F. Lockovsky and W. Ma. Instance-based Schema Matching for Web Databases by Domain-specific Query Probing, VLDB 2004.

[10] W. Wu, C. Yu, A. Doan, and W. Meng. An Interactive Clustering-based Approach to Integrating Source Query Interfaces on the Deep Web**.** SIGMOD Conference, 2004.

[11] Z. Zhang, B. He, and K. Chang. Understanding Web Query Interfaces: Best-Effort Parsing with Hidden Syntax. SIGMOD Conference, 2004.