SG-WRAP: A Schema-Guided Wrapper Generator

Xiaofeng Meng[1]   Hongjun Lu[2]   Mingzhe Gu[1]   Haiyan Wang[1]

[1]Information School, Renmin University of China, Beijing, China
xfmeng@public.bta.net.cn

[2]Hong Kong University of Science & Technology, China
luhj@cs.ust.hk

1. Introduction

With the development of the Internet, the World-Wide-Web has become everyone's invaluable information source. However, most of data on the Web is currently in the form of HTML pages, which is neither well-structured nor associated with schema. It is almost impossible to use such data efficiently. Web wrapper technology has been developed to transform unstructured /semi-structured data to semi-structured/structured data, which can be queried and analyzed using matured techniques developed in database and other fields. The main issues of wrapper generation include (1) to identify semantics of the data contained in an HTML document and, (2) to establish the mappings between its structure and its semantics. Various wrapper generation tools developed addressed these issues in different ways [HBG+97, KMA+98, KWD97, SA99, LPH00].

SG-WRAP, the system to be demonstrated was design and developed based on the following observations. First, it is true that user interaction is probably the best way to help the generation of wrappers for a specific HTML source quickly and accurately because of the diversity of HTML pages and limited semantic information expressed in HTML tags. However, user's effort should be minimized as much as possible. Most existing systems can still be much improved in this regard. Second, the ultimate goal for wrapper generation is to transform the original data into some structure that is easy to consume rather than to understand the structure of the original data. When a user gathers data from the Web, s/he must have her/his needs in her/his mind. It is often not necessary to generate wrappers for entire HTML document. Therefore, SG-WRAP, adopts a novel, schema guided, approach for wrapper generation. With this approach, a user defines the schema of data to be extracted from an HTML page in terms of data type descriptors (DTD) of XML [W3C00]. The user also provides example mappings by associating data in the HTML page and elements in DTD. The system then induces the mapping rules and generates a wrapper that extracts data from the HTML page and produce an XML document conformed to the specified DTD.

Compared with other available wrapper generators, SG-WRAP approach has the following unique features.

- With the guidance of user-defined schema, the wrapper generated could be more accurate and better reflect the users requirements. For the same source page, different users may be interested in different portion of the data, which can be obtained by different user-defined schema.
- The extracted data follows the user-defined schema and is ready to populate to database directly. As such, the wrapper can be easily integrated into the data integration process.
- Minimum user interaction is required. After a user defines the schema, s/he only needs a few clicks to establish the correspondences between data elements in the HTML document and elements in the schema. The system will induce the data extraction rules from the given sample matching instances.
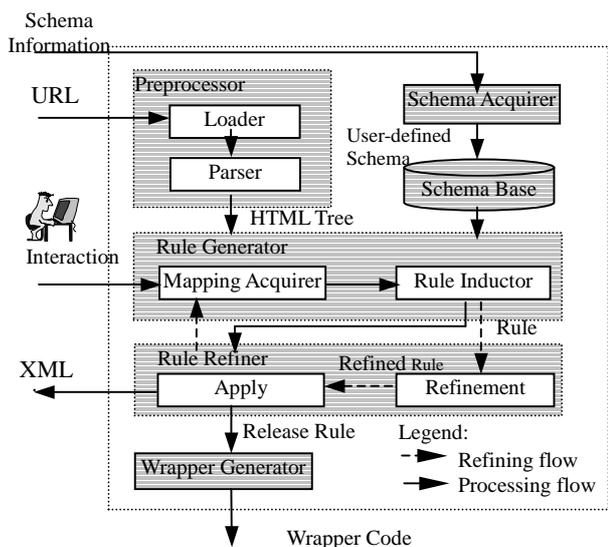

Figure 1: SG-WRAP: The system Architecture

2. SG-WRAP: The System

Figure 1 depicts the architecture of SG-WRAP. The system consists of five major components: *Preprocessor*, *Schema Acquirer*, *Rule Generator, Rule Refiner* and *Wrapper Generator*. The main user interface is shown in Figure 2. Module Preprocessor is responsible for setting up the environment for the system. It fetches the Web page using the URL (Figure 2, A) given by the user. The fetched HTML page is displayed (Figure 2, B). Another input of the system is a user-defined schema for extracted data, which is obtained by the Schema Acquirer and displayed on the screen (Figure 2, C). A user entered schema can be saved in a Schema Base to be used later, or shared for other sources from which the same kinds of data to
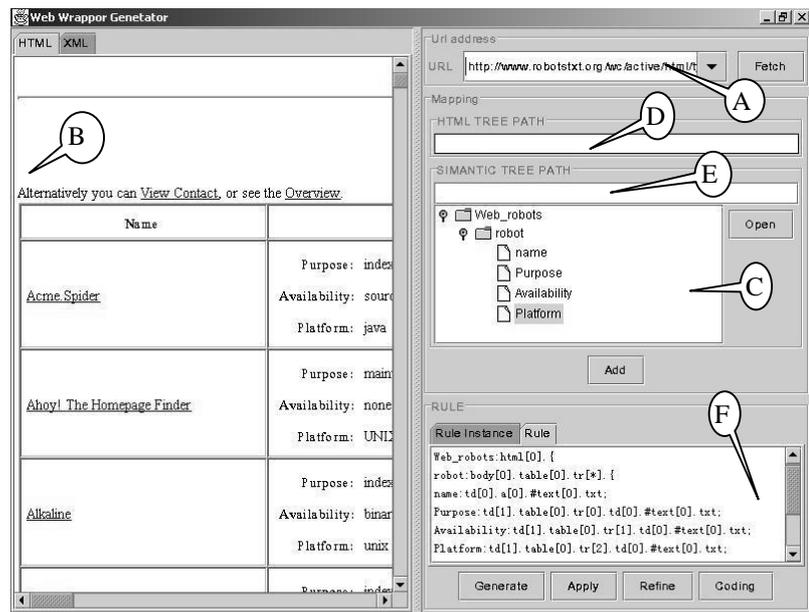

Figure 2: SG-WRAP: The main interface

be extracted. The fetched HTML page is parsed. Syntax errors, such as missing tags in the original HTML document are fixed during the parsing process. Internally the HTML page is represented as a tree using the document object model (DOM) [W3C98], where a data item is a leaf node in the tree and its position in the document can be described by the path from the root to the leaf. As such, when the user highlights a string in the page shown, its path can be identified, which is displayed as an HTML tree path (Figure 2, D). Similarly, when the user clicks an element in the DTD, it is displayed as a semantic tree path in the schema (Figure 2, E). The Mapping Acquirer of the system captures those user clicks that associate strings in the HTML page and their corresponding elements in the DTD. For example, with the sample page in Figure 2, the following mapping can be captured

```
html.body.table[0].tbody[0].tr[0].a[0] <=> Web_robots.robot.name  "Alkaline"
```

where `Alkaline` is a string in the HTML page, `html.body.table[0].tbody[0].tr[0].a[0]` is the tree path in the HTML tree, and `Web_robots.robot.name` is the associated element in the DTD, respectively. In other words, from user interactions, the system obtains a set of instances of rules that map strings in the HTML page to elements in the schema. With this set of instances of mapping rules, the Rule Generator generates data extraction rule by an induction algorithm, which takes the list of mapping rules instances *L* as input and returns a candidate rule by incorporating the similar mapping rule instances into a new extraction rule. A sample data extraction rule for the HTML page shown in Figure 2 is as follows

```
Web_robots:html{
    Robot: body.table[0].tbody[0].tr[*] {
            Name:           a[0].txt ;
            Purpose:        table[0].tbody[0].tr[0].td[0].txt;
            Availability:   table[0].tbody[0].tr[1].td[0].txt;
            Platform:       table[0].tbody[0].tr[2].td[0].txt
    }
}
```

The data extraction rules are induced from a limited set of instances. In order to guarantee that the data extraction rule is applicable for the entire HTML document, SG-WRAP includes a refining process. The Rule Refiner generates an XML document by applying the induced rule on the input page. The generated document is displayed in Figure 2, window A for users to preview. From the displayed document, user determines whether the current extraction rule correctly extracts required data from the source page. If not, s/he can identify more mapping instances. The induction process will repeat and the Refiner will merge the previously generated rules with the refined rules. The system will display a new version of result to the user for checking. This refining process continues until the user is satisfied with the data extracted. The wrapper is generated based on the final

data extraction rule by the Wrapper Generator.

From the above brief description, we can see the following:

- Compared with [HBG+97, KMA+98, KWD97], our system focuses on the users requirements. Let users, rather than the system, choose what users want. The wrapper generated could be more accurate and better reflect the users requirements.

- Although W4F [SA99] provides a declarative language to define extraction rules that are very expressive, the generation of the rules must rely on experts' experience. So it is very difficult for non-expert users to use it. Our system provided a friendly interface for user to interact with the system by clicks.

- XWRAP [LPH00] generates wrappers for Web sources semi-automatically through the interaction with the user. However, it still requires quite a lot of works from the users. In our system, minimum user interaction is required.

Although wrapper generation work has been reported in the literature, there seem no standard ways to evaluate the performance of such systems. We conducted a series of experiments to evaluate the usability, correctness and efficiency of SG-WRAP. The usability tests selected a number of users to use the system. The results indicated that, with minimal introduction of the system, DTD definition and structure of HTML pages, even naï ve users could quickly generate wrappers without much difficulty. For correctness, we adapted the precision and recall metrics in information retrieval to data extraction. The results show that, with the refining process, the system can generate wrappers with very high accuracy. Finally, the efficiency tests indicated that the wrapper generation process is fast enough even with large size Web pages.

3. Plan of the Demonstration

The system to be demonstrated is implemented in Java (Sun JDK1.3). During the demo,

- we will demonstrate the major algorithms used in SG-WRAP, i.e., the rule induction and refining algorithms.

- a complete version of SG-WRAP will be available so that users can use the system to generate wrappers for Web pages of their choices to appreciate the schema-guided wrapper generation approach and the friendliness and accuracy of the system.

- the experimental results conducted will also be reported.

References

[HBG+97] J. Hammer, M. Brenning, H. Garcia-Molina, S. Nestorov, V. Vassalos, and R. Yerneni. Template-based wrappers in the TSIMMIS system. In Proceeding of ACM SIGMOD Conference, Tucson, Arizona, May 1997, 532-535.

[KMA+98] C. A. Knoblock, S. Minton, J. L. Ambite, N. Ashish, P. J. Modi, I. Muslea, A. Philpot, and S. Tejada. Modeling web sources for information integration. In *Proceedings of AAAI Conference*, Madison, WI, 1998, 211--218.

[KWD97] N. Kushmerick, D. Weil, and R. Doorenbos. Wrapper induction for information extraction. *In Proceedings of Int. Joint Conference on Artificial Intelligence (IJCAI)*, volume 1, Nagoya, Japan, 1997, 729-735.

[LPH00] L.Liu, C. Pu, W. Han. XWRAP: An XML-enabled Wrapper Construction System for Web Information Sources, In *Proceedings of the International Conference on Data Engineering*, San Diego, CA USA, 2000, 611-621.

[RA00] D. Raggett. Clean Up Your Web Pages with HTML TIDY. *http://www.w3.org/People/Raggett/tidy/*, 2000.

[SA99] A. Sahuguet and F. Azavant. WysiWyg Web Wrapper Factory (W4F). In *Proceedings of WWW Conference*, Toronto, Oct. 1999.

[W3C98] World Wide Web Consortium (W3C). *The Document Object Model*, http://www.w3.org/DOM, 1998.

[W3C00] World Wide Web Consortium (W3C). *Extensible Markup Language (XML1.0).*