By Dawn G. Gregg and Steven Walczak

ADAPTIVE WEB INFORMATION EXTRACTION

The Amorphic system works to extract Web information for use in business intelligence applications.

Web mining has the potential to dramatically change the way we access and use the information available on the Web. Tools for mining the Web allow users to query and combine data based on its semantic content. Services already exist that utilize Web mining to wrap, mediate, and restructure information from the Web into a form that provides added value for users (for example, Internet price services extract prices from a variety of sources and provide it in a unified framework). In addition, Web business intelligence applications are an emerging type of decision support software that "leverages the unprecedented content on the Web to extract actionable knowledge in an organizational setting" [12].

Illustration by Carmen Segovia

Ideally, Web information services and Web business intelligence applications would have access to structured information that could be easily extracted and incorporated into their value-added services, but currently this is not the case. The Web provides access to an enormous volume of semi-structured HTML data in a variety of ever-changing formats. This presents several major challenges to developers interested in using Web data in their applications: First, HTML documents containing interesting data must be located. Second, data of interest must be located within the Web page and rules that can be used to reliably extract the data must be created. Third, the mechanism used to create data extraction rules must either be sufficiently general or be easy to implement so that data can be extracted from the wide variety of page formats available on the Web. Finally, the information extraction system must be able to cope with changes to Web page structure since Web content providers frequently change the configuration and content of their pages. Figure 1 illustrates the adaptive information extraction process envisioned in this research.

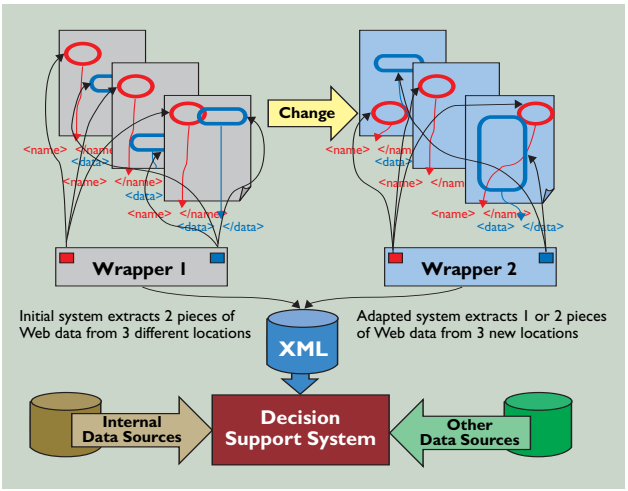

Figure 1. Adaptive Web information extraction process.

Adaptive Information Extraction System Requirements

Accurate: The system must extract the correct data.

Resilient: The system must continue to work properly even when Web pages change.

Self-repairing: The system should automatically repair its extraction rules when a Web page changes.

General: Information extraction rules need to work for most Web sites in an application domain.

Extensible: Information extraction rules should be easy to build for a variety of domains.

Open: The system should allow for platform-independent data exchange.

Today's information extraction systems usually rely on extraction rules or wrappers tailored to a particular information source. These wrappers translate semi-structured HTML data into a regular form so that it can be written to a database or consumed directly by other applications. Currently most automatic information extraction systems can only cope with a limited set of document formats and do not adapt well to changes in document structure. As a result, many real-world information sources with complex document structures cannot be consistently interpreted using a single information extraction system.

An effective Web information extraction system must interpret a wide variety of HTML pages and adapt to changes without breaking. An information extraction system should recognize different Web page structures and act on this knowledge to modify the information extraction techniques employed. In addition, the system should be customizable for a variety of domains and data-object types [6]. We call this type of information extraction system adaptive because it has the capability of adjusting to the wide variety of document formats used to distribute Web-based information (see the box here).

Current Web Information Extraction Systems

Web information extraction involves locating documents and identifying and extracting the data of interest within the documents. Information extraction systems usually rely on extraction rules called wrappers that are tailored to a particular information source. A wrapper is defined as a program or a rule that understands information provided by a specific source and translates it into a regular form as, for instance, XML or relational tables. Wrappers are specific to a given Web site and are tightly linked to the mark-up and structure of provider pages. The most challenging aspect of wrappers is they must be able to recognize the data of interest among many other uninteresting pieces of text (for example, mark-up tags, inline code, and navigation hints, among others [9]).

The simplest information extraction systems utilize extraction rules that are constructed manually. These systems require a human developer to create a new wrapper for each information source or for information sources that are structurally changed. This limits users to accessing information only from predefined information sources. Wrapper induction has been suggested to overcome the lack of scalability in

the manual wrapper generation process [8]. The wrapper induction method automatically builds a wrapper by learning from sample pages.

Currently there are two principal methods for identifying interesting data within Web pages: ontology-based extraction and position-based extraction.

Ontology-based Extraction. Ontology-based information extraction tools feature many of the properties desired for an adaptive Web information extraction system. An ontology-based tool uses domain knowledge to describe data. This includes relationships, lexical appearance, and context keywords. Wrappers generated using domain ontologies are inherently resilient (that is, they continue to work properly even if the formating features of the source pages change) and general, (they work for pages from many distinct sources belonging to a specific application domain) [4].

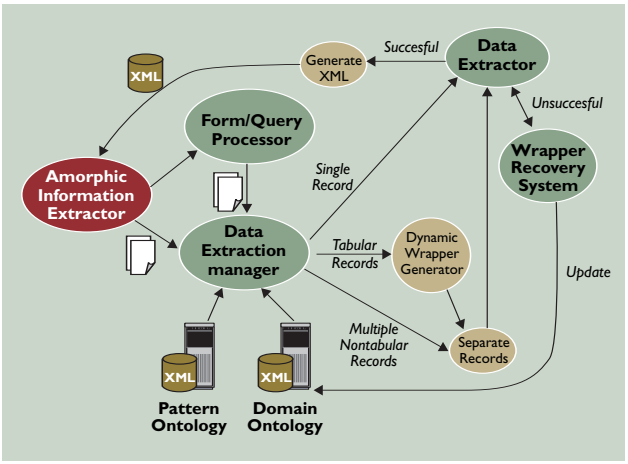

Figure 2. The Amorphic architecture.

However, ontology-based tools require that the data be fully described using page-independent features. This means the data must either have unique characteristics or be labeled using context keywords. Unfortunately, all interesting Web data does not necessarily meet these requirements. Some data is freeform and cannot be identified using a specific lexical pattern and also is not labeled. This type of data can only be extracted using its specific location in the HTML page.

Position-based Extraction relies on inherent structural features of HTML documents to accomplish data extraction. Under a position-based extraction system, a HTML document is fed to a HTML parser that constructs a parsing tree that reflects its HTML tag hierarchy. Extraction rules are written to locate data based on the parse-tree hierarchy. If a collection of items is to be retrieved (as from a search results page), a regular expression is constructed to allow multiple items to be retrieved for a hierarchical pattern.

Position-based extraction lacks the resilience of ontology-based extraction. When there are changes to the structure of the target Web pages, it frequently fails. However, it does guarantee a high accuracy of information extraction, with precision and recall being at least 98% [2]. In addition, it is possible to use wrapper induction to create position-based wrappers based on a sample of regularly formatted Web pages. This can greatly speed the development and update of position-based wrappers [11]. Thus, position-based extraction can be appropriate when the data to be extracted can only be identified based on its location within a Web page and not on domain information.

Wrapper Recovery and Repair. The Web is a dynamic medium, and, as such, Web pages are frequently altered in structure and appearance. These changes are made by ISPs to offer additional content and functionality, increase ease of use, or make the Web page more attractive to new users. When a Web page's structure is changed, a wrapper can fail to find keywords or path expressions in the page and thus cannot complete the information extraction. In most information extraction systems, once a wrapper fails it must be manually recreated to conform to the new page structure, which slows the recovery process [1].

An important characteristic for an adaptive infor-

An effective Web information extraction system must interpret a wide variety of HTML pages and ADAPT TO CHANGES WITHOUT BREAKING.

mation extraction system is for it to repair itself when an information extraction error occurs [7]. The problem of wrapper repair and maintenance is only beginning to be addressed by researchers (for example, [2, 9]); and has not been addressed for data extraction systems that utilize domain ontologies instead of position-based extraction rules.

In a system capable of wrapper recovery, the wrapper processor triggers a recovery routine when an error is detected during data extraction. This recovery routine attempts to repair the wrapper and resume the extraction process. Wrapper recovery and repair consists of two steps. First, the recovery routine must attempt to locate the target data within the revised page structure. If successful, the extraction rules must then be regenerated to match the new page format [2, 10]. If the extraction recovery is not successful, or the wrapper cannot be repaired automatically, the system should generate an error message so that a human can assist in the recovery process.

The processes involved in adaptive information extraction are discussed here in the context of an Amorphic Web information extraction system prototype.

An Integrated Web Information Extraction Solution

To illustrate the processes involved in Web information extraction, an information extraction system that combines position-based extraction, ontology-based extraction, and wrapper recovery was created. The Amorphic system can locate Web data of interest based on domain knowledge or page structure, can automatically generate a wrapper for an information source, and can detect when the structure of a Web-based resource has changed and act on this knowledge to locate the desired information. One key feature of the Amorphic system is that both the extraction rules and the output data are represented by XML documents. This approach increases modularity and flexibility by allowing the extraction rules to be easily updated (manually or automatically), and by allowing the retrieved data to either be converted to HTML for consumption by a human or returned in a SOAP envelope as part of a Web Service. The current Amorphic prototype represents a cost-effective approach to developing large-scale adaptable information extraction systems for a variety of domains. The Amorphic system, shown in Figure 2, consists of several modules:

- The *form/query processor* creates a user query by parsing the site's search form, combining the user query with the site's form elements, and sending the resulting search parameters to obtain the HTML search result pages.
- The *data extraction manager* examines the page structure and determines how best to parse the site. This module analyzes the content of the HTML page, and constructs extraction rules using the domain knowledge. The extraction rules are used to locate data of interest (tokens) within the HTML page.
- The *data extractor* pulls the specific data from the HTML pages.
- The *wrapper recovery system* is invoked when the Amorphic system cannot locate tokens within the Web pages.

A prototype Amorphic information extraction system has been implemented using Java.

Data Preprocessing. The HTML page undergoes several preprocessing steps before the data extraction is performed. First, a document is retrieved from the Web. The document is then processed using a HTML parser to obtain a representation of the Web page's structure. As HTML pages are composed of tags and text enclosed by tags, it is possible to represent a HTML page's layout by a tree of nested HTML tags that follows the Document Object Model (DOM). The parsing process separates HTML tags, attributes, and content. The Amorphic HTML parser uses the DOM parse-tree to create a *location-key* to identify the *content-text* found in the Web page. The location-key is a path expression that defines the set of nested tags that the content-text resides within [3]. Once the parsing process is complete, the Amorphic system examines the page structure to determine how

The current Amorphic prototype represents a COST-EFFECTIVE APPROACH to developing large-scale adaptable information extraction systems for a variety of domains.

to extract the tokens from the Web page.

- *Tabular Data*: If the Amorphic system detects that a page contains tabular data, a temporary wrapper is generated to map the appropriate columns to the correct token.
- *Multiple Data Records:* If the page contains multiple records, the Amorphic system separates the data into groups, each of which contains data for an individual record. The separation of data into single record groups is accomplished either using a <*Record Delimiter*> defined in the domain ontology or using a set of heuristics to locate the record boundaries dynamically (for example, [5]).

Data Extraction. Following completion of the data preprocessing steps, a data extraction process is initiated to locate the tokens within the parsed HTML document. The Amorphic system uses a three-step location process to correctly identify and extract the tokens. The system searches the set of location-key/content-text pairs generated by the HTML parser (and if appropriate grouped in the record separation process) for any of the keywords defined. If a keyword represents a path expression, this indicates that position-based extraction is being used and the location-key is used for the search. Otherwise, the data is being extracted using ontology-based extraction and the content-text is searched to locate the keyword.

Once a keyword is located, the Amorphic data extraction module searches the content-text before and after the keyword location to find data that matches a pattern defined in the domain ontology. When Web data containing an appropriate pattern is found, the data type is used to extract the desired token from the content-text. The extracted data is enclosed in XML tags and returned as a single XML record.

Automatic Wrapper Recovery and Repair. When the data extraction module cannot locate a required token using the standard extraction procedures, the basic recovery strategy is to locate the token using a new set of keywords. The wrapper recovery system uses a thesaurus to generate additional keywords to locate the data of interest. In the Amorphic system a thesaurus entry represents a special pattern set that can be defined for any word or group of words. These word patterns may then be used to replace a single word or set of words found in a keyword list for a domain. After new keywords are generated, a three-step location process is used to identify candidate Web data that could be the tokens of interest. First, the set of location-key/content-text pairs generated by the HTML parser is scanned to identify content-text that contains one of the thesaurus-generated keywords. When a keyword is located, candidate tokens are identified by searching the content-text before and after the keyword to find the first occurrence of a pattern defined in the domain ontology.

Following candidate token identification, a ranking process is initiated to select which candidate tokens are the most likely to contain the data of interest, and to determine what new keywords/key phrases are being used to label the data. In the ranking process, thesaurus keywords are used to identify the beginning and ending of each keyword phrase found. For example, if "Product Description" were changed to "Item Details & Description" the entire phrase would be the new keyword for the token. Valid tokens are separated from the noise by ranking each candidate token based on assumptions about the structure of Web pages.

Table 1. Information extracted using Amorphic.

Page type	% Records Retrieved	% Data Items Retrieved
Search Results Page	100.00%	99.99%
Single Item Page	100.00%	99.20%

Table 2. Information extracted with and without wrapper recovery.

Page type	% Records Retrieved without Recovery	% Records Retrieved with Recovery	% Data Items Retrieved without Recovery	% Data Items Retrieved with Recovery
Search Results Page	3.81%	100.00%	1.28%	76.45%
Single Item Page	100.00%	100.00%	25.00%	83.08%

When a Web page is changed it is possible that data of interest within that page could either be removed entirely or changed so much that it cannot be located using automatic wrapper repair and recovery procedures. In such cases the wrapper recovery and repair procedure will fail and an error message will be generated so that appropriate human intervention can be taken to repair the broken wrapper.

Adaptive Information Extraction in Practice

The Amorphic data extraction process and automatic wrapper recovery and repair was tested in the online auction field; preliminary results of which are summarized here.

In the proof of concept demonstrations, an XML ontology was developed for the online auction

The Amorphic system has been used to extract data relevant to the study of online auctions. IN THE FUTURE, SIMILAR SYSTEMS CAN BE USED TO EXTRACT DATA related to financial markets, online travel, or benefits, just to name a few.

domain. Since the domain ontology allows more than one set of keywords and more than one pattern to be specified, it can be used to extract data from several different Web sites in the auction domain. Once appropriate domain ontologies were created, the Amorphic information extraction system was used to extract data from 1,609 search-results pages and 626 single-item pages from the eBay, Yahoo, and Amazon online auction sites. Table 1 shows the prototype Amorphic system showed excellent performance for three Web sites tested.

To test the wrapper recovery procedures the Amorphic system was used to extract data from six additional online auction sites: Bidz.com; uBid.com; DellAuctions.com; CompUSAAuctions.com; BidVille.com; and ZBestOffer.com. The prototype Amorphic system demonstrated the ability to adapt to six additional Web sites it was not originally designed to support. The testing of the six additional auction sites did not require changes to the Amorphic program or the online auction domain ontology. Table 2 shows the information extraction results both with and without wrapper recovery. It shows the Amorphic agent was able to extract substantially more information from the six new Web sites using automatic wrapper recovery.

Conclusion

The use of external information for business decision making is not new. What is new is the abundance of information freely available via the Internet. However, this information is not being systematically included in current decision-making applications [8]. This research demonstrates that it is possible to reliably extract Web information for use in Web business intelligence applications. It will be possible for organizations to use a system like Amorphic to extract information of interest from Web pages for a wide variety of domains. These potential business intelligence applications will allow a deep and detailed look at small portions of the Web relevant to specific domains.

The Amorphic system has been used to extract data relevant to the study of online auctions. In the future, similar systems can be used to extract data related to financial markets, online travel, or benefits, just to name a few. Use of an information extraction system, like Amorphic, has the potential to provide businesses with access to up-to-date, comprehensive, and ever-expanding information sources that can in turn help them make better strategic decisions. C

References

1. Arasu A. and Garcia-Molina, H. Extracting structured data from Web pages. A*CM SIGMOD Record* (June 2003), 337–348.
2. Chidlovskii, B. Automatic repairing of Web wrappers by combining redundant views. In *Proceedings of IEEE Conf. Tools with AI* (Nov. 2002), 399–406.
3. Cohen, W., Hurst, M., and Jensen, L. A flexible learning system for wrapping tables and lists in HTML documents. In *Proceedings of the Conf. on WWW* (2002), 232–241.
4. Embley, D., Campbell, D., Smith, R., and Liddle, S. Ontology-based extraction and structuring of information from data-rich unstructured documents. In *Proceedings of the Conf. on Info. and Knowledge Management* (Nov. 1998), 52–59.
5. Embley, D.W., Jiang, Y., and Ng, Y.K. Record-boundary discovery in Web documents. *ACM SIGMOD Record 28*, 2 (June 1999), 467–478.
6. Gregg, D. and Walczak, S. Exploiting the Information Web. *IEEE Trans. on System, Man and Cybernetics Part C* (forthcoming 2006).
7. Knoblock, C., Leramn, K., Minton, S., and Muslea, I. Accurately and reliably extracting data from the Web: A machine learning approach. *Bulletin IEEE Computer Society Technical Committee on Data Engineering* 23, 4 (2000), 33–41.
8. Kushmerick, N., Weld, D., and Doorenbos, R. Wrapper induction for information extraction. In *Proceedings of the Conf. on AI* (1997), 729–735.
9. Laender, A.H.F., Ribeiro-Neto, B.A., da Silva, A.S., and Teixeira, J.S. Surveys: A brief survey of web data extraction tools. *ACM SIGMOD Record 31*, 2 (June 2002), 84–93.
10. Lerman, K., Minton, S., and Knoblock, C. Wrapper maintenance: A machine learning approach. *J. of AI Research 18* (Feb. 2003), 149–181.
11. Muslea, I., Minton, S., and Knoblock, C. A hierarchical approach to wrapper induction. In *Proceedings on Autonomous Agents* (1999), 190–197.
12. Srivastava J. and Cooley, R. Web business intelligence: Mining the Web for actionable knowledge. *J. on Computing 15*, 2 (2003), 191–207.

Dawn G. Gregg (dawn.gregg@cudenver.edu) is an assistant professor of information systems management in the Business School at the University of Colorado at Denver and Health Sciences Center. **Steven Walczak** (swalczak@carbon.cudenver.edu) is an associate professor of information systems management in the Business School at the University of Colorado at Denver and Health Sciences Center.