# An Ontology-based Semantic Extraction Approach for B2C eCommerce

Ali Ghobadi[1] and Maseud Rahgozar[2]
[1]Database Research Group, University of Tehran, Iran
[2]Control and Intelligent Processing Center of Excellence, University of Tehran, Iran

**Abstract:** *Although varieties of investigations have been done on human semantic interactions with Web resources, no advanced and considerable progresses have been achieved. It could be said that comparative shopping systems are the last generations of B2C eCommerce systems that connect to multiple online stores and collect the information requested by the user. In some cases, the information is extracted from the online store sites through keyword search and other means of textual analysis. These processes make use of assumptions about the proximity of certain pieces of information. These heuristic approaches are error-prone and are not always guaranteed to work. In this paper, we propose an ontology-based approach to extract the products' information and the vendors' price from their public Web sites' pages. Although most vendors on the Web present their products' information in HTML documents that are not semantic formats. However, our approach is based on understanding semantics of HTML documents and extracting the information automatically.*

**Keywords:** *Semantic correspondence, ontology, and schema.*

*Received November 28, 2008; accepted May 17, 2009*

## 1. Introduction

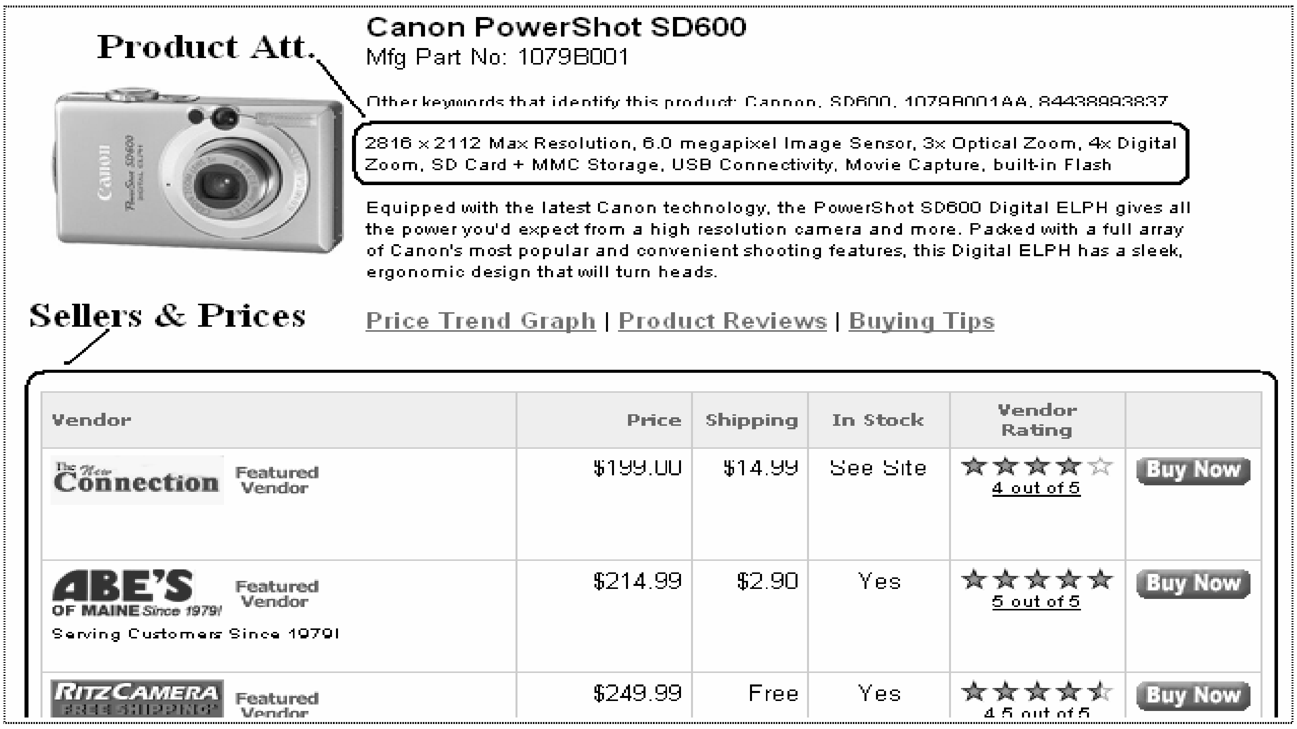
Electronic Data Interchange (EDI) between companies and Automatic Teller Machines (ATM) for banking were the first introductions of the electronic commerce (eCommerce). Introduction of the Web Browsers opened up a new age by combining open internet and easy user interface approaches [11].

Business-to-Consumer (B2C) eCommerce is the predominant commercial experience of Web users. A typical scenario involves a user's visiting one or several online shops, browsing their offers, selecting and ordering products. Ideally, a user would collect information about price, terms, and conditions (such as availability) of all or at least all major, online shops and then proceed to select the best offer. But manual browsing is too time-consuming to be conducted on this scale. Typically a user will visit one or a very few online stores before making a decision.

However, the evolution of B2C eCommerce has been formed through various generations. Last models of B2C eCommerce are comparative shopping catalogs. Models such as pricescan.com [17] and nextag.com [16] that visit several shops, extract product and price information, and compile a market overview. The comparative result obtained is then displayed in a tabular format in the user's browser. Their functionality is provided by wrappers, programs that extract information from an online store. One wrapper per store must be developed. This approach suffers from several drawbacks. First, it's necessary for these models to get access grant from vendors before to access their databases for retrieving any information. Since some vendors may not give access grant to their databases, their product information will not appear in the information provided by these models. Second, in some cases, the information is extracted from the online store site through keyword search and other means of textual analysis. This process makes use of assumptions about the proximity of certain pieces of information (for example, the price is indicated by the word price followed by the symbol *$* followed by a positive number). This heuristic approach is error-prone; it is not always guaranteed to work. Because of these difficulties only limited information is extracted. In addition, programming wrappers is time-consuming, and changes in the online store outfit require costly reprogramming.

We have proposed an ontology-based approach to resolve these problems. In this approach, products and price information are understood and extracted from Web pages of vendors' sites to build virtual catalog directly.

Most vendors issue their products information in HTML formats. Though HTML is used as the language for markup, even documents that conceptually follow a common schema are marked up for visual rendering purposes only, and in different ways due to diverse authorship and goals of the people writing these documents. This makes it more difficult to automate the processing of HTML documents in terms of semantic retrieval and integration. This problem is a little easier when data on the HTML pages is represented in HTML tables. Data in HTML tables is mostly structured, but we usually do not know the structure in advance. Thus, we cannot directly query for data of interest. In addition, Web pages are often cluttered with some other contents like advertisements, navigation-panels, copyright notices etc., surrounding the main content of the Web page. Therefore, extracting structured data from Web sites is not a trivial task. Suppose the HTML pages of Figures 1 and 2 retrieved from two comparison shopping sites. These pages show the information about a digital camera named Canon Powershot SD600 and its sellers. A segment of these pages contains the product information (i.e., attributes and values) and another segment (represented in HTML table) contains information of sellers and their price about this article.



Canon PowerShot SD600
Mfg Part No: 1079B001

Other keywords that identify this product: Cannon, SD600, 1079B001AA, 8443899383?

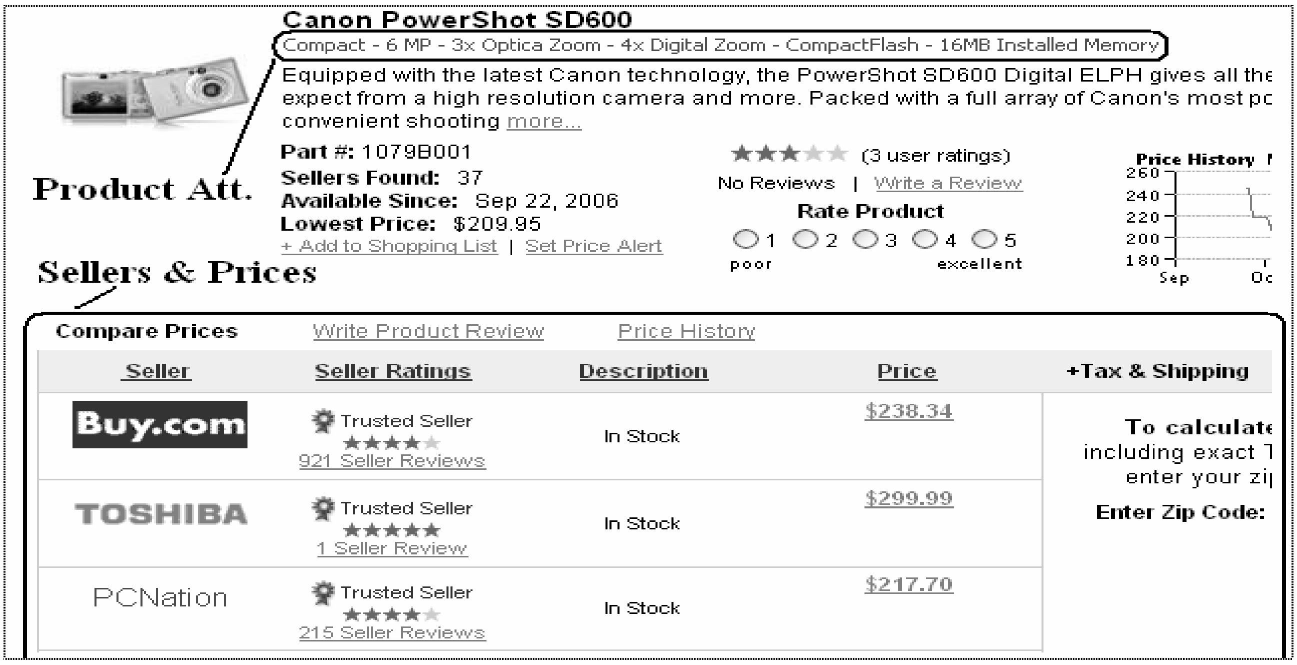
2816 x 2112 Max Resolution, 6.0 megapixel Image Sensor, 3x Optical Zoom, 4x Digital Zoom, SD Card + MMC Storage, USB Connectivity, Movie Capture, built-in Flash

Equipped with the latest Canon technology, the PowerShot SD600 Digital ELPH gives all the power you'd expect from a high resolution camera and more. Packed with a full array of Canon's most popular and convenient shooting features, this Digital ELPH has a sleek, ergonomic design that will turn heads.

Price Trend Graph | Product Reviews | Buying Tips

| Vendor | Price | Shipping | In Stock | Vendor Rating | |
|---|---|---|---|---|---|
| The New Connection Featured Vendor | $199.00 | $14.99 | See Site | ★★★★☆ 4 out of 5 | Buy Now |
| ABE'S OF MAINE Since 1979! Featured Vendor Serving Customers Since 1979! | $214.99 | $2.90 | Yes | ★★★★★ 5 out of 5 | Buy Now |
| RITZ CAMERA FREE SHIPPING Featured Vendor | $249.99 | Free | Yes | ★★★★½ 4.5 out of 5 | Buy Now |

Figure 1. A digital camera and its sellers' information from www.pricescan.com [17].



Canon PowerShot SD600
Compact - 6 MP - 3x Optica Zoom - 4x Digital Zoom - CompactFlash - 16MB Installed Memory
Equipped with the latest Canon technology, the PowerShot SD600 Digital ELPH gives all the expect from a high resolution camera and more. Packed with a full array of Canon's most pc convenient shooting more...

Part #: 1079B001
Sellers Found: 37
Available Since: Sep 22, 2006
Lowest Price: $209.95
+ Add to Shopping List | Set Price Alert

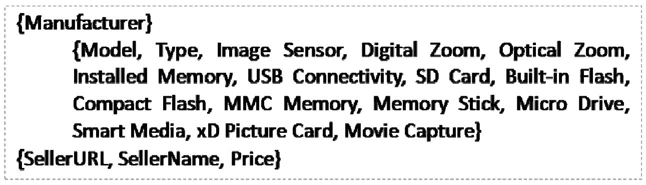
★★★☆☆ (3 user ratings)
No Reviews | Write a Review
Rate Product
○1 ○2 ○3 ○4 ○5
poor excellent

Compare Prices Write Product Review Price History

| Seller | Seller Ratings | Description | Price | +Tax & Shipping |
|---|---|---|---|---|
| Buy.com | Trusted Seller ★★★★☆ 921 Seller Reviews | In Stock | $238.34 | To calculate including exact T enter your zi Enter Zip Code: |
| TOSHIBA | Trusted Seller ★★★★★ 1 Seller Review | In Stock | $299.99 | |
| PCNation | Trusted Seller ★★★★☆ 215 Seller Reviews | In Stock | $217.70 | |

Figure 2. Same digital camera and sellers' information from www.nextag.com [16].

The goal of the information extraction is to find out a semantic correspondence between one or more source schemas and a target schema. For example, suppose that we are interested in viewing and querying digital cameras' information through the target schema represented in Figure 3.



**{Manufacturer}**
**{Model, Type, Image Sensor, Digital Zoom, Optical Zoom, Installed Memory, USB Connectivity, SD Card, Built-in Flash, Compact Flash, MMC Memory, Memory Stick, Micro Drive, Smart Media, xD Picture Card, Movie Capture}**
**{SellerURL, SellerName, Price}**

Figure 3. Target schema of digital cameras information.

In the simplest form, semantic correspondence is a set of mapping elements, each of which binds a concept or an attribute in a source schema to a concept or an attribute in a target schema or binds a relationship among concepts in a source schema to a relationship among concepts in a target schema. Such simple forms, however, are rarely sufficient, and researchers thus use queries over source schemas to form concepts/attributes and relationships among concepts to bind with target concepts/attributes and concepts relationships. We must resolve the following main issues while performing the information extraction process:

- Locating segments of interest: While it is easy for a human to locate the data segments of interest on a HTML page, doing the same action programmatically on a HTML page, is not generally a trivial task.
- Semantic parsing of the segments: It is also easy for a human and not a program to parse the data segments and determine their meanings, independent from their viewing formats.
- Identifying semantic correspondence of the terms: Some terms appeared on the similar pages have an identical meaning. Again, identifying this semantic correspondence between two terms is a non-trivial task and needs some initial knowledge introduced by a human being.

We present the details of our approach in the remainder of the paper as follows. After a short overview of the related work in section 2, section 3 describes a model for representing ontology in our virtual catalog. Section 4 explains how we locate segments of interested data and extract their information. In section 5, we report the experiments we conducted involving digital camera advertisements on the Web. Finally, section 6 presents the conclusion of this work.

## 2. Related Work

HTML document wrappers are in some point of views related to our approach of generating virtual catalogs for comparative shopping. Several types of document wrappers have been suggested. YAT [5] and WysiWyg Web Wrapper Factory (W4F) [19] are manual wrappers which require users to specify exactly how to extract data from HTML documents through some wrapping languages. TSIMMIS [10] allows users to generate wrappers according to declarative specifications. The specification part states where the data of interest is located on the HTML pages.

A number of semi-automatic approaches [9, 12, 13, 14, 18] to wrapper generation use the idea of learning by examples. The user, first, labels a number of examples of extracted data, and the software then generates extraction rules based on these examples. XWRAP [15] is a semi-automatic wrapper-generator that builds on the structural meaning of specific HTML tags (e.g., headings and tables) and how they are used for data layout. Heuristics are used to determine the parent-child relationships between data items, for instance table names, field names, and values.

Related to automatic wrapper generation, several systems [2, 6] deserve special discussion because they support fully automatic generation of wrappers. These systems examine the structures of sample Web pages and automatically generate a template for the data contained in these pages.

All of the above mentioned systems are based purely on syntax and do not take advantage of the semantics of the specific domain. However, our approach is based on understanding and extracting the semantics of HTML documents. Related to semantic understanding and extraction, we can mention AUTOBIB [8], an approach that has been proposed to automate extraction of bibliographic information on the Web that (like some other automated extraction systems) bootstraps itself with an initial knowledge of bibliographic records. In [1, 3, 4, 7], some ontology and knowledge based approaches have been introduced for question answering systems that can extract concepts and their relations based on human plausible reasoning.

## 3. Ontology Representation

Ontology is defined as concepts, their relationships, and concepts instances of specific domain. Concepts and relationships are identified and defined by domain experts. When we apply the ontology to a Web page, the objects and relationships are identified and associated with concepts and relationships in the ontology's conceptual-model. Thus the strings on a

Web page are recognized and understood in terms of the target schema.

The ontology's conceptual-model of the system is represented by a semantic network. Figure 4 depicts a partial view of semantic net for domain of digital cameras. Each node in the semantic network may represent a concept, a concept attribute, or an attribute value. Each relation along with its connected nodes forms a logical statement. For instance we can enumerate several statements in Figure 4: ATT (Digital Camera) = {Model, Image Sensor …}, VAL (Model) = {Canon Powershot SD600, Canon Powershot SD550 …} and so on.

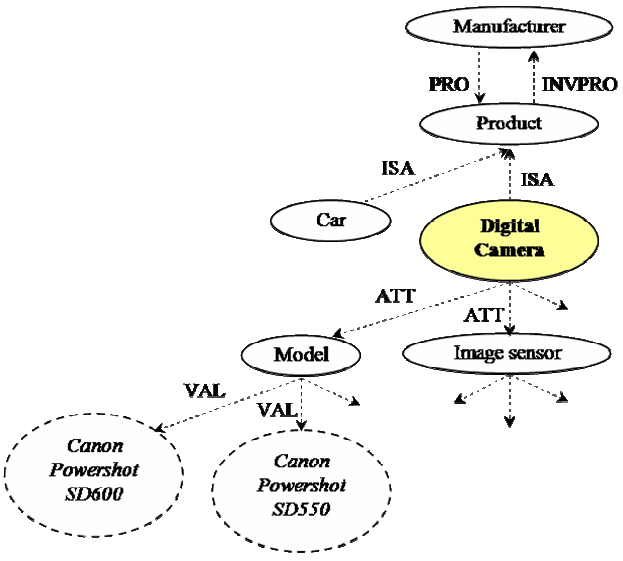


Figure 4. Ontology's conceptual-model for domain of digital cameras (partial).

We have defined 10 relations between concepts in this phase. Most of them have been used to describe UML associations. The main relations defined in our ontology's conceptual-model are PRO, OFR, OFD, ATT, VAL, SIM, and ISA. PRO means a manufacturer produces a product. OFR has been used to define offer relations between sellers and costs. OFD means a price is offered by a seller. A concept has some attributes which has been defined by ATT relations between concepts and their attributes. For some attributes, we have defined some values by VAL relations. SIM which is the core of ontology's conceptual-model has been used to define semantic correspondence between concepts or values. Finally, we have used ISA to define kind of relations between concepts.

Some values (displayed by dashed ellipses in the ontology's conceptual-model) have been defined for some attributes in the ontology. These values are used as an initial knowledge to bootstrap the system. Other values will be extracted from Web pages. Therefore, system's ontology will be enriched while performing the information extraction process.

## 4. Information Extraction

The task of information extraction is to find a semantic correspondence between one or more source schemas and a target schema. For example, consider the target schema shown in Figure 3 and the source schemas shown in Figure 1 or Figure 2 for digital cameras' domain. We must extract the information of digital camera shown in Figure 1 or Figure 2 in terms of attribute/value, combine and refine these terms, and map to the target schema. In this process, we should attend the three mentioned problems of locating segments of interest, semantic parsing of the segments, and identifying semantic correspondence of the terms.

In our approach of information extraction, the extraction process of product information is different from the sellers' information extraction process. Therefore, we describe these two parts of information extraction in the two following subsections.

### 4.1. Product Information Extraction

First, the segment of HTML page that contains the product information must be found. Since, different tags may be used for displaying the product information on the Web pages, extraction process of product information cannot be based on any particular tag. In our approach, we use a tag-independent method to find the interesting segments while we suppose that HTML pages are well formatted. For badly formatted HTML documents, a structure checker tool like HTML Tidy [21] (a free utility from W3C) can detect missing and mismatching end tags.

Some product attributes may have values in the ontology but not for some others. The process of information extraction for these two groups of product attributes would be different. Thus, we define two groups of attributes as follows:

- Static attributes: product attributes which have values in the ontology. For example, attribute Model of digital camera has a number of values in the ontology such as Canon Powershot SD600.
- Variable attributes: product attributes which have no values in the ontology and their values must be extracted from Web pages. For example, attribute optical zoom of digital camera has no value in the ontology. Therefore, for each digital camera, this information must be extracted from the corresponding Web pages.

For each static attribute of a product, its values (retrieved from ontology) must be searched on the Web pages and for each variable attribute, it must be searched on.

Product title (attribute model) is usually displayed in a separated tag followed by the other characteristics of product on the Web pages. Therefore, we describe our process of information extraction for a product title and the other attributes in the following two subsections.

### 4.1.1. Finding and Extracting Product Title

Using the ontology, we search the attributes of the Model of product on the Web page. If no attributes of the Model are found, the Web page would be discarded. Model usually contains the manufacturer name and then the product name. The first part of the Model is assigned to the Manufacturer and the remainder part is assigned to the product attribute (i.e., Model attribute) respectively. Therefore, the target schema shown in Figure 3 updated as shown in Figure 5.

**{(Manufacturer: Canon)}**
**{(Mode: Powershot SD600), (Type: Null) …}**
**{(SellerURL: Null), (SellerName: Null), (Price: Null)}**

Figure 5. Target schema after extracting product title.

In some Web pages, attributes of the Model appear in different orders. For example, Canon Powershot Rebel XT digital camera in some Web pages is displayed in this order but in other cases displayed in Canon Powershot XT Rebel. Since Model attributes of products are only different in order of words, hence we can consider all orders of the words (except the first word, because this word shows manufacturer name and is always the first word of product model) as attributes of the Model. If Model has been composed from n words, all orders of (n-1) remainder words (i.e. (n-1)!) are considered as attributes of the Model and must be searched on the Web pages.

### 4.1.2. Finding and Extracting Other Attributes of a Product

In this step of process, the segment which contains the product information must be found. All of the static and variable attributes of product would be considered to find the segment. There are some attributes with values of Yes/No that if the value is Yes then the attribute appears on the segment of product information, otherwise it does not appear. For example, if a digital camera has capability of USB connectivity, the attribute of USB Connectivity for that digital camera will be displayed on the segment of product information. We cannot say that a specific segment of the Web page is not an interesting segment, if these attributes (part or all of them) don't appear on that segment. Therefore, in another classification, we can group the attributes of a product in two sets, as follows:

A= {Variable attributes with value of Yes/No}

B= {(Total attributes - A - Values of *Model* attribute)}

Static attributes have no value of *Yes/No*. For the set B, all values of static attributes must be retrieved from the ontology and added to the set. Values of Model attributes are discarded because they have already been extracted.

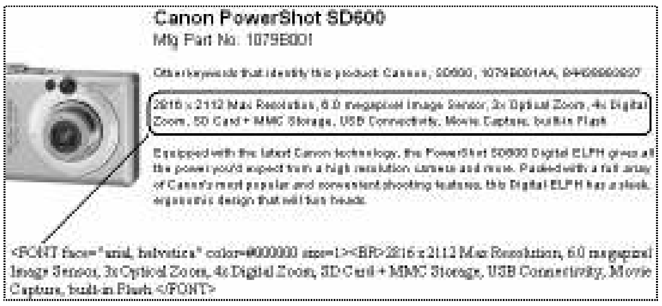
Now, we can find the interesting segment of product information by searching the elements of the sets A and B on the Web page. In other words, for each tag, the elements of the sets A and B are searched on the tag and the probability of this tag as an interesting tag of product information ($P(t_i)$) is computed as follow:

$$P(t_i) = \frac{1}{l+k-f}\Big[\sum_{j=1}^{l+n} Found(x_j \in B, t_i) + \sum_{c=1}^{k-n} Found(y_c \in A, t_i)\Big] \qquad (1)$$

Where: $l$ and $k$ are the number of static and variable attributes of product respectively; $n$ is the number of variable attributes with no value of *Yes/No*; $x_j$ and $y_c$ are attributes of the product with and without the value of *Yes/No* respectively; $f$ is the average number of variable attributes with value of *Yes/No* that product doesn't has the corresponding capability.

The tag with maximum $P$ is the segment of interest. Some tags might have the same values of $P$ (i.e., parent tags in a tree structure); in these cases we keep the nested tags (i.e., child tags in the tree) and discard the others. We consider 0.1 as minimum value of $P$. The value less than minimum value shows that the tag (segment) has too little information about product and can be discarded.

Using this method, the segment of interest for product information of the Web page shown in Figure 1 is identified as shown in Figure 6.


Canon PowerShot SD600
Mfg Part No: 1079B001

Other keywords that identify this product: Canon, SD600, 1079B001AA, [illegible]

2816 x 2112 Max Resolution, 6.0 megapixel Image Sensor, 3x Optical Zoom, 4x Digital Zoom, SD Card + MMC Storage, USB Connectivity, Movie Capture, built-in Flash

Equipped with the latest Canon technology, the PowerShot SD600 Digital ELPH gives all the power you'd expect from a high resolution camera and more. Packed with a full array of Canon's most popular and convenient shooting features, this Digital ELPH has a sleek, ergonomic design that will turn heads.

<FONT face="arial, helvetica" color=#000000 size=1><BR>2816 x 2112 Max Resolution, 6.0 megapixel Image Sensor, 3x Optical Zoom, 4x Digital Zoom, SD Card + MMC Storage, USB Connectivity, Movie Capture, built-in Flash </FONT>

Figure 6. Segment of product information from the page shown in Figure 1.

After identifying an interesting segment, we extract the product information on this segment. There are three steps for this state:

- For each variable attribute with Yes/No value, if the name (or a semantic correspondence of the name) is found on the segment, the value of *Yes* is assigned to the corresponding element of the product object in the target schema.

- For each variable attribute without Yes/No value, if the name (or a semantic correspondence of the name) is found on the segment, the former string (before the found point) is considered as a value for this attribute and assigned to the corresponding element of the product object in the target schema.
- Static attributes' values found on the segment are assigned to the corresponding element of the product object in the target schema.

For example, after extracting the product information on the Web page shown in Figure 1, the target schema for product object is updated as shown in Figure 7.

{(Model: Canon Powershot SD600), (Type: Null), (Image Sensor: 6.0), (Digital Zoom: 4x), (Optical Zoom: 3x), (Installed Memory: Null), (USB Connectivity: Yes), (SD Card: Yes), (Built-in Flash: Yes), (Compact Flash: Null), (MMC Memory: Yes), (Memory Stick: Null), (Micro Drive: Null), (Smart Media: Null), (xD Picture Card: Null), (Movie Capture: Yes)}

Figure 7. Target schema for product object after extracting product information from the page shown in Figure 1.

This product information will be more complete during the extraction process from another Web page as shown in Figure 2. For each Web page that contains information about this product, we should perform the searching and the extraction of product information.

## 4.2. Sellers' Information Extraction

As shown in Figures 1 and 2, the sellers' information (i.e., seller's URL, price, etc.,) are almost displayed in the HTML tables, after the product information segment on the Web pages. In some cases, there are columns in these tables on different Web pages, that have the same semantic with different titles (e.g., first columns of sellers' tables in Figure 1 and Figure 2 are identical but their titles are Vendor and Seller respectively). We resolve this issue through the ontology by extracting semantic correspondence relationships between concepts (e.g., Vendor has a semantic correspondence relationship with Seller).

Each table block in HTML documents uses the same specific tags (i.e., <TABLE>, <TH>, <TR>, <TD>) that we can exploit during our extraction process. However, to perform the sellers' information extraction process four steps must be followed:

1. Finding the table of interest: The sellers' information table almost has the columns with specific titles (e.g., Seller, Price, Shipping, etc.,). We use these titles and their synonyms (via the ontology) to find the table of interest on the Web page. The nested table with maximum probability *P* is identified as the interesting table, using the same approach as in subsection 4.1.
2. Deleting the columns of no interest: We are interested in two columns which contain sellers' URLs and their price (and sellers' names). Other columns must be deleted from the table block.
3. Deleting the extra tags: From logical view, all the HTML tags except <a href=..> and <img src=..> can be considered unnecessary and hence removed [20]. The tags <TABLE>, <TH>, <TR>, <TD>, and their end tags are considered as delimiters in table block and will not be removed. We can also remove all extra information in the tags (e.g., font styles, font size, etc.,).
4. Extracting Information and mapping to target schema: After performing the steps above, two remained columns of each row (i.e., placed in a <TR></TR> tag) will contain a seller's URL, name, and price with no extra information. From the first column, sellers' URLs and names are extracted respectively. The second column contains sellers' price that can be extracted as product price and mapped to the corresponding element of seller object in the target schema.

Figure 8 shows the target schema after extracting sellers' information of two first rows for Canon Powershot SD600 digital camera from page shown in Figure 1.

{(Manufacturer: Canon)}
{(Model: Canon Powershot SD600), (Type: Compact), (Image Sensor: 6.0), (Digital Zoom: 4x), (Optical Zoom: 3x), (Installed Memory: 16MB), (USB Connectivity: Yes), (SD Card: Yes), (Built-in Flash: Yes), (Compact Flash: Null), (MMC Memory: Yes), (Memory Stick: Null), (Micro Drive: Null), (Smart Media: Null), (xD Picture Card: Null), (Movie Capture: Yes)}
{(SellerURL: http://www.ritzcamera.com), (SellerName: Ritzcamera), (Price: $349.99)}
{(SellerURL: http://www.buydig.com), (SellerName: Buydig), (Price: $287.00)}

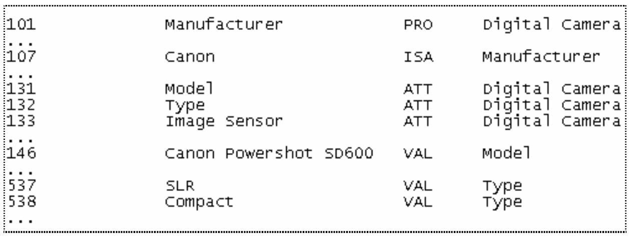
Figure 8. Target schema after extracting sellers' information (partial).

## 5. Preliminary Experiments

This section explains our experiments conducted to verify the validity of our approach. First we describe the process in which the underlying ontology was created and implemented. Then we present the evaluation of our proposed approach.

### 5.1. Creation of the Ontology

Determining and defining the requisite ontology for ontology-based systems is a cumbersome task. Classical expert systems required years to be crafted by perfect and highly skilled knowledge engineers. For our system, some digital camera domain experts were asked to fill simple templates with triple relations they were familiar with. The basics of system's ontology-conceptual model were explained to them in advance to make them understand what types of relations were needed. Finally, we implemented the concepts and their relationships (defined by domain experts) in the text format that its partial view has been shown in Figure 9.

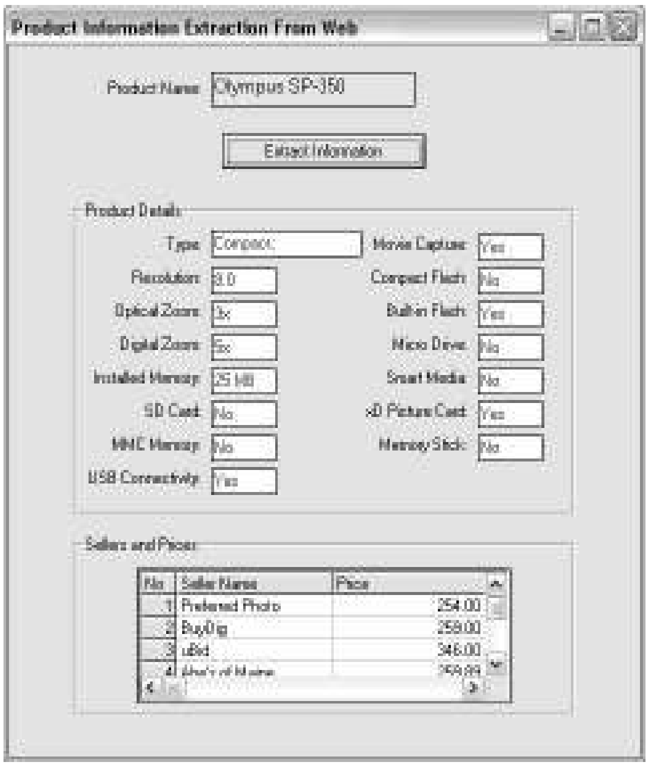


```
101    Manufacturer           PRO   Digital Camera
...
107    Canon                  ISA   Manufacturer
...
131    Model                  ATT   Digital Camera
132    Type                   ATT   Digital Camera
133    Image Sensor           ATT   Digital Camera
...
146    Canon Powershot SD600  VAL   Model
...
537    SLR                    VAL   Type
538    Compact                VAL   Type
...
```

Figure 9. Digital camera domain ontology (partial).

## 5.2. Evaluation of Approach

We have tested our approach of virtual catalog generation by developing a tool. All components of this tool have been developed in Java. First, we retrieved 140 Web pages (i.e., product & sellers' information) in domain of digital cameras manually. Typical pages have been shown in Figure 1 and Figure 2 from pricescan.com and nextag.com respectively. Then, we applied these pages to our system and asked 70 questions. We use the recall and precision measures to evaluate the performance of our system for information extraction. Recall and precision were obtained 0.91 and 0.94 respectively. Figure 10 shows the typical output of our virtual catalog generated from some Web pages.



Product Name: Olympus SP-350

| | | | |
|---|---|---|---|
| Type: | Compact | Movie Capture: | Yes |
| Resolution: | 8.0 | Compact Flash: | No |
| Optical Zoom: | 3x | Built-in Flash: | Yes |
| Digital Zoom: | 5x | Micro Drive: | No |
| Installed Memory: | 25 MB | Smart Media: | No |
| SD Card: | No | xD Picture Card: | Yes |
| MMC Memory: | No | Memory Stick: | No |
| USB Connectivity: | Yes | | |

| No | Seller Name | Price |
|---|---|---|
| 1 | Preferred Photo | 254.00 |
| 2 | BuyDig | 258.00 |
| 3 | uBid | 346.00 |
| 4 | [illegible] | [illegible] |

Figure 10. Typical output of virtual catalog generated from web pages.

## 6. Conclusions

We proposed a new ontology-based approach for generating virtual catalogs. Although the problem has been studied by several researchers, existing techniques are limited to specific heuristics and databases. An effective method is proposed to locate the interesting segments of information in a Web page and extract the information automatically. We proposed a probabilistic method that can correctly identify the data segments. We avoid the tag-oriented approaches to make the solution as generalized as possible. An ontology provided by domain experts is used for identification of the interesting segments and information extraction processes.

## References

[1] Angele J., Monch E., Oppermann H., Staab S., and Wenke D., "Ontology-based Query and Answering in Chemistry: Ontonova@Project Halo," *in Proceedings of the 2nd International Semantic Web Conference*, Berlin, 2003.

[2] Arasu A. and Garcia-Molina H., "Extracting Structured Data from Web Pages," *in Proceedings of the ACM SIGMOD International Conference on Management of Data*, California, 2003.

[3] Chung H., Song Y., Han K., Kim S., Yoon D., Lee J., and Rim H., "A Practical QA System in Restricted Domains," *in Proceedings of the ACL Workshop on Question Answering in Restricted Domains*, Spain, pp. 566-568, 2004.

[4] Clark P., Thompson J., and Porter B., "A Knowledge-based Approach to Question Answering," *in Proceedings of AAAI'99 Fall Symposium on Question-Answering Systems*, pp. 43-51, 1999.

[5] Cluet S., Delobel C., Siméon J., and Smaga K., "Your Mediators Need Data Conversion!," *in Proceedings of ACM SIGMOD*, pp. 177-188, 1998.

[6] Crescenzi V., Mecca G., and Merialdo P., "RoadRunner: Towards Automatic Data Extraction from Large Web Sites," *in Proceedings of the International Conference on Very Large Data Bases*, pp. 109-118, Italy, 2001.

[7] Darrudi E., Rahgozar M., and Oroumchian F., "Human Plausible Reasoning for Question Answering Systems," *in Proceeding of Advances in Intelligent Systems Theory and Applications*, Luxembourg, 2004.

[8] Geng J. and Yang J., "AUTOBIB: Automating the Extraction of Bibliographic Information on the Web," *Computer Journal of International*

*Database Engineering and Application Symposium*, vol. 3, no. 2, pp.155-157, 2004.

[9] Golgher B., Laender F., Da S., and Ribeiro-Neto A., "An Example-based Environment for Wrapper Generation," *in Proceeding of the 2nd International Workshop on the World Wide Web and Conceptual Modeling*, pp. 152-164, 2000.

[10] Hammer J., Garcia-Molina H., Cho J., Crespo A., and Aranha R., "Extracting Semistructured Information from the Web," *in Proceeding of the Workshop on Management of Semistructured Data*, pp. 18-25, 1997.

[11] Kalakota R. and Whinston B., *Electronic Commerce, A Manager's Guide*, Addison Wesley Professional, 1997.

[12] Knoblock A., Lerman K., Minton S., and Muslea I., "Accurately and Reliably Extracting Data from the Web: A Machine Learning Approach," *Computer Journal of IEEE Data Engineering Bulletin*, vol. 23, no. 4, pp. 33-41, 2000.

[13] Kushmerick N., "Wrapper Induction: Efficiency and Expressiveness," *Computer Journal of Artificial Intelligence*, vol. 118, no. 1, pp. 15-68, 2000.

[14] Kushmerick N., Weld S., and Doorenbos B., "Wrapper Induction for Information Extraction," *in Proceeding of the International Joint Conference on Artificial Intelligence*, Japan, pp. 729-737, 1997.

[15] Liu L., Pu C., and Han W., "XWRAP: An XML Enabled Wrapper Construction System for Web Information Sources," *in Proceedings International Conference on Data Engineering*, California, pp. 22-26, 2000.

[16] Price Comparisons, Product Reviews in NexTag, http://www.nextag.com, Last Visited 2009.

[17] Price Comparisons, Product Reviews in Pricescan, http://www.pricescan.com, Last Visited 2009.

[18] Ribeiro-Neto A., Laender F., and Da Silva S., "Topdown Extraction of Semi-Structured Data," *in Proceeding of the 6th Symphony on String Processing and Information Retrieval*, Mexico, pp. 176-183, 1999.

[19] Sahuguet A. and Azavant F., "Looking at the Web through XML Glasses," *in Proceeding of the 4th IFCIS International Conference on Cooperative Information Systems*, pp. 148-159, 1999.

[20] Seo H., Yang J., and Choi J., "Knowledge-based Wrapper Generation by Using XML," *In IJCAI-Workshop on Adaptive Text Extraction and Mining (ATEM2001)*, Seattle, USA, pp.1-8, 2001.

[21] Structure Checker Program, Tidy Tool for HTML Correction, http://www .w3.org /People/ Raggett/tidy, Last Visited 2009.

**Ali Ghobadi** received his BSc degree on software engineering from University of Tehran and in 2006 he received his MSc degree on software engineering from Islamic Azad University in Tehran. He has joint the academic community of Database Research Group of University of Tehran on September 2005, after over 8 years of professional career in IT consultancy as an IT project manager and senior consultant. His current fields of interests are automatic ontology building, Agent-based and Intelligent Systems, natural language processing, question answering systems, service oriented software development, and service oriented enterprise architecture.

**Maseud Rahgozar** has joint the academic community of Tehran University on September 2000, after over 19 years of professional career in French software house companies as R&D manager, senior consultant, etc. His current fields of interests are database systems, designing CASE tools for object oriented programming, designing CASE tools for database normalization and modernization of legacy applications and their environments. In 1979, he received his BSc degree on electronics engineering from Sharif University of Technology in Tehran, and in 1987 he received his PhD on database systems from Pierre and Marie Curie University in Paris.