Realization Mechanism of Intelligent Comparison-Shopping Systems based on Web Information Extraction

Xun Wang, Haiwei Jin, and Zhenyue Chen

College of Computer & Information Engineering, Zhejiang Gongshang University, Hangzhou, China

Summary

Based on DOM, two-stage work pattern of information extraction and the conception of page information unit has been proposed. PIU is extracted and classified by classified algorithm, and information is extracted in PIU. It is implemented finally that the key information of the online goods is extracted. A intelligent Comparison Shopping System based on Web Information Extraction is implemented with analysis and design about flow and frame structure. It is showed that the algorithm is steady and has higher Recall and Precision with the sample testing.

Key words:

Comparison-shopping, Information extraction, Intelligent business, Document object model

Introduction

With the development and spread of Internet, the mode of online shopping has been accepted by common people and its scale becomes bigger and bigger. But it is not easy for ordinary customers to search through different web shops, compare the goods and prices, and make decisions in the end. And fraudulent conduct occurs on the web now and then. Intelligent Comparison-Shopping System (ICSS) can be used to gather the product information from the Internet and compare that information according to customers' demand. ICSS can easily locate the products that meet users' requirements, evaluate the products and web shops and influence customers' purchase and confidence through the evaluation results, thus aiding customers' in gathering data and comparing products with qualified intelligent service.

Intelligent Comparison Shopping Systems based on traditional information retrieve techniques can be used to gather the product information from the Internet and compare that information according to customers' demand. The key technology is specific Web information extraction. But the tag of HTML just informs the browser how to display the information without any semantics, so the web pages written in HTML is only suitable for browsing after browser resolution, while being unsuitable for data communication processed by computers [1].

Under web environment, normally it is the wrapper that is responsible for extracting the information contained in HTML documents and transforming that information into data structure storage that can be further processed. Generating the wrapper has become the research hot spot of intelligent information processing domain [1]. In recent years, many famous international conferences such as SIGMOD, VLDB, ICDE and etc. have also published several relevant articles [2-4]. Document [5] has categorized those methods from different point of views, while most of those methods bases on different rules to generate wrapper. According to producing the difference of rules, wrapper can be roughly divided into the following two categories:

(1) Developing special grammars to illustrate the distribution of data in HTML pages and data extraction[6-8] ;

(2) Applying induction technique in generating extraction rules automatically or semi-automatically [9-10].

Document [6] designs a declaration advanced language in writing templates to define extracting rules, and describes the queries wrapper would receive and the objects would be returned. Once the query matches template, the action associated with that template would be activated, transforming the query of integrated system into query of data source oriented and translating the results into the form that integrated system could discern. This method must be adapted to the designing requirements of the integrated system, so it could not be generally applied. Document [8] regards the Web as a large distributed database system composed by non-structured and semi-structured files, so constructing hypertext relation view and using query languages could fulfill queries, but the results of queries have flaws that the granules are too big and are not precise enough.

Document [9] introduces the research on incorporating induction into the wrapper, so the wrapper could analysis Web data source efficiently and has enough capability to represent real data source. But it handles mainly table-layout data source with much limitation. Document [10] uses Landmark from induction learning method to represent extraction rules, including the beginning rule and ending rule, which is especially suitable for extracting information delimited by the start tag and the end tag, but unsuitable for other complex information extraction.

After analyzing the drawbacks of various wrappers and features of comparison-shopping systems, this paper

Manuscript received June 5, 2006.
Manuscript revised June 25, 2006.

introduces the tree structure path expression of document object model to position the information in HTML pages to be extracted, and proposes a two-stage information extraction algorithm, and designs an Intelligent Comparison-Shopping System(ICSS) based on this algorithm.

2. Overview of ICSS

Intelligent comparison-shopping system based on Web orients towards specific industries, extracting information from web pages located by professional search engine and returning the query results while meeting various detailed demands of the products that user is interested in. The main object of the system is to extract and regroup information of the web pages, while the data source is being provided by professional search engine. The detailed implementation technique of professional search engine will be discussed in other papers.

2.1 ICSS system workflow

Since the forms of product information in web vary greatly and the structures embedded in web pages differ, it is quite difficult to induce data pattern for all the web pages. In the application of ICSS, the types of information to be extracted are relatively fixed, focusing mainly on extraction of information related to products while rejecting irrelevant information contained in the web pages. With the research on sampling web pages, following features have been spotted:

(1) In the web pages that contain product information, most product information is densely located under certain web page tags, such as <table>、<tr>、<td>、<th>, etc,. So these four tags could be considered as key basis for extracting product information.

(2) If the web page contains information about more than one product item, then the structures of these information are usually organized in similar mode or similar classified mode.

(3) The components of product information are usually in the same hierarchy, being arranged in parallel. In the DOM tree structure, the relationship then can be drawn as sibling nodes connecting to parent node.

According to the research of above sampling web pages, the whole extraction process could be divided into two stages. At the first stage, extracting the page information unit (PIU) that contained in web pages and classifying PIU according to the different structure

The task of professional search engine is to retrieve web page information and update web page database according to the query key words, its specific operating principles would be presented in other papers.

features when these PIU appear in the web pages. At the second stage, applying different extraction rules to extract product information from classified PIU information and knowledge domains.

As analyzed above, the workflow is drafted as follows:

(1) Establishing domain knowledge database for different industries and products.

(2) The user can submit specific query through the interface.

(3) With the key query words from the user, the professional search engine is activated to primarily gather information and update the content of web database.

(4) The web analyzer is activated, trying to classifying the web pages according to the key query words and extracting PIU into database for further use.

(5) With different PIU structure categories, invoking different extraction rules to extract product information and keep those information in database for further use.

(6) Regrouping and reorganizing the product information and returning the result web page to user.

2.2 ICSS system frame

From input of key words to return of results, ICSS system can be divided into several modules: user interface, domain knowledge database, Web page analyzer and information extraction module. The overall architecture is shown in Fig. 1.

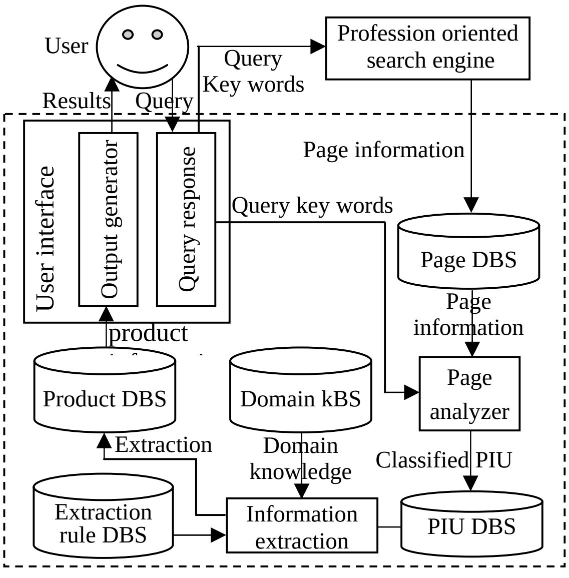


Fig.1. The overall architecture

User interface is in charge of accepting user query. On the one hand it transfers the key words to search engine for updating web pages, and on the other hand it transfers the key words to web page analyzer for analyzing web page. Output generator will group the information retrieved from product database after the analyzing and information extracting process into uniform output format.

Web page analyzer tries to obtain the query key words from the user interface, retrieves the web page information from database, extracts the relevant PIU of the products queried, classifies PIU with respect to different web structure and stores the classified results into PIU database for later use of information extraction.

Domain knowledge database, which contains the knowledge about one specific domain, supports the information extraction and should be set up according to the standard of that domain and be improved continuously in functioning.

Information extraction module tries to retrieve categorized PIU from PIU database and relational domain knowledge from domain knowledge database. The module applies different extraction rules to extract product information according to the key query words and classified PIU information, while the product information obtained is stored into the product information database

Web page analysis

3.1 Page Information Unit

Page information unit, which is separated from the web page, contains the corresponding code of the product information needs to be extracted. So PIU should possess following features:

(1) PIU should contain at least one piece of complete product information. Completeness refers that the product information elements appeared within key query words and other relevant product information are totally included. Product information elements will vary according to different product category and system requirements. In this paper, we use steels as example, whose elements always include designation, specifications, material quality, price, producer, vendor, transaction site and mode, etc,.

(2) PIU contains some web page tags, whose various structural forms can classify PIU into different categories. Since the tags in PIU are closely correlated with the hierarchical layout of product information, the number of categories of corresponding DOM tree structures is relatively fewer.

(3) PIU begins from the web page start tag and finishes at the end tag, with the contents enclosed between the pair of tags. Accordingly, in DOM tree structure, tag node is considered as root node, while text node is treated as leaf node.

3.2 PIU classifying

Information extraction rules need to be applied to PIU of various web page formats and web structures. So it is necessary for one PIU corresponds to various classifying rules from different points of view, and delimiting the structures and categories of product information contained. In database, this multiple classifying rule represents several attributes. Detailed classifying rules are listed as follows:

(1) classifying by PIU product categories

The source data of ICSS comes from professional search engine, and the product categories could be obtained from the relevant source data. Classifying by PIU product categories can make the system select rules from corresponding domain knowledge database.

(2) classifying by web page types

Though many web pages such as html、asp、jsp、php besides XML are of different code and grammar at the server side, when those pages are loaded at the client side, there are not much differences at all. Since all web page tags are without semantics, those tags could be grouped into the same category and will not influence the extraction of information

Since XML pages include semantics and are of different code structures from those four above mentioned web pages, so it should design separately a method to extract information. Since research has discovered that the usage of XML pages is not extensive enough in native commercial web sites and the sampling pages are rare, so this paper will not take it into consideration.

(3) classifying by PIU structural features

Through extensive research on huge volumes of web pages, the web pages containing product information can be classified into: list type, table type, mixed type of list and table and a few irregular types. So the PIU extracted from web pages can be categorized into two types: list and table types. Further study on those two types of page code has shown that they can be further divided according to their structural features.

PIU of list type can be divided further into those with header information and those without header information, while the contents of those two PIU list types are all enclosed between tags of <table> and </table>. The text nodes of corresponding DOM trees usually contain few materials and are in uniform format, and every column represents one element of the product information, while every line represents the entire information of one product. PIU of list type with header information can analysis and distinguish the contents of every column by analyzing the header information, while PIU of list type without header information relies on the domain knowledge to distinguish the product elements of every column.

With the corresponding DOM tree for the PIU of table type, the text nodes are more complex. Since web page of table types cannot represent product information elements in uniform identification using header information, text nodes usually contain some interpretations explaining the information contained. For example, the designation: XXX, specification: XXX, etc,. The text node of the PIU of table type can also include more than one product information elements, so trying to distinguish the product information of PIU of table type usually demands additional product domain knowledge.

The code fragment of web page used to extract PIU contains at least a entire product information, so the PIU of table type can be further divided into four categories according to different start web tags:

(1) Code fragment in tags <td> or <th>
(2) Code fragment in tags <tr>
(3) Code fragment in tags <table>
(4) Besides the three types listed above, elements of an entire product information located at different <table> tags

3.3 PIU classifying extraction algorithm

PIU classifying extraction means to extract and classify the PIU contained in web pages using the classifying rules for further use in extracting information. The algorithm is described below:

```
Input: Keyword($key_1$, $key_2$, ⋯, $key_m$), page code;
        //key words、code of source pages
Output: classified PIU;
        //classified PIU
To pretreat page;            //preprocessing of web pages
To make TableSet($table_1$, $table_2$, ⋯, $table_n$);
i=i+1;
loop1:  if $key_1 \notin table_i$   then   goto loop3
        To make KeywordNodes($node_1$, $node_2$, ⋯ , $node_p$);
loop2:  if condition1 then { if condition5
        then PIU$\in$type(a)
                j=j+1;
                if j<=p then goto loop2
                else goto loop3}
        else if condition2 then { if condition6 then PIU$\in$type(b)
                j=j+1;
                if j<=p then goto loop2
                else goto loop3}
        else if condition3 then { if condition4 then
                { if condition7 then {if condition8 then
        PIU$\in$list type with header
                else PIU$\in$list type without header}
                                    }
        else if condition7 then PIU$\in$type(c)
loop3:  { i=i+1;
          if i<=n then goto loop1 else exit}
```

The symbols used in the algorithm are explained below:

TableSet($table_1$, $table_2$, ⋯, $table_n$): the set of all the <table> units in the web page, n being the number of <table> units in the web page

Keyword(key_1, key_2, ⋯, key_m): the set of all the key words provided by user, while m being the number of key words, and key1 represents the product designation

KeywordNodes($node_1$, $node_2$, ⋯, $node_p$) : the DOM tree structure of $table_i$ contains the node key_1, while p being the number of nodes that contain key_1

condition1: the columns of $node_j$ (<tr> or <th>) contain other elements of product information, and can constitute at least an entire product information, while containing explanatory text information about elements of product information;

condition2: the line of $node_j$ (<tr>) contains other elements of product information, and can constitute at least an entire product information, while containing explanatory text information about elements of product information;

condition3: the table of $node_j$ (<table>) contains other elements of product information and can constitute at least an entire product information

condition4: $node_j$ and key_1 fully match, while nodes containing other elements of product information do not include explanatory text information about elements of product information;

condition5: the other elements of product information of the column of $node_j$（<td> or <th>）contains key_2, ⋯, key_m;

condition6: the other elements of product information of the line of $node_j$（<tr>）contains key_2, ⋯, key_m;

condition7: the other elements of product information of the table of $node_j$（<table>）contains key_2, ⋯, key_m;

condition8: the first line would be the header information if the table of nodej（<table>）and the columns of the line of nodej match.

4. PIU information extraction algorithm

In Section 3, the classified PIU contains several product elements, while the user is interested in comparing at least one element in order to do comparison-shopping. Key elements of product information could be more than one and depend on user requirements. Various PIU information extraction algorithms are listed below.

4.1 List type PIU information extraction algorithm

The web page structure of list type PIU is with regular patter and the content of the text node of DOM tree is plain, so matching the key words could locate the corresponding text in the text node, while the elements of product information are at just the same line of the text (the text contained in <tr> node of DOM tree). The difference is that if it is a list with header information, then the elements of product information could be defined with the prompt of the header information; if it is a list without header information, then matching the elements of product information with related domain knowledge to define exact meanings. The information extraction algorithm for PIU with header is relatively simple, so the algorithm for PIU without header is described as follows.

```
Input: PIU, Keyword($key_1, key_2, \cdots, key_m$);
Output: ProductInfo($info_1, info_2, \cdots, info_t$);
To make DOM for PIU;
To make NodeSet($text_1, text_2, \cdots, text_n$);
i=1; j=1;
while i<=n do{
    To make $Text_i$_tr($td_1, td_2, \cdots, td_t$);
    while j<=t do{
    $info_j$=$td_j$.toString();
    if ($info_j \in$Keyword) then (To extract $info_j$ by Keyword)
        else{ r=1;
        while r<=q do{
            if ($info_j \in knowledge_r$) then {To extract $info_j$ by knowledge$_r$; goto loop1}
            else r=r+1;}
        r=1;
        while r<=q do{
            if similitude($info_j$, $knowledge_r$)> $\varepsilon$ then
                //the $info_j$ and $knowledge_r$ resembles enough
               {To update $knowledge_r$ and extract $info_j$ by
          $knowledge_r$; goto loop1}
            else r=r+1;}
        To append semantic information of $info_j$;
        goto loop2;}
loop1:   j=j+1}
        print(ProductInfo($info_1, info_2, \cdots, info_t$));
loop2:   i=i+1}
```

The symbols used in the algorithm are explained as follows:

NodeSet($text_1$, $text_2$, $\cdots$, $text_n$): the set of text nodes in DOM tree containing key_1, while n being the number of text nodes that contains key_1.

$Text_i$_tr(td_1, td_2, $\cdots$, td_t): lines in the list that contain $text_i$, the content between <tr> and </tr> in DOM tree that contains $text_i$, while t being the number of lines.

td_j.toString(): the string of elements of product information in the text nodes contained in td_j.

ProductInfo($info_1$, $info_2$, $\cdots$, $info_t$): the set of elements of product information extracted form PIU, while t is the number of elements and equals the number of columns in $Text_i$_tr.

tr_k.toString(): the string of text node contained in tr_k, namely the string with text information in the kth <tr> of the <table>.

KnowledgeSet(knowledge1, knowledge2, … $\cdots$, knowledgeq): the knowledge set of a specific domain, while q being the numbers of sub items in the knowledge set.

4.2 Table type PIU information extraction algorithm

The web page structure of table type PIU is much more diversified and the content of text nodes in DOM tree is quite complicated, including not only the elements of product information, but also the explanatory text information of those elements. So when extracting information, it is necessary to distinguish between explanatory text information and elements of product information, trying to match the elements with relevant domain knowledge in order to define specific meanings.

In the web page structure of table type PIU, type (c) is the most complicated one, so the discussion below tries to give out corresponding algorithm. Since types (a) and (b) are relatively simple, they will not be discussed any more.

The elements of product information of PIU type (c) are distributed in various <tr> tags in <table>, which means that from the view of web code programming, PIU type (c) contains many pieces of product information. After analyzing sampling web pages, it can be concluded that almost all PIUs of type (c) have several pieces of product information and the text information in the same <tr> containing various products' elements listed in sequence. For example, in the nth <td> of the first <tr> would be the designation of the nth product, while the nth <td> of the second <tr> would contain the specification of the nth product. Benefiting from this feature, just counting the occurrences of key elements of product information of table type PIU (c) well determines the numbers of complete product information contained. Then through the judgment of locations of the queried key words will obtain the elements of product information. The algorithm is described as follows:

```
Input: PIU, Keyword($key_1, key_2, \cdots, key_m$);
Output: ProductInfo($info_1, info_2, \cdots, info_t$);
To make DOM for PIU;
KeyInfo.count()=$tr_{KeyInfo}$.children().size();
To make NodeSet($text_1, text_2, \cdots, text_n$);
i=1;
```

```
        while i<n do{
              x=1; k=1;
              To make $Text_i$_tr($td_1$, $td_2$, ⋯, $td_t$);
              To make trList($tr_1$, $tr_2$, ⋯, $tr_p$);
              while k<=p do{
loop:         word=$tr_k$.child(z+x-1).toString();
              x=x+1;
              To separate “description” from “core” in “word”;
              if x <=$tr_k$.children().size()/KeyInfo.count() then goto loop;
              x=x+1; k=k+1;}
        i=i+1;}
print(ProductInfo($info_1$, $info_2$, ⋯, $info_t$));
```

The symbols used in the algorithm are explained as follows:

KeyInfo: key elements of product information.

KeyInfo.count(): the number of occurrences of key elements of product information in $tr_{KeyInfo}$.

tr_k.children().size(): the number of kth <tr> sub-nodes in the DOM tree, namely the number of columns with kth line.

trk.child(j).toString(): the string of text nodes contained in the jth sub-node of kth <tr> of the DOM tree, namely the string of the jth <td> of the kth <tr> in <table>

5. Experimental results

The test pages are chosen randomly from web pages about two specific groups (steels and cosmetics) after casting XML pages and keeping totally 386 pages of types such as HTML,. asp, .jsp,. php. There are 216 pages about steels, mainly of list type; 170 pages about cosmetics, mainly of table type. At the same time, price is chosen to be the key element of product information.

In the first stage, test on classifying pages and extracting PIU is carried out, and the classification result is given in recall and precision rates. The precise processing rate is defined as the number of pages precisely classified divided by the number of pages processed, while the processing rate is defined as the number of pages precisely classified divided by the number of pages processed in the test. The results is listed in table. 1.

Table. 1 The results of web pages classification and extraction

Web page type	Manual classification	Classifying algorithm		Precisely processing rate	Processing rate
		Web page processed	Web pages precisely processed		
List type	224	224	224	100	100
Table type(a)	85	85	84	98.8	98.8
Table type(b)	37	36	36	100	97.3
Table type(c)	29	26	25	96.2	86.2
Table type(d)	11	15	11	73.3	100

It is shown that the classifying and extraction algorithm is efficient and its various indexes reach the criteria of comparison-shopping system.

In the second stage, comprehensive testing based on two-stage extraction algorithm is applied to sampling pages, the result is listed in table. 2.

Table. 2 The comprehensive testing results based on two-stage extraction algorithm

Web page type	Total number of page	Web page processed	Average precision rate	Average recall rate
List type web page	224	224	93.7%	98.5%
Table type web page	162	147	81.4%	92.6%
Web page of steel	216	215	95.7%	99.3%
Web page of cosmetic	170	156	78.2%	92.3%

From table. 2, it can be concluded that the recall rate of pages containing PIU of table type is a bit lower than that of PIU of list type. Further analysis reveals the reason being that the sampling pages selected contain some pages of table type (d). The reason why the precision rate of pages of PIU of table type is much lower than that of pages of PIU of list type is that the pages of list type is more regular and the contents of text nodes of DOM tree is more plain, while parts of the outcome from information extraction algorithm applied on PIU table type needs further to be processed with manual appended semantics.

The precision rate of pages of steels is much higher than that of cosmetics; the reason is that the domain knowledge of steel is more standard than that of cosmetics besides the differences of page structures, so the domain knowledge database of cosmetics needs to be improved.

6. Conclusions

This paper presents a intelligent comparison-shopping system based on Web information extraction and analyses the basic formation, frame structure and workflow of the system. Two-stage work pattern of information extraction method and the concept of PIU are proposed, and at the same time the PIU classifying extraction and the extraction algorithm of product elements from PIU are presented too. This algorithm can greatly reduce the workload of manual appended semantics in web page information extraction while keeping high extraction efficiency. The experiment outcome shows that the two-stage information extraction technique meets the expected results on the whole.

With the further development of e-commerce, product information pages in XML will increase. The superiority of XML over HTML makes it definitely the new web information exchange standard. Further research would be centered on making the ICSS process XML and other complicated web pages, optimizing the quality of PIU classifying and information extraction algorithm and improving the coverage of pages to be handled.

Acknowledgments

This work is supported by in part the foundation of science and technology department of Zhejiang province under grant No. 2005C34002, Zhejiang natural science foundation under grant No.M603149.

References

[1] Florescu D, Levy A Y, Mendelzon A. Database techniques for the World-Wide Web: A Survey[A]. In: ACM The SIGMOD Record, 1998.59-74.

[2] Liu L, Pu C, Han W. XWRAP: An CML-enabled wrapper construction system for web information sources[A]. In: Proc International Conference on Data Engineering(ICDE), San diego, California, 2000. 611-621.

[3] Cham berlin D, Robie J, Florescu D. Quilt: an XML query language for the heterogeneous data sources. In: Proc International Workshop on the Web and Database(WebDB'2000), Dallas, Texas, 2000. 53-62.

[4] Sahuguet A, Azavant F. Building light-weight wrappers for legacy web datasources using w4f. In: Proc International Conference on Very Large Database, Edinburgh, Scotland, 1999. 738-741.

[5] Laender A, Ribeiro-Neto B, Silva A. A brief survey of web data extraction Tools[J], SIGMOD Record, 2002，31(2): 84-93.

[6] Hammer J, Garcia-Molina H, Nestorov S et al. Template-based wrappers in the TSIMMIS system[A]. In: Proc of ACM SIGMOD Conference on Management of Data, Tucson, Arizona, 1997. 523-535.

[7] Gruser Jean-Robert, Raschid L, Vidal M E et al. Wrapper generation for web accessible data sources[A]. In: Proc the Coopis, 1998. 14-23.

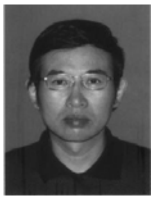
[8] Liu M, Ling T M. A conceptual model and rule-based query language for HTML. World Wide Web, 2001, 4(1): 49-77.

[9] Kushmerick N, Weld D et al. Induction for information extraction. In: Proc the 15th International Joint Conference on Artificial Intelligence, Nagoya, Japan, 1997, 2: 729-737.

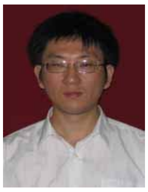
[10] Ashish N, Knoblock C. Wrapper generation for semi-structured internet sources[A]. In: Proc of Workshop on Management of Semi-Structured Data, Tucson, Arizona,1997,10-17.

[11] Liu L, Pu C, Han W. XWRAP: An CML-enabled wrapper construction system for web information sources[A]. In: Proc International Conference on Data Engineering(ICDE), San diego, California, 2000. 611-621.

Xun Wang received the B.S. and the M.S. degrees in Mechanics and Computer Science from Zhejiang University in 1990 and 1999, respectively. He now is associate professor in Zhejiang Gongshang University, and PH.D. candidate in Zhejiang University. His research interests include intelligent information processing, interactive graphics, multimedia information security.



Jin Haiwei received the B.S. and M.S. degrees in Computer Science and Management Engineering from Zhejiang University in 1982 and 1988, respectively. He now is associate professor in Zhejiang Gongshang University. His research interests include Management Information System, Decision Support System, and Artificial Intelligent.



Zhenyue Chen received the B.S. and the M.S. degrees in Computer Science from Zhejiang University in 1999 and 2003, respectively. He now is assistant professor in Zhejiang Gongshang University. His research interests includes Web data mining, GIS, computer graphics.