Architecture and Performance of the Rule based Comparison Shopping: Delivery Cost Experience

Gyoo Gun Lim
School of Business,
Hanyang University
17 Haengdang-dong,
Seongdong-gu, Seoul
133-791, Korea
+82-2-2220-2593
gglim@hanyang.ac.kr

Ju Young Kang
School of Business,
Ajou University
San 5, Wonchon-Dong
Yeongtong-Gu, Suwon
443-749, Korea
+82-31-219-2910
jykang@ajou.ac.kr

Jae Kyu Lee
Business School
KAIST
207-43 Cheongryangri,
Dongdaemun, Seoul,
130-722, Korea
+82-2-958-3612
jklee@kaist.ac.kr

Dae Chul Lee
School of Business,
Hanyang University
17 Haengdang-dong,
Seongdong-gu, Seoul
133-791, Korea
+82-2-2220-2593
dclee@hanyang.ac.kr

ABSTRACT
Comparison shopping is one of the important ingredients in electronic commerce, because it saves time and efforts of searching for product information and costs. However, most shopping malls still rely on the simple data such as price and other descriptive specifications, and cannot support the services of comparing the exact cost which requires rule based computation such as delivery cost. The comparison of delivery costs requires tailored computation because each bookstore has different free shipping rules, delivery options, and shipping rates. The purpose of this paper is to propose a rule based comparison shopping framework using XRML(eXtensible Rule Markup Language) approach. We propose the architecture, named *ConsiderD*, of maintaining rules in the comparison portal site to be consistent with its original pages. We experiment the benefit of using the rule based exact computation in comparison of book buying with the result by estimated average price such as BestWebBuy.com does. The result shows that providing the benefit of rule based comparison is significant and the hidden information such as delivery cost can be effectively processed with XRML framework.

Categories and Subject Descriptors
H.4.3. [Communications Applications]: Information Browsing (Electronic Commerce)

General Terms
Management, Measurement, Design, Experimentation,

Keywords
Comparison Shopping, XRML, Semantic Web, Knowledge Based System

1. INTRODUCTION
One of the most successful business models in electronic commerce era is the comparison shopping over various shopping malls and the number of online shopping stores is exploding in these days. Customers can save the time and efforts of searching the shopping information such as price and services.

The current comparison sites have limitation that they cannot support the rule based computation of exact price such as delivery costs. The comparison sites merely collect simple data like prices. However, calculating the delivery cost of multiple online bookstores is not an easy task, because each bookstore has its own free shipping rules, delivery options, and shipping rate policy. If a customer selects a set of books, the comparison site should calculate the delivery cost for each bookstore with different rules, options, and shipping rates of each bookstore. However, this is impossible with the data based comparison framework.

Some comparison shopping sites provide the average delivery cost of each shopping site, but the results are not accurate enough. Thus the rule based comparison shopping is one of the plausible solutions for this kind of situation.

There are several issues in using rule based architecture for the comparison portal site. First of all, how and who acquires the necessary rules? Another issue is to design the appropriate architecture that can share and maintain rules between shopping stores and comparison sites. We are also concerned if the rule based exact computation makes significant differences from the data based average estimates of the current comparison system.

XRML(eXtensible Rule Markup Language) framework[13] can provide some clues of solving the issues of rule acquisition and maintenance. XRML consists of three components, Rule Identification Markup Language (RIML), Rule Structure Markup Language (RSML) and Rule Triggering Markup Language (RTML) [13]. The merits of XRML framework can be summarized in three components: Support the rule acquisition process from Web pages, Use the acquired rules, and Maintain consistency between rules and Web pages. RIML and RSML supports the rule acquisition that consists of two steps − rule component identification and rule structuring - from Web pages[12]. The procedure is semi-automated by using ontology[17] so that we can adopt the XRML based system to solve the rule acquisition issue. Moreover, the RIML and RSML pair provides consistency maintenance between rules and Web pages [11]. The detailed procedure of applying XRML approach to comparison shopping domain is discussed in section 2 and section 3.

The research objective of this paper is to suggest rule based comparison shopping architecture by using XRML approach. The architecture is implemented in the domain of online bookstores and the comparison target is delivery cost of packaged books. We make a prototype of our suggested architecture with 5 real-world

10th Int. Conf. on Electronic Commerce (ICEC) '08 Innsbruck, Austria

online bookstores. Moreover, we experiment if the differences of the total cost including book price and delivery cost. We confirm that the function of rule based computation significantly outperforms the average data-retrieved delivery cost.

2. RELATED WORKS

2.1 Comparison Shopping

The electronic commerce market is exponentially expanding as the internet and Web technologies rapidly grow. US online retail sales will grow to $328.6 billion in 2010, up from $172.4 billion in 2005, according to a Forrester research report[10]. Moreover, the products and information about them in internet shopping malls are exponentially increasing. As the result of that, the role of comparison shopping sites is getting more important by saving the time and effort of customers for product information search over tremendous shopping malls. Online traffic to a custom category of leading comparison shopping websites increased 56 percent for the week ending Nov. 3, 2007 compared to Nov. 4, 2006[8].

There is numerous research about comparison shopping. One of them is implementing real-time comparison[5], but many comparison sites are still using batch style manual updates because real-time comparison requires excessive time and network traffics just before the service. Another research proposed a personalized comparison engine that recommends products based on ranking algorithm and user behavior model[27]. However, it did not handle complicated features that require additional processing.

A research about applying semantic Web technology to comparison shopping is also interesting[16]. It can significantly decrease manual jobs of matching products over different shopping malls by automatic matching of products and product attributes with ontology mapping techniques.

Even though above works about comparison shopping are very interesting, the problems we suggested in section 1 are still unsolved in related works because they are the first generation comparison shopping using only data. We expect that we will show the second generation comparison shopping based on rules and that compares the exact delivery costs.

2.2 Knowledge Acquisition

In the delivery cost of online bookstore domain that we have chosen, the rules for free shipping, delivery methods, and shipping rates are already explained in the Web pages of each bookstore. The process of acquiring rules from Web pages is similar to the knowledge acquisition process from natural languages because the rules are explained with texts and tables. However, knowledge acquisition has been an everlasting bottleneck in building rule based systems. Moreover, achieving perfect natural language processing is difficult because the natural language texts may imply more than one valid interpretation as Wetter and Nűse pointed out [25].

Recently, ontology has become popular for specifying the knowledge of a particular domain on the Web [6, 23, 24]. The natural language processing may be used to automatically extract terms to add to the ontology using grammar analysis [3, 15, 18], linguistic patterns [19, 20] with predefined templates [22, 26]. However the quality of automatically extracted knowledge from natural language sources is not accurate enough yet, so the draft should be manually refined by knowledge engineers [20, 25].

Machine learning techniques such as inductive learning, neural networks, and statistical models may be applied under the umbrella term of data mining - specifically Web mining when the log data are collected from Web pages [9, 13]. If a structured data set is available, we can induce them to more generalized and abstract knowledge. Several methods and tools [1, 2, 4, 21] have been developed using this approach. However, since extracting rules from the natural texts and tables usually aims to acquire the knowledge at the same level of abstraction, the inductive learning [2, 4, 21] is not the primary issue in extracting rules from Web pages.

Ontology was used in rule acquisition to support extracting rule components from Web pages in another research[17]. An interesting idea of this research is that it used ontology as a link for repeated rule acquisitions by summarizing rules of one site into compact ontology and using it on the other sites.

2.3 Rule Acquisition in XRML Approach

Rule acquisition is essential in our approach because the suggested second comparison shopping framework is based on rules. Moreover we should consider rule consistency between comparison sites and shopping malls. We are going to describe rule acquisition and consistency maintenance procedure of XRML approach here.

As we discussed in above section, acquiring rules from Web pages is a difficult task because the text in Web pages is unstructured and direct conversion of natural language text to formal representation is extremely complex as Hulth [7] pointed out. Therefore, the *eXtensible Rule Markup Language (XRML)* approach proposed a framework of acquiring rules from Web pages with divided steps using knowledge engineer's help instead of trying direct conversion. XRML rule acquisition framework consists of two components: *Rule Identification Markup Language (RIML*) and *Rule Structure Markup Language (RSML)* [27]. RIML identifies the rules implicitly expressed in Web pages; RSML represents the formal rule structure which corresponds to the rule syntax in commercial rule based systems [12].

The procedure of the XRML approach is depicted in Figure 1 with the step numbers in parentheses. The function of each step is the following: (1) *Identify Rules:* S/he identifies the rules from the browsed Web pages which display the text, tables, and pictures. Then the *knowledge editor* generates HTML/RIML files where the RIML statements are embedded in the HTML files. (2) *Automatic Transformation:* The identified rules in HTML/RIML can be automatically transformed to the rule syntax in RSML. The set of identified rules derived in this manner may not be complete, so these are called *draft rules*. (3) *Interactive Refinement:* The draft rules may need refinement to make them complete. This interactive refinement will be much easier than making the rules from scratch. (4) *Rule Base Generation:* The RSML rules may be transformed to the syntax of a commercial rule based system. (5) *Inference:* The rule base can be used to make an inference for human. (6) *Maintenance with Consistency:* If any changes happen on the Web pages or rule bases that are generated by the XRML framework, we can detect their

counterparts effectively and efficiently. Thus maintaining consistency between them can be easily implemented [12].

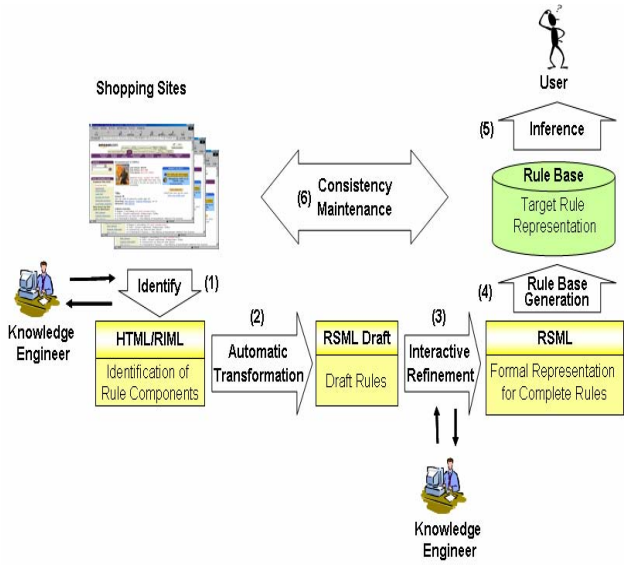


Figure 1: Rule Acquisition Process by the XRML Approach [12]

3. RULE BASED COMPARISON SHOPPING FRAMEWORK - *CONSIDERD*

This section describes the architecture of rule based comparison shopping site called *ConsiderD* System which uses XRML for rule representation. This comparison architecture can compare exact delivery costs of selected goods in different shopping malls as well as prices. By adopting the XRML approach, the architecture can guarantee the consistency of rules between the original shopping malls and the comparison site and reuse of rules without modification. The important points of our approach is that our framework compares the exact calculation of delivery costs based on rules and we provide methods of acquiring rules and rule consistency maintenance to support that. We have chosen the book comparison domain for our prototype because the domain is familiar to normal Internet users and it is easy to find and acquire delivery rules from various on-line book stores.

3.1 Design Issues of Rule Based Comparison Architecture

The first issue that was discussed at section 1 is about rule acquisition. To provide a solution for this issue, we have an important assumption that every online bookstore in the architecture built RIML and RSML for its own rule based system to provide shipping cost calculation service. This assumption is our recommendation for a reasonable architecture at the same time. Figure 2 shows an architecture of Web based KBS(Knowledge Based System) of each bookstore. In this architecture, the XRML framework plays a role that helps the knowledge engineer to acquire rules from Web pages and manage them. The brief procedure was explained at section 2.3, and an example in the delivery cost domain will be explained at section 3.3.

The second issue is about share and maintenance of rules between stores and comparison sites. They might be able to share the rule bases of bookstores, but the problem is that a comparison site may have different type of rule base. In that case, we have to be able to transform every possible type of commercial rule base. Therefore, the answer is to share RIML and RSML instead of rule bases, so that the only transformation that the comparison site should do is the one from RSML to its own rule base. Moreover, fetching and transforming RSML make the maintenance easier.

The last issue is about a significance test of the difference between our approach and the current comparison sites. We have conducted experiments on that at section 4 and discussed about the experiments results at section 5.

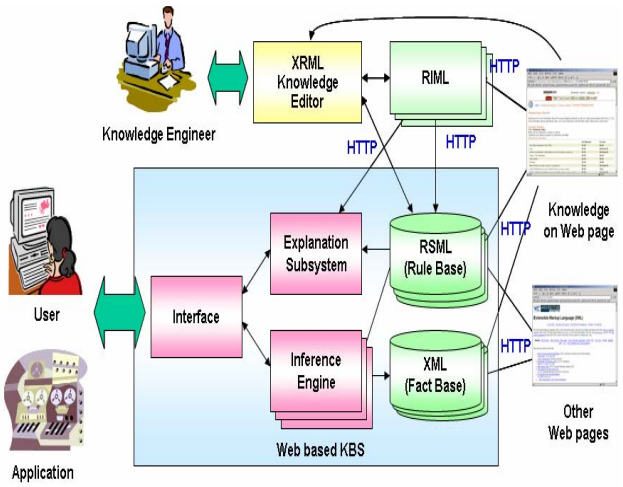


Figure 2: Architecture of Web based KBS System

3.2 Overall Architecture of *ConsiderD* System

The purpose of *ConsiderD* system is to provide and compare the overall cost for customers by adding the exact delivery cost to the total price of selected books in various online stores. Therefore we have chosen five representative online bookstores which are Amazon.com, BarnsandNoble.com, Powells.com, BooksAMillion.com and eCampus.com. After that, we found the Web pages explaining delivery cost of books and other items of each bookstore, and acquired delivery rules from the pages by using the XRML approach. The detailed procedure will be explained in the next section.

The exact calculation of delivery cost of books requires rule based reasoning system that uses acquired rules from the five online bookstores as we discussed in section 1. Our prototype *ConsiderD* is a comparison shopping site which includes the rule based reasoning system. We designed *ConsiderD* with knowledge share structure based on XRML that is explained in section 2.2 to fetch and manage rules from bookstores.

Figure 3 shows the overall architecture of *ConsiderD*. There are 5 stores in our prototype. Every online bookstore in the architecture built RIML and RSML for its own rule based system to provide shipping cost calculation service to customers. There is an assumption in the architecture that every online bookstore built RIML and RSML for its own rule based system to provide

shipping cost calculation service to customers. We expect that this assumption makes sense in the point of view that XRML architecture provides an advantage to online bookstores on acquisition of rules from existing Web pages.

ConsiderD fetches RIML and RSML from the stores and merge and transform them into one rule base as shown in Figure 3. After that, *ConsiderD* provides comparison of shipping costs and book prices between different shopping malls by doing inference on the merged rule base. The important factor of the architecture is that *ConsiderD* does not need to acquire rules by itself from Web pages of each bookstore. Moreover, the each bookstore can easily maintain the consistency of rules by comparing RIML and RSML. If there is any change in HTML files that explain the rules, the rule manager updates RIML and finds the corresponding parts of RSML for the changed part of RIML. After that, the manager can update the rules in RSML. *ConsiderD* checks RIML and RSML files in each bookstore. If there is any change, it repeats the fetch and merge of them into the rule base.

Because of this architecture, the customer can easily compare the exact total price including delivery costs of chosen books over 5 online bookstores instead of visiting each book store one by one.

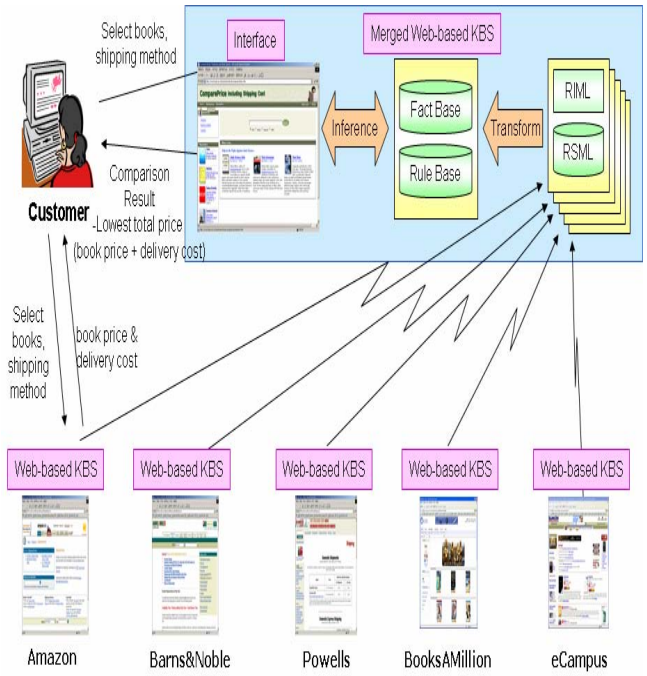


Figure 3: Architecture of *ConsiderD* System

3.3 Acquiring and Maintaining Rules from Online Stores by Using XRML Approach

Acquiring delivery rules from Web pages happens in each bookstore. For example, Amazon.com acquires delivery rules from the Web pages explaining delivery policies for its own rule based system. Figure 4 shows a part of the page that explains "Priority International Courier" shipping rates for CDs and books. There are two types of shipping rates, "Per Shipment" and "Per Item". If a customer who lives in "Asia & Pacific Islands" put 5 books in a shopping basket and selected the delivery method "Priority International Courier", then the per shipment charge of "$29.99" will be added once to the delivery cost while the per item charge of "$8.99" will be added 5 times resulting $74.94 of total delivery cost .

The rule acquisition starts with rule component identification by extracting rule components such as variables and values into an RIML document which is shown in Figure 5. A part of the RIML document in Figure 5 shows that "Items", "Per Shipment", and "Per Item" are variables and "CDs", "DVDs", "Books", and "$24.99" are values in rules.

In the next step, the RIML document is converted to an RSML document that contains reasoning rules in the form of XML syntax as shown in Figure 6. The second rule with rid 3 is what was explained at the first paragraph of this section. The architecture of Web-based KBS of each bookstore is shown in Figure 2. Now, we can automatically calculate the delivery cost with the acquired rules and rule based system at Amazon.

The core of our architecture is *ConsiderD* system, so the most important step is fetching the rules from bookstores and transforming them into the rule base of Web-based reasoning system as shown in Figure 3. Firstly, every RIML and RSML pairs of the bookstores is fetched and stored in *ConsiderD* system. After that, each RSML is transformed to a group of rules for the corresponding bookstore in the rule base. The rules are categorized into 5 categories according to their sources. The inference is repeated 5 times for each category of rules, and produces the delivery cost calculation results to the customer.

The maintenance of rules is also an important part of our architecture. As shown in Figure 3, the rule base of *ConsiderD* is connected to the bookstores' rule bases via RIML and RSML. If there is change with the delivery policy of a bookstore, the bookstore will apply the change to the Web page. The XRML framework provides a consistency maintenance method by coupling RIML and RSML documents, so *ConsiderD* can acquire changed rules by re-fetching the changed RIML and RSML documents.

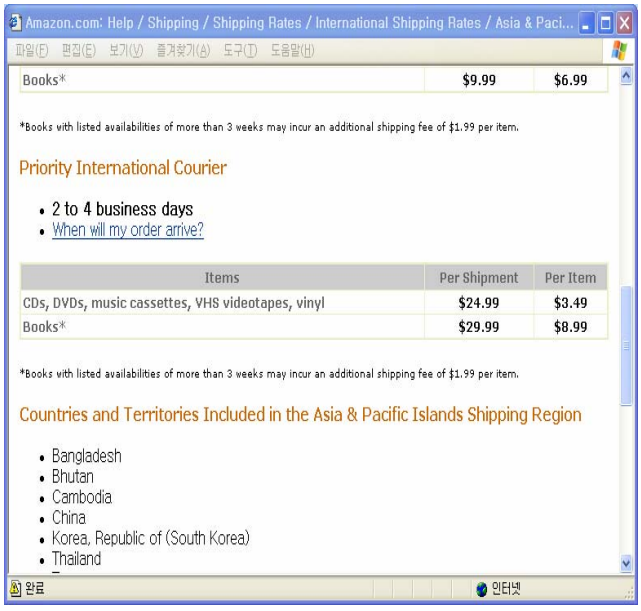


Figure 4: A Web Page Explaining Priority International Courier at Amazon

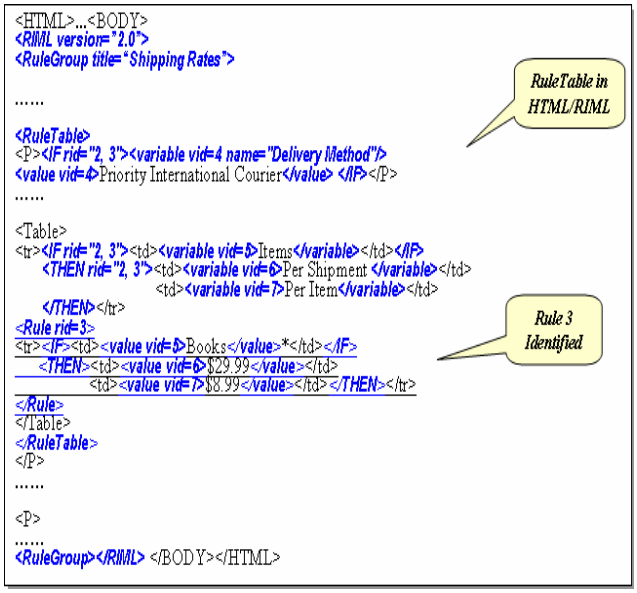


Figure 5: A Part of the RIML Document Extracted from the Web page of Figure 4

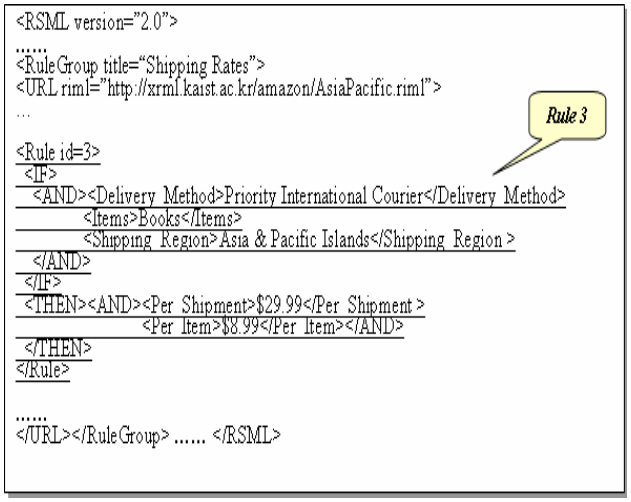


Figure 6: A Part of RSML Document Generated from Figure 5

3.4 Scenario of Using *ConsiderD* for Customers

The following steps are the procedure of using *ConsiderD* for customers to compare the total cost including book price and delivery cost over 5 online bookstores.

(1) A customer selects books from *ConsiderD* and put them into a shopping basket.

(2) The customer selects online bookstores that s/he wants to compare.

(3) The customer selects a preferred delivery method.

(4) The customer executes comparison shopping.

(5) *ConsiderD* calculates and shows the total price including book price and delivery cost for selected books and bookstores.

(6) If needed, the customer can refer the Web pages that related to the calculation.

The user interface of *ConsiderD* is very easy and intuitive. Figure 7 shows the page where the customer can select preferred bookstores and a delivery method for each shopping mall. Each bookstore provides different types of delivery methods. Therefore, the customer should select appropriate corresponding delivery methods over every bookstore. For example, the customer selected "*Priority International Courier*" at Amazon and "*International Express*" at BarnesAndNoble in Figure 7 because the required delivery times are similar to each other.

By clicking comparison button with the label of "*Calculate Total Cost Including Shipping Cost*" in the lower part of Figure 7, the customer can see the comparison results as shown in Figure 8. The results that contain delivery time, total book price, delivery cost per shipment, delivery cost per item, and final total price for the selected package of books are categorized by bookstores. The customer can compare the final costs of 5 online bookstores in the Web page of Figure 7. The price at Amazon is the lowest with $363.64 of the total cost. The result in Figure 8 shows the exact costs of books while the comparison result of BestWebBuy shows the average delivery cost of the book and book price. Showing a comparison result for only one book is another drawback of BestWebBuys.com.

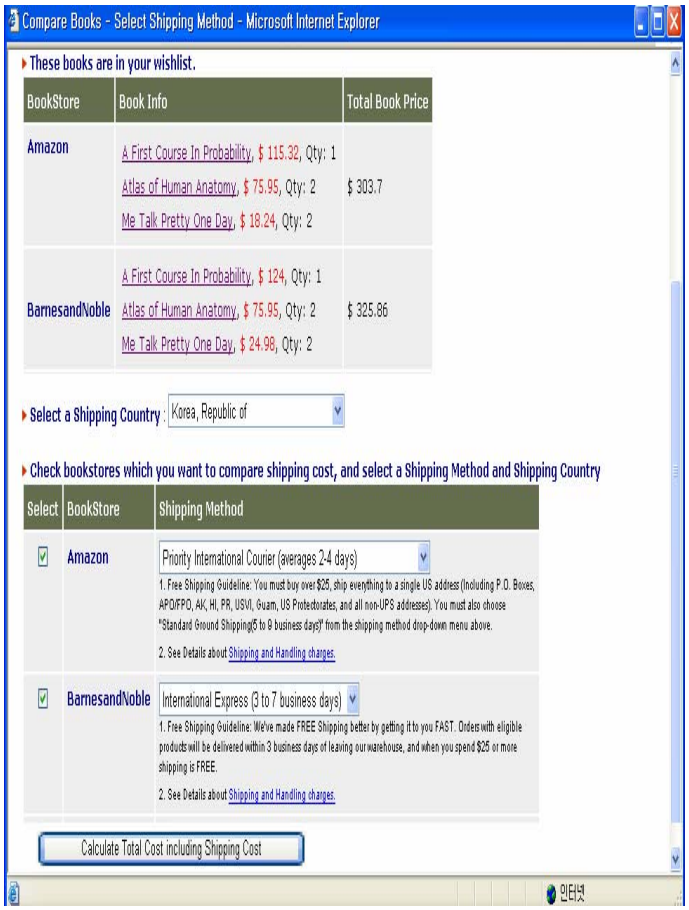


Figure 7: Choosing Bookstores and Delivery Methods in *ConsiderD*

Figure 8: Comparison Results in *ConsiderD*

4. EXPERIMENT SETTINGS

To prove the efficiency of *ConsiderD* comparing to the traditional comparison approaches, we did a simulation with statistical methodology. From now on we will use the terminologies of *CD approach* for *ConsiderD* and *BW approach* for BestWebbuy.com as the representative of the traditional comparing approach. We chose five well known online book stores which can be found on the BestWebbuy.com website such as Amazon, Barnes&Noble, Powell, BooksAmilion, and eCampus. We selected 100 different books from the suggested best sellers on BestWebbuy.com for the experiment. Fourteen books from the Business and Investing category, eight from Computers and Internet, sixteen from Fiction, two from Mystery and Thrillers, 21 from Non-Fiction, 36 from Professional & Technical, and three from Science Fiction and Fantasy. For a more complete experiment, we adjusted the selection that the number of lowest price books of a web site is equal, that is we selected 20 different books on each web site that sells the books with the lowest price among the book stores. The characteristics of the selected books are: The average price of books is $63.47, the minimum price is $9.42, and the maximum price is $169.60 with 48.12 standard deviation as shown in Table 1.

The delivery methods of each site are various. However the delivery methods can be categorized into three such as standard, expedited, and priority delivery according to the speed and price. The delivery cost is also varying according to the destination. We used four different delivery destinations such as the United States (domestic), Canada, United Kingdom, and Korea. The characteristics of delivery costs are: The average delivery costs of standard delivery are $2.99(the U.S), $4.30(Canada), $6.49(Korea), and $7.49(U.K). The average delivery costs of expedited delivery are $3.49(the U.S), $7.00(Canada), $9.50(Korea), and $7.75(U.K). The average delivery costs of priority delivery are $10.49(the U.S), $27.50(Canada), $31.40(Korea), and $30.40(U.K).

We did the experiment with the cases of buying from one to ten books. For every case we selected books randomly from the population and tried 12 different delivery methods (four destinations X three delivery methods). Also, for each case we tried the experiment 100 times. Therefore we had 12,000 experiment data. However, because some sites do not support certain delivery methods, so we used a valid set of 9,615 data.

We will use the following notations.

BW: means that case of using BestWebbuy.com

CD: means that case of using ConsiderD

TP(x): Total Price of approach x

BP(x): Book Price of approach x

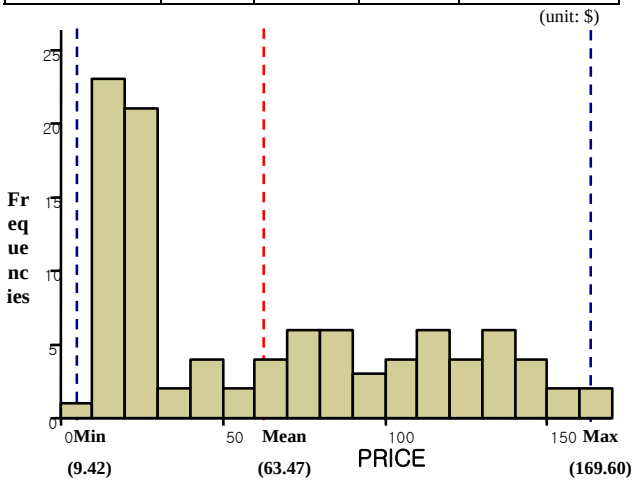
DC(x): Delivery Cost of approach x

TP(x) = BP(x) + DC(x)

x – y: x minus y. For example, TP(BW) – TP(CD)

Table 1. Books chosen for the experiment (unit: $)

N	Mean	Min	Max	Std. Deviation
100	63.47	9.42	169.60	48.12



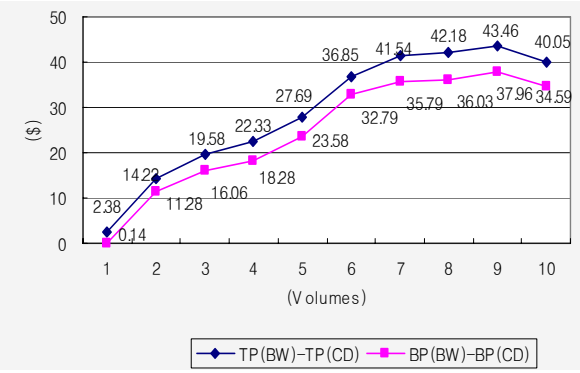
5. THE RESULTS OF THE EXPERIMENT

5.1 CD Approach vs. BW Approach

Table 2 shows the result of the experiment. It shows TP(BW), TP(CD), TP(BW) – TP(CD), BP(BW) – BP(CD), and DC(BW) – DC(CD). The average total price of BW was $423.85 and that of CD was $394.82. For the case of TP(BW) - TP(CD), the average of it was $29.03. CD approach can reduce the cost of buying books by $5.33 on the average for each book. As we see in Figure 9, the amount of saving increases as the buyer buys more books from $2.38 to $40.04, when buying from one to ten books.

Table 2. Statistical summary of the experiment results

#of Books	BW				CD				BW – CD					
	BP(BW)	DC(BW)	TP(BW)		BP(CD)	DC(CD)	TP(CD)		BP(BW) – BP(CD)		DC(BW) – DC(CD)		TP(BW) – TP(CD)	
			Average	Price per book			Average	Price per book	Average	Price per book	Average	Cost per book	Average	Price per book
1	56.41	17.87	74.28	74.28	56.27	15.63	71.90	71.90	0.07	0.07	2.30	2.30	2.38	2.38
2	144.1	23.06	167.16	83.58	132.82	20.12	152.95	76.47	11.15	5.57	3.07	1.53	14.22	7.11
3	208.29	27.77	236.06	78.69	192.22	24.25	216.47	72.16	15.98	5.33	3.60	1.20	19.58	6.53
4	276.85	31.75	308.60	77.15	258.57	27.69	286.26	71.57	18.21	4.55	4.12	1.03	22.33	5.58
5	363.86	35.69	399.55	79.91	340.27	31.58	371.86	74.37	23.52	4.70	4.17	0.83	27.69	5.54
6	434.96	39.53	474.49	79.08	402.18	35.47	437.65	72.94	32.77	5.46	4.08	0.68	36.85	6.14
7	507.76	44.31	552.07	78.87	471.9	38.67	510.52	72.93	35.76	5.11	5.78	0.83	41.54	5.93
8	571.62	49.04	620.66	77.58	535.59	42.89	578.48	72.31	36.02	4.50	6.16	0.77	42.18	5.27
9	634.49	52.44	686.93	76.33	596.53	47.16	643.47	71.50	37.97	4.22	5.49	0.61	43.46	4.83
10	662.03	56.64	718.67	71.87	627.44	51.19	678.63	67.86	34.57	3.46	5.48	0.55	40.05	4.00
Avg.	**386.04**	**37.82**	**423.85**	**77.73**	**361.38**	**33.47**	**394.82**	**72.40**	**24.60**	**4.30**	**4.43**	**1.03**	**29.03**	**5.33**



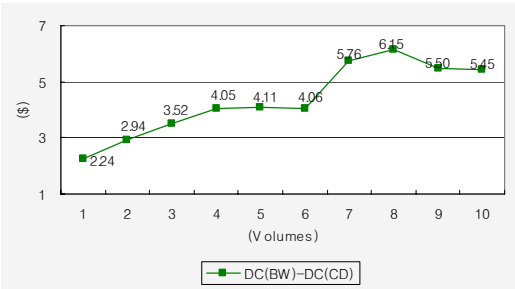


Figure 9: TP(BW) – TP(CD), BP(BW) – BP(CD), and DC(BW) – DC(CD)

To check the statistical validity of the experiment, we performed a t-test as shown in Table 3. The result shows that buying books in CD approach is significantly more efficient than that in BW approach, with a 99% confidence interval (p-value < 0.01). The reason of the significant difference between BW and CD approaches is while BW approach estimates the total price based on the average delivery cost per volume, CD approach calculates the exact price considering book price and delivery methods.

5.2 The Portion of Book Price and Delivery Cost in TP(BW) - TP(CD)

From Table 2, we know that BP(BW) - BP(CD) is $24.6 and DC(BW) - DC(CD) is $4.43 on the average. To find how much the delivery cost affects the total price we checked the portion of book price and delivery cost in TP(BW) - TP(CD) as shown in Figure 10. The portion of book price and delivery cost in the total price was 76.86% and 23.14%, on the average. As we see in Figure 10, when we buy a small amount of books the delivery effects is high. That means that consumers pay more than the necessary cost when they buy a small amount of books at one time, which is the general case in online book buying. And the main reason for the distorted price is the delivery cost. As Figure 10 shows, 94.14% is the portion of the delivery effect in the total price between BW approach and CD approach when we buy just one book. It is because if we buy only one book, it is not necessary to compare the rules provided by each online bookstore so that there is no significant book price difference between BW and CD approach. Therefore, the delivery cost differences looks bigger than the book price difference in this situation.

Table 3: t-test for TP(BW) – TP(CD)

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	99% Confidence Interval of the Difference				
				Lower	Upper			
Pair 1 TC(BW)-TC(CD)	29.06930	33.51745	.34182	28.18866	29.94994	85.043	9614	.000

Confidence interval = 99% (α =0.01)

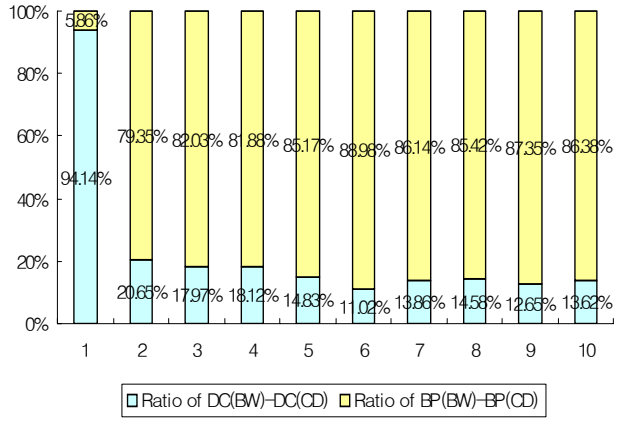


Figure 10: The portion of book price and delivery cost in TP(BW) - TP(CD)

5.3 Analysis of the Delivery Effect: According to Region

As a result, we can see that the delivery cost in buying books plays an important role in the overall price, though in general consumers neglect that. We tried experiments according to regions.

To compare the regions we compared domestic (inside of the U.S) delivery costs with international (outside of the U.S) delivery costs. Figure 11 shows that the delivery portion is 18.99% for the domestic delivery cases and the delivery portion of the international delivery cases is 12.86%. It means that the CD approach is better than the BW approach in domestic delivery cases which are more frequent than that of international delivery. And we want to note that the delivery effect appears very high even with only one book.

To check the significance, we did t-tests for domestic and international cases, as shown in Table 4 and Table 5. The cost of the BW approach is $5.17 more, on average, than the CD approach, with 99% confidence interval (α=0.01) (p-value < 0.01) in domestic cases. While in the international cases the cost of the BW approach is $3.88 more, on average, than the CD approach, with 99% confidence interval (α=0.01) (p-value < 0.01).

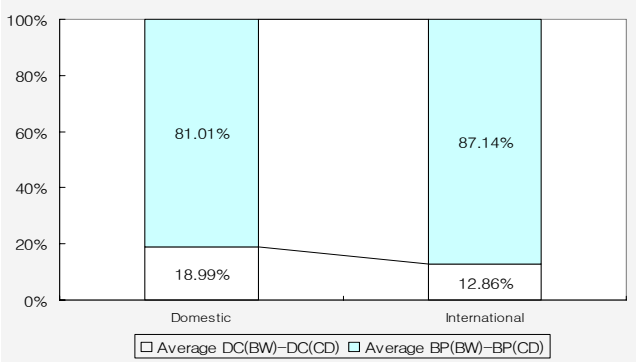


Figure 11: The portion of difference of the book price and delivery cost in the case of domestic & international delivery

Table 4: t-test for DC(BW) and DC(CD) for domestic delivery

Domestic	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	99% Confidence Interval of the Difference				
				Lower	Upper			
Pair DC(BW)-DC(CD)	5.17280	9.78573	.16133	4.75702	5.58859	32.063	3678	.000

Confidence interval = 99% (α =0.01)

Table 5: t-test for DC(BW) and DC(CD) for international delivery

International	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	99% Confidence Interval of the Difference				
				Lower	Upper			
Pair DC(BW)-DC(CD)	3.88002	7.83119	.10164	3.61812	4.14192	38.173	5935	.000

Confidence interval = 99% (α =0.01)

5.4 Analysis of the Delivery Effect: Types of Delivery

To compare the types of delivery we compared standard, expedited, and priority delivery costs. Figure 12 shows that the delivery effect portion in standard delivery, expedited delivery, and priority delivery were 7.18%, 18.26%, 17.09% respectively. To check the significance, we did ANOVA test for three groups of different delivery types as shown in Table 6. The delivery types are significantly different with a 95% confidence interval (α=0.05) (p-value < 0.05). Also, the costs of delivery of special and fast delivery methods are higher than that of the normal delivery type. It means that the effect of using CD approach is greater when we use it for special delivery methods.

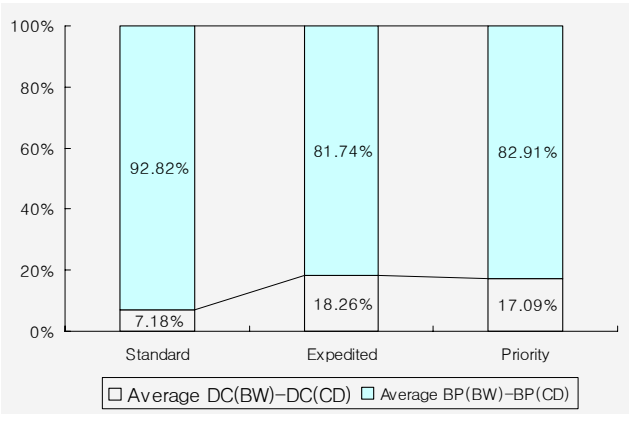


Figure 12: The portion of difference of the book price and delivery cost, using Standard, Expedited and Priority delivery

Table 6: ANOVA test of delivery methods

BW-CD

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	7744.331	2	3872.165	52.257	.000
Within Groups	712237.975	9612	74.099		
Total	719982.306	9614			

Duncan[a,b]

Ship Method	N	Subset for alpha = .05		
		1	2	3
Standard	2902	3.0491		
Expedited	1898		4.5394	
Priority	4815			5.1086
Sig.		1.000	1.000	1.000

Means for groups in homogeneous subsets are displayed.
a. Uses Harmonic Mean Sample Size = 2779.980
b. The group sizes are unequal. The harmonic mean of the group sizes is used. Type I error levels are not guaranteed.

Confidence interval = 95% (α=0.05)

5.5 Analysis of the Delivery Effect: Delivery Cost

From this study, we realized that the delivery cost is hidden in the total price of online book selling. Furthermore, if we use a rule-based comparison then we can compare more exactly and buy at an optimal price. Figure 13 shows the delivery cost per book. It shows that DC(BW) – DC(CD) is $2.30 on the average when we buy one book and the effect of the CD approach is greater when we buy from one to three books, which are the most usual cases when people buy book online.

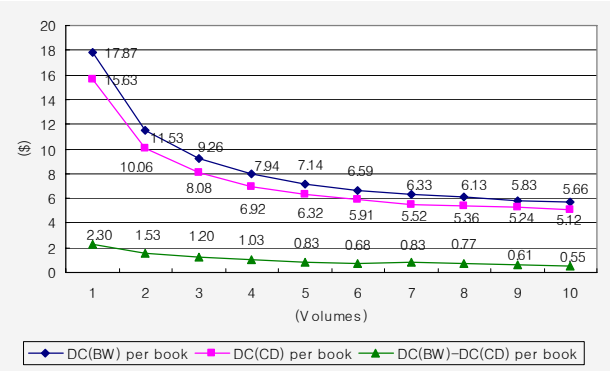


Figure 13: Delivery cost per book

6. CONCLUSION

Even though comparison shopping is one of the most important and rapidly growing business areas in electronic commerce, most shopping malls still rely on simple data such as price and other specifications. As the result, the current comparison sites have a limitation that they cannot compare complicated features that require a complicated processing system. For example, the calculation and comparison of delivery costs requires complex processing because each bookstore has different free shipping rules, delivery options, and shipping rates due to its own delivery policies. Therefore, we proposed a rule based comparison shopping framework that enables the complicated comparison. To achieve our goal, we suggested architecture of *ConsiderD* deploying XRML approach that is useful in rule acquisition and maintenance from Web pages. Also, we implemented a prototype of *ConsiderD* and proposed a scenario of using it for comparison of book prices.

For this study, we chose five well known online book stores which can be found on the commercial BestWebbuy.com website. By defining notations and setting simulation environment, we tried experiment to verify the efficiency of the new comparison approach, CD approach. From this research we found some useful and interesting results, as follows:

First, it showed that CD approach is better than conventional BW approach. Because the CD approach uses the XRML-based rule in making the comparison, it uses full information for the comparison, so it can give an optimal solution for the buying decision. Whereas BW approach just estimates the total price based on the average prices provided on the seller's sites.

Second, the advantage of the CD approach is based on the calculation of the hidden information. So not only the book price but also the delivery cost can be the matter of the difference between BW approach and CD approach.

Third, The delivery effect portion was 23.14% of the total price on average. In the BW approach we can not check fully the delivery cost in various cases. However, the CD approach can find the total price with optimal delivery cost.

Fourth, the delivery effect portion was 18.99% for domestic delivery and 12.86% for international delivery. It means that the CD approach is better than the BW approach in domestic delivery cases, which are more frequent than that of international delivery.

Fifth, the delivery effect portion of standard delivery, expedited delivery, and priority delivery were 7.18%, 18.26%, 17.09% respectively. That means that the effect of using CD approach is greater with special delivery.

Sixth, it showed that DC(BW) – DC(CD) is $2.30 on average and 94.14% is the portion of the delivery effect in TP(BW) – TP(CD) when we buy one book. In general, consumers buy only one book or two, or three. From the research of the Kyobook book publishing company(www.kyobobook.co.kr) which is the biggest book store in Korea, it showed that the portion of buying one book was 29.4%, two books was 28.4%, and three books was 20.2% of the total book buying. It also showed that the effect of the CD approach is high when we buy a small amount of books, which are the usual cases in daily life.

We did this experiment using the setting of online book buying. The result shows that the benefit of rule based comparison is significant and it can be extended to various cases. We hope that more advanced research and implementation about XRML-based e-business can provide more optimal and transparent commerce in the future e-business world.

7. REFERENCES

[1] Alani, H., Kim, S., Millard, D.E., Weal, M.J., Hall, W., Lewis, P.H. and Shadbolt, N.R. 2003. Automatic Ontology-Based Knowledge Extraction from Web Documents. IEEE Intelligent Systems 18(1) (2003) 14-21.

[2] Apte, C., Damerau, F. and Weiss, M.S. 1994. Automated Learning of Decision Rules for Text Categorization. ACM Transactions on Information Systems, 12 (3) (1994) 233-251.

[3] Babowal, D. and Joerg, W. 1999. From Information to Knowledge: Introducing WebStract's Knowledge Engineering Approach. Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering, Edmonton, Alberta, (May 1999) 1525 -1530.

[4] Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K. and Slattery, S. 2000. Learning to Construct Knowledge Bases from the World Wide Web. Artificial Intelligence 118(1-2) (2000) 69-113.

[5] dos Santos, S. C., Angelim, S. and Meira, S. R. L. 2001. Building Comparison-Shopping Brokers on the Web. Lecture Notes in Computer Science, no.2232 (2001), 26-38.

[6] Guarino, N. 1997. Understanding, Building and Using Ontologies. International Journal of Human and Computer Studies 46 (1997) 293-310.

[7] Hulth, A., Karlgren, J., Jonsson, A., Boström, H. and Asker, L. 2001. Automatic Keyword Extraction using Domain Knowledge. Proceedings of the Second Computational Linguistics and Intelligent Text Processing, Mexico City, Mexico, (2001) 472-482.

[8] Jasra, M. 2007. Comparison Shopping Traffic Increases 56 Percent. http://manojjasra.blogspot.com/2007/11/comparison-shopping-traffic-increases.html

[9] Jicheng, W., Yuan, H., Gangshan, W. and Fuyan, Z. 1999. Web Mining: Knowledge Discovery on the Web, Proceedings of the IEEE Conference on Systems, Man, and Cybernetics, Tokyo, Japan, (Oct. 1999).

[10] Johnson, C., Mulpuru, S., Mendelsohn, T. and Baird, N. 2005. Topic Overview: US Online Retail. http://www.forrester.com/Research/Document/Excerpt/0,7211,38360,00.html

[11] Kang, J. and Lee, J. K. 2003. Extraction of Structured Rules from Web Pages and Maintenance of Mutual Consistency: XRML Approach. Lecture Notes in Computer Science, no.2876 (2003), 150-163.

[12] Kim, D., Jung, H. and Lee, G. 2003. Unsupervised Learning of mDTD Extraction Patterns for Web Text Mining, Information Processing & Management 39(4) (2003) 623-637.

[13] Lee, J.K. and Sohn, M. 2003. Extensible Rule Markup Language – toward Intelligent Web Platform. Communications of the ACM 46 (May 2003) 59-64.

[14] Maedche, A. and Stabb, S. 2000. Mining Ontologies from Text, Proceedings of the European Knowledge Acquisition Workshop. Lecture Notes in Artificial Intelligence 1937 (2000) 189-202.

[15] Park, S. and Kim, W. 2007. Ontology Mapping Between Heterogeneous Product Taxonomies in an Electronic Commerce Environment. International Journal of Electronic Commerce, Vol. 12, No. 2 (Winter 2007-2008) 69 – 87.

[16] Park, S. and Lee, J.K. 2007. Rule Identification Using Ontology While Acquiring Rules from Web pages. International Journal of Human-Computer Studies, Vol.65, No.7 (2007), 659-673.

[17] Rau, L.F., Jacobsa, P.S. and Zernika, U. 1989. Information Extraction and Text Summarizatrion using Linguistic Knowledge Acquisition. Information Processing & Management 25 (4) (1989) 419-428.

[18] Ruiz-Sanchez, J.M., Valencia-Garca, R., Fernandez-Breis, J.T., Martnez-Bejar, R. and Compton, P. 2003. An Approach for Incremental Knowledge Acquisition from Text. Expert System with Applcations 25(1) (2003) 77-86.

[19] Schmidt, G. and Wetter, T. 1998. Using Natural Language Sources in Model-Based Knowledge Acquisition. Data & Knowledge Engineering 26 (1998) 327-356.

[20] Soderland, S. 1999. Learning Information Extraction Rules for Semi-structured and Free Text. Machine Learning 34(1) (1999) 233-272.

[21] Szpakowicz, S. 1990. Semi-automatic Acquisition of Conceptual Structure from Technical Texts. International Journal of Man-Machine Studies 33(4) (1990) 385-397.

[22] Sánchez-Carreño, R., Fernández-Breis, J.T., Martínez-Béjar, R. and Cantos-Gómez, P. 2000. An Ontology-Based Approach Knowledge Acquisition from Text. Cuadernos de Filologia Inglesa 9(1) (2000) 191-212.

[23] van Heijst, G., Schreiber, A.T. and Wielinga, B.J. 1997. Using Explicit Ontologies in KBS Development. International Journal of Human-Computer Studies 45 (1997) 183-292.

[24] Wetter, T. and Nűse, R. 1992. Use of Natural Language for Knowledge Acquisition: Strategies to Cope with Semantic and Programatic Variation. IBM Journal of Research and Development 36 (3) (May 1992) 435-468.

[25] Yang, J., Oh, H., Doh, K.G. and Choi, J. 2002. A Knowledge-Based Information Extraction System for Semi-structured Labeled Documents. Proceedings of the 4th Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science 2412 (2002) 105-110.

[26] Yuan, S.T. 2003. A Personalized and Integrative Comparison-Shopping Engine and Its Applications. Decision Support Systems, v.34, no.2 (2003), 139-156.