Intelligent Product Information Search Framework Based on the Semantic Web

Wooju Kim[a], Dae Woo Choi[a] and Sang-un Park[b]

[a]Department of Information Industrial Engineering, Yonsei University
134 Shin-Chon Dong, SeoDaeMun-gu, Seoul 120-749, Korea
Tel: +82-2-2123-5716, Fax: +82-2-364-7807, E-mail: wkim@yonsei.ac.kr, qorwkr@nate.com

[b]Department of Management Engineering, Korea Advanced Institute of Science and Technology
207-43 Cheongryangri-dong, Dongdaemoon-gu, Seoul 130-012, Korea
Tel: +82-2-958-3662, Fax: +82-2-960-2102, mascon@kgsm.kaist.ac.kr

Abstract

The Semantic Web and Web Services provide many opportunities in various applications such as product search and comparison in electronic commerce. We implemented an intelligent meta-search and comparison system for products through multiple attributes by using ontology mapping and Web Services. Under the assumption that each shopping mall offers product ontology and product search service with Web Services, we proposed a meta-search framework to configure a customer's purpose, make and dispatch proper queries to each shopping mall, evaluate search results from malls, and show the customer the product list with a ranking. Ontology mapping is used for generating proper queries for malls that have different taxonomies of product category. Also we implemented an inference based search engine using ontology and Web Services for each mall.

1. Introduction

The Semantic Web and Web Services provide many opportunities in various applications using the Internet. We expect that product search and comparison in electronic commerce can be one of the killer applications of such technologies. So far most product comparison systems over multiple shopping malls depend on a keyword-based search or manual collection of comparison information [4]. The meta-search engine [1, 3, 5] is one of those systems. However, shopping malls are offering search and directory service for their products in their Web pages, so the keyword based meta-search for those malls may become redundant. Moreover, it brings inaccurate results because of difficulties of natural language processing. We want to note that Web Services can provide a new framework of the meta-search engine. Amazon.com is offering a service that allows functions from retrieving information about products to adding an item to a shopping cart with Web Services[1]. As more shopping malls are expected to offer Web Services, the meta-search engine will be able to use Web Services to collect structured product information. Especially when the customer wants to compare products based on various attributes rather than simple price, Web Services will be the most appropriate alternative for the meta-search infrastructure.

But, there is still a problem to integrate and compare the results from various malls with Web Services. The meta-search system needs semantic interoperability among malls because malls are using different product taxonomies. While the standardization and integration of ontology is still progressing, it seems impossible that a single taxonomy will be used in every mall. Therefore, we employed ontology mapping techniques [2] to assure semantic interoperability among malls.

In this paper, we will suggest a meta-search and comparison framework using multiple attributes. This framework uses Web Services of shopping malls to acquire structured product information, and compare products based on multiple attributes. Also we use ontology mapping to handle different taxonomies. We named the meta-search and comparison system ***Intelligent Product Information Search (IPIS)***. Section 2 briefly describes the architecture of IPIS. And section 3 explains the intelligent product search and comparison

[1] http://www.amazon.com/exec/obidos/tg/browse/-/1086640/qid=1093862859/sr=1-1/002-1300672-1360811

procedure using IPIS. The last section summarizes the conclusions.

2. Architecture of the IPIS (Intelligent Product Information Search) System

IPIS System architecture (see Fig. 1) consists of four key areas. First, ***Interface*** represents a customer's search purpose in a structured format. The customer can input the desired product, attributes, and priorities of attributes in ***Interface***. ***Query Generator*** transforms the customer's configuration into queries for each shopping mall by using ontology mapping because shopping malls have different taxonomies. Shopping malls process the queries to search for proper products satisfying the customer's input and return detailed information of the products to ***Product Evaluation Agent*** through ***Web Services***. ***Product Evaluation Agent*** evaluates and compares the results from shopping malls and shows the customer the list of products with a ranking and attributes. Detailed functions of each module will be described in section 3.

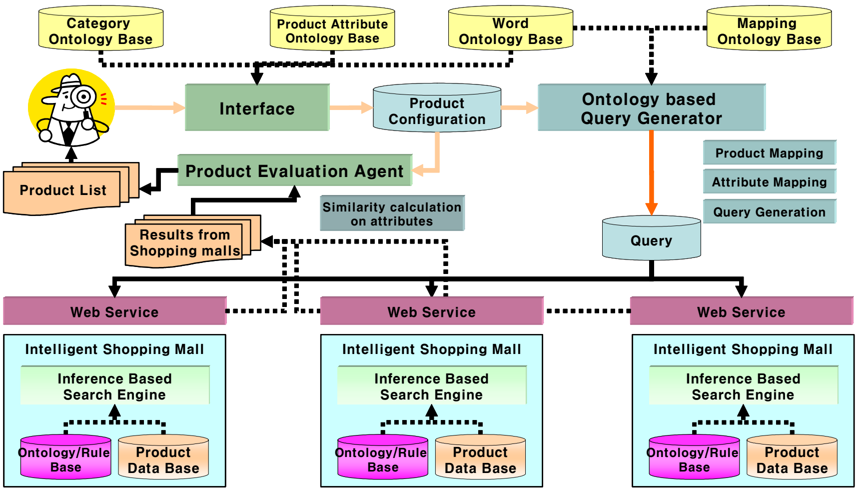

Fig. 1. Architecture of the IPIS system

3. Intelligent Product Information Search Procedure

The IPIS procedure consists of four steps as follows: configuration of a customer's purpose, query generation, query execution, and product evaluation.

3.1. The Configuration of a Customer's Search Purpose

To represent a customer's configuration, we used *Open Directory Project*[2] as the category ontology base. Let us describe the configuration process. First, a customer selects the desired product from the category ontology base, and then inputs attributes and his own priorities of attributes. For example, a customer selects 'TV' from category ontology and selects 'monitor size', 'manufacturer', and 'price' as concerned attributes for decision-making. Then, the customer inputs the criteria of attributes such as '> 15 inch' for 'monitor size and '< $600' for 'price'. Finally, the customer assigns his personal priority to attributes with numbers among 1 and 10 such as 8 for 'price' and 10 for 'monitor size' which means that 'monitor size' is more important to him than 'price'. We employ a semantic tree structure to represent the customer's configuration including product category, names, attributes, criteria, and priorities of attributes in structured form. Fig. 2 shows an example of the customer's configuration with a semantic tree structure.

3.2. Query Generation

Query Generation is the most important step in the IPIS system because an appropriate query for each

---

[2] http://www.dmoz.com/

shopping mall can give an appropriate answer. To make an appropriate query, the IPIS system transforms an original query generated from the customer's configuration into individual queries for each mall because they are using different taxonomies for a product category. The differences of taxonomies come from a product category hierarchy, product names, and name, type, and unit of attributes.

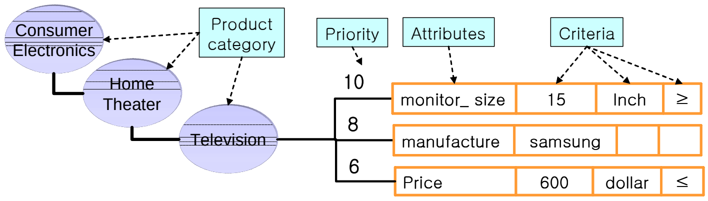

Fig. 2. An Example of a Semantic Tree for Customer Configuration

The ***Query Generation*** procedure consists of three modules: ***Product Mapping***, ***Attribute Mapping***, and ***Query Generation***. ***Product Mapping*** searches for the product name from the mapping ontology base for each shopping mall. If it fails, the ***Product Mapping*** module extends the product category structure to find a similar product for each shopping mall. The structure of the product category can be extended to various structures by using hypernyms and hyponyms of the product name in WordNet[3] [1]. The suitability of a product for the modified query is determined by our measures such as co-occurrence and order consistency. We omitted detailed algorithm because of the page limit. After ***Product Mapping***, the ***Attribute Mapping*** module determines attributes for the modified query in a similar manner. If there is any difference of type and unit of attributes between the original query and the modified queries, ***Attribute Mapping*** converts the type and unit properly. As the last step, ***Query Generator*** generates queries for each shopping mall after the completion of all mappings. In the demo, the IPIS system transforms the original query into two types of queries for Amazon.com and Buy.com. Fig. 3 shows an example of original query and modified queries for each mall which are represented with RDQL[4]. ***Query Generator*** sends modified queries to corresponding malls.

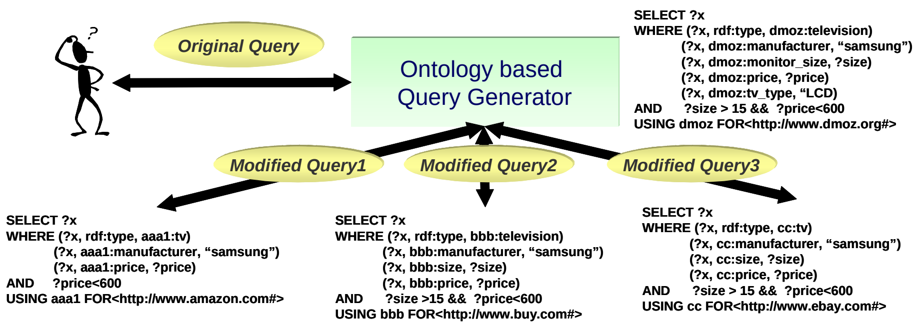

Fig. 3. An Example of Interpreted Queries for Various Malls

3.3. Query Execution

In the real world, the search system of each shopping mall is heterogeneous without any standard. Some shopping malls might provide just a keyword search and directory service, and others might provide extended search functions on various product properties using a database system. But we expect that search services using Web Services will be available in the near future regardless of the internal structures of search systems.

We implemented the product search system by employing an ontology base and rule base which consists of OWL rules for reasoning. The search system exploits an inference based search engine as one of the Semantic Web search engines that can perform various searches on product properties like a database query. Each shopping mall returns the search results after query execution. We constructed two prototypes of intelligent

[3] http://wordnet.princeton.edu/
[4] http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/

shopping malls by implementing the above search engine and ontology with the information from Amaon.com and Buy.com.

3.4. Evaluation of Products

We used general a similarity calculation method to evaluate products. First, we calculated distances between the customer's configuration and searched for products based on every attribute. The distance of each attribute is calculated by comparing names, values, and units of the attribute. Second, the customer's priorities for each attribute are changed to attribute weights. Third, we calculated the weighted sum of distances. Finally, we ranked products by their similarities to the customer's configuration. In using our prototype system, users can search and compare products of Amazon.com and Buy.com, which we have constructed as examples of intelligent shopping malls. Fig. 4 shows the example of product search results in the IPIS system which is implemented using java language. In Fig. 4, the upper right corner shows the customer's configuration and the lower part shows the search results for the configuration from Amazon.com and Buy.com.

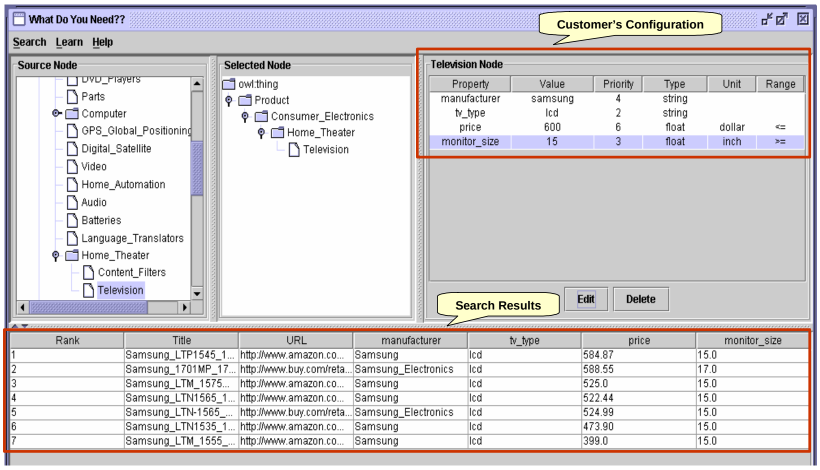

Fig. 4. Product Search Results in the IPIS system

4. Conclusion

We have designed and implemented an intelligent product information search framework using ontology mapping and Web Services which has a taxonomy free architecture for meta-search and comparison. Although shopping malls of our systems have the same inference system, our architecture can be easily extended to heterogeneous systems by using the Semantic Web and Web Services. From our experiment, we were able to find five products satisfying all criteria from Amazon.com with the IPIS system while we found four satisfactory products out of 19 returned products through several trials of the keyword search. Consequently, the accuracy of search results is increased and customers can save the time required for various keyword searches. We expect that our system will show the opportunities and challenges of the Semantic Web and Web Services in electronic commerce.

References

[1] Howe, A. E. and Dreilinger, D., "Savvy Search: A Metasearch Engine that Learns which Search Engines to Query," AI Magazine, vol. 18, no. 2, 1997, pp. 19-25.

[2] Kalfoglou, Y. and Schorelmmer, M., "Ontology mapping: the state of the art," The Knowledge Engineering Review, v.18, no.1, 2003, pp.1-32.

[3] Lawrence, S. and Giles, C. L., "Context and Page Analysis for Improved Web Search," IEEE Internet Computing, vol. 2, no. 4, 1998, pp. 38-46.

[4] O'Keefe, R. M. and McEachern, T., "Web Based Customer Decision Support Systems," Communications of the ACM, vol. 41, 1998, pp. 71-78.

[5] Selberg, E. and Etzioni, O., "The MetaCrawler Architecture for Resource Aggregation on the Web," IEEE Expert, vol. 12, no. 1, 1997, pp. 11-14.