Available online at www.sciencedirect.com

SCIENCE DIRECT®

Electronic Commerce Research and Applications 5 (2006) 16–28

Electronic Commerce Research and Applications

www.elsevier.com/locate/ecra

Building an operational product ontology system

Taehee Lee [a,*], Ig-hoon Lee [a], Suekyung Lee [a], Sang-goo Lee [a], Dongkyu Kim [b], Jonghoon Chun [b], Hyunja Lee [c], Junho Shim [c]

[a] School of Computer Science and Engineering, Seoul National University, Seoul 151-742, Republic of Korea

[b] CoreLogix Corp. RIACT #414, Seoul National University, Seoul 151-172, Republic of Korea

[c] Department of Computer Science, Sookmyung Women's University, Seoul 140-742, Republic of Korea

Received 29 November 2004; received in revised form 19 June 2005; accepted 16 August 2005
Available online 18 November 2005

Abstract

A base of clearly defined product information is a key foundation for an e-commerce system. The manipulation and exchange of semantically enriched and precise product information can enhance the quality of an e-commerce system and offer a high level of interoperability with other systems. Product information consists of product attributes and the relationships between products. Product categorization (or classification) is one type of such relationships. Ontology can play an important role in the formalization of product information. Although the idea of utilizing ontology for e-Catalogs has been raised before, we are yet to find an operational implementation of applying ontology in the domain. In this paper, we report on our recent effort to build an operational product ontology system for a government procurement service. The system is designed to serve as a product ontology knowledge base; not only for the design and construction of product databases but also for search and discovery of products. Especially, the keyword-based searching over product ontology database demands different techniques from those over conventional document databases or relational databases, and should be designed to reflect particular characteristics of product ontology. We also introduce some other issues that we have experienced in the project, and those issues include product ontology modeling, ontology construction and maintenance, and visualization. Our work presented herein may serve as a reference model for similar projects in the future.

Keywords: Product ontology; Ontology; E-catalog; E-commerce; Product information management

1. Introduction

Creating ontology for a domain provides an opportunity to analyze domain knowledge, make domain assumptions explicit, separate domain knowledge from operational knowledge, provide common understanding of the information structure, and enable reuse of domain knowledge [1]. The created ontology provides a reference domain model that both human and software can refer to for various purposes such as search, browsing, interoperability, integration, and configuration [2]. Research has been very active in this area in recent years including the efforts for building foundation for using ontology (e.g., standard semantic markup languages such as RDF and OWL), especially in the context of W3C's Semantic Web [3].

Although there have been a rich amount of research in ontology, there are still gaps to be filled in actual deployment of the technology/concept in a real life commercial environment. The problems are hard especially in those applications that require well-defined semantics in mission critical operations. In this paper, we present some of the challenges in building an ontology for a commercial based on our current project; building of a product ontology for a procurement system.

Product information (often in the form of e-catalogs) is an essential component in e-commerce. A base of precisely and clearly defined products and services is a necessary foundation for collaborative business processes. By sharing

* Corresponding author. Tel.: +82 2 883 9235; fax: +82 2 873 5516.
E-mail address: thlee@europa.snu.ac.kr (T. Lee).

1567-4223/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.elerap.2005.08.005

a precise product model containing rich semantics, a high-level of interoperability can be offered for e-business systems. One possibility for improvement is dramatically improved supply chain management. With more accurate and up-to-date information on current inventories and demands, more accurate production plans can be formed with significant savings in costs. A strong support for semantics of product data and processes will allow for dynamic, real-time data integration, and also real-time tracking and configuration of products, despite differing standards and conventions at each stage.

E-procurement is another area that can benefit from well defined product information. The processes of registering a product, searching for a product, registering a buy request, adding new suppliers, and ordering and settlement all require accurate product information. In January 2004, we have launched a project to build an ontology based e-catalog system, named KOCIS for the Public Procurement Services of Korea.[1] The purpose is to provide a universal ontology repository with browsing, searching, and downloading capabilities in order to facilitate e-catalog sharing and interoperability.

Public Procurement Services has built the online G2B (Government-to-Business) e-procurement system[2] for procurement of commodities for public and government organizations. Since its launch in September 2002, the system has been a huge success taking care of over 90% of the government procurement transactions. However, there are still rooms for improvements. Product registration still relies on manual efforts of more than 50 domain experts. It not only makes maintenance costs high, but also results in inconsistent product database; incorrect product classification, duplicated product registration, inadequate attribute assignment, etc. Additionally, the G2B system must support interoperation with commercial marketplace systems. Thus, the database must be well organized, semantically rich, and promote standardization. We believe an ontology based product database provides a solution to these requirements.

Fig. 1 presents our business model showing the interactions around our product ontology system. The participants are classified into three types: Administrators, Internet Shops, and Consumers.

- **Administrators**. Administrators should do several cleansing processes such as duplication check and product classification whenever to build an e-catalog for new product information. Because ontology includes not only product schemes and product information but their semantic relationships, it may respond dynamically to on-line requests for product information. For administrators, KOCIS provides a low-overhead and semi-automated cleansing processes for their product information. Reduced building costs and extended search capabilities are the principal benefits for them.
- **Internet shops**. An Internet shop is a distributor, wholesale or retail, who typically sets up a Web site of products and sells the products. Such a shop can download from KOCIS the core catalogs of products it carries and customize them to meet its own particular style and requirements, instead of building them from scratch. Automated procedures can be set up to periodically poll KOCIS for updated product information. Reduced catalog building costs and access to up-to-date product information are the principal benefits for these shops.
- **Consumers**. A consumer may be an individual purchasing personal items or the procurement department of a company ordering industrial supplies. Consumers will typically search and browse the shops for products and services they need. For those shops that conform to catalog exchange standards, one could query multiple shops concurrently and construct an integrated catalog or compare prices. Whether to conform to the standards and participate in such queries is a business decision that an individual shop has to make. However, should a shop decide to conform to the standards, KOCIS can facilitate such conformance by providing the core e-catalogs in standard forms in the first place. In addition, consumers may query KOCIS directly and be referred to the shops that have downloaded core e-catalogs of the resulting product(s), in which case, KOCIS plays the role of a shopping directory. So, the benefit to the consumer is that KOCIS facilitates catalog interoperability, which in turn will enable services for more effective decision making.

The main focus is the development of a system of ontologies representing products and services, which includes the definitions, properties, and relationships of the concepts that are fundamental to products and services. The system will function as a standard reference system for e-catalog construction. It will supply tools and operations for managing catalog standards. The ontology will function as a knowledge base, not only for the design and construction of product databases but also for search and discovery of products and services.

We allow product search by merely issuing keyword queries without any knowledge of the underlying product ontology schema. Processing keyword search queries over product ontology databases introduces new challenges. First of all, the ranking function for it must be different from the one used for conventional document search. It must be able to augment the notion of relationships that exist among different products. In addition, keywords used in a query may be values, attribute names, category names, and even relationship names; whereas in a typical keyword search in relational databases they are assumed to be confined in values of attributes only. Our approach enables

[1] Public Procurement Services (PPS, http://www.pps.go.kr.) is a government agency responsible for procurement for government and public agencies in Korea.

[2] http://www.g2b.go.kr.

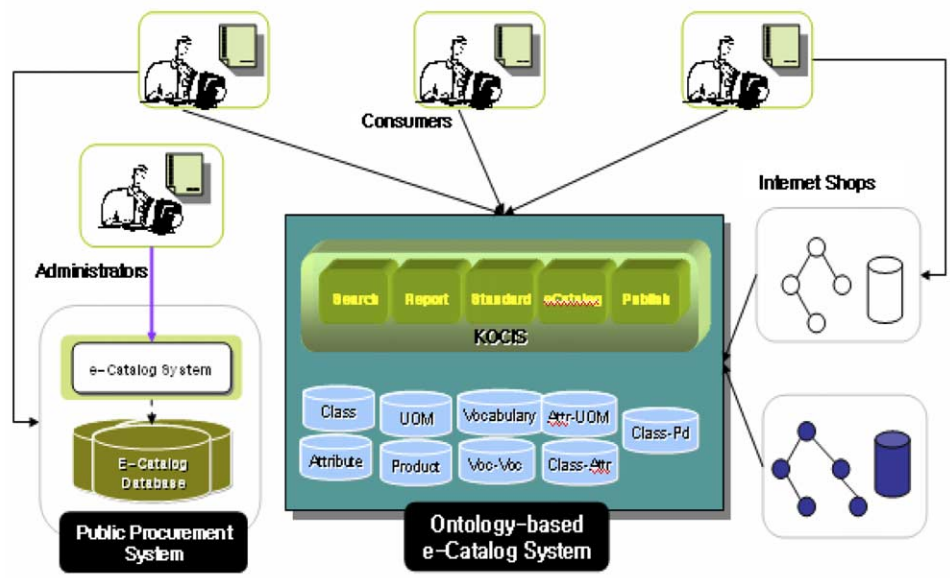

Fig. 1. Business model of our product ontology system.

users to access product ontology directly from keyword search interface. We concentrate ourselves to present a method to rank product ontology keyword search results. Our ranking is based on probabilistic model which naturally generalizes a naïve Bayesian belief network. We propose how this can be adapted to rank query results in product ontology databases and delineate how the ranked results can be computed in a timely manner.

The rest of this paper is structured as follows: Section 2 describes our ontology model and logical design for a product information management. In Section 3, we present implementation of ontology-based information system, and discuss several related issues and their solutions. After some related works are presented in Section 4, conclusions are drawn and future work is outlined in Section 5.

2. Product ontology models

2.1. Meta modeling of product ontology

The ontological modeling is an inherent process for building an ontology application regardless of the application domain. After the domain analysis, one needs to first conceive the key concepts and their relationships which may best portray the domain. In our product ontology, we regard *products*, *classification scheme*, *attributes*, and *UOMs* as the key concepts. The products, the most important concept, are for the goods or services. The classification scheme and the attributes are used for the classifications and descriptions of products, respectively. The UOM is short for the unit-of-measures and it is associated with the attributes.

Fig. 2(a) illustrates our view for product ontology using the meta-model approach [4,5]. The meta-modeling approach enables a product ontology model to be more extensible and flexible. Our product model follows the basic meta-model which employs three modeling-levels: M0 meta-class level, M1 class level, and M2 instance level. Within the M0 level which describes a high level conceptual product ontology, we have the aforementioned key concepts as the meta-classes. As illustrated in the figure, the meta-classes may have relationships (meta-relationships) each other. The relationships are shown in Fig. 2(b). They include semantic relationships from general domain; such as *class inclusion (isa)*, *meronymic inclusion* (*component, substance, and member*), *attribution*, and *synonym*. In addition, product domain specific relationships such as *substitute*, *complement*, *purchase-set*, *mapped-to* are also considered. As an example of the relationships, a pencil is a *substitute* of a ballpoint pen in that each may role as a replacement of the other. Note that in the figure meta-class and meta-relationships are named with the prefix ‘M_’ to indicate that they are of the meta concepts.

The M1 class level contains a snapshot or instance of the product ontology model in M0. That is, it illustrates a class schema of our PPS product ontology database. The conceptual class schema is then translated into its logical schema managed by the operational DBMS. For example, we use Oracle 8i as the operational DBMS, and therefore the logical schema in our case is a set of object-relational tables and views. Fig. 3 illustrates a part of the class schema of our PPS product ontology database. Note that our logical schema is not included in the figure.

And finally, the M2 instance level refers to the physical ontology data managed by the system. For example, *notebook* and *LCD panel* products in M1 level are instances of *products* meta-class in M0 level, and there is a *component* relationship between them, i.e., a notebook *contains* a LCD panel and a LCD panel is a *component of* a notebook. Note that *contains* and *component-of* are in inverse relation each other and they are the instance relationships of *com-*

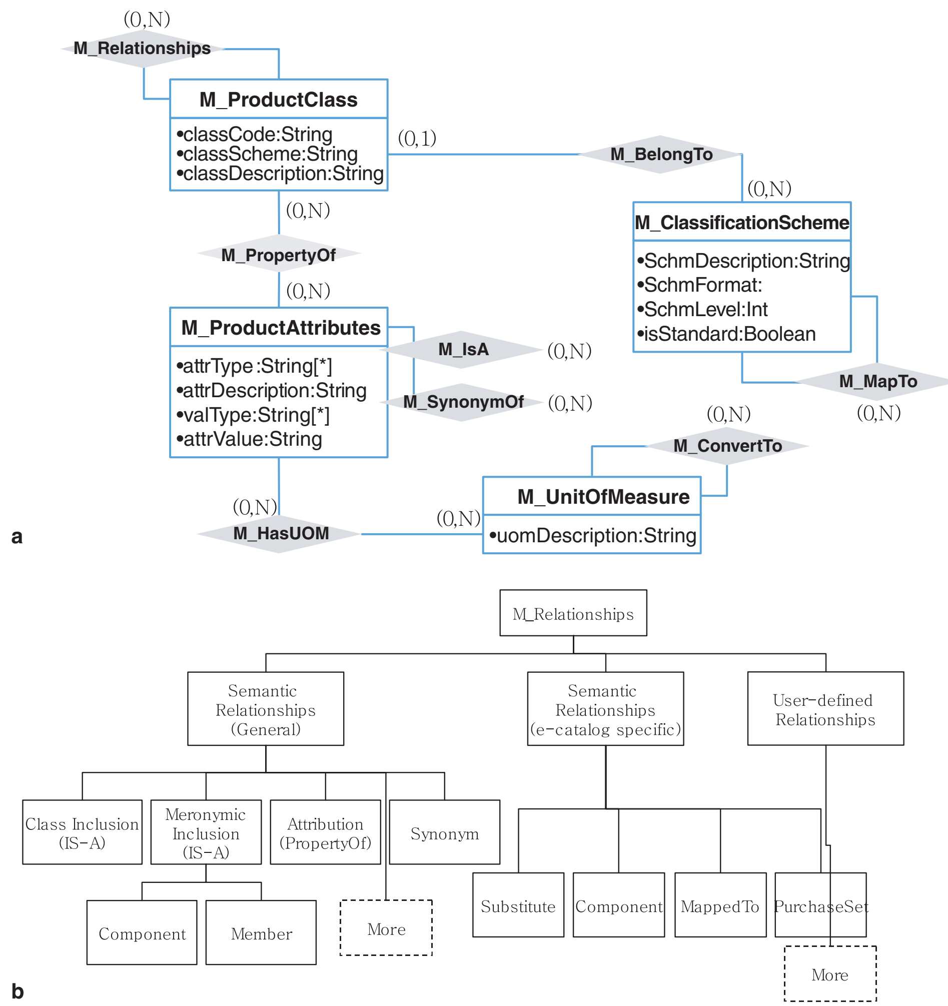

Fig. 2. (a) Product ontology model in meta-level. (b) Meta-relationships hierarchy for product ontology.

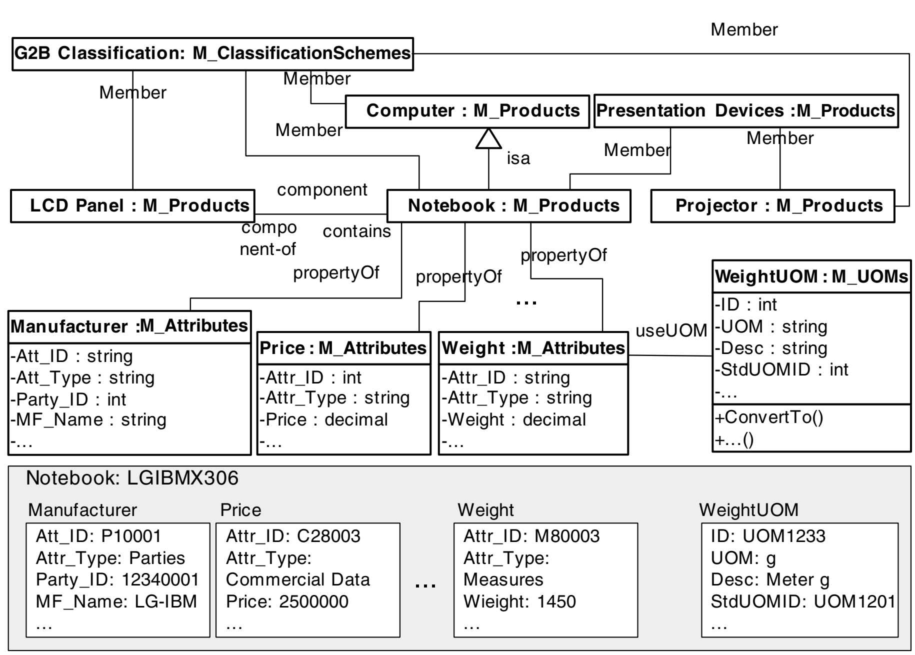

Fig. 3. Product ontology model in conceptual level and an exemplary instance product.

ponent meta-relationships existing between *products* meta-classes. A notebook has attributes (described as *propertyOf* relationship) such as manufacturer, price, weight, and so on. Then an individual notebook product, IBMX306 should appear in M2 level. Readers who are interested in the detail of our product ontology model including the types of semantic relationships for the product domain are referred to [6].

2.2. Model implementation and technical dictionary

Our modeling goal is not only to design a 'conceptual' product ontology model but also to implement it as an operational ontology database model. One way to achieve this goal may be through using an ontology language such as OWL and building an OWL knowledgebase that represents the intensional and extensional concepts and relationships for our product ontology. This approach was not taken into the consideration simply because we were not able to find any robust enough OWL engine to practically handle a large knowledgebase as ours. In addition, we only needed rather a limited-set of reasoning capabilities such as transitivity and inverse. Naturally the general purpose reasoning capability such as having an OWL engine was considered as an over-kill.

Forth, we have built our ontology database on top of a commercial object-relational DBMS, and implement the ontology subsystem to provide just enough reasoning capabilities along the products, classification scheme, attributes, UOMs, and their relationships. Fig. 4 shows an example to illustrate this reasoning (or inference). For example, when a user inquires any concept containing the word "LCD", the ontology subsystem may find two concepts: *LCD Panel* product class and *Panel Type* attribute class. Let's say that the LCD Panel product is mapped to the 43172410 commodity class under a certain standard classification scheme, and according to the same classification scheme, 43172410 class is supposed to have attributes A1, A2, A3,...,Ak. Now we can infer the association between the word LCD and the attribute A2 and this illustrates the reasoning path between them.

We create an object-relational product ontology database consisting of more than 40 base tables to reflect the aforementioned conceptual and class schemas. The ontology construction subsystem (Section 3.1) populates the data into our product ontology database from the existing PPS relational database. One novel feature that our ontology database provides is to organize and view the product information in the form of technical dictionaries. Technical dictionaries, shortly TDs, are quite used in the e-Catalog domain to describe the products and their properties. The well-known TDs may include eOTD, GDD, and RNTD of ECCMA, EAN/UCC, and RosettaNet, respectively [7–9]. Compared to these TDs, our TD holds far more rich set of information that are required for machines to operate on products intelligently; such as the product's attributes and its relationship to other products (related products, similar alternatives).

We have constructed a number of technical dictionaries, in a way along which we can reflect the core concepts and relationships explained in the conceptual product ontology model. The contents of each TD are extracted and organized from the different sets of underlying ontology base tables.

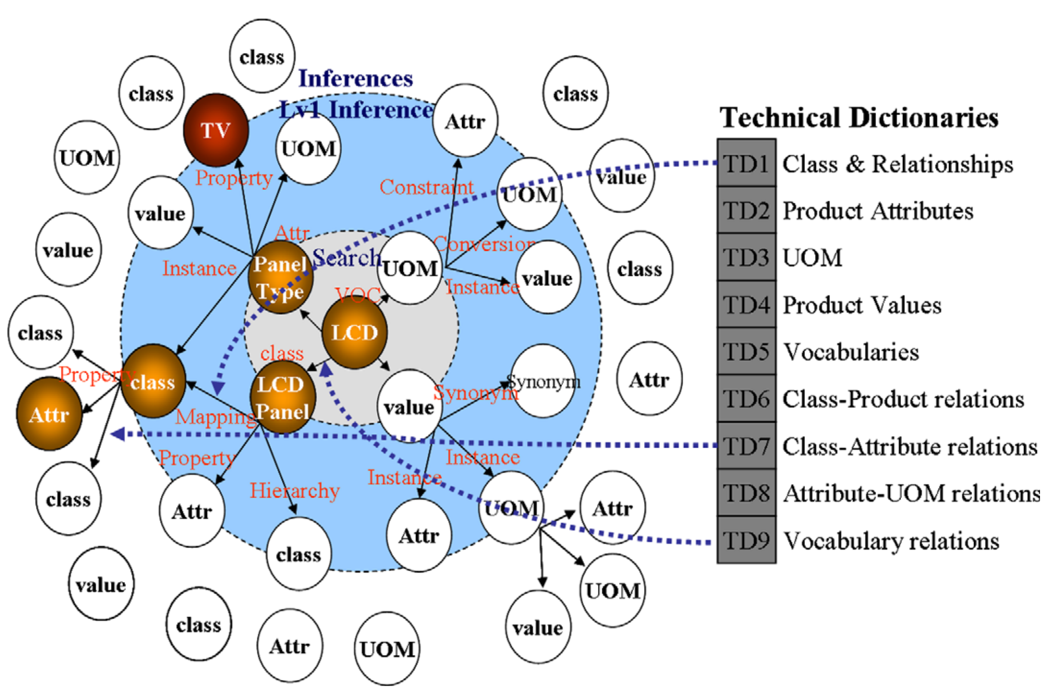

Fig. 4. Reasoning path between concepts and technical dictionaries.

As an example of a TD, Fig. 5 shows a part of the classification technical dictionary. Within the dictionary, we have a product classification for *personal computers*. Under this specific classification scheme (g2b), it has a class code of 43171825, and it is also called as PC, desktop computers, or workstations (synonyms field). In general, a product class may be defined or classified differently depending on classification schemes. We can find such information in the code mappings. For example, the product *personal computers* is mapped to 43171803 in UNSPSC classification system, and 8471–10 or 8471–41 in HS code system. It has component (contains) relationships with CPU, HDD, RAM, and monitors, and substitute relationships with notebook computers. And we may also see that such products as monitors, OS, and mouse are also purchased with a personal computer.

Note that each item within a same TD might have different columns to the ones that other items have, in that they can have different classification code mappings and relationships in actual contexts. For example, while the item *personal computer* has three columns of code mappings for UNSPSC, HS, and GUNGB, another item *LCD monitors* has just two columns of code mappings for UNSPSC and GUNGB. This means that in the actual product ontology database the item *LCD monitors* does not have any mapping relationship to HS. Similarly, the types of relationships that personal computers and LCD monitors have are different; personal computers are to have components but no supplements while LCD monitors are to have supplement but no component products.

3. Ontology subsystems

In this section, we introduce implementation issues for developing our product ontology system. The ontology system consists of two major subsystems; *ontology construction and maintenance system* and *ontology search system* (Fig. 6).

Ontology construction and maintenance subsystem populates the data into the product ontology database from the existing PPS relational database, and maintains the consistency and the synchronization of the ontology database. Ontology search subsystem helps users to navigate through or search for the domain knowledge stored in the ontology database. We use Oracle 8i object-relational DBMS to manage and operate our product ontology database. For compatibility with existing PPS system, we use Java2EE 1.3 and Bea WebLogic 6.1 to build the system applications, and our system is currently running on Solaris OS v5.9.

class name	G2B code	description	synonyms	code mappings			relationships			etc
personal computers	43171825	A computer built around a microprocessor for use by an individual …	PC, desktop computers, workstations	UNSPSC	HS	GUNGB	component	substitute	purchase-set	…
				43171803	8471-10, 8471-41	7010300	CPU, HDD, RAM, monitors,…	notebook computers	monitors, OS, mouse,…	
LCD monitors	43172410	A low-power flat-panel display used for …	LCD, liquid crystal displays, flat panel displays	UNSPSC	GUNGB		supplement	substitute	purchase-set	…
				43172402, 4317240, 431724	7025366, 7025302		arm stand, antiGlare filter	CRT monitors	personal computers	
…	…	…	…	…			…			…

Fig. 5. An example of a technical dictionary: G2B classification TD.

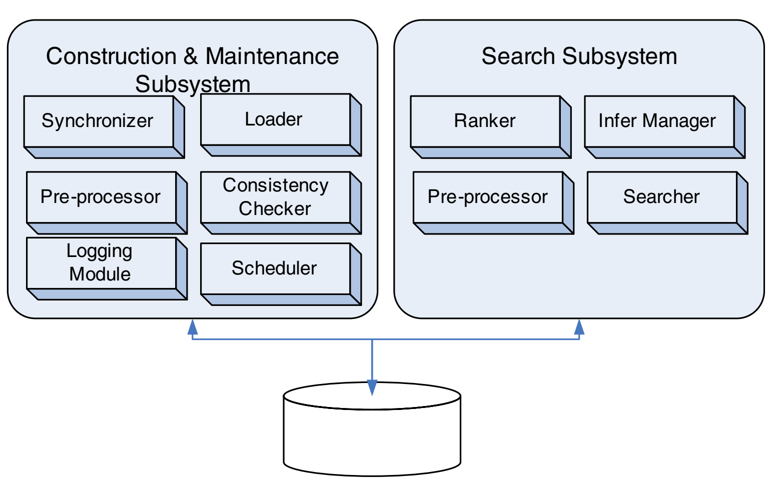

Fig. 6. The components of ontology subsystems.

3.1. Ontology construction and maintenance system

Our product ontology database is built in a batch fashion by bulk-loading and transforming the data from the existing PPS catalog database. The transformation at this phase is done automatically by the system. Since entities that have references to other entities should be built later than the referenced entities, transformations should be performed in a specific order to preserve the reference dependency of each entity. In our case, the order is classification schemes, attributes, UOMs, and products. At the end of the construction process, we performed the quality checking and cleaning manually. Then, technical dictionaries are built upon the product ontology database.

Technical dictionaries are implemented as database views in our system. That is, the actual contents of TDs are not materialized but queried on-the-fly on corresponding views. Materialized views are not taken into account for the system portability purpose in that some object-relational DBMSs do not allow the materialized view to contain objects within a table. Nevertheless, with the tradeoff the data redundancy and maintenance cost, TDs can be implemented using separate tables and triggers to synchronizing the TD tables and base tables.

Since the automatically constructed ontology may not show the quality that the domain experts expect, methods for improving the quality are required. Among them we discuss the following subjects: catalog preprocessing, consistency checking and synchronizing with the catalog databases.

- **Catalog preprocessing**. We considered some conventional IR (information retrieval) techniques to parse and extract meaningful vocabularies from source catalogs. The techniques we employed include lexical analysis, elimination of stopwords, and stemming [10]. In addition to these typical document preprocessing techniques, techniques for the catalog annotation and measure phrase extraction are added to improve the preprocessing quality.

  The catalog annotation is to choose a set of words that best describe the concepts in the catalog. In [11], a universal dictionary for all topics of the domain was constructed and indexing terms were selected from the dictionary for a document representation. Similarly, we brought in a general-purpose dictionary and performed a lexical analysis on every vocabulary appeared in the product catalog to annotate it. By doing this, a set of smaller words could be extracted from those terms in the catalog. For example, we can extract "CD" and "writer" from "CDWriter" by the lexical analysis if the dictionary has either of the words. This was effective to extract meaningful words from unknown terminologies and compound words, which are not in the dictionary, especially in Asian languages where the word spacing rules are not strictly followed.

  The measure phrase extraction is to identify a sequence of measures from the product descriptions which include numeric values, UOMs, or measure conjunctions (e.g., 120 cm × 80 cm × 70 cm to describe the size of a desk). Numeric descriptions with UOM or measure conjunctions should be carefully managed in that some people prefer to use a single measure while others do a series of measures; e.g., 120 cm × 80 cm × 70 cm vs. 120 × 80 × 70 cm × cm × cm. We regard each pair of numeric value and UOM as well as a conjunction of them as a vocabulary that describes the product.

  A numeric measure can be converted to other UOMs that have same meaning (e.g., 1 m equals to 100 cm). We defined a set of standard UOMs and a set of conversion rules between standard UOMs and their related UOMs Then each finding of non-standard UOMs is substituted with its standard UOM. For example, we used "meter m" as a standard UOM for the measure of lengths and built a set of conversion rules such as "cm = m*0.1" or "km = m*1000". And numeric descriptions such as 10 cm or 0.1 km were converted to 0.1 or 100 m, respectively, to have its standard "meter" measure.
- **Synchronization**. From the system architectural perspective, the ontology system is a part of the PPS e-catalog system, which works as an ontology knowledgebase (Fig. 1). All updates on the source catalog database should be propagated towards the ontology database as soon as they are detected. We considered two approaches for implementing the online synchronization; a database-level approach and an application-level approach.

  In a database-level approach, as an update occurs on a source table the DBMS automatically triggers the corresponding maintenance module. This approach is a simple and easy way of implementing synchronization process if it is supported by the DBMS functionalities. It is, however, inappropriate due to the performance degrade as the database schema grows or more number of maintenance modules are required.

  In an application-level approach, specially designed logging modules are implemented to cover every update cases. It gives flexibilities on the system design by allowing database schema evolving, efficient scheduling of batch updates and distributing servers over the network. Pacitti and Simon [12] suggested the application-level architecture for maintaining replica's consistency while minimizing the performance degradation due to the synchronization of refresh transactions. We adopted the consistence criteria and the system architecture from [12] and optimized the algorithm to work in a single copy environment.

  Fig. 7 illustrates the architecture of our synchronization system. As the catalog database changes, the Propagator sends the ontology system Java RMI messages that describe the types of changes and changed contents. The RMI messages are in XML format. The Receiver stores the reception logs in a database table. The

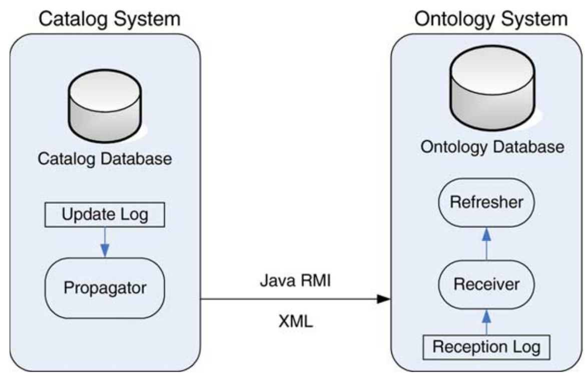

Fig. 7. Architecture of synchronization subsystem.

Refresher schedules the synchronization process considering the system load accordingly.

3.2. Ontology search system

Product ontology has a large number of concepts and relationships. For naive users to search and navigate the ontology efficiently, the ontology system should provide good search and visualization functionalities. We allow product search by merely issuing keyword queries without any knowledge of the underlying database schema. This is significant because database schema for product ontology varies in a wide variety of forms, ranging from universal tables to attribute-value pairs. In reality, it is virtually impossible for any product ontology users to have enough knowledge on the underlying schema to be able to issue queries in query languages such as OWL-QL or SQL anyways. Besides, those query languages are too complex to learn and standard SQL may not be sufficient enough to effectively reference ontology data.

In this subsection, we consider some searching related issues when the ontology database is large and operated by a relational database management system, as most of successful e-commerce systems nowadays operate on e-catalogs set up on relational DBMSs'. Our approach enables users to access product ontology directly from keyword search interface. In contrast, other approaches let users to access product ontology either by specifically-tailored ontology SQL operators or via APIs.

3.2.1. Ranking keyword query results

The problem of processing keyword search queries over product ontology databases is very much different from that of processing queries over conventional document databases, relational databases, or Semantic Web. Obviously, the ranking function for it must be altered because it now deals with product ontology instead. It must be ripened to reflect peculiar characteristics of product ontology, such as augmenting the notion of relationships that exist among different products while utilizing IR-style information such as term proximity.

Consider the following example of product ontology in Fig. 8, in which we show the necessity of probabilistic approach and keyword proximity. If we issue a product class search query "IBM P4 3.0 GHz" intending to find a UNSPSC code for it, the system would return both 48171803 (desktop computers) and 48171801 (notebook computers). Conventional cumulative ranking functions such as PageRank [13] would rank desktop computers higher than notebook computers simply because they would determine the global importance of desktop computers is greater than that of notebook computers, i.e., the number of incoming edges for desktop computers is greater. However, from a probabilistic point of view, the keywords "IBM, P4 3.0 GHz" are more likely relevant for notebook computers since every product in notebook computers is "IBM P4 3.0 GHz" whereas only half of the products in desktop computers are "IBM P4 3.0 GHz".

In a product ontology search, keywords used in a query may be values, attribute names, category names, and even relationship names, etc; whereas in a typical keyword search in relational databases they are assumed to be confined in values of attributes only. Therefore, the notion of keyword proximity in product ontology contributes even more significantly to the computation of ranking than in conventional document search. Naturally, we do not want to disregard that information. Even for a simple query "IBM graphic card ATI X300" to find a product with brand name IBM with a graphic card "ATI X300",

"IBM ThinkPad G41" must be ranked higher than "IBM S50" since the distance between "IBM ThinkPad G41" and "ATI X300" is closer than the distance between "IBM S50" and "ATI X300".

3.2.2. Probabilistic similarity computation

We model the product ontology as a Bayesian belief network [10] as in Fig. 9. In this model, the user query q is modeled as a binary random variable which is pointed to by the index term nodes which compose the query concept. Ontology data (concepts) are treated analogously to user queries. They are also modeled as nodes which are pointed by the index term nodes in the concept. In this model, $P(d_j|q)$ is adopted as the rank of concept d_j with respect to the query q, and it is computed as

$$P(d_j|q) \cong \sum_{\forall k} P(d_j|k)P(q|k)P(k). \qquad (1)$$

In Eq. (1), $P(d_j|k)$ can be estimated by $n_{d,k}/n_k$, where $n_{d,k}$ and n_k denote the number of occurrences of k in d_j and the number of occurrences of k in entire ontology, respectively. Alternatively, if we normalize the value by the size of data, $|d_j|$, then $P(d_j/k)$ can be estimated as $k/|d_j|/\sum_{d\in D} k/|d_j|$, where D is the set of concepts in the ontology.

To model the semantically related concept r with the d_j, we extend the Bayesian network model to incorporate the relevant concept node r. As shown in Fig. 9, we add a relevant concept node r_i, which represents a semantically related concept between the index term nodes and the concept node d_i. Then we can compute the $P(d_j|k)$ as follows:

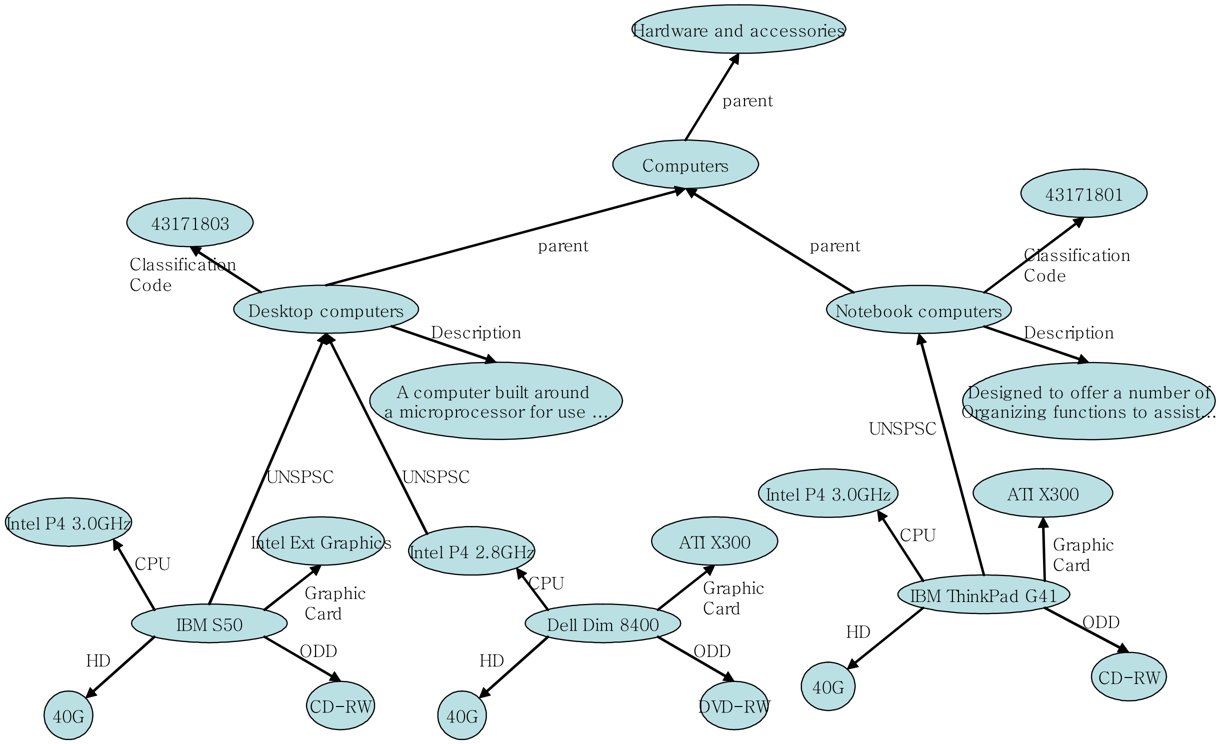

Fig. 8. An example of the product ontology graph.

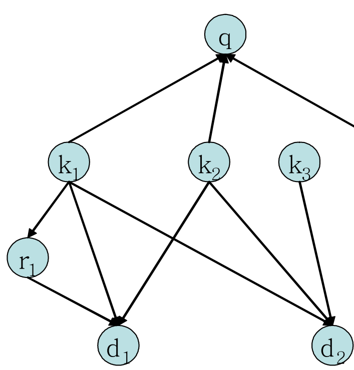

Fig. 9. Bayesian belief network.

$$P(d_j|q) \cong \sum_{\forall k} P(d_j|q)P(q|k)P(k)$$
$$= \sum_{\forall k} P(q|k) \times \left(P(d_j|k) + \sum_{\forall r} P(r|k) \times P(d_j|r) \right) \times P(k). \quad (2)$$

In Eq. (2), $P(d_j|r)$ represents the degree of belief on d_j when r occurs. Even though the exact value is not known at the time of ontology construction, it can be estimated by using heuristics or can be specified by an administrator. For example, $d_{\text{in},r}/r_{\text{out}}$, where $d_{\text{in},r}$ is the number of incoming edge of d from r and r_{out} is the number of outgoing edges from r, can be used. Notice that the Bayesian Network becomes naïve Bayesian classifier when $d_{\text{in},r}/r_{\text{out}}$ is 1.

Given the probabilistic similarity computation defined in Eq. (2), we need an algorithm that ranks the query results. We provide a simple ranking algorithm which iteratively approximates the ranks of concepts. The NaïveAlgorithm is summarized in Fig. 10. The weighted ontology graph, target concept class, and the maximal level of inferences must be given to the algorithm. By maximal level of inferences k, we denote the number of inferences that can be applied to each instance. Initially, it computes $P(d_i|k)$ at each node d_i ($1 \leqslant i \leqslant n$) of ontology schema graph and makes a vector S: $\langle P(d_1|k), P(d_2|k), \ldots, P(d_n|k)\rangle$. Then it makes a $n \times n$ vector E where each element $e_{i,j}$ is $P(d_i|d_j)$ defined by the administrator or estimated by link analysis. By $E \times S$, we can compute the score of instances after an inference. It iterates $E \times S$ l times, where l is a user-defined threshold. The NaiveAlgorithm is very similar to PageRank and HITS in that it iteratively computes instances'

```
NaïveAlgorithm(G, C, k, l, n)
        G : ontology of n instances;
        k : a set of keywords
        l: maximal level of inferences
        n : the number of returned results

        For each node d_i in G, compute P(d_i|k);
        Let S denote the vector : <P(d_1|k), P(d_2|k), ...,
        P(d_n|k)>;
        Let E denote the n * n vector where each element
        e_i,j is P(d_i|d_j);
        For i = 1, 2, ..., l
            S = ExS;
        End
        Return top-n elements of type C from S
```

Fig. 10. Probabilistic ranking algorithm: NaïveAlgorithm.

score by vector calculation. Kleinberg [14] shows that the S converges to an equilibrium state, an eigenvector of E, when the number of inferences increases arbitrarily. Analogously our NaïveAlgorithm converges to the equilibrium state, but we limit the maximum number of loops because it is shown that relatively small value (about 20) of l is sufficient for the vectors to become stable in [14].

The ranking method presented here is only a part of the problems that need to be addressed in keyword search systems for product ontology. However, we believe that our ranking method is intuitive and practical to use on reasonably large ontology databases. Currently, we are developing a pruning algorithm to improve the efficiency of query processing, which will later be validated by conducting a performance study.

3.2.3. Ontology visualization

Various visualization techniques have been introduced for exploring ontology entangled with relations and concepts. For the majority of ontologies where taxonomy resides predominantly, hyperbolic style diagram with few cross-taxonomical links and with very few relationships is mostly suitable for navigating data structures. However, it may not be suitable to the product ontology since a concept related with multiple relationships may have a large number of instances, and those instances may participate in different types of many relationships. A graphical method employing cross-taxonomical links for this environment seems inapplicable due to its hard readability.

Instead we employ a navigation approach to provide paths for searching other information. In our navigation interface, each concept is associated with a set of TAB link each of which represents paths to other concepts. A path associated with a concept means that the concept has relationship (path) with other concepts. The snapshot of our ontology navigation interface is shown in Fig. 11.

Let us say that you are looking at a page about a product classification 24131503 *refrigerator*. The page also has TAB links for a list of its instances (i.e., individual refrigerator commodities, related product classifications), a list of its properties and their associated UOMs. Displaying lightweight sub-ontology as a hyperbolic tree is under development.

3.2.4. Indexing

Every meaningful piece of words from the product information can be a target of keyword search. The size of vocabulary database increases rapidly as the number of products increases. Although, the basic keyword searching mechanism is supported by the underlying DBMS, the keyword searching cost is still very high. Techniques are required to improve search efficiency in consideration of difficulties in partial match. For example, we consider dividing a vocabulary table into several sub-tables each of which has a hash-function index.

Indexing techniques such as multi-dimensional indexing (B + -tree) and path indexing are under consideration. Multi-dimensional indexing (B +-tree) techniques [15] may be useful since queries on the product ontology are often made on multiple-attributes, and the order of attributes that each query include is diverse. Path indexing techniques [16] may be suitable for processing queries of which access patterns follow the certain types of navigation paths.

4. Related work

The utility and potential benefits of ontology to e-commerce have been conjectured by many practitioners and

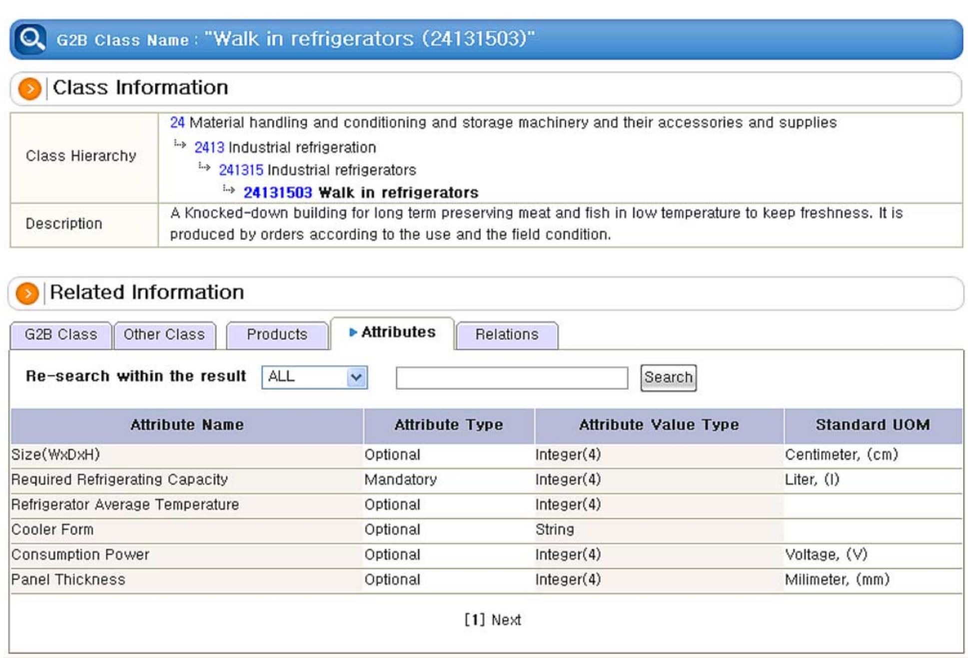

Fig. 11. A snapshot of product ontology navigation.

researchers in recent years. Two of the earlier papers are [17,18]. Fensel and Omelayenko [17] present the issues of B2B integration, focusing on product information. They list the difficult aspects of building, maintaining, and integrating product information, and propose that ontological approach may be the answer. Obrst et al. [18] proposes to use cross industry standard classifications such as UNSPSC and eCl@ss as the upper ontology and industry specific classifications as lower ontology, thus achieving the generality and specificity. These two early works are representative of a group of works (including [19,20]) focusing mainly on classification standards as the shape of ontology for product information. Classification hierarchies are essential part of product information semantics but make up only one piece of the picture. Also, their proposals are still in an abstract level.

An interesting effort is presented in [21]. They try to utilize the ISO standard for product library (PLIB [22]) to model product ontologies. However, their view of a product ontology is still limited to classification hierarchy and the PLIB standard provides a set of meta data specifications for such hierarchies.

Bergamaschi et al. [23] utilizes ontology (WordNet and designer supplied relationships) in mapping classification schemes. They do not fall into the habit of equating ontology as product classification, but ontology is used in a rather general level (than product information specific) and the focus of their work is in having applied the MOMIS system [24] in an interesting domain.

Hepp [25], Kim et al. [26], and Lee [27] are important works that emphasize the importance of attributes in product information management. Hepp evaluates the quality of product classification standards based a number of factors including the quality of their attribute lists. In [26], a data model for classification hierarchies is presented where specification of attributes and semantics is a requirement. In [27], it is pointed out that a classification hierarchy is a representation of just one of many views over the set of products. A product's identity and property does not depend on how the product is classified. Product database design issues and guidelines are presented, where the focus is on properties (attributes) rather than on classification hierarchies. Guo and Sun [28] proposes a "concept-centric engineering" approach to product representation and exchange, which also emphasizes the semantics and properties of products.

We believe that our work can be characterized as property/semantics oriented, in the sense that our view of a product ontology is the entire set of bits and pieces that make up the contents and semantics of a product database. This includes not only the classification hierarchies but also the attributes requirement for each class, the domain constraints on each attribute, the unit of measures and their relationship among each other as well as to other concepts. Another contributing feature of our work would be that our approach is bottom-up. We try to capture most of the information that is represented in a product database, either explicitly or implicitly.

5. Conclusion

We have developed a product ontology system for the product databases of the PPS, a central government procurement agency in Korea. The system consists of product ontology database and ontology subsystem. Our product ontology database is modeled using meta-modeling approach, and includes the key semantic concepts such as products, classification schemes, attribute requirement for each class and domain constraints on each attribute, unit of measures and their relationships among each others. We also organized and built a number of TD(technical dictionary)s to view the contents of the product ontologies.

The ontology subsystem provides the *ontology construction and management system* to build the ontology database from the product databases of the PPS and to manage the real-time processing for update operations while maintaining a consistency of the ontology data. The *ontology search system* retrieves and navigates the product ontology information. We also include other applications such as *personal catalog management system* to build/manage personalized e-catalogs and enable downloading of the G2B classification hierarchies and product catalogs.

A dominant and well-known modeling approach for building an ontology application is to directly employ a formal ontology language such as DAML + OIL or OWL, to represent the domain knowledge. This approach, mainly favored by the research community with deep computer science background, may be beneficiary for integrating the domain ontology model with an inference engine for the language. However, this approach is deficient in that it is technically too complicated to represent and comprehend the domain for a domain expert who has little knowledge in the formal language. More importantly, there is no publicly known robust inference engine to provide the reasoning on such a large ontology knowledgebase as our product ontology database. As of early December, 2004, our product ontology database contains 881,951 concepts, each of which is assigned to one or more semantic concepts. These concepts are linked by the semantic relationships, and the total number of the semantic links is 21,069,028. And the size of the database keeps growing.

Instead, we implement our product ontology model using an object-relational model. At the same time, we have also developed a mechanism to translate each concept and relationship into a corresponding description language in SHIQ(d) which is known reasonably practical with regard to its language expressiveness and algorithmic complexity [6]. The formal ontology model in DL may be also written in OWL-DL. That is, our PPS ontology database can be wholly exported into a standard knowledge base purely written in OWL-DL and RDF.

Our main focus was to develop an ontology system representing products and services. Our ontology includes the definitions, properties, and relationships of the concepts that are fundamental to products and services. The system will naturally function as a standard reference system for

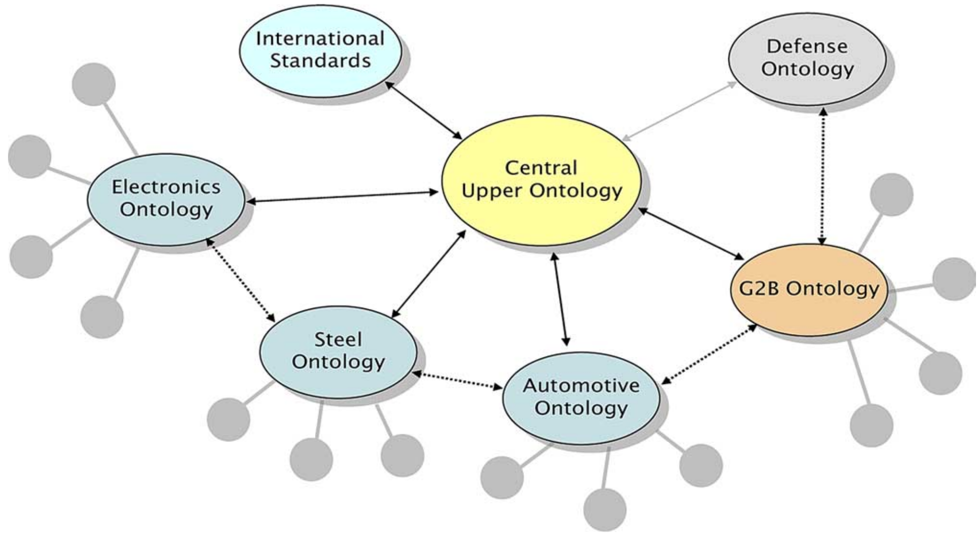

Fig. 12. e-Commerce content ontology framework.

e-catalog construction, and supply tools and operations for managing catalog standards. The ontology can function as a knowledge base, not only for the design and construction of product databases but also for search and discovery of products and services.

With the increasing importance of product information management in every day e-commerce environment, it is vital that precise definition of product and services readily available in sharable, manageable, flexible, and scalable form, namely in the form of an ontology. The paper addresses the problem of ranking keyword search results by modeling the product ontology as a Bayesian belief network. Our approach enables users to reference product ontology directly through simple keyword search interface, thus opening up the door for people with little knowledge on product ontology systems.

In the near future, we will apply the ontology system to automation of the product management process such as creation, registry, and classification of products data and to integration/interoperation among heterogeneous e-catalog systems. We also plan to develop the web services to serve the contents and functionality to the public.

Ultimately, we intend to construct an e-commerce content ontology framework (a web of ontologies for various industry sectors and purposes) as shown in the Fig. 12. A wide range of diverse business processes and product standards will be supported. The central upper ontology contains the semantic descriptions of common product catalogs and process specifications. It will provide services to suppliers and consumers of various industries who want their own business processes to be inter-linked with others.

Acknowledgements

We thank Hyung-jong Min, Hee-mun Kim, Seong-bu Han, Chong-hwan Ahn, Hyeok-ki Kwon and Sam-hyun Jo of the Public Procurement Services, Korea, for their cooperation to fulfill this project. We also thank the whole team of this project for their sincere effort to implement the system. This work was supported in part by the Ministry of Information & Communications, Korea, under the Information Technology Research Center (ITRC) Support Program.

References

[1] N.F. Noy, D.L. McGuinness, Ontology Development 101: A guide to creating your first ontology, Available from: <http://protege.stanford.edu/publications/ontology_development/ontology101.html>.
[2] D.L. McGuinness, Ontologies come of age, in: D. Fensel et al. (Eds.), The Semantic Web: Why, What, and How, MIT Press, 2001.
[3] W3C, Semantic Web, http://www.w3.org/2001/sw/.
[4] C. Atkinson, T. Kühne, The essence of multilevel metamodelingProceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools, Springer-Verlag, Berlin, 2001.
[5] J. Shim, S. Lee, C. Wu, A unified approach for software policy modeling: incorporating implementation into a modeling methodology, in: Proceedings of the 22nd International Conference on Conceptual Modeling, 2003.
[6] H. Lee, J. Shim, D. Kim, Ontological modeling of e-catalogs using EER and description logic, in: Proceedings of the International Workshop on Data Engineering Issues in E-Commerce (DEEC 2005), IEEE Society 2005.
[7] eOTD, ECCMA Open Technical Dictionary, ECCMA, http://www.eccma.org/eotd/index.html.
[8] GDD, Global Data Dictionary, http://www.ean-ucc.org/global_smp/global_data_dictionary.htm.
[9] RosettaNet, The RosettaNet Technical Dictionary, http://www.resettanet.org.
[10] R. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, Reading, MA, 1999.
[11] C. Apte, F. Damerau, Automated learning of decision rules for text categorization, ACM Transaction On Information Systems 12 (3) (1994).
[12] E. Pacitti, E. Simon, Update propagation strategies to improve freshness in lazy master replicated databases, The International Journal on Very Large Data Bases 8 (3-4) (2000).

[13] S. Brin, L. Page. The anatomy of a large-scale hyper-textual web search engine, in: Proceedings of the 7th WWW Conference 1998.
[14] J.M. Kleinberg, Authoritative sources in hyperlinked environment, J. ACM 48 (1999).
[15] V. Gaede, O. Günther, Multidimensional access methods, ACM Computing Surveys 30 (2) (1998).
[16] X. Meng, Y. Jiang, Y. Chen, H. Wang, XSeq an indexing infrastructure for tree pattern queries, in: Proceedings of the ACM SIGMOD conference, 2004.
[17] D. Fensel, B. Omelayenko, Y. Ding, E. Schulten, G. Botquin, M. Brown, A. Flett, A product data integration in B2B electronic commerce, IEEE Intelligence System 16 (3) (2001).
[18] L. Obrst, R.E. Wray, H. Liu, Ontological engineering for B2B e-commerce, Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS'01), ACM Press, Maine, USA, 2001.
[19] E. Schulten, H. Akkermans, G. Botquin, M. Dörr, N. Guarino, N. Lopes, N. Sadeh, The e-commerce product classification challenge, IEEE Intelligence System 16 (4) (2001).
[20] J. Leukel, V. Schmitz, F. Dorloff, A modeling approach for product classification systems, in: Proceedings of the DEXA Workshops, 2002.
[21] J. Leukel, Standardization of product ontologies in B2B relationships – on the role of ISO 13584, in: Proceedings of the America's Conference on Information Systems, 2004.
[22] ISO, ISO 13584 Industrial automation systems and integration – Parts library, http://www.iso.ch.
[23] S. Bergamaschi, F. Guerra, M. Vincini, A data integration framework for E-commerce product classification, in: Proceedings of the 1st International Semantic Web Conference, 2002.
[24] I. Benetti, D. Beneventano, S. Bergamaschi, F. Guerra, M. Vincini, An information integration framework for e-commerce, IEEE Intelligent Systems 17 (1) (2002).
[25] M. Hepp, Measuring the quality of descriptive languages for products and services, in: Dorloff, F.-D. et al. (Eds.), E-Business–Standardisierung und Integration, Tagungsband zur Multikonferenz Wirtschaftsinformatik, 2004.
[26] D. Kim, S. Lee, J. Chun, A semantic classification model for e-catalogs, in: Proceedings of the IEEE Conference on E-Commerce, 2004.
[27] S. Lee, Design & implementation of an e-catalog management system, Tutorial at DASFAA, 2004.
[28] J. Guo, C. Sun, Collaborative product representation for emergent electronic marketplace, in: Proceedings of the 16th Bled e-Commerce Conference e-Transformation, 2003.