CATALOG MANAGEMENT IN E-COMMERCE SYSTEMS*

Dongkyu Kim[1], Sang-goo Lee[2], Jonghoon Chun[3], Sangwook Park[1], Jaeyoung Oh[3]

[1] CoreLogiX, Inc.
SNU RIACT 138-414
Kwanakgu Shillimdong
Seoul, Korea

[2] Sch. of Computer Sci. & Eng.
Seoul National University
Kwanakgu Shillimdong
Seoul, Korea

[3] Dept. of Computer Eng.
Myongji University
Yongin Namdong San 38-2
Kyongkido, Korea

ABSTRACT*

Electronic catalogs are information about products and services in the electronic commerce environment, and require diverse and flexible schemas. A catalog management system supports the definition, storage, retrieval, and management of product information throughout the e-commerce process, and thus, is an essential component in almost all e-business applications. However, the diversity in product types, applications, and vocabulary (language) makes it a difficult system to design and implement. We present in this paper the modeling and design issues in an e-catalog management system and introduce a system that implements our approaches to these issues.

KEY WORDS

electronic commerce, e-business, catalog, database, product classification

1. INTRODUCTION

The Internet is an open public network with endless commercial opportunities. The exciting environment of electronic commerce (or *e-commerce*, for short) encompasses a broad range of interaction processes among the various market participants; order, transport and delivery, mediation, invoice and payment, etc. A typical e-commerce transaction consists of suppliers and consumers gathering information and exploring potential market partners for goods and services, exchanging bids and offers, movement of goods, payments, and post transaction activities. Electronic catalogs (or *e-catalogs*) hold most of the information required in this stream of processes and the computerized nature of e-catalogs allows for opportunities to automate and streamline many of the processes involved. Consequently, e-catalogs form the basis of an e-commerce transaction and catalog (or product information) management is a function that is required in almost all e-commerce systems; e-procurement, e-marketplace, supply chain management (SCM), enterprise resource planning (ERP), etc. We envision that eventually catalog management will become a separate individual system supporting the required functionality to the various applications.

Catalog management is complicated by a number of factors. One of these factors is the diversity of schemas for products. A typical catalog containing 100,000 products may contain thousands of different schemas [1]. For example, a *TV* may have a *voltage* attribute, while a *pen* may not. Another factor is diversity in users' views. A manufacturer will tend to view products in terms of the materials and processes associated with the manufacturing of the product, while a distributor would be more interested in the product's size and weight. Difference in views can be clearly witnessed in the variety of product classification schemes that are in use nowadays. A third factor that contributes to the complexity of catalog management is the diversity in vocabulary; the synonyms and homonyms, the various abbreviations and notations, different unit of measures (UOM), the different languages and dialects.

Standardization can only be part of the solution, albeit an important one. No single standard can serve all the diverse purposes in the e-commerce environment. The other part of the solution must be provided by the system by being flexible enough to accommodate and manage diversity. A catalog management system must provide the means for defining, storing, and retrieving e-catalogs, while supporting multiple views and vocabulary in order to transparently support the various processes of an e-commerce application. Critical issues in the design and implementation of a catalog management system include modeling & design, standardization, translation, publishing, indexing, and version management.

In this paper, we address two of the modeling & design issues in catalog management, and introduce a system that

* This work was supported in part by Ministry of Information & Communications, Korea, under the Information Technology Research Center (ITRC) Support Program.

Please address all correspondence to: Prof. Sang-goo Lee, School of Computer Science & Engineering, Seoul National University ENG4190, Kwanakgu Shillimdong, Seoul, Korea; sglee@snu.ac.kr; Tel:+82-2-880-5357

implements our solutions to these issues. Database design issue is dealt with in section 2 and product classification issue in section 3. In section 4, we introduce eCliX™, a commercial catalog management system, to demonstrate how these issues are implemented in a commercial system.

2. DATABASE DESIGN

Although XML (eXtensible Markup Language) [2] is considered important for e-commerce applications including e-catalogs, the storage and management of large-scale catalog databases is still the realm of relational databases [1, 3]. In this section we address the catalog database design issue in the relational database model.

2.1 Terminology

We use the terms defined in [3]. A *product group* (or *class*) is a group of products that are treated as the 'same type', i.e., have the same set of attributes. *Product group id* uniquely identifies a product group while *product id* uniquely identifies each individual product in the database. The set of *common attributes*, denoted as $\boldsymbol{C} = \{c_1, c_2, ..., c_n\}$, is the set of attributes required for every product group in the database, such as `product-group-id` and `product-id`. The set of *dependent attributes,* $\boldsymbol{D}_i = \{d_{i1}, d_{i2}, ..., d_{jm}\}$, for product group *i* is the set of attributes d_{ij} that are specific to the product group and not shared by every product group (e.g., `voltage` for refrigerators and `diameter` for bearings). The schema for product group *i*, $\boldsymbol{P}_i = \boldsymbol{C} \cup \boldsymbol{D}_i$, is the set of attributes required for product group *i*.

2.2 Design Process for e-Catalog Databases

We propose a design process tailored for designing relational database schemas for large-scale electronic catalogs. The process consists of five phases as shown in figure 1**.**

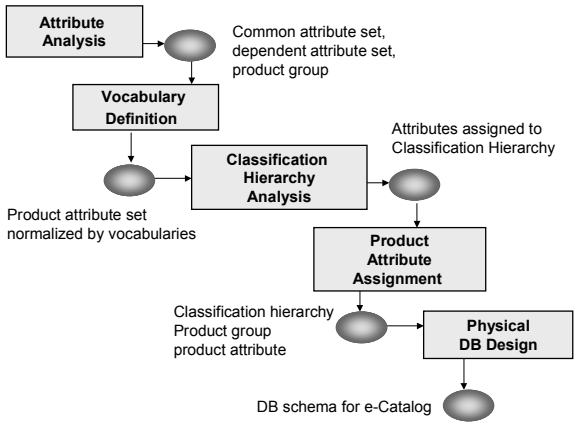


Figure 1: e-Catalog Design Phases

Attribute analysis phase: Electronic catalog systems deal with many products, each with its own set of attributes. This diversity makes it extremely difficult for any database designer to define a coherent set of database columns. So, in the first phase, the database designer should confine himself to analyzing attributes of products for the purpose of differentiating *common* from *dependent attributes*. For example, the `product-name` attribute of "washers" and "telephones" is a common attribute since it is common for all electronic appliances, whereas `capacity` and `caller-id` are examples of dependent attributes for product groups "washers" and "telephones" respectively. Result of this design phase is a set of product groups, a set of common attributes, along with a set of dependent attributes for each product group.

Vocabulary definition phase: As different catalogs may use different names for similar attributes a standard vocabulary must be defined for each attribute. The vocabulary definition specifies the vocabulary name itself, the attribute domain such as type and constraints, and possibly transformation information between different units of measures (UOM). The result of this phase is a set of normalized vocabularies for attributes.

Classification hierarchy analysis phase: In this phase, the database designer may need to decide whether to use a standard classification scheme such as UNSPSC [4], to modify an existing one, or to create a new one. To create a new one, she must define each and every product group's information using the path from the root node to the node that the product group belongs to. For example, as shown in figure 2, the definition for product group "washers" would be *washers = products + electronic appliances + washers*. The result of this phase is a classification hierarchy. The vocabularies defined in the previous phase are used here.

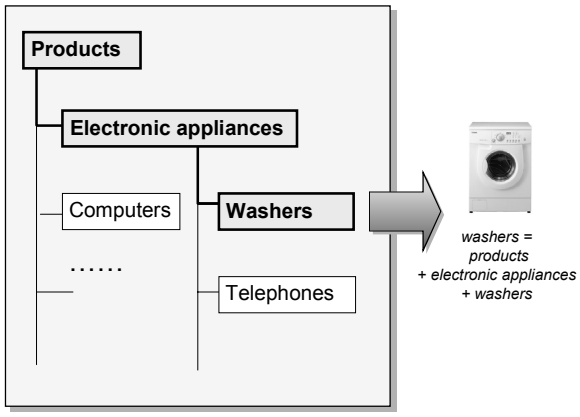


Figure 2: Product Group Definition

Product group attribute assignment phase: In this phase, the database designer actually assigns attributes to each node of the classification hierarchy derived from the previous phase. For example, common attributes `product-name`, `manufacturer`, and `price` are assigned to the root node "products." A dependant attribute `voltage` is assigned to the "Electronic appliances" node.

Physical database design phase: This is the phase where an actual relational database schema gets generated. A product database schema is frequently modeled as a vertical schema in the form of *<id, attribute-name, attribute-value>*. However, there are alternative schemas that are more appropriate for our purpose as presented in the following subsection.

2.3 Electronic Catalog Data Models

We now turn our attention to the various vertical database models for electronic catalogs.

Universal Table (UT): The UT model stores product data in a table $\boldsymbol{T}_{\text{UNIVERSAL}}$ with a schema consisting of the union of common attributes and dependent attributes of all product groups. $\boldsymbol{T}_{\text{UNIVERSAL}} = \boldsymbol{C} \cup (\cup \boldsymbol{D}_i)$. Each tuple in the table represents one product. This model requires another table $\boldsymbol{T}_{\text{UNIVERSAL-META}}$ to keep the metadata identifying the attributes relevant for each product group; $\boldsymbol{T}_{\text{UNIVERSAL-META}}$ = (*product-group-id, attribute-name*) [1]. Thus, the UT model is defined as $\boldsymbol{M}_{\text{UT}} = (\boldsymbol{T}_{\text{UNIVERSAL}}, \boldsymbol{T}_{\text{UNIVERSAL-META}})$. To introduce a new product group, new attributes that are not shared by existing products must be added to $\boldsymbol{T}_{\text{UNIVERSAL}}$ and corresponding metadata inserted into $\boldsymbol{T}_{\text{UNIVERSAL-META}}$. The merit of this model is its simplicity. Only two tables are created. In addition, we can use the integrity constraints provided by the DBMS since the model uses the relational schema directly. On the other hand, there will be too many nulls in $\boldsymbol{T}_{\text{UNIVERSAL}}$ since each tuple will use only some of the attributes in the table. However, because of its simplicity, UT is currently a popular model in actual implementations [5].

Name-Value Pair Table (NV): The NV model stores all product data in table $\boldsymbol{T}_{\text{NV}}$ = (*product-id, product-group id, attribute-name, attribute-value*). The meta-data about the attributes of each product group must be maintained in $\boldsymbol{T}_{\text{NV-META}}$=(*product-group-id, attribute-name*, $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, ..., $\boldsymbol{c}_n$), where $\boldsymbol{c}_k$ is a constraint for the attribute such as data type, non-null, unique, etc. The NV model is defined as $\boldsymbol{M}_{\text{NV}} = (\boldsymbol{T}_{\text{NV}}, \boldsymbol{T}_{\text{NV-META}})$. This is the most flexible model currently used in a relational database as new attributes can be added without changing the schema of the table. To introduce a new product group, we simply insert tuples describing the metadata into $\boldsymbol{T}_{\text{NV-META}}$. Although NV is popular, DBMS provided integrity constraints cannot be used directly. Another drawback is that this model uses too many tuples because it requires one tuple for each attribute rather than for each product.

HYBRID MODELS: The above two primitive models deal with all attributes in the same way. The following hybrid models handle common attributes and dependent attributes differently. Since common attributes do not vary with product groups, common attributes of all products are stored in a single table. This table is $\boldsymbol{T}_{\text{COMMON}}$ which is a projection of common attributes from $\boldsymbol{T}_{\text{UT}}$; $\boldsymbol{T}_{\text{COMMON}} = \prod_{\boldsymbol{C}}(\boldsymbol{T}_{\text{UT}})$.

- **HYBRID_1** combines UT and NV. It is composed of three tables. $\boldsymbol{M}_{\text{HYBRID-1}} = (\boldsymbol{T}_{\text{COMMON}}, \boldsymbol{T}_{\text{NV}}, \boldsymbol{T}_{\text{NV-META}})$. $\boldsymbol{T}_{\text{COMMON}}$ stores common attributes of all the products while $\boldsymbol{T}_{\text{NV}}$ and $\boldsymbol{T}_{\text{NV-META}}$ store dependent attributes in the same way as in the NV model. In contrast to NV, $\boldsymbol{T}_{\text{NV}}$ and $\boldsymbol{T}_{\text{NV-META}}$ in this model do not contain tuples for common attributes because they are already stored in $\boldsymbol{T}_{\text{COMMON}}$.

- **HYBRID_2** introduces $\boldsymbol{T}_{\text{OPTION}}$ as $\boldsymbol{T}_{\text{OPTION}}$ = (*product id, product group id, optional field$_1$, optional field$_2$,..., optional field$_n$*). Each product group can use *optional field*s in $\boldsymbol{T}_{\text{OPTION}}$ for its attributes. Another table is used to manage the meta-data; $\boldsymbol{T}_{\text{OPTION-META}}$ = (*product group-id, optional-field#, attribute-name*, $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, ..., $\boldsymbol{c}_n$), where $\boldsymbol{c}_k$ is a constraint for the attribute. $\boldsymbol{T}_{\text{OPTION-META}}$ is similar to $\boldsymbol{T}_{\text{PT-META}}$ except that it contains an attribute that indicates a specific optional field in $\boldsymbol{T}_{\text{OPTION}}$. Combining (joining) $\boldsymbol{T}_{\text{COMMON}}$ and $\boldsymbol{T}_{\text{OPTION}}$, we get $\boldsymbol{T}_{\text{COMMON-OPTION}}$ which is defined as a table containing the common attributes and optional fields. The HYBRID_2 model is defined as $\boldsymbol{M}_{\text{HYBRID-2}} = (\boldsymbol{T}_{\text{COMMON-OPTION}}, \boldsymbol{T}_{\text{OPTION-META}})$. Values of dependent attributes are stored in *option field*s of $\boldsymbol{T}_{\text{COMMON-OPTION}}$, with $\boldsymbol{T}_{\text{OPTION-META}}$ indicating which option field corresponds to which attribute for the product group. In this case, the number of *option field*s must be large enough to accommodate the maximum of the number of elements in $\boldsymbol{D}_i$, for all *i*.

- **HYBRID_3** is composed of three tables. $\boldsymbol{M}_{\text{HYBRID\_3}} = (\boldsymbol{T}_{\text{COMMON}}, \boldsymbol{T}_{\text{OPTION}}, \boldsymbol{T}_{\text{OPTION\_META}})$. HYBRID_3 is similar to HYBRID_2 except that $\boldsymbol{T}_{\text{OPTION}}$ and $\boldsymbol{T}_{\text{COMMON}}$ are kept as separate tables. It performs better for queries with conditions only on common attributes or only on dependent attributes, but worse for queries with conditions on both common attributes and dependent attributes because of join operations.

Although NV is a proper model for managing diverse and flexible catalogs in relational databases, it has been shown that the model is inefficient in practical applications [3]. In addition, applications take on most of the responsibility of integrity constraint enforcement.

UT is another popular model. It requires the least amount of storage of all the models. However, as the number of attributes increases, it becomes harder to manage. It also performs worse in DBMSs that handle nulls poorly.

[1] Primary keys will be underlined.

Hybrid models, especially HYBRID_2 and HYBRID_3, have been shown to have good performance and space complexity [3], and are the preferred choices in our implementation.

3. PRODUCT CLASSIFICATION

As discussed in section 1, each participant dealing with e-catalogs has its own perspective of the whole product universe. This perspective, in most cases, takes the shape of a classification (categorization) hierarchy. Examples of internationally recognized classification schemes include UNSPSC[2], HS[3], and Nice[4]. As suggested by these examples, products can be classification in more than one ways depending on the purpose of the application. Classification serves a number of purposes including the facilitation of product search and product definition.

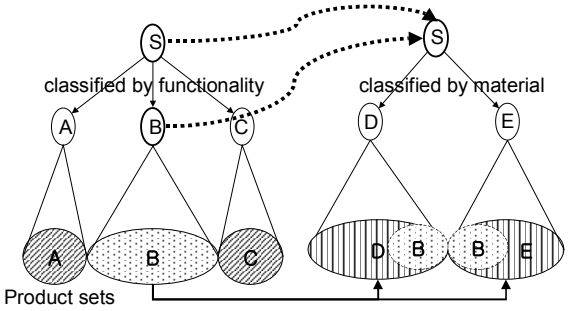
Among the various classification schemes UNSPSC seems to be the most general industry-wide classification that can be used for general e-commerce transactions. However, it cannot represent all the different perspectives of the participants in an e-commerce transaction, nor can it encompass the level of detail required by a direct material procurement system in a specific industry section (say, electronics or ship-building). So, UNSPSC should be considered as an important attribute of a product that tells us its general characteristics using an internationally accepted classification. For a specific process within a transaction, the product may have to be classified differently using a specific classification scheme corresponding to the process and participants. Thus, it is imperative that a catalog management system be able to accommodate multiple classification schemes and provide translation/mapping between the different perspectives.

A simple and widely used method of mapping is the use of *crosswalk tables*. A crosswalk table is a table where each row specifies a correspondence of a category in one scheme to a category in another scheme. Figure 3 shows part of the crosswalk table that defines a mapping from CPV classification to UNSPSC classification [8].

CPV	LEV	TITLE	UNSPSC	LEV	TITLE
60110000-2	3	Passenger land transport serv	781118	3	Road transportation
60113300-6	5	Patient transport services.	92101902	4	Ambulance services
60113310-9	6	Non-emergency patient transp	781118	3	Road transportation
60113400-7	5	Transport of handicapped per	781118	3	Road transportation
60114000-0	4	Regular passenger transport s	78111802	4	Scheduled bus service
60115000-7	4	Passenger transport services	78111804	4	Taxicab services
60115100-8	5	Rental services of passenger	78111804	4	Taxicab services
60115110-1	6	Taxi services.	78111804	4	Taxicab services
60116000-4	4	Special-purpose land passen	78111804	4	Taxicab services
60116100-5	5	Non-scheduled passenger tra	781118	3	Road transportation
60116200-6	5	Sightseeing bus services.	78111803	4	Chartered bus services

Figure 3: A Crosswalk Table

It is not difficult to see the drawbacks of this simple method. For example, in figure 3, 'Transport of handicapped persons' class in CPV is mapped to 'Road transportation' in UNSPSC, which is too general. 'Passenger land transport services', the parent of 'Transport of handicapped persons', is also mapped to 'Road Transportation'. Crosswalk tables introduce information loss and inconsistent mapping. Figure 4 shows a conceptual depiction of the situation.



Because of different perspectives, class B can only be mapped to S which corresponds to the parent of B.

Figure 4: Imprecise mapping with simple correspondence

We propose a semantic classification model that minimizes information loss and enforces consistency. A classification scheme CL is a 4-tuple $< S, C_0, \{C_0, C_1, ..., C_n\}, IC >$, where S is the set of all products, C_0 is the root class that includes all of S, $\{C_0, C_1, ..., C_n\}$ is the set of classes, and IC is the set of integrity constraints. Each C_j is a 3-tuple $< P_j, M_j, A_j >$, where P_j is the parent class C_i of C_j, M_j is the membership function defined in terms of attributes of its parent class C_i, and A_j is the set of attributes common to all members of C_j. For the scheme to have practical utility, we require the following set of integrity constraints to hold;

- P_0=null (root has no parent) and M_0=TRUE (satisfied by all products)
- the parent-child relationship does not form a cycle
- $A_i \supseteq A_j$ if C_i is the parent of C_j (i.e., attributes are inherited)

[2] United Nations Standard Products and Services Classification; Built and managed by United Nations Development Programme for the purpose of providing industry-wide classification scheme for effective identification of goods [4].

[3] Harmonized Commodity Description and Coding System.; Managed by the World Customs Organization and is widely used in trade and customs; Now adopted by the United Nations Statistics Division in collecting international merchandise trade statistics [6]

[4] International classification of goods and services for the purposes of the registration of marks (trademarks and service marks) under the Nice Agreement; Managed by World Intellectual Property Organization [7].

- M_j subsumes M_i if C_i is the parent of C_j, i.e., set of products in C_j is a subset of products in C_i

For example in UNSPSC (see figure 3), class 'Taxicab services' can be defined as follows;

- P_j, = *'Road Transportation'* (C_i)
- $M_j = M_i \wedge$ *(vehicle_type='automobile')*$\wedge$ *(load_type='passenger')*$\wedge$*(commercial='Y')*
- $A_j = A_i \cup \{fleet, make, model, year, \ldots\}$

In this model, a classification scheme is not merely a mechanical assignment of class codes but is a semantic representation of a perspective on the product set *S*. Given two classification schemes CL_1 and CL_2 in this model, mapping between class C_{1i} of CL_1 and C_{2j} of CL_2 can be specified precisely. If the two schemes share the same set of vocabulary (attribute names, values, etc.), the mapping becomes the process of finding common subexpressions in M_{i1} and M_{j2}. In practice, even when the schemes do not share a common vocabulary, this model can be used to further qualify the simple correspondence defined in a crosswalk table. An example is shown in figure 5.

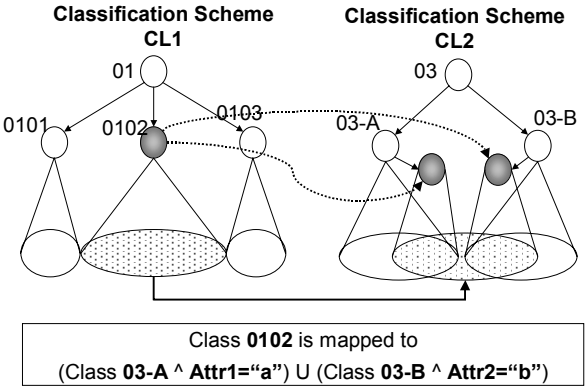


Figure 5: Correspondence mapping with qualification

4. IMPLEMENTATION

eCliX™, by CoreLogiX Inc., Seoul, Korea, is a commercial electronic catalog management system [9]. It supports multiple database schemas and classification schemes by adopting a hybrid verticalized schema presented in section 2. Thus, it has the ability to handle catalog data in different formats. eCliX provides simple drag-and-drop GUI to easily transform catalog data to different formats between different classification schemes, with custom mapping rules to capture various semantics employing the semantic model presented in section 3. It also allows intelligent personalization of catalog to be published to multiple recipients with custom business rule declaration capability to reflect various business policies. eCliX is equipped with a powerful web search engine tailored for hybrid verticalized schema, which ensures quick and efficient response for ad hoc queries as well as multi-keyword-type queries. Large volume of electronic catalogs is efficiently managed and all information in eCliX can be published in XML.

eCliX includes a number of components as shown in figure 6. We give a brief overview.

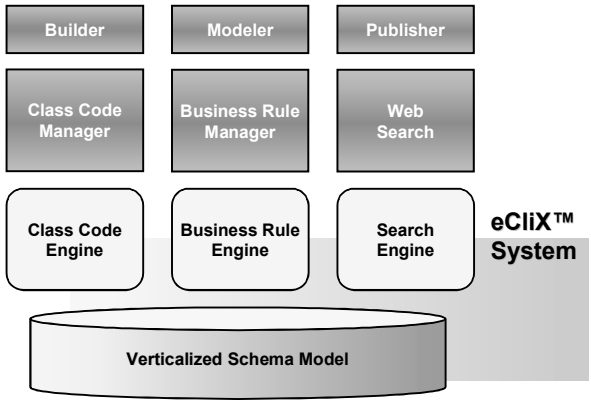


Figure 6: eCliX™ architecture

Class Code Engine (CCE) is responsible for managing the various classification schemes used in the application. CCE is an essential component supplying services to the other components such as Builder and Modeler since most components use one classification scheme or the other in carrying out their functions. For example, when defining a product group (class) in the Modeler, the user would locate the appropriate parent class in one of the classification schemes and inherit the attributes. Users may import, extract, merge, update, or newly build multiple classification schemes and define mappings using our semantic classification model discussed in section 3. A typical CCE session is shown in figure 7. Merging, extracting, and version management are complicated operations requiring well-defined semantics and algorithms [10].

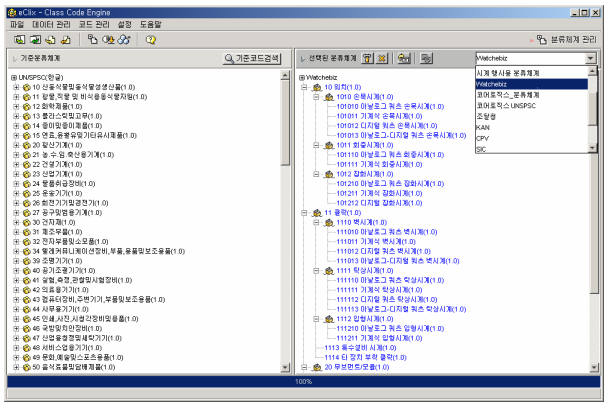


Figure 7: Class code management interface

eCliX Modeler is used in designing an e-catalog database. It implements the 5 phases of design presented in section 2.2. Software wizards guide the user through the assignment of attributes to the proper product groups of the classification hierarchy. Multiple classification schemes may be used in defining the same set of products thus allowing multiple views. An optimized verticalized schema is automatically generated at the end of the design session.

eCliX Builder imports product information from multiple sources into the eCliX database. A number of input formats are supported. Pre-specified mapping information between heterogeneous classification schemes (using CCE) is used to ensure proper catalog data loading. Expert users may want to define and use rules for detecting dirty data and cleansing them while importing.

eCliX Business Rule Manager and Engine allows specification of various business policies, which are available for the Publisher to show multiple versions of the same product information according to the policies. **eCliX Publisher** transforms the catalog data to be published for different recipients in the form of XML documents. **eCliX Web Search Engine** provides an efficient search service for the catalog database. It implements the Mass Catalog Indexing method [11] specifically designed for indexing large catalog databases.

eCliX is currently being used in building a Central e-Commerce Ontology Repository for the Korean Government's Public Procurement Services. Multiple classifications schemes are imported and mappings are being defined. Metadata for the catalogs are defined for 25,000 product classes currently in service. eCliX is also being installed in a vertical B2B pilot project for the ceramic industry.

5. CONCLUSIONS

We have presented two aspects of modeling and design of product information. The diversity of product types complicates the database design in that the schema must support diverse schema. We have proposed a schema model that supports diverse product types in an effective and efficient manner. A related and important issue in the design is product classification. We have introduced a formal classification model in order to represent the semantics of a classification scheme. The model allows us to specify more precisely the mappings between classification schemes. eCliX™ is a commercial catalog management system which has been built on these ideas. It allowed us to validate the proposed model and methods in a commercial setting.

We believe that catalog management is an essential component in almost all e-commerce applications, which will eventually establish its status as an e-commerce system software. The issues presented in this paper are only a subset of the technical problems that need to be addressed in such a system. Search efficiency in multiple schemas is a challenging topic. Catalog publishing and XML issues is another prospective research area. The utility and importance of vocabulary (or ontology) in e-commerce has only recently been recognized. Vocabulary modeling and management is another area that merits attention.

REFERENCES

[1] A. Jhingran, Moving up the food chain: Supporting e-commerce applications on databases, *Technical Report, IBM Almaden Research Center*, 2000.

[2] WWW Consortium, Extensible markup language (XML), *http://www.w3c.org/XML*, 2001.

[3] K. Kim, et al, An Experimental Evaluation of Dynamic Electronic Catalog Models In Relational Database Systems, *Proc. of the Information Resources Management Association International Conf.,* 2002.

[4] UNDP, United Nations Standard Products and Services Code, *White paper*, *http://www.un-spsc.net*, 1998 (revised 2001).

[5] J. Jung, et al, EE-Cat: Extended electronic catalog for dynamic and flexible electronic commerce, *Proc. of the Information Resources Management Association International Conf.,* Anchorage, AL, 2000, 303-307.

[6] World Customs Organization, Harmonized System explanatory Notes, *http://www.wcoomd.org*, 2002.

[7] World Intellectual Property Organization, Classification treaties, *http://www.wipo.org*, 2002.

[8] ECCMA, Conversion crosswalk table: CPV to UNSPSC, *http://www.eccma.org*, 2000.

[9] CoreLogiX, Inc., eCliX™ product overview, *http://www.corelogix.co.kr*, 2001.

[10] D. Kim, et al, Catalog integration for electronic commerce through category-hierarchy merging technique, *Proc. 12th IEEE Workshop on Research Issues on Data Engineering, Engineering E-Commerce/E-Business Systems (RIDE-2EC '2002)*, Los Angeles, CA, 2002.

[11] S. Park, et al, MCI: Mass catalog indexing for searching large-scale catalog databases, *Technical Report, CoreLogiX Inc*., Seoul, Korea, 2002**.**