Journal of Network and Computer Applications (2001) **24**, 201–228
doi:10.1006/jnca.2001.0132, available online at http://www.idealibrary.com on IDEAL®

Deployment of personalized e-catalogues: An agent-based framework integrated with XML metadata and user models

Duen-Ren Liu, Yuh-Jaan Lin, Chung-Min Chen*[*] and Ya-Wen Huang

Institute of Information Management, National Chiao-Tung University, Hsinchu 300, Taiwan
[*]*Telcordia Technologies, Morristown, NJ, USA*

(Received 28 June 2000; accepted 25 June 2001)

In e-commerce applications, supporting electronic catalogues (e-catalogues) to provide product-search services on the Internet is emerging as an important function of electronic brokers (e-brokers). However, as the Internet expands, an increasing number of virtual stores are selling products on-line. The explosive growth of information is creating difficulties for customers searching for the product information they desire. In this work, an integrated framework is proposed to develop personalized e-catalogues. The proposed framework integrates XML-based metadata models, user models and agents. The XML-based metadata model can facilitate resource-discovery and format translation, and can flexibly model the definitions of diverse product attributes in heterogeneous data resources. The proposed user model can model a user's shopping interests to resources, categories and products. This work integrates the agent technology with the proposed metadata model and user model to develop a personalized e-catalogue system. The novel system effectively provides an integrated search for heterogeneous product information, as well as supporting personalized product search and browsing for each user based upon their profiles. © 2001 Academic Press

1. Introduction

In electronic commerce, electronic brokers [1–3] provide various important brokerage services, such as merchandise filtering, suggestions, search, negotiation and electronic catalogues. The primary function of an electronic catalogue (e-catalogue) system is to support Internet-based product searches. However, with the growth of Internet virtual stores and product information, a number of businesses are making their services available on the Internet. Currently, many companies have constructed electronic catalogues to provide navigation of product information to Web shoppers. However, the characteristics of goods may vary in distinct product categories. Moreover, resources (virtual stores) may also employ different methods and data formats to query and store goods information. Notably, a (data) resource is defined as a store (e.g. virtual store or virtual mall) capable of providing search for product data stored in its repository. Relevant issues include the integration of heterogeneous goods information as well as interoperation among the search system and diverse resources. Moreover, the explosive growth of information is creating difficulties for customers in searching for the goods information they require. Finding possible data resources (resource discovery) and conducting format translation are important issues to search product from heterogeneous resources.

Metadata [4,5] describes the characteristics of data that has been collected for a specific purpose. It can be used to facilitate the search service and integrate heterogeneous information. There are two types of metadata, which are directory-level metadata and inventory-level metadata. Directory-level metadata describes a data resource or a collection of data. Inventory-level metadata describes the actual raw data, which a resource contains. Directory Interchange Format (DIF) [5] is a de-facto standard used to create directory entries, which describe a group of data. STEP [4], which is an inventory-level metadata, applies EXPRESS to specify the product information.

The extensible markup language (XML), is a subset of an international standard, Standard Generalized Markup Language (SGML) which is designed to facilitate the interchange of structured documents over the Internet [6]. XML uses a flexible, open, and standard-based format to provide interoperability, with novel means of accessing legacy databases and delivering data to clients. XML is quite appropriate for certain electronic commerce and metadata applications [7].

Software agents are programs that act on behalf of their human users to perform laborious information gathering tasks [8,9]. Agents communicate through the agent communication language (ACL) [10]. ACL consists of three parts; vocabulary (ontology), KIF (Knowledge Interchange Language), and KQML (Knowledge and Query Manipulation Language). Software agent technology is well suited to search and integrate data from heterogeneous environments. Stanford Digital Library [11] uses methods including proxies, CORBA and attribute models, while InfoSleuth [12] applies methods including agent technology and information brokerage.

Electronic brokers provide various brokerage services such as merchandise filtering, suggestion, search, negotiation and electronic catalogues [2]. Recently, Web-based e-catalogue systems in B2B procurement have been proposed [13]. Keller *et al.* have also proposed smart catalogues and virtual catalogues to support multi-company cross-catalogue searches [14]. In their study, a brokering architecture is employed to retrieve and integrate data from heterogeneous environments. The brokering architecture utilizes agent technology and information brokerage. Some issues are not addressed, such as resource discovery and the modelling of separate levels of goods-related information, and these are vital to providing effective integrated search services across multiple virtual retailers. Moreover, enabling customers to search for goods information with specific attributes of a product category are not addressed. Lincke *et al.* also propose a similar work on electronic catalogues [15]. Their work presents an architecture for mediating electronic product catalogues without further detailing the design and implementation of the system.

Although the search function of an e-broker has been discussed, few studies have mentioned the information filtering function, which is substantial, as the search result may still be too large to be of use. The filtering of goods information can be realized by incorporating user profiles into search systems. Though many topics regarding user profiles have been discussed [16–19], most have focused on text-based document filtering. Barra *et al*. have proposed a vector of configurations indicating customer preference to buy product items from specific categories [20]. However, their model only describes user preference to a specific category, which is inadequate to model user needs. For example, a user may prefer to buy goods from a certain resource (virtual store). Therefore, a user's preference may also vary in distinct product categories.

In this work, an integrated framework is proposed to develop personalized e-catalogues. The proposed framework integrates XML-based metadata models, user models and agents. The proposed metadata model is a multi-level architecture to describe separate levels of goods-related information. The XML standard defines a syntax that enables the users to create markup languages to specify information structures. In the multi-level metadata model, XML is employed as the meta-language to describe shopping domains, resources (e.g. virtual stores), product categories provided within a resource, and the goods (product) information, respectively. Resource discovery and format translation are vital to providing effective integrated search from heterogeneous data resources. The proposed metadata model can facilitate resource-discovery and format translation, and can flexibly model the definitions of diverse attributes to describe goods in various product categories and data resources. Furthermore, user models are designed to support personalized filtering within e-catalogues. The proposed user model employs three-level user profiles to model shopping interests to a certain level of goods information, including preferred resources, categories and goods. The user profile records the user's preference (weight) to the characteristics (described by attribute and value) of goods information. To provide personalized filtering, a match process is designed to rank users' preferred product items.

On the basis of the proposed metadata model and user model, this work integrates the agent technology to develop a personalized e-catalogue system. The proposed metadata is used as the basis of resource discovery and the ontology (vocabulary) translation. The proposed e-catalogue system comprises a resource agent, broker agent, user agent, category model agent, control agent, virtual catalogue agent and domain agent. Agents communicate and cooperate with one another to achieve search results that satisfy the needs of customers. The multi-level design flexibly models goods-related information and users' shopping interests. Based on the metadata model, the system provides a structured query by the attributes of the selected product category. Moreover, the novel system is able to support personalized product search and browsing for each user via referencing the user profiles. The system not only conducts product search according to the query constraints, but also uses the personalized constraints from the user profiles for further personalized filtering. The novel e-catalogue system provides an effective, integrated search for heterogeneous goods information located in various virtual stores.

The remainder of this paper is organized as follows. Section 2 presents the XML-based metadata model. User models for personalized e-catalogues are discussed in Section 3. Personalized virtual catalogues are also described. Section 4 illustrates the design of the e-catalogue system by applying agent technology. In Section 5, we discuss the implementation of the proposed system. A demonstration of the prototype system is also presented. Section 6 compares our design with related research work and systems. Conclusions and proposals for future work are finally given in Section 7.

2. XML-based metadata model

To capture valuable information on a wide variety of resources for various purposes, the designed metadata must be sufficiently flexible. As XML [6] is extensible, flexible, and readable, and can easily be distributed over the Internet via HTTP, it is considered appropriate for defining metadata. Hence, XML [21] is used herein to define the proposed multi-level metadata framework. The design of the novel metadata is not implied to be comprehensive, and further work is required to complete the design.

Domain-level metadata describes the subject interest of each shopping community (domain). Resource-level metadata is used to describe the information regarding resources (e.g. virtual stores). Category-level metadata describes the information of a product category provided within a resource. Goods-level metadata describes the goods (product) information. Fig. 1 illustrates the graphical overview of relationships among different levels of goods information.

2.1 *Domain-level metadata*

In general, users with the same interest may form an Internet community. To promote Internet shopping, shopping communities constructed by varying subjects of interests will be an inevitable trend. The communities may be formed according to areas and/or products as well as other criteria. For example, virtual stores in Taiwan that sell electronic products may join together to form a Taiwan-Electronic community. The domain-level metadata has been designed to describe each community (domain). Table 1 illustrates the XML document type definition (DTD) of the domain-level metadata.

The optional character following a name or list governs whether the element or the content particle may occur once or more (+), zero or more (*), or zero or one time (?). The absence of such an operator means that the element or content

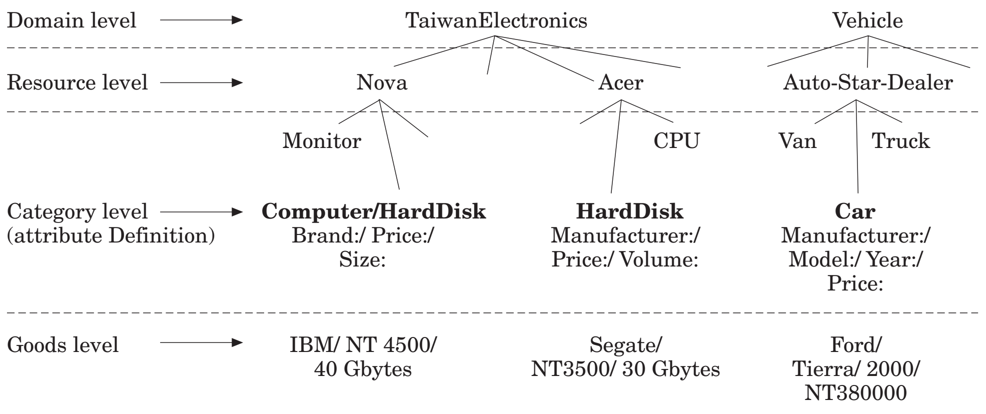

Domain level
Resource level
Category level (attribute Definition)
Goods level
TaiwanElectronics
Vehicle
Nova
Acer
Auto-Star-Dealer
Monitor
CPU
Van
Truck
Computer/HardDisk Brand:/ Price:/ Size:
HardDisk Manufacturer:/ Price:/ Volume:
Car Manufacturer:/ Model:/ Year:/ Price:
IBM/ NT 4500/ 40 Gbytes
Segate/ NT3500/ 30 Gbytes
Ford/ Tierra/ 2000/ NT380000

Figure 1. An overview of multi-level metadata.

Table 1. *Domain-level metadata*

```
<?xml version='1.0'?>
<!DOCTYPE DOMAINS [
<!ELEMENT DOMAINS (DOMAIN)* >
<!ELEMENT DOMAIN (URL, SUBJECT+) *>
<!ATTLIST DOMAINS
      ID    CDATA #REQUIRED       TITLE CDATA #REQUIRED>
<!ELEMENT URL (#PCDATA)>
<!ATTLIST SUBJECT
      TYPE  CDATA #REQUIRED       VALUE CDATA #REQUIRED>
]>
```

particle must appear exactly once. The DTD of domain-level metadata shows that the ‘domains’ element contains ‘domain’ sub element. Each domain element has URL and ‘SUBJECT’ sub-elements. The ‘SUBJECT’ element can be comprised of multiple entries, which describe the type and value of each subject interest.

2.2 *Resource-level metadata*

The resource-level metadata describes the sorts of product a virtual store or virtual mall sells. A virtual mall may contain numerous virtual stores. The resource-level metadata provides resource information, such as store profile, contact information, categories, and keyword location. Table 2 depicts the partial DTD of the resource-level metadata. The ‘RESOURCE’ element contains two kinds of sub elements; virtual store and virtual mall. The ‘CATEGORY’ element can be comprised of multiple entries to indicate the categories contained in the resource. The element is a hierarchical name with ‘|’ to represent a Category | Sub-category relationship.

Table 2. *Resource-level metadata*

```
<?xml version= '1.0'?>
<!DOCTYPE RESOURCE [
<!ELEMENT RESOURCE (VIRTUAL_STORE | VIRTUAL_MALL)*>
<!ELEMENT VIRTUAL_MALL ((VIRTUAL_STORE)+, PHONE, ADDRESS, URL, ADMINISTRATOR,
       SERVICE_CONTACT, SUMMARY?, CATEGORY+, KEYWORD+, LOCATION)*>
....
<!ELEMENT VIRTUAL_STORE (PHONE, ADDRESS, URL, ADMINISTRATOR,
            SERVICE_CONTACT, SUMMARY?, CATEGORY*, KEYWORD+, LOCATION?)*>
<!ATTLIST VIRTUAL_STORE
      ID    CDATA #REQUIRED     TITLE  CDATA #REQUIRED >
....
<!ELEMENT SUMMARY (#PCDATA)>
<!ELEMENT CATEGORY (#PCDATA)>
<!ELEMENT KEYWORD (#PCDATA)>
<!ELEMENT LOCATION (#PCDATA)>
]>
```

2.3 *Category-level metadata*

Products that have common attributes are grouped within the same category. Category-level metadata is used to describe searchable attributes of each category and class relationship within categories. A category in a different virtual store may support varying attributes. The category-level metadata includes both general and extensible fields. General fields are used to describe the basic information of a category, such as store ID, catalogue ID, title, category hierarchy, keyword, and summary. Extensible fields model the definitions (data format and constraints) of additional searchable attributes that will appear in goods-level metadata. The same product category in distinct virtual stores may support different extensible attributes. Table 3 shows the DTD of the category-level metadata.

The DTD of category-level metadata reveals that the ‘CATEGORIES’ element contains ‘CATEGORY’ sub elements. Each category element has class, keyword, summary, and attribute sub-elements. The ‘CLASS’ element describes the class hierarchy of each category. The definitions (name, type, size, constraint) of extensible attributes for describing goods are modelled by the ‘ATTRIBUTE’ sub-elements. Table 4 shows an example of a hard disk category that supports the search of price and volume attributes.

For each category supported in a data resource, the category title (name) is recorded in the CATEGORY element of the corresponding resource-level metadata. In addition, a corresponding category-level metadata is created to model each category. The identity of the data resource is recorded in the store_ID element of category-level metadata, while the category title is recorded in the title element of the resource-level metadata. This provides the linkage between the resource-level metadata and category-level metadata.

Table 3. *Category-level metadata*

```
<?xml version='1.0'?>
<!DOCTYPE CATEGORIES [
<!ELEMENT CATEGORIES (CATEGORY)*>
<!ELEMENT CATEGORY (CLASS, KEYWORD+, SUMMARY, ATTRIBUTE+)*>
<!ATTLIST CATEGORY
     ID CDATA #REQUIRED STORE_ID CDATA #REQUIRED TITLE CDATA #REQUIRED>
<!ELEMENT CLASS (#PCDATA)>
<!ELEMENT KEYWORD (#PCDATA)>
<!ELEMENT SUMMARY (#PCDATA)>
<!ELEMENT ATTRIBUTE EMPTY>
<!ATTLIST ATTRIBUTE
    NAME        CDATA #REQUIRED   TYPE        CDATA #REQUIRED
    SIZE        CDATA #REQUIRED   CONSTRAINT CDATA >
]>
```

Table 4. *Example of category-level metadata*

```
< CATEGORIES>
  <CATEGORY ID=''NOVA_COST001'' STORE_ID=''NOVA001'' TITLE=''HARD DISK''/>
  <CLASS>Computer | Hard Disk</CLASS>
  <KEYWORD>discount</KEYWORD>
  <SUMMARY>sell Seagate, IBM, Quantum</SUMMARY>
  <ATTRIBUTE NAME=''PRICE'' Type=''NUMBER'' SIZE=''6'' CONSTRAINT=''>0''/>
  <ATTRIBUTE NAME=''VOLUME'' Type=''NUMBER''SIZE=''5''CONSTRAINT=''>0''/>
  </CATEGORY>
</ CATEGORIES>
```

A virtual store may belong to many different malls. The resource-level metadata of each virtual mall contains the information of the shared virtual store. Furthermore, many different stores may sell the same product. The major issue is that different stores provide different query capabilities such as different data formats and searchable attributes for the same product. For example, store A only allows querying a monitor by price and brand, while store B is able to support more searchable attributes including price, brand, type and size.

The category-level metadata is designed to model various stores' query capabilities. For each product (category) sold in a virtual store, a corresponding category-level metadata is defined to model the virtual store's query capabilities with respect to the particular product (category). The definitions of searchable attributes of a category supported in a data resource are recorded in the ATTRIBUTE elements of the corresponding category-level metadata. Fig. 2 shows different sets of searchable attributes supported in different stores. The searchable attributes of the Hard-Disk category in the Nova/E-Shop category are different from those in the Acer/AOpen and ISN store. Notably, a category-level metadata of Hard-Disk is created for each store ISN, Nova/Apollo, Nova/E-Shop, Acer/AOpen and Acer/E-Shop, respectively. Each set of searchable attributes is modeled by the ATTRIBUTE elements of corresponding category-level metadata.

2.4 *Goods-level metadata*

Goods-level metadata provides detailed product information. To facilitate a search, product information formats of diverse virtual stores must be translated to a unified format that is described by the goods-level metadata. Table 5 illustrates the partial DTD of the goods-level metadata. Table 6 shows an example of goods-level metadata.

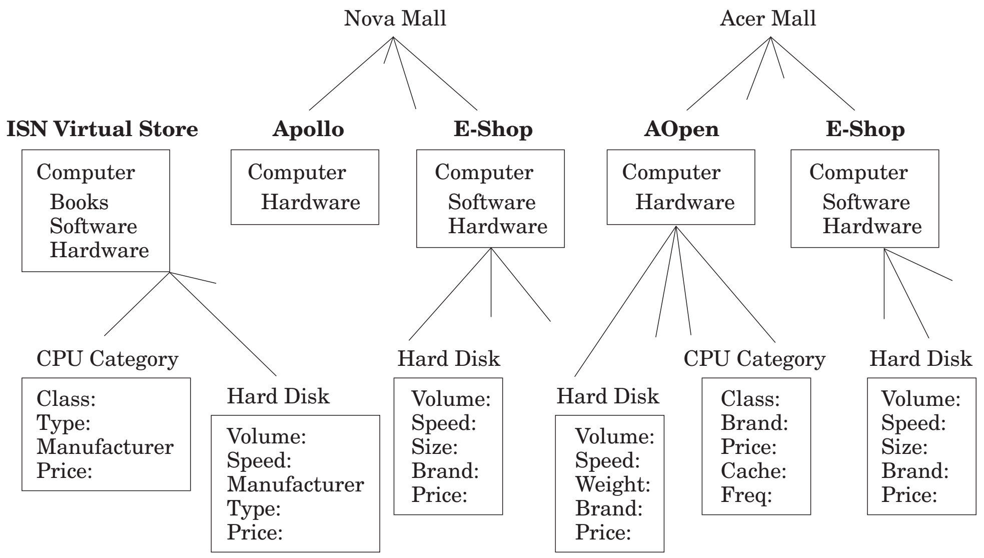


Figure 2. Diverse categories supported in different virtual stores.

Table 5. *Goods-level metadata*

```
...
<!ELEMENT GOODS (PRODUCT)*> ......
.....
<!ELEMENT PRICE (#PCDATA)> .....
<!ELEMENT KEYWORD (#PCDATA)>
<!ELEMENT ATTRIBUTE EMPTY>
<!ATTLIST ATTRIBUTE
     NAME    CDATA  #REQUIRED VALUE CDATA #REQUIRED>
<!ELEMENT DESCRIPTION (#PCDATA)>....
<!ELEMENT URL (#PCDATA)>
<!ELEMENT IMAGE (#PCDATA)>...
```

The goods-level metadata describes that a product element contains manufacturer, price, keyword, attribute, guarantee, URL, and image sub elements. The ATTRIBUTE elements allowing multiple entries of specific (searchable) attribute name and value pairs are used to describe goods of a specific product category. We note that the definitions (name, type, size and constraint) of specific attributes supported in goods-level metadata are modelled by the ATTRIBUTE elements in the corresponding category-level metadata.

3. Personalized e-catalogue design

Based upon their preferences, a personalized electronic catalogue can utilize user profile information to serve each user differently. The user profile would assist the system in establishing the search, retrieval, and processing for the user. The setting options of the user profile are based upon the metadata architecture. The information stored in the user profile includes the basic information of each user, password, and preferred locations, virtual stores, and categories. In addition, we propose a three-level user profile that has a corresponding level to the merchandise metadata, to describe a user's interest to a certain level of merchandise. Under an attribute-based description, the profile describes users' shopping interests in detail. This implies that the profile records every weight of an attribute and its value of a certain item. Section 3.1 illustrates the detailed design of user models.

In our current design, users need to select or enter their preferences to create or refine user profiles. For instance, a student who lives in Taipei may be interested in computer hardware, and books that are sold in Taipei. He can construct his own user profile by selecting the preferred locations, stores, and categories. In addition, the system provides both general search services and personalized search services. The general search service conducts product search according to the

Table 6. *An example of goods-level metadata*

```
<GOODS>
 <PRODUCT ID=''CostHD001'' STORE_ID=''COST001'' NAME=''HardDisk IDE''>
 <MANUFACTURER TITLE=''IBM''>.. </MANUFACTURER>
 <BRAND>DeskStars 4</BRAND>
 <PRICE >7500</PRICE>
 <KEYWORD>fast </KEYWORD>
 <ATTRIBUTE NAME=''VOLUME'' VALUE=''7.4G''/>
 <DESCRIPTION> high-performance </DESCRIPTION>
 <GUARANTEE TYPE=''YEAR''>3 </GUARANTEE>
 ...
 <URL>http://www.ibm.com.tw/storage...<URL>
 <IMAGE>http://www..storage/ibm-1.jpg<IMAGE>
 </PRODUCT>
</GOODS>
```

query constraints specified in the user's query, without considering personalized filtering, while the personalized search service conducts product search based on the query constraints and user profiles for further personalized filtering. In personalized search services, the system not only conducts product search according to the query constraints, but also uses the personalized constraints from the user profile for further personalized filtering. The search of products is not solely according to the user profiles. Users can flexibly find products they require via entering new search constraints in the query. Notably, the constraints specified in the original query surpass the personalized constraints in the user profile. The system finds product items that satisfy the query constraints. In addition, product items of the query result will be further filtered and ranked according to the personalized constraints in the user profile.

Section 3.2 describes how the system provides general search services without considering personalize filtering. Section 3.3 illustrates a personalized search service based on user profiles. In addition, by establishing a personal user profile, the system not only can filter the desired information based on the needs and preferences of each user, but can also construct a personalized virtual catalogue. Section 3.4 presents the personalized virtual catalogue.

3.1 *User model*

To separate the interests of a user into resource, category and goods level, the user model is, therefore, a three-level profile. Fig. 3 shows the structure of user profiles. The resource-level user profile describes the user shopping interest to virtual stores when buying items of a certain product category. In addition, the goods-level user profile can filter out certain product items that the user may not be interested in.

The designed e-catalogue system is a category-oriented advising system. Herein, every advising request from a user must always specify the product category, under the assumption that shopping always starts with a motivation of buying a product item in which its product category is predetermined. However, even within the same product category, each category may still vary. For example, a hard disk category may have various types of hard disks for sale, such as a second-hand hard disk in virtual store A or an IBM-only hard disk in virtual store B. Hence, to achieve a symmetric modelling and provide a more specific filtering to category-level goods information, we still enter the category-level user profile into our system. The user profile describes user's shopping interests in as much detail as possible. As a result, a strategy to describe user's preference to a certain product item is required. Therefore, a 'weight' concept is applied to record the importance of each attribute and value.

Every node in a user profile has an essential attribute weight, which is used to determine the importance of a certain attribute to a user. Important values of the attribute also have corresponding weights to determine their significance.

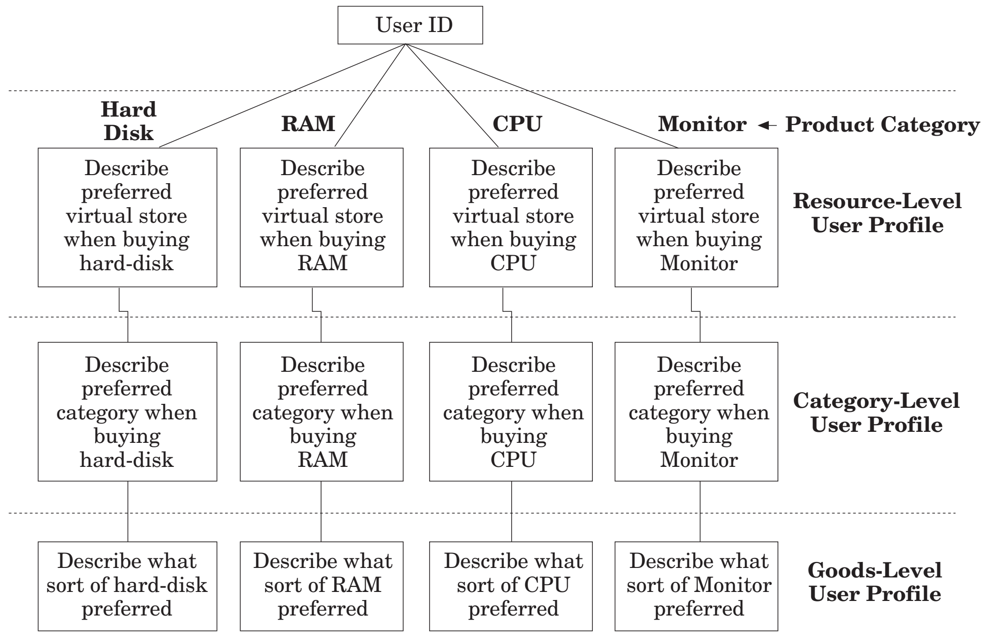

User ID
Hard Disk
RAM
CPU
Monitor
Product Category
Describe preferred virtual store when buying hard-disk
Describe preferred virtual store when buying RAM
Describe preferred virtual store when buying CPU
Describe preferred virtual store when buying Monitor
Resource-Level User Profile
Describe preferred category when buying hard-disk
Describe preferred category when buying RAM
Describe preferred category when buying CPU
Describe preferred category when buying Monitor
Category-Level User Profile
Describe what sort of hard-disk preferred
Describe what sort of RAM preferred
Describe what sort of CPU preferred
Describe what sort of Monitor preferred
Goods-Level User Profile

Figure 3. The user model.

For example, a user profile may have a 'colour' attribute and is weighted as 0.3. It indicates that the importance of the attribute 'colour' to a user is 0.3. Also, the attribute value of 'colour' may be 'red', 'blue', or 'green', which is weighted as 0.1, 0.1, and 0.8, respectively. This means that the user prefers a green product to others. If a product is green, the user treats this feature of the product as important as $0.3{*}0.8{=}0.24$, which implies a 24% influence on buying the product. The weights can be scaled values or normalized values. The weights will add up to 1.0 if they are normalized. If no user weight is specified, a default weight (with very low weight) is used. A formal user profile can be described as the following formulation, which is a vector space model:

$$\left\{ C, L, \begin{bmatrix} (A_1, w_{A_1}, \langle\langle v_1^{A_1}, w_1^{A_1}\rangle, \langle v_2^{A_1}, w_2^{A_1}\rangle, \cdots, \langle v_{m_1}^{A_1}, w_{m_1}^{A_1}\rangle\rangle), \\ (A_2, w_{A_2}, \langle\langle v_1^{A_2}, w_1^{A_2}\rangle, \langle v_2^{A_2}, w_2^{A_2}\rangle, \cdots, \langle v_{m_2}^{A_2}, w_{m_2}^{A_2}\rangle\rangle), \\ \vdots \\ (A_n, w_{A_n}, \langle\langle v_1^{A_n}, w_1^{A_n}\rangle, \langle v_2^{A_n}, w_2^{A_n}\rangle, \cdots, \langle v_{m_n}^{A_n}, w_{m_n}^{A_n}\rangle\rangle) \end{bmatrix} \right\}$$

C refers to the product category that this user profile describes, L indicates the level of the user profile, which can be resource, category or goods level. The meaning of category in C and category-level in L is different. That is, C represents the product category that this user profile describes, while category-level in L means a group of some common feature under the product category. For example, C can be a hard disk, while L can be a group of all second-hand hard disks. A_1 to A_n are the attributes in this user profile, and w_{A1} to w_{An} represent the weight of each corresponding attribute. The list following A and W describes attribute A's important values and their corresponding weights. Each entry $<v, w>$ indicates that the weight of value v is w. Each attribute A_i has m_i values. The following is an example of a user model that describes the user's preference when buying a jacket category:

(1) Resource-level user profile:

$$\left\{ \begin{array}{c} Jacket, \\ resource, \end{array} \left[\begin{array}{l} (location, 0.2, \langle\langle Taipei, 0.7\rangle, \langle TauYuan, 0.01\rangle, \cdots, \\ \quad \langle PanChiau, 0.1\rangle\rangle), \\ (payment, 0.6, \langle\langle master, 0.4\rangle, \langle visa, 0.2\rangle, \cdots, \\ \quad \langle cash, 0.1\rangle\rangle), \\ (keyword, 0.1, \langle\langle \text{``}WellService'', 0.1\rangle, \cdots, \\ \quad \langle \text{``}SatisfactionGuarantee'', 0.5\rangle\rangle) \end{array} \right] \right\}$$

(2) Category-level user profile:

$$\left\{ \begin{array}{c} Jacket, \\ category, \end{array} \left[\begin{array}{c} \left(summary, 0.5, \left\langle \begin{array}{l} \langle WellKnownBrand, 0.7\rangle, \ldots, \\ \langle SpringWear, 0.01\rangle \end{array} \right\rangle \right), \\ (keyword, 0.5, \langle\langle Discount, 0.6\rangle, \cdots, \\ \langle WaterResistance, 0.1\rangle\rangle) \end{array} \right] \right\}$$

(3) Goods-level user profile:

$$\left\{ \begin{array}{c} Jacket, \\ goods, \end{array} \left[\begin{array}{c} (colour, 0.5, \langle\langle red, 0.1\rangle, \langle green, 0.8\rangle, \cdots, \langle blue, 0.05\rangle\rangle), \\ (texture, 0.2, \langle\langle nylon, 0.4\rangle, \langle cotton, 0.2\rangle, \cdots, \langle silk, 0.1\rangle\rangle), \\ (size, 0.3, \langle\langle S, 0.1\rangle, \langle M, 0.2\rangle, \cdots, \langle XL, 0.5\rangle\rangle) \end{array} \right] \right\}$$

Occasionally, there are certain attributes without discrete values, such as an item price. This will render the representation of all possible attribute values in such a user profile difficult. Thus, a range is employed to represent such a kind of attribute value. For example, a price attribute within user profile can be expressed as the following:

$$(price, 0.2, \langle\langle <1000, 0.6\rangle, \langle 1000\sim 2000, 0.2\rangle, \cdots, \langle >100000, 0.1\rangle\rangle)$$

3.2 *General search service*

Fig. 4 depicts how the system can establish a *Directory Information Tree (DIT),* through processing the resource-level metadata and category-level metadata, which was described in Section 2. DIT is essential for managing the resources and categories information and is hierarchical in describing the resource and what categories it sells in. The top-level node is a root. Each node below the root is an information object that represents the information of a resource (virtual store or virtual mall). If a node is a virtual mall, the members (virtual stores) of the virtual

mall are its child nodes. A node containing the information of a resource is defined as a resource node. Nodes below the resource nodes, denoted as category nodes, are information objects to represent the information of product categories the resource sells. The resource nodes are created with the Resource-level metadata, while the category nodes are created with the Category-level metadata. Notably, according to the CLASS fields of the Category-level metadata, category nodes below a resource node form a hierarchy of category/sub-category relationships.
The system employs DIT to ascertain possible data resources that may sell goods, thereby satisfying the customer's request. Depending on the customer's query criteria, the system may search the resource nodes with/without further searching the category nodes for filtering possible (target) resources. By this approach, the system only needs to submit a customer's query to target resources without submitting to all resources. The integrated search on the Internet can thus be conducted effectively. The primary advantage of using DIT is to simplify the search for a certain product. Owing to the hierarchical relation, rather than travelling all nodes, we can simply bind our search into only a specific sub-tree.

3.3 *Personalized search service*

A user profile describes the user's interests in product items. However, applying that information and determining which product items meet users' demand is required. A matching algorithm is designed to filter and rank product items and determine the items in the highest demand. To support the personalized search service, the system needs to obtain the personalized constraints from the user model, prior to issuing the query to target resources. Before explaining the

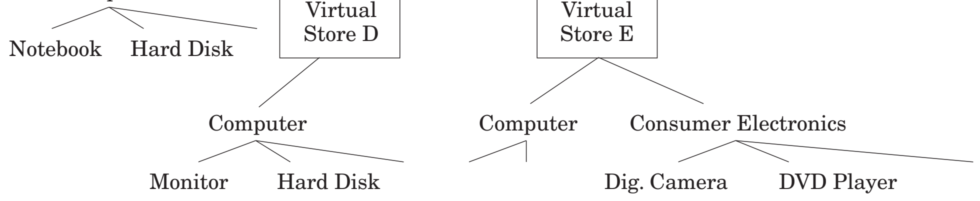


Figure 4. Directory Information Tree (DIT).

detailed design of the match process, we initially illustrate the system's ranking of a certain product item in the following section.

3.3.1 *Ranking*. Every product item has an attribute set and a corresponding value set. The system ranks the product item by referencing user profiles. The rank value reveals the user's preference to that product item. The following vector space calculation shows how a certain item is ranked. Notably, this ranking formula contains attributes with multi-values. The value n represents the number of attributes in the item, while m_i indicates the number of important values of attribute A_i. The ranking formula is expressed as follows:

$$Rank = \sum_{i=1}^{n} \sum_{j=1}^{m_i} w_{A_i} e_j^{A_i} w_j^{A_i},$$

where

$$W = (w_{A_1}, w_{A_2}, \cdots, w_{A_n}), \text{ the weight list of attribute } (A_1, A_2, \cdots, A_n)$$
$$V_a = (w_1^{A_a}, w_2^{A_a}, \cdots, w_{m_a}^{A_a})^T, \text{ the weight list of value } (v_1^{A_a}, v_2^{A_a}, \cdots, v_{m_a}^{A_a})$$
$$M_a = (e_1^{A_a}, e_2^{A_a}, \cdots, e_{m_a}^{A_a}), \text{ where } e_j^{A_a} = \begin{cases} 0 \text{ if item does not have value } v_j^{A_a} \\ 1 \text{ if item has value } v_j^{A_a} \end{cases}$$

The detailed calculation of the rank value is illustrated in the following:

$$Rank = W * \begin{pmatrix} M_1 & 0 & \cdots & 0 \\ 0 & M_2 & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ 0 & 0 & \cdots & M_n \end{pmatrix} * (V_1, V_2, \cdots, V_n)^T$$
$$= (w_{A_1} M_1, w_{A_2} M_2, \cdots, w_{A_3} M_n) * (V_1, V_2, \cdots, V_n)^T$$
$$= w_{A_1} M_1 V_1 + w_{A_2} M_2 V_2 + \cdots + w_{A_n} M_n V_n$$
$$= \sum_{i=1}^{n} w_{A_i} * (e_1^{A_i}, e_2^{A_i}, \cdots, e_{m_i}^{A_i}) * (w_1^{A_i}, w_2^{A_i}, \cdots, w_{m_i}^{A_i})^T$$
$$= \sum_{i=1}^{n} w_{A_i} * (e_1^{A_i} w_1^{A_i} + e_2^{A_i} w_2^{A_i} + \cdots + e_{m_i}^{A_i} w_{m_i}^{A_i}) = \sum_{i=1}^{n} \sum_{j=1}^{m_i} w_{A_i} e_j^{A_i} w_j^{A_i}$$

3.3.2 *Match process*. The system initially selects the proper user profiles that describe the user's preferences of a product category that is being sought after. Then, the system retrieves the personalized constraints, including resource-level and category-level user profiles that are related to the resource-level metadata and category-level metadata, to discover the probable resources. By referencing the query constraints and user profile, the matching algorithm will branch and bound the directory information tree (DIT). The method will filter out all nodes having a rank value lower than a specified threshold.

The algorithm first removes all resource-level nodes (stores) that do not satisfy the resource-level constraints specified in the user query. The remainder of the resource nodes in DIT is ranked via using the resource-level user profile to eliminate unqualified resource nodes with rank values lower than a specified threshold. Consequently, the algorithm restricts the search and concentrates on a more suited category. Next, the algorithm removes category-level nodes that do not satisfy the category-level query constraints. Similarly, the category-level user profile provides a better view of what sort of category a user prefers, which can also be used to rank category nodes in DIT. Target resources are those resource nodes that both the resource node and one of its descending category nodes have ranked with a value higher than the specified threshold value.

Finally, the system issues the query to target resources. Product items of a query result will be further ranked and filtered according to the goods-level user profile. The system maintains the rank values of corresponding resource and category of each product item. The item's final rank value will be computed by adding the rank value of goods itself, the rank value of corresponding resource node and the rank value of corresponding category node.

By separating the user profiles into three levels, the system can use resource-level and category-level user profiles to constrain the search to only certain possible resources satisfying a matching threshold, rather than search all resources. Applying this approach, the integrated search on heterogeneous data resources can thus be conducted effectively. In our current design, all three levels of user profiles are given an equal weight. In practice, a user may have a higher preference on attributes of resource level than those of goods level. Our design can be easily extended to assign different weights to different levels of user profiles.

3.4 *Personalized virtual catalogue*

By establishing a personal user profile, the system can filter the desired information based on the needs and preferences of each user. The system can also construct a personalized virtual catalogue according to the user profile. Virtual catalogues dynamically retrieve product data by combining information from multiple manufacturers' catalogues and present this product data in a unified manner. According to the user profile, each user can have a personalized virtual catalogue. The system constructs a tree structured virtual catalogue by consulting the *Directory Information Tree* (DIT). Once a user profile is established, the system will use the personalized information stored therein as filtering constraints to search the DIT and dynamically construct a personalized virtual catalogue. In general, to construct the virtual catalogue, the system employs the basic information of each user‘s preferred shopping locations, virtual stores, and categories as filtering constraints to construct the virtual catalogue. In addition, resource-level user profile and category-level user profile can be further used as constraints to construct the virtual catalogue. Fig. 5 depicts the construction procedure of a personalized virtual catalogue.

The system constructs a new directory information tree, DIT′, which satisfies the constraints specified in the user profile. Distinct settings within a user profile will produce distinct DIT′, which yields a personalized virtual catalogue. The data stored in directory information tree are searchable. After a user selects a supported constraint, the system will search the DIT to acquire more detailed information based on the user’s selection. This detailed information can be used to set up a profile easier. Users can apply the virtual catalogue to search the products that are sold in each virtual store. After a virtual store is selected, the coverage will focus on that specific virtual store. If the user wants to search product information from all virtual stores, before issuing a query, a user can move to the most outer layer of the virtual catalogue.

4. Agent-based e-catalogue framework

On the basis of the XML-based metadata and user models, we apply the agent technology to develop a personalized e-catalogue system. The system provides a structured query by the attributes of a selected product category. Based upon the user profile, the system also supports the personalized virtual catalogue. The multi-level metadata facilitates searching and retrieving of product data from diverse virtual stores. The domain-level metadata is useful for the system to forward the query to other proper communities. The system uses the resource-level metadata and category-level metadata to discover probable resources within a community. The searchable attributes of diverse categories can be obtained from the category-level metadata. The goods-level metadata translates diverse

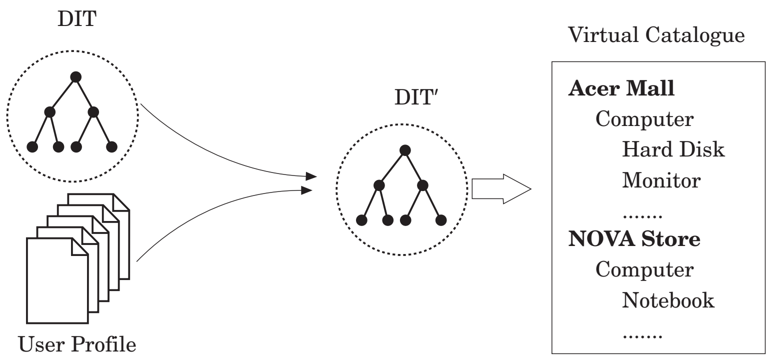


Figure 5. Personalized virtual catalogue.

product information formats from virtual stores into a unified XML format that is used in the e-catalogue system.

Notably, using XML does not eliminate the issue of integrating heterogeneous data formats. The system needs to maintain the ontology and mapping information between the XML-based metadata format and diverse data formats of data resources (virtual stores). The resource agent can then translate data format between the system's goods-level metadata and data format of data resources, by using existing mapping information.

Fig. 6 shows the agent-based electronic catalogue framework. The XML formatted multi-level metadata proposed in Section 2 is applied to describe communities, resources, categories, and products. The proposed brokering architecture consists of the user agent, the facilitator, and the resource agent. The facilitator comprises five components, the control agent, broker agent, category model agent, virtual catalogue agent and domain agent. The user agent connects the system to the client. The facilitator performs routing, resource discovery, data collection, and translation. The resource agent is responsible for retrieving data from a particular resource.

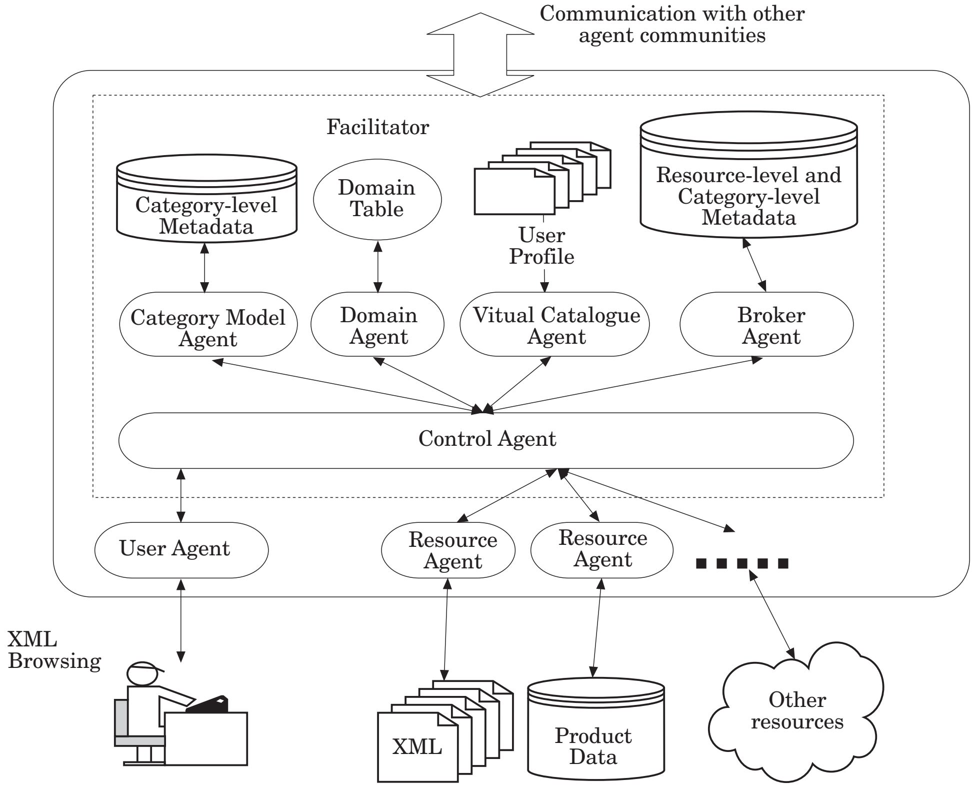


Figure 6. Agent-based personalized e-catalogue framework.

A domain implies an autonomous integrated agent community. The community can be formed by the same subject interest, such as the area or product category. The domain agent makes the architecture more extensible. Once an agent community is initiated, it will advertise within other agent communities. The domain agent uses a domain table to maintain information regarding other agent communities. By classifying resources into different domains, a query can be processed more effectively. A control agent may forward a user's query to control agents in other domains that may contain the probable resources. Once a control agent receives the query, it will collect the result within its domain and then translate the result in XML form to the control agent that initiated the query. Without the domain concept, the cost of communication will increase, as the control agent must directly communicate with each resource agent even though some resources are located in domains (e.g. clothes) unlikely to contain the probable goods (e.g. 128 MB RAM).

To support the personalized service, a virtual catalogue agent is designed to manage the user profile. The virtual catalogue agent maintains user profiles to construct personalized virtual catalogues. Furthermore, several functions on the agents were enhanced to support the personalized filtering service. Each agent performs specific functions and cooperates with other agents. Agents use personalized constraints including the resource, category and goods-level user profiles, to cooperate in the ranking and filtering of product items. The detailed functions of each agent are illustrated in the following.

User agent. A user agent is a bridge between the client and the agent community. A user agent performs the following functions: (1) communicate with the category model agent to obtain supported categories and supported attributes of a category; (2) forward the query to the control agent; (3) receive the XML-formatted result from the control agent and forward the query result to the client; (4) send the DTD of the result to the client system, if necessary.

Besides, the user agent may communicate with the virtual catalogue agent to retrieve the personal profile. To enhance the user agent, the subsequent functions were added: (1) translate the user identity and password to the control agent; (2) translate the personalized virtual catalogue to the client; (3) send the new setting of personal preferences to the virtual catalogue agent.

Control agent. Control agent is the core of the facilitator. It controls the query flow, gathers the result from resource agents and communicates with other agent communities. A control agent performs the following functions: (1) issue the query to the domain agent to discover probable facilitators in additional agent communities; (2) query the broker agent for resource discovery; (3) query the category model agent to verify the capabilities of the probable resources; (4) issue the query to probable resource agents; (5) collect and integrate the results; (6) translate the result to the user agent.

The personalized search service is conducted as described in Section 3.3. To support a personalized service, the control agent will obtain the personalized constraints from the virtual catalogue agent before issuing the query. First, the control agent retrieves the personalized constraints, including the resource-level and category-level user profile to discover the probable target resources satisfying the query constraints and the preference. Then, the control agent issues the query to target resource agents. The returned query result will be further ranked and filtered according to the user profile. The following functions were added to enhance the control agent: (1) retrieve personalized profile from the virtual catalogue agent; (2) merge the query constraints with personalized constraints to query the broker agent for resource discovery; (3) issue the query to probable target resource agents; (4) rank and filter out product items of the query result.

Category model agent. A category model agent manages the capability of each resource, including the specific attributes of product categories for each resource, as well as the translation ontology that defines the translation of attributes between the system and resources. The category model agent was employed to manage specific (searchable) attributes of diverse product categories. A category model agent performs the following functions: (1) register the category-level metadata of each resource; (2) provide information regarding supported categories and supported searchable attributes of each product category to the user agent; (3) verify the capabilities of probable resources.

Broker agent. To maintain the resource-level metadata and category-level metadata, the broker agent constructs a directory information tree (DIT). DIT is employed to check possible data resources that may sell goods, which satisfy the request. Depending on the query criteria, the broker agent may search the resource nodes with/without further searching the category nodes for filtering possible (target) resources. By this approach, the system only needs to submit the query to target resources, without submitting to all resources. The integrated search on the Internet can thus be conducted in an effective way. In our design, a broker agent has the following functions: (1) manage the information regarding resources and categories via DIT; (2) provide the resource-discovery service to find possible resources and return the results to the control agent.

Notably, the broker agent will conduct a resource-discovery service with two different search modes, as discussed in Section 3. In the general search service, the broker agent uses the query criteria to locate possible target resources. In the personalized search service, to discover the probable resources, the broker agent uses the query criteria, resource-level user profile and category-level user profile. The broker agent will branch and bound the DIT by filtering out nodes having a rank value lower than a specified threshold, as illustrated in the match process of Section 3.3.

Resource agent. Resource agents retrieve product information from each resource and forward it to the control agent in XML format. A resource agent performs the following functions: (1) advertise resource-level metadata to the broker agent; (2) advertise category-level metadata to the category model agent;

(3) submit translated queries to a virtual store; (4) transfer the result to XML-formatted goods-level metadata.

Domain agent. The domain agent uses a domain table to record the domain-related information of agent communities, including the URL and the subject interests, such as the supported product categories of each agent community. To forward queries to the control agents in other domains, the domain agent verifies the supported product categories of other domains to discover the probable agent communities. A domain agent performs the subsequent functions: (1) advertise the URL and subject interests in the community to other domain agents; (2) register the domain-related information in other communities into the domain table; (3) search the domain table to discover probable communities and return the result to the control agent.

Virtual catalogue agent. The virtual catalogue agent is applied to manage the user profile and dynamically construct a personalized virtual catalogue. We record each of the preferences in the user profile. With the personalized information, the system can serve each user differently based on their setting. A virtual catalogue agent performs the following functions: (1) manage the user profile; (2) verify the identity and password; (3) dynamically construct a virtual catalogue and translate the virtual catalogue to the user agent; (4) send the personalized information, including resource-level, category-level and goods-level user profiles to the control agent.

5. System implementation

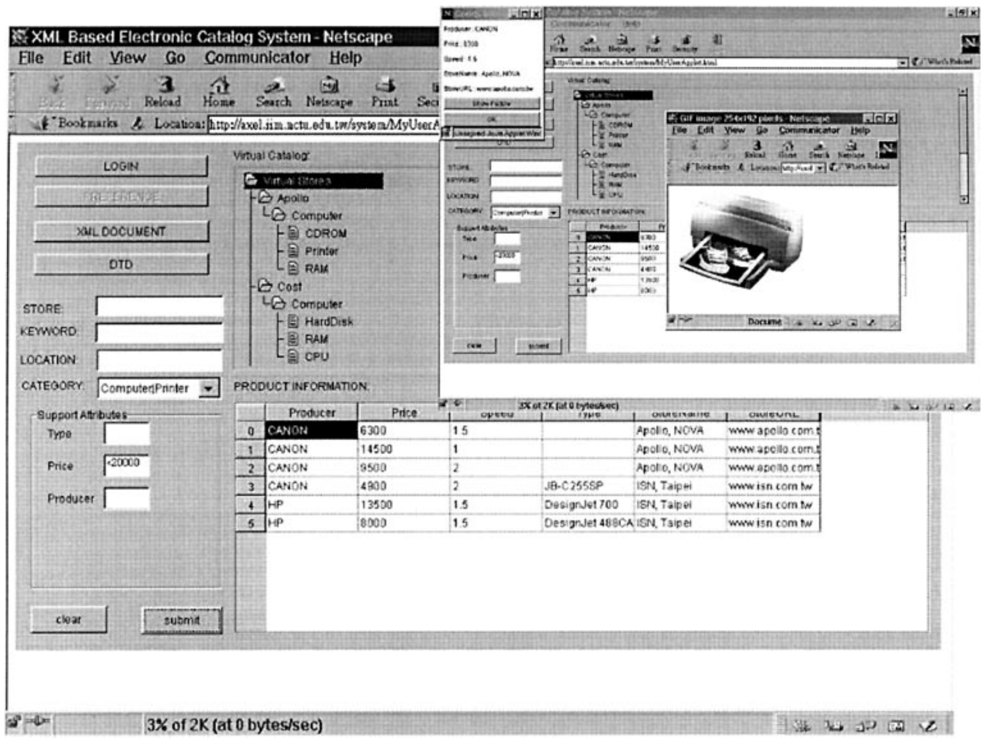
A prototype system is implemented to demonstrate our approach. In the implementation, we consider three heterogeneous data resources in which each supports query capability of XML query and database query by SQL server and MySQL, respectively. Notably, we focus on the implementation necessary to demonstrate the main idea and the feasibility of our approach. Further work is required to carry out a thorough implementation of our system.

A personalized e-catalogue system is designed and implemented to search for product information. The system supports searching and browsing of XML documents. Software agents are used to retrieve data from various resources and translate the result into a unified XML format. Meanwhile, XML Query Language (XQL) [22] is used to filter the desired information from XML-related resources. XQL is a general-purpose query language designed specifically for XML documents. The ability to directly query XML content gives customers the powerful capability to select only the XML content of interest rather than process a set of irrelevant data. The implementation of querying XML documents is based on a package of JAVA-based XQL classes developed by Datachannel Inc. [23]. An XQL parser is created to process the query. The parser parses the XML document and translates the result into XML format. XQL does not specify the physical representation of the result, and the query result may be stored and further processed locally for various applications.

JAVA and Voyager, a JAVA-based agent class developed by ObjectSpace Inc. [24], were used to develop the proposed system. An applet was also developed in a Web browser to communicate with the user agent. The resource agent accesses the related database using JDBC if necessary. MySQL was used as the database herein. To improve the performance, when the control agent translates a query into corresponding queries and then sends queries to several resource agents, the control agent will use multi-threads to perform the work concurrently rather than sequentially.

The system provides a structured search and presents the result in a Web browser. Users can select one of the supported categories, and then enter constraints into the fields of the supported attributes. After submitting the request, the result will be returned in XML format. The applet parses the XML result, and presents it in the Web browser. Fig. 7 shows a snapshot of a screen after the user submitted a request.

In this example, the user wishes to find printers with price less than NT20000 dollars. The supported searchable attributes of the 'Printer' category are 'Price',


XML Based Electronic Catalog System - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print
Bookmarks Location: http://axel.iim.nctu.edu.tw/system/MyUserA
LOGIN
PREFERENCE
XML DOCUMENT
DTD
Virtual Catalog:
Virtual Stores
Apollo
Computer
CDROM
Printer
RAM
Cost
Computer
HardDisk
RAM
CPU
clear
submit
3% of 2K (at 0 bytes/sec)

STORE:
KEYWORD:
LOCATION:
CATEGORY: Computer|Printer

Support Attributes
Type:
Price: <20000
Producer:

PRODUCT INFORMATION:

	Producer	Price	Speed	Type	StoreName	StoreURL
0	CANON	6300	1.5		Apollo, NOVA	www.apollo.com.t
1	CANON	14500	1		Apollo, NOVA	www.apollo.com.t
2	CANON	9500	2		Apollo, NOVA	www.apollo.com.t
3	CANON	4900	2	JB-C255SP	ISN, Taipei	www.isn.com.tw
4	HP	13500	1.5	DesignJet 700	ISN, Taipei	www.isn.com.tw
5	HP	8000	1.5	DesignJet 488CA	ISN, Taipei	www.isn.com.tw

Figure 7. Query product information.

'Type' and 'Producer'. The system discovered six goods satisfying the criteria. To acquire the detailed information of a particular product, the user can click on the row header of the table to demonstrate the particular product information. Notably, the searchable attributes are distinct for various product categories. If the user selects a 'clothes' category, the support attributes shown on the query window will include 'style', 'colour', and 'fabric', which differ from the support attributes of the 'printer' category.

Furthermore, a user can set up personal preferences and dynamically construct a virtual catalogue. The virtual catalogue provides each user with the stores and categories of interest, and hides the information that is of no interest. To set up the personal user profile, the user may initially select a location. The system then dynamically searches the directory information tree and lists the virtual stores and supported categories within the selected location. The user is thus able to browse the stores and the categories of interest. According to the settings of the user profile, the system dynamically constructs a tree structured virtual catalogue.

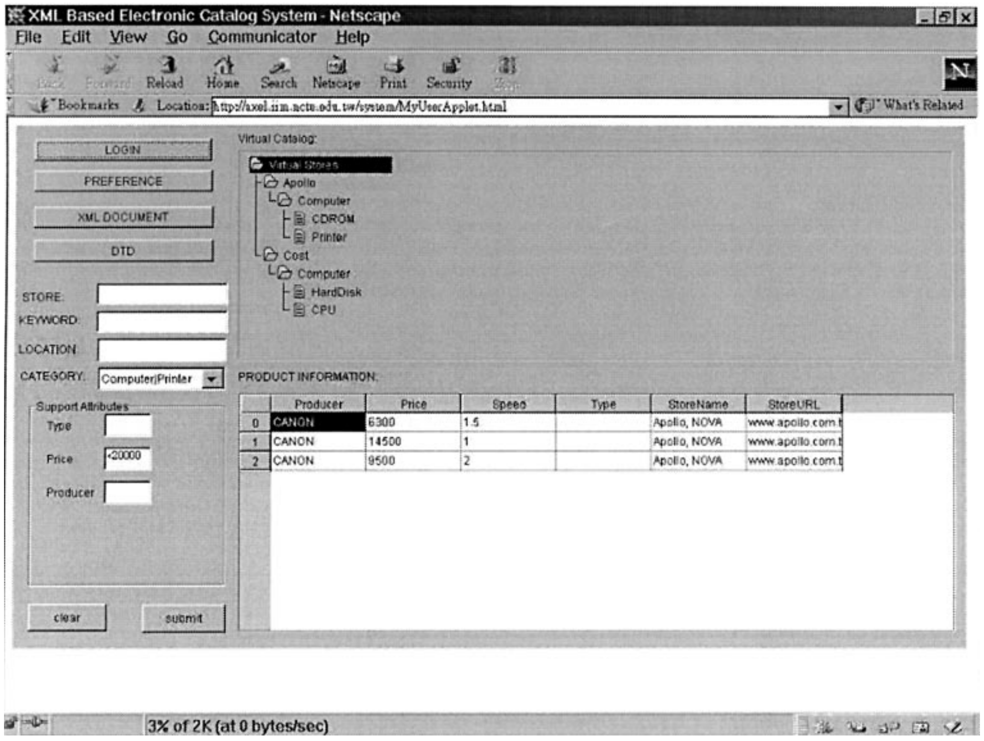
To acquire the product information of a particular store, the user can use the virtual catalogue to select the store and category of interest in. The query will then focus on the product information within the virtual store, as selected by the user. If the user wants to search product information from all virtual stores' before issuing a query, a user can move to the most outer layer of the virtual catalogue. Moreover, with the support of personalized services, based on user profiles, the system is able to filter and rank product items for each user. Figure 8 demonstrates that three CANON printers were found to satisfy the query criteria and the user's preference. The system ranks and filters product items according to the user profile, showing that the preferred printer product is CANON. Notably, the personalized virtual catalogue in Fig. 8 is different from the one in Fig. 7.

6. Discussion and comparison

We have compared our design with related research work on e-catalogue design [13–15] and existing commercial search tools such as Amazon [25], Barnes&Noble [26], and Yahoo [27]. Notably, the design of our system is not targeted for commercial use. The comparison is mainly conducted to demonstrate the characteristics of the research approach proposed in this work.

6.1 *Comparison with related research work on e-catalogues*

Keller *et al*. have proposed smart catalogues and virtual catalogues to support multi-company cross-catalogue searches [14]. Lincke *et al*. also propose a similar work on electronic catalogues [15]. In addition, Web-based e-catalogue systems in B2B procurement have also been proposed [13]. They all use a brokering architecture to retrieve and integrate data from heterogeneous environments. The brokering architecture utilizes agent technology and information brokerage, which


XML Based Electronic Catalog System - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks
What's Related
LOGIN
PREFERENCE
XML DOCUMENT
DTD
Virtual Catalog:
Virtual Stores
Apollo
Computer
CDROM
Printer
Cost
Computer
HardDisk
CPU
clear
submit
3% of 2K (at 0 bytes/sec)

Location: http://axel.iim.nctu.edu.tw/system/MyUserApplet.html

STORE:
KEYWORD:
LOCATION:
CATEGORY: Computer|Printer

Support Attributes
Type
Price <20000
Producer

PRODUCT INFORMATION:

	Producer	Price	Speed	Type	StoreName	StoreURL
0	CANON	6300	1.5		Apollo, NOVA	www.apollo.com.t
1	CANON	14500	1		Apollo, NOVA	www.apollo.com.t
2	CANON	9500	2		Apollo, NOVA	www.apollo.com.t

Figure 8. Personalized filtering and browsing.

is similar to our work. Our approach is different from previous work in several aspects as described in the following.

6.1.1 *Resource discovery and multi-level modelling of goods information*. Previous work on e-catalogue systems do not address issues such as resource discovery and the modelling of separate levels of goods-related information, and these issues are vital to providing effective integrated search services across multiple data resources. Although they had considered the issues of heterogeneous data resources, they did not address how to model multi-category product data stored in heterogeneous data resources. In contrast, our design provides flexible modelling of multi-level goods information to support effective attribute-based searches for multi-category product data from heterogeneous data resources.

- The resource-level metadata provides resource information, such as store profile, contact information, supplied product categories, and location. The information is necessary to provide resource discovery. The system uses resource-level metadata to check possible data resources that may sell goods satisfying the customer's request. For example, the customer may like to purchase product from data resources that provide guarantees of lowest price and full refund policy.

- The category-level metadata is used to model searchable attributes of a product category supplied in a particular data resource. Attribute definitions of a product category, such as attribute names, types and domains, are specified in the category-level metadata. According to the category-level metadata for each data resource, the system is able to obtain the searchable attributes of a product category supplied in the data resource.
- According to the category-level metadata defined for each product category supplied in each data resource, the system is able to define the ontology and translation (mapping information) of attributes between the system and the data resource. Furthermore, the goods-level metadata describes the goods information. To facilitate an integrated search, product information of diverse virtual stores must be translated to a unified format that is described by the goods-level metadata, using the ontology and mapping information.

6.1.2 *Personalization*. Previous works of research do not consider personalized search services. In contrast, our approach provides multi-level user profiles to support personalized search services. To achieve such capabilities, the brokering architecture of our e-catalogue system has been enhanced to integrate with the multi-level metadata design and user profiles. The designed agents are thus able to perform resource recovery, attribute-based search for heterogeneous multi-category data and personalized search services.

6.2 *Comparison with related systems*

Most commercial search tools support keyword search as well as keyword search by category. Both Amazon and Barnes&Noble provide attribute-based searches by category. However, they mainly support searches for product data stored in its own data resource and do not consider heterogeneous data resources. The comparison is illustrated in following aspects.

6.2.1 *Agent-based approach*. Our approach is an agent-based design targeted to conduct an integrated attribute-based search for multi-category product data from heterogeneous data resources.

6.2.2 *Attribute-based search for multi-category data from heterogeneous data resources*. Most search tools provide a keyword-based search. Few have provided an attribute-based search. However, those search tools do not consider attribute-based search for product data from heterogeneous data resources. Their design mainly provides an attribute-based search for each product category stored in its own resource. In contrast, our design is targeted to conduct an integrated attribute-based search for multi-category product data from heterogeneous data resources.

6.2.3 *Handle diverse searchable attributes supported by distinct data resources.* Existing search tools mainly consider product data stored in its own data resource, and do not handle diverse searchable attributes supported by distinct data resources. Several issues arise for designing an integrated attribute-based search. First, the characteristics (attributes) of goods may vary in distinct product categories. Second, data resources (virtual stores) may also employ different methods and data formats to query and store goods information. As a result, for the same product category, the searchable attributes supported by distinct data resources may also be different. Our system employs a multi-level metadata model and agent-based approach to handle the above issues.

6.2.4 *Resource discovery & data translation.* Existing attribute-based search tools such as Amazon and Barnes&Noble are mainly designed to search for product data in its own data resource. In contrast, our design considers attribute-based search for product data from multiple data resources. Our design handles issues of resource discovery and data translation among heterogeneous data resources based on the multi-level metadata model and agent-based approach.

6.2.5 *Personalization.* The personalization has been evolved extensively in research and commercial tools. However, the personalization is mainly targeted for personal Web site and shopping recommendation and is not combined with the search service. Most search tools provide search services without considering personalized filtering. Our system provides both general search services and personalized search services. The general search service conducts a product search according to the query constraints specified in the user's query, without considering personalized filtering, while the personalized search service conducts a product search based on the query constraints and user profiles for further personalized filtering.

7. Conclusion and future work

The proposed XML-based metadata can facilitate resource discovery and format translation, and flexibly model the definitions of diverse attributes to describe goods in various product categories and data resources. Furthermore, in supporting personalized filtering within e-catalogues, the designed user model, with three-level user profiles, can model users' shopping interests. With the multi-level design of both a user and merchandise model, the match process can be carried out efficiently to search and rank users' preferred product items. The main advantage of using DIT in the search and match process is to travel to only a certain sub-tree rather than traveling to all nodes to locate possible target resources. Applying this approach, the system only needs to submit a customer's query to target resources, rather than search all resources. The integrated search on the Internet can thus be conducted effectively.

Moreover, with the integration of XML-based metadata and user models, the designed agents can communicate and cooperate with other agents in resource discovery, format translation, ranking and filtering to achieve search results that satisfy customer needs. The novel system provides a structured query via the attributes of a selected product category, as well as supporting personalized product searches and browsing for each user based on their profiles. Also, with the support of personalized services, the novel e-catalogue system facilitates an effective, integrated search for heterogeneous goods information located in various virtual stores. A prototype system was developed to demonstrate the proposed work. This work not only facilitates business-to-consumer commerce on the Internet, but also contributes to the research on electronic commerce.

In the future, we will expand our system with a more general matching structure to provide a bi-directional suggestion to both customers and merchants. Additionally, the refinement of user profile is a very vital issue to a profile-based information filtering system. However, the refinement of a user profile is a very challenging task. Since users may occasionally alter their preferences, maintaining a current user profile is difficult. In addition, if a user purchases a product item that does not match the behaviour described by user profile, the adjustment of user profile and the re-evaluation of each attribute and value need to be tackled carefully. Hence, user profile refinement provides a challenging topic that is worth future study. Furthermore, data mining techniques can be applied in e-catalogue applications to construct user profiles and provide suggestion services. In the future, we plan to integrate data mining techniques into our system.

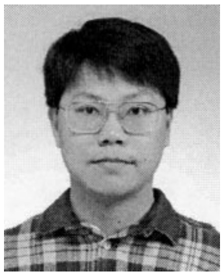
Acknowledgments

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC 89-2213-E-009-019. In addition, the authors want to thank the four reviewers of JNCA of a first draft of this paper for their valuable comments that have assured that this paper has been expanded, corrected and modified to better fit into the framework of similar research efforts.

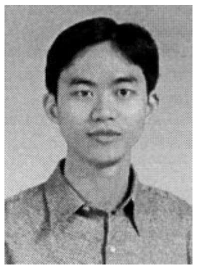
References

1. M. Bichler, C. Beam & A. Segev 1998. OFFER: A broker-centered object framework for electronic requisitioning. In *Proceeding of Intl. IFIP Working Conference: Trends in Electronic Commerce*. Germany: Hamburg, June 1998.
2. S. Caughey, D. Ingham & P. Watson 1998. Metabroker: A Generic Broker for Electronic Commerce. Technical Report 635, *Department of Computing Science, University of Newcastle upon Tyne*, At URL: *http://w3objects.ncl.ac.uk/pubs/mbgbec/TR635/html/index.html*
3. I. Gallega, J. Delgado & J. J. Acebron 1998. Distributed models for brokerage on electronic commerce. In *Proceedings of Intl. IFIP Working Conference: Trends in Electronic Commerce.* Germany: Hamburg, June 1998.
4. International Organization for Standardization. ISO 10303: Product Data Representation and Exchange (STEP). At URL: *http://www.diffuse.org/products.html#STEP*
5. National Aeronautics and Space Administration. Directory Interchange Format (DIF) Writer's Guide, Version 7.0. At URL: *http://gcmd.gsfc.nasa.gov/difguide/difman.html*

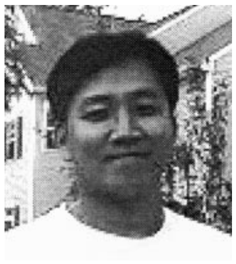
6. World Wide Web Consortium. Extensible Markup Language (XML). At URL: *http://www.w3.-org/XML*
7. B. Meltzer & R. Glushko 1998. XML and electronic commerce: enabling the network economy. *ACM SIGMOD Record* **27**(4), 21–24.
8. M. R. Genesereth & S. P. Ketchpel 1994. Software agents. *Communications of the ACM* **37**(7), 48.
9. R. V. Guha & D. B. Lenat 1994. Enabling agents to work together. *Communications of the ACM* **37**(7), 126–142.
10. T. W. Finin, R. Fritzson, D. McKay & R. McEntire 1994. KQML as an agent communication language. In *Proceedings of the third ACM Intl. Conference on Information and Knowledge Management (CIKM'94).* Maryland: Gaithersburg, Nov. 29–Dec. 2, 1994, 456–463.
11. C.-C. K. Chang, L. Gravano, H. Garcia-Molina & A. Paepcke 1997. STARTS: Stanford proposal for internet meta-searching. In *Proceedings of the ACM SIGMOD Intl. Conference on Management of Data*, USA: Tucson, Arizona, May 1997, 207–218.
12. R. J. Bayardo Jr, W. Bohrer & R. Brice 1997. InfoSleuth: Agent-based semantic integration of information in open and dynamic environments. In *Proceedings of the ACM SIGMOD Intl. Conference on Management of Data*. USA: Tucson, Arizona, May 1997, 195–206.
13. J. P. Baron, M. J. Shaw & A. D. Bailey Jr. 2000. Web-based e-catalogue systems in B2B procurement. *Communications of the ACM* **43**(5), 93–100.
14. A. M. Keller & M. R. Genesereth 1996. Multi-vendor catalogues: smart catalogues and virtual catalogues. *EDI FORUM: Journal of Electronic Commerce* **9**(3), 87–93.
15. D. M. Lincke & B. Schmid 1998. Mediating electronic product catalogues. *Communications of the ACM* **41**(7), 86–88.
16. S. Laine-Cruzel, T. Lafouge, J. P. Lardy & N. B. Abdallah 1996. Improving information retrieval by combining user profile and document segmentation. *Information Processing & Management* **32**(3), 305–315.
17. P. W. Foltz & S. T. Dumais 1992. Personalized information delivery: an analysis of information filtering methods. *Communication of the ACM* 35(12), 51–60.
18. Q. Lu, M. Eichstaedt & D. Ford 1998. Efficient profile matching for large scale webcasting. *Computer Networks and ISDN Systems* **30**(1–7), 443–455.
19. S. Ramanathan & P. V. Rangan 1994. Architectures for personalized multimedia information services. *IEEE Multimedia* **1**(1), 37–46.
20. M. Barra, G. Cattaneo, M. Izzo, A. Negro & V. Scarano 1998. Symmetric adaptive customer modelling for electronic commerce in a distributed environment. In *Proceedings of Intl. IFIP Working Conference: Trends in Electronic Commerce*, Germany: Hamburg, June 1998.
21. World Wide Web Consortium. Guide to the W3C XML Specification ("XMLspec") DTD, Version 2.1. At URL: *http://www.w3.org/XML/1998/06/xmlspec-report.htm*
22. World Wide Web Consortium. XML Query. At URL: *http://www.w3.org/XML/Query*
23. Datachannel Inc. DataChannel XML Parser for Java. At URL: *http://www.datachannel.com/*
24. ObjectSpace Inc. Voyager 2.0B2. At URL: *http://www.objectspace.com/products/voyager*
25. Amazon.com. At URL: *http://www.amazon.com*
26. Barnes&Noble.com. At URL: *http://www.bn.com*
27. Yahoo!. At URL: *http://www.yahoo.com*



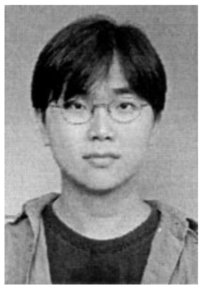
Duen-Ren Liu received the BS and MS degree in Computer Science and Information Engineering from the National Taiwan University, Taiwan, in 1985 and 1987, respectively, and the PhD degree in Computer Science from the University of Minnesota in 1995. He is currently an associate professor of the Institute of Information Management, National Chiao Tung University, Taiwan. His research interests include database systems, electronic commerce, workflow systems, and Internet applications. Dr Liu is an associate member of the IEEE and a member of the ACM.



Yuh-Jaan Lin received the BS degree in Computer Science from the Tung-Hai University, Taiwan in 1997, and the MBA degree in Information Management from the National Chiao Tung University, Taiwan in 1999. He is currently a software engineer of AVerMedia Technologies, Inc. His research interests include electronic commerce, Internet applications, and embedded systems.



Chung-Min Chen received a BSc degree in Computer Science and Information Engineering from the National Taiwan University and a PhD degree in Computer Science from the University of Maryland, College Park. Currently, he is a research scientist at Telcordia Technologies (formerly known as Bellcore). His research interests include database systems and telecommunications.



Ya-Wen Huang received the BS degree in Computer Science from the National Tsing Hua University, Taiwan in 1997, and the MBA degree in Information Management from the National Chiao Tung University, Taiwan in 1999. His research interests include electronic commerce, information systems, and Internet applications.