

DESIGNING A CATALOG MANAGEMENT SYSTEM– AN ONTOLOGY APPROACH

Hadi Khosravi Farsani, Mohamad Ali Nematbakhsh

Department of Computer Engineering, University of Esfahan, Esfahan, Iran

Emails: hadi_farsani@yahoo.com, nematbakhsh@comp.ui.ac.ir

ABSTRACT

When enterprises collaborate, a common frame of understanding of all products and catalogs in each organization is indispensable. Suppliers who virtually collaborate should be able to share product related data, create new products, and update old products from their own catalog. In this paper, we describe the development of a model for presenting electronic catalog based on OWL ontology language. This model uses WordNet ontology to distinguish classes and the relationships between them. We use SPARQL query language to introduce three types of search for the catalog management system. The concepts of this classification system are mapped to the concept of current standard classification systems such as UNSPSC, ECL@SS, and etc. We use VSM to diagnose class of one product. For customization aspect of electronic catalog, we introduce a semantic recommendation procedure which is more efficient when applied to Internet shopping malls. The suggested procedure recommends the semantic products to the customers and is originally based on Web usage mining, product classification, association rule mining, and frequently purchasing. We applied the procedure to the data set of MovieLens Company for performance evaluation, and some experimental results are provided. The experimental results have shown superior performance in terms of coverage and precision.

Keywords: *Electronic Catalogs, OWL Ontology, Classification, Personalization, Recommendation, Query language.*

1.0 INTRODUCTION

Product information (often in the form of e-catalogs) is an essential component in e-commerce. Clear and precise products and services definition is a necessary foundation for collaborative business processes. By sharing a precise product model containing rich semantics, a high level of interoperability can be offered for e-business systems. One possibility for improvement is dramatically improved supply chained management with more accurate and up-to-date information on current inventories and demands. More accurate product plans can be formed with significant saving in costs.

However, researches have shown that searching products on Internet is a difficult process [1]. Customers find their requirements with some difficulty and in most cases they give up their product searching. Consequently, e-catalogs cannot fulfill their duty successfully. One of the factors for this lack of success is the existence of unstructured product information and huge bulk of this information with no search tool. Most companies believe that the existence of more information such as multimedia product information will help to attract more customers, but it is mostly confusing to the customers. Therefore, in order to make use of the maximum potential of markets, providing a comprehensive approach to the organization and presentation of product information is necessary.

The paper-based catalogs form the basis of the electronic catalogs. The paper-based catalogs have facilitated purchasing from customers. These catalogs have colored and structured presentation of products. E-catalogs are direct translation of the paper-based catalogs to the electronic type. Their styles gradually change with respect to the type of information they carry. The first type of the electronic product catalogs was the compact disk-based catalogs. Most of them are still not online, but in comparison with the paper-based catalogs, they have a sophisticated search functionality and also can present the multimedia presentation of the products [2].

IEPCs (Internet Electronic Product Catalogs) are new versions of electronic catalogs that are online and can integrate with other activities of the company and its commercial partners. These catalogs are up-to-date and change with respect to the requirements of the user and establish direct communication channels between buyers and sellers.

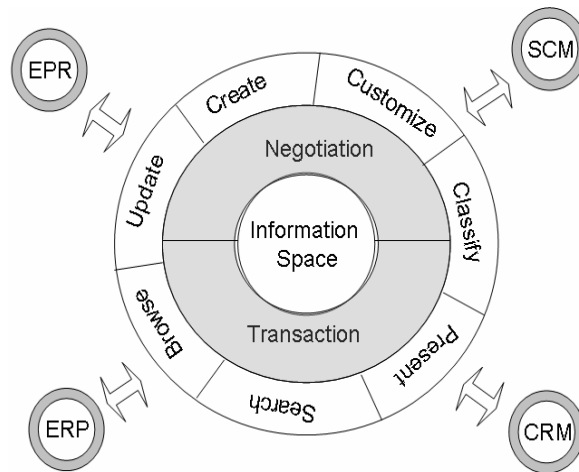


Fig. 1: Catalog Function

Electronic catalog management systems (ECMSs) are mediator between customers and electronic catalog. There are several definitions for electronic catalog management system (ECMS). Segev [3] defined ECMS as "virtual gateway for the buyer to obtain information about the product, purchase the product, make payment, access support services, and cooperate with the sellers...". A more narrow definition is the one provided by Timm and Rosewithz [4]. They defined ECMS as systems, which "allow customers to browse through multimedia product presentations and to get relevant information concerning the products". Keller [5] has defined ECMS "as the reference for product selection and which can assist with source selection and description of terms and conditions". The above mentioned definitions focus on the functions of e-catalogs: product representation, product classification and product search. In this paper, we define e-catalogs which are shown in Fig. 1, as "ECMS with a high volume of interactive and multimedia information on the Internet that have several abilities such as creating, updating, browsing, presenting, classifying, and customizing the information that assist users to maintain and retrieve the product information".

Principal elements of catalog management system include the modeling of Web objects (for example, products or pages) and subjects (customers), categorization of objects and subjects, matching between and across objects and/or subjects, and determination of a set of actions to be recommended for personalization. In this paper, we employ all of these elements for our model. In section 2, we review related works on this context and some challenges for electronic catalog. We model and classify products based on primitives of OWL ontology language in section 3. Section 4 contains modeling of the customers as well as product model. Search and navigation on catalog is another challenge which is presented in section 5. Recommendation procedure is discussed in section 6. The experimental results are presented in section 7. Finally, section 8 is devoted to the conclusions derived from this study.

2.0 RELATED WORKS

Designing electronic catalog management system (ECMS) is complicated by a number of factors. Presenting product information, designing classification system, search and navigation aspects of these systems, diagnosing class of one product and customization or recommendation system are challenges for ECMS. Cui discussed the requirements and issues in catalog management system and the limitations of currently available solutions [6]. He believes that heterogeneity at the system, syntax, structural and semantic level increase the time and cost of catalog publication. He has developed a shared ontology using the terms from UNSPSC (United Nations Standard Products and Services Code). Each company imports this shared ontology and customizes it to cover its needs. This approach focuses specially on semantics and does not consider other catalog capabilities such as searching and customization. Using UNSPSC as a shared ontology is another problem because not all of the companies may accept this classification system.

Presenting product information is an aspect of ECMS which affects another aspect of ECMS systems. There are several challenges for presenting product information. One of them being the use of different vocabularies for describing the products: the synonyms and homonyms, the various abbreviations and notations, different languages, etc. Another challenge contributing to this complication is the huge amount of product information. E-catalogs must present necessary and sufficient information based on user types. The third challenge is different schema for products; different attributes and different data types for an attribute of one product lead to many problems in these

systems. And the last challenge is diversity in the users' views. For example, manufacturers will tend to view products in terms of materials, while a distributor would be more interested in the size and weight of the products.

Each ECMS must use one classification system which classifies products and customers. In the real world, it is better to use one standard classification system in e-commerce among all institutions. However, it seems to be impossible due to at least two reasons. Products come and go and consequently the classification systems are continually being developed. The other reason is that institutions cannot wait long for one standard classification system to arise. That is why many product classification systems have been released.

The first product classification is UNSPSC. It is a horizontal classification system and has five levels [7]. Each level contains two numerical characters and one textual description. A part of this system has been shown in Fig. 2. The most important challenges that exist in product classification systems are as follows. First, they did not have any attribute for concepts. Second, there is no inheritance between products and finally, they have not provided different views of classification. The lack of the complete and reference model for product presentation has led to different types of researches in this context. The challenges of classification systems have been analyzed in [8]. Leukel has explained standard set of attributes for such systems [9].

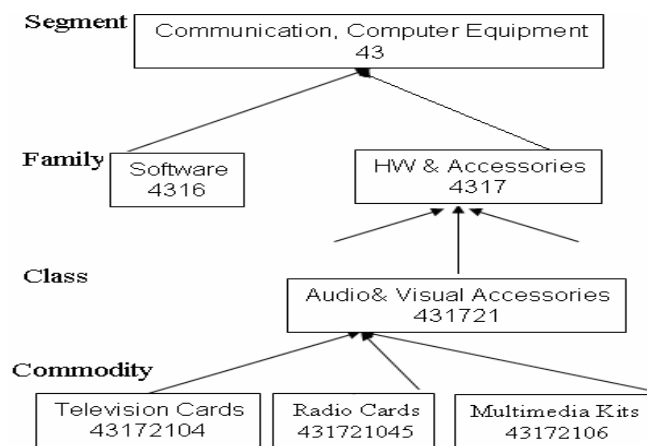


Fig. 2: UNSPSC Classification System

In order to reach an effective model, the process of database design for electronic catalogs is described by Kim [10]. In Kim's approach, a model for product classes has been presented. The attributes of a model for electronic catalogs has been numerated in [11] and then semantic model has been proposed for classifying products in catalogs. This model was used as a reference model for implementing product ontologies [12]. In this implementation, ontology model has been mapped to relational databases. Using ontology with different styles is one of the best approaches for modeling electronic product catalogs. Mapping between types of ontologies is one of these styles [13]. In this approach, each e-market makes its own ontology and maps each concept of its own ontology to one concept in the other ontology of upper level. The study of catalog lifecycle and role of ontology in this cycle has been performed by Fensel [14]. Mapping between different classification systems is possible using WordNet ontology [15] and using ontology primitives for presenting product catalog is declared in [16].

Search and navigation mechanisms of the ECMSs are another challenge for the catalog management systems. Hierarchical, parametric and free keyword search are three types of navigation for ECMS. Customer uses hierarchical type when he does not know exactly what he requires and uses parametric search to restrict the attributes of one product. Keyword search is also another search method and is used when the customer only know some keyword of his/her product. These three types of search mechanism were used in database systems. The precision of the database searching depends on the query and query languages used in it.

Growing products and services for the World Wide Web have pushed designers of electronic catalog management systems to develop recommender systems [17, 18]. So, recommendation system is another challenge of ECMSs. Recommender systems are personalized information filtering technology used to either predict whether a particular user will like a particular item (prediction problem) or to identify a set of N items that will be of more interest to a certain user (top-N-recommendation problem). In recent years, recommender systems have been used in a number of

different applications such as recommending products a customer will most likely buy; movies, TV programs, or music a user will find enjoyable; identifying web pages that will be of interest; or even suggesting alternate ways of searching for information [19].

To date, most recommendation systems fall into three major categories; manual decision rule systems, collaborative filtering systems, and content-based filtering systems [20]. Manual decision rule systems enforce administrator to specify rules based on static profiles or session histories. The rules are used to affect the content served to a particular user. Collaborative filtering systems take information in the form of user ratings or preferences, afterwards try to discover similarity between items or users and accordingly, recommend the product. Content-based filtering approach relies on content similarity of items to personal profiles obtained explicitly or implicitly from user [21].

Two approaches have been developed to realize the collaborative filtering-based systems. The first approach, referred to as *user-based*, relies on the fact that each person belongs to a larger group of similarly behaving individuals. As a result, the items frequently purchased by various members of the group can be used to form the basis for recommended items [22, 23, and 24]. The second approach, known as *model-based*, analyzes historical information to identify relations between different items, e.g., the purchase of an item often leads to the purchase of another item, and then uses these relations to determine the recommended items. User-based collaborative filtering approaches tend to produce systems that lead to higher quality recommendations in comparison with model-based schema [25]. But the complexity of computing each recommendation grows linearly with the number of users and items. These systems also have "new item" problem which means that new items will not be recommended until they are bought or visited by other customers.

Designing an effective recommender system must tackle a number of challenges. One of them is the modeling and presentation of recommended items. Modeling of customers' behavior is another challenge. Up to now, there is no comprehensive model for customers' behavior, but they can almost represent customers' behavior. The third challenge is the recommender algorithm and its quality. The last challenge is evolution of recommender system. There is no criterion accepted by all for comparing recommendation systems [26, 27].

The new generations of Web personalization tools are attempting to incorporate techniques for pattern discovery from Web usage data [28, 29]. Web usage mining systems run any number of data mining algorithms on usage data which is gathered from one or more Web sites in order to discover user profiles. The increasing focus on Web usage data is due to several factors. The input is not a subjective description of the users by the users themselves, and thus is not prone to biases. The profiles are dynamically obtained from user patterns, and thus the system performance does not degrade over time as the profiles grow. Furthermore, using only the content similarity as a way to obtain aggregate profiles may result in missing important semantic relationships between Web objects. Thus, Web usage mining can reduce the necessity for obtaining subjective user ratings or registration-based personal preferences [30, 31, 32, and 33].

3.0 PRODUCT CLASS BY OWL

A (*product*) class is a set of products. Taking P as the set of all products in consideration, this class is a subset of P . A class can be defined by enumerating its members. Each class has several attributes. From among of these attributes, SubClassOf, WordNet, UNSPSC, ECL@SS are very important. SubClassOf attribute is used for classification hierarchy. We use WordNet attribute for the meaning and purpose of the class. UNSPSC and ECL@SS are used for mapping between classes of these classification system and standard classification systems. We define product in consideration as below:

Definition: A *product set* P_s over a set of products P is a 3-tuple $\langle P, PC, PC_1 \rangle$, where P is the set of all products (in consideration), $PC = \{PC_1, PC_2, \dots, PC_x\}$ is the set of all product classes that each class is in the form of $\{(A, V), R\}$. $P = \{P_1, P_2, \dots, P_m\}$ is the set of all products. PC_1 is the root product class. $PC_i = \langle \{(A_1, V_1), (A_2, V_2), \dots, (A_n, V_n)\}, R \rangle$. A_1, A_2, A_3 and A_4 attributes are used as SubClassOf, WordNet, UNSPSC and ECL@SS property.

Using ontology and ontology languages for product classification system is the best choice. According to the studies that have been performed, OWL is selected as the best language from among the current ontology languages [34,

35]. The OWL ontology language takes into consideration the requirements of electronic catalog. In this paper, we present electronic catalog using OWL ontology language.

We must use ontology primitives for modeling e-catalogs. Each of the current concepts in e-catalogs is mapped equivalently in OWL ontology language. Classes of products will be presented by *class* in ontology language. Product specifications will be described using object attributes. In this model, class hierarchy in electronic catalogs will be implemented with *ISA* or *SubClassOf* primitive. And finally, restrictions on the class or attribute are modeled using *Domain*, *Range*, etc.

Product Information is a set of attributes and values. For example, 'Camera' product has color attribute that has 'silvery' value. This data modeling was supported by RDF (Resource Description Framework). RDF data modeling is A (O, V) where 'O' object has 'A' attribute that its value is 'V'. Camera class is a subclass of electronic devices which its type is 'camera'. It has sensor, prodNum, lens, dimension and retailPrice attributes. The camera class with its corresponding attributes is shown in Fig. 3.

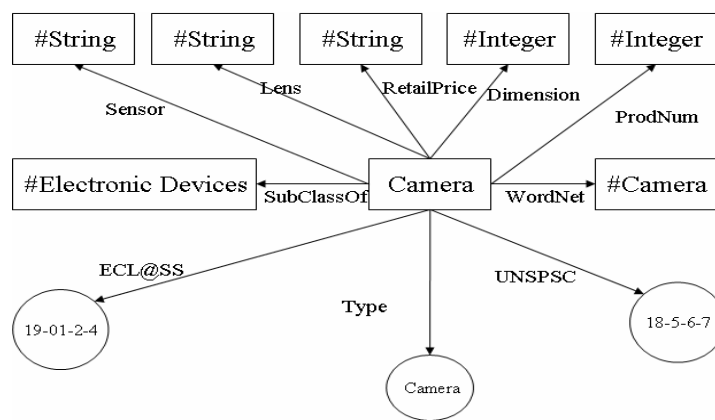


Fig. 3: Camera Class Definition in RDF

In this model, each class also has UNSPSC, ECL@SS and other attributes. These attributes help customers to compare products among electronic markets. This class and one of its attributes are modeled as below.

```
<OWL: Class RDF: ID= 'Camera'>
  <RDFS: SubClassOf RDF: resource = '#Electronic Device ' />
</OWL: Class>

<RDF: Property RDF: ID='ProdNum' >
  <rdfs:Domain rdf:Resource = '#Camera' />
  <rdfs:Range rdf:Resource='#integer' />
</RDF: Property>
```

'Film Camera' class is a subclass of 'camera' class. This class inherits all of the attributes of super class and has other attributes such as 'filmSize', 'loading', and 'filmSpeed'. As you can see in Fig. 4, the value of sensor attribute of this class is 'film'. OWL ontology language will provide this restriction by using <OWL: restriction> tag. Implementation of this class in OWL ontology language is similar to 'camera' class.

```
<OWL: Class RDF: ID = 'Film Camera'>
  <RDFS: SubClassOf RDF: Resource='#Camera'>
  <OWL: Restriction >
    <OWL: onProperty RDF:Resource='#sensor' />
    <OWL: HasValues> film </OWL: HasValue>
  </OWL: Restriction >
  </RDFS: SubClassOf >
</OWL: Class>
```

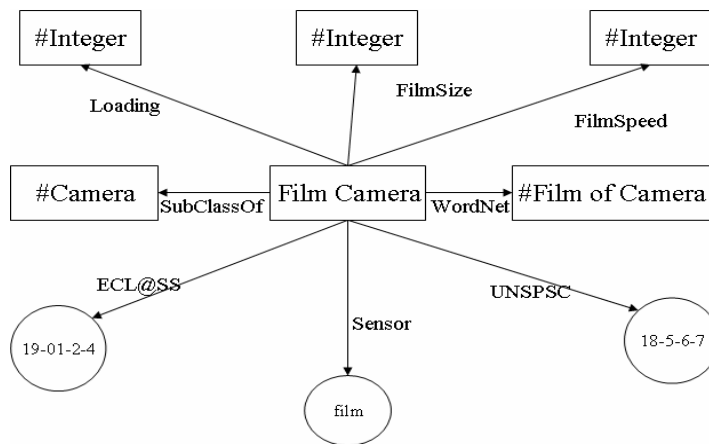


Fig. 4: Film Camera Class Definition

'High end digital camera' is subclass of both 'digital camera' and 'high end camera'. We can present this situation using <OWL: Unionof> tag in OWL. If a class is defined by this tag, it inherits all of the attributes from super classes. Also, <OWL: Intersectionof> tag is used for intersection of two classes.

```
<OWL: Class RDF: ID = 'High End Digital Camera'>
  <OWL: Unionof RDF: ParseType='collection'
    <OWL: Class RDF: about='#Digital Camera' />
    <OWL: Class RDF: about='#High End Camera' />
  </OWL: Unionof>
</OWL: Class>
```

We can define class instances in OWL language (See Fig. 5). For example, 'Camera-Sony-2040' is an instance of 'camera' class.

```
< Camera RDF: ID = ' Camera_Sony_2040' >
  <Pro_num> 2040 </Prod_num>
  <manufacture rdf:resource= '# Sony Company' />
  ...
</Camera>
```

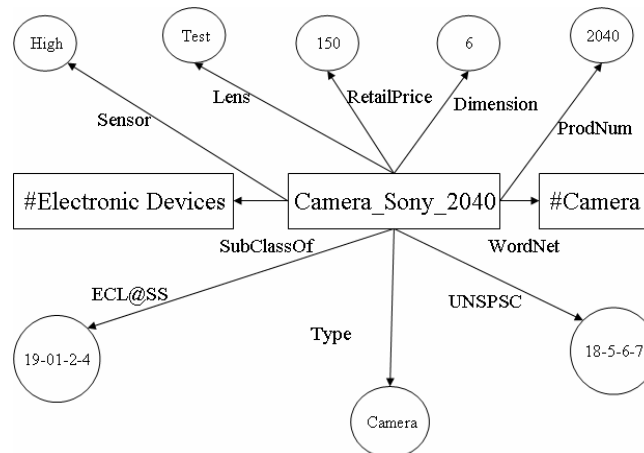


Fig. 5: Instance of Camera Class

We can improve this model using other primitives of OWL. <OWL: disjointwith> tag is used for defining independent classes. We use <OWL: equivalentclass> tag for two class that are the same. In OWL, the maximum

and minimum of an attribute will be determined using <OWL: maxcardinality> and <OWL: mincardinality> tags. Finally, we have a classification hierarchy for products after using these primitives from OWL language as you can see in Fig. 6.

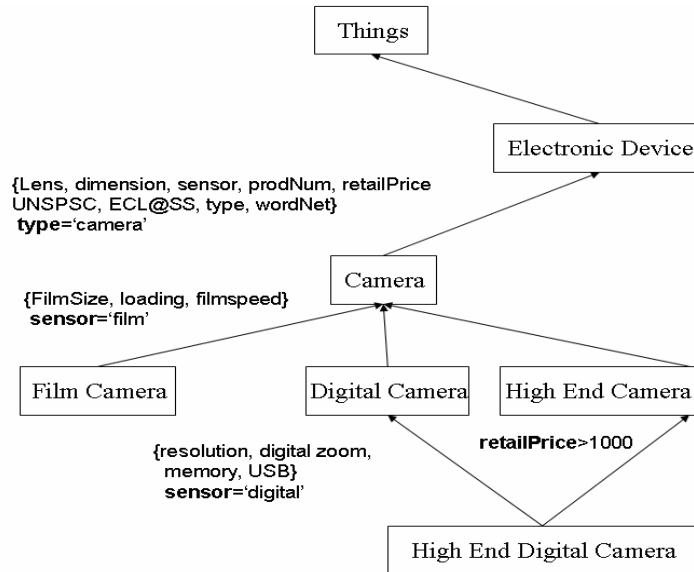


Fig. 6: Class Hierarchy of Products

4.0 CUSTOMER CLASS BY OWL

The proposed system classifies customers using the same model which is used for product classification. It has defined a number of attributes for each customer class such as name, description, genre, and so on.

Definition: A customer set C_s over a set of Customers C is a 3-tuple $\langle C, CC, CC_1 \rangle$, where C is the set of all products (in consideration), $CC = \{CC_1, CC_2, \dots, CC_y\}$ is the set of all customer classes that each class is in the form of $\{(A, V), R\}$. $C = \{C_1, C_2, \dots, C_z\}$ is the set of all customers. CC_1 is the root customer class. $CC_i = \{(A_1, V_1), (A_2, V_2), \dots, (A_n, V_n)\}, R$. A_1, A_2, A_3 and A_4 attributes are used for SubClassOf, WordNet, UNSPSC and ECL@SS property.

We used Protégé tool for implementing product and customer classes. This tool is very powerful for creating OWL ontology and other ontology languages such as DAML+OIL, RDF, RDF(S) and OIL. It is developed in Stanford medical informatics at year 1995.

We take UNSPSC classes and its structure in order to present product catalog using OWL ontology language. Part of the implementation has been shown in Fig. 7. There are 30 classes at first level and each of them is divided to other classes. This approach continues until five levels are created (It is also possible for creating extra levels). For example, 'Printing and photographic and audio and video and visual equipment and supplies' class at first level has been divided to several other classes. 'Digital Camera' class is subclass of 'cameras' class which has several properties. Some of these properties have been inherited from parent class. All of the properties of this class are shown at the right corner of Fig. 7. Also, you can define several restrictions on class properties (refer to the middle of Fig. 7).

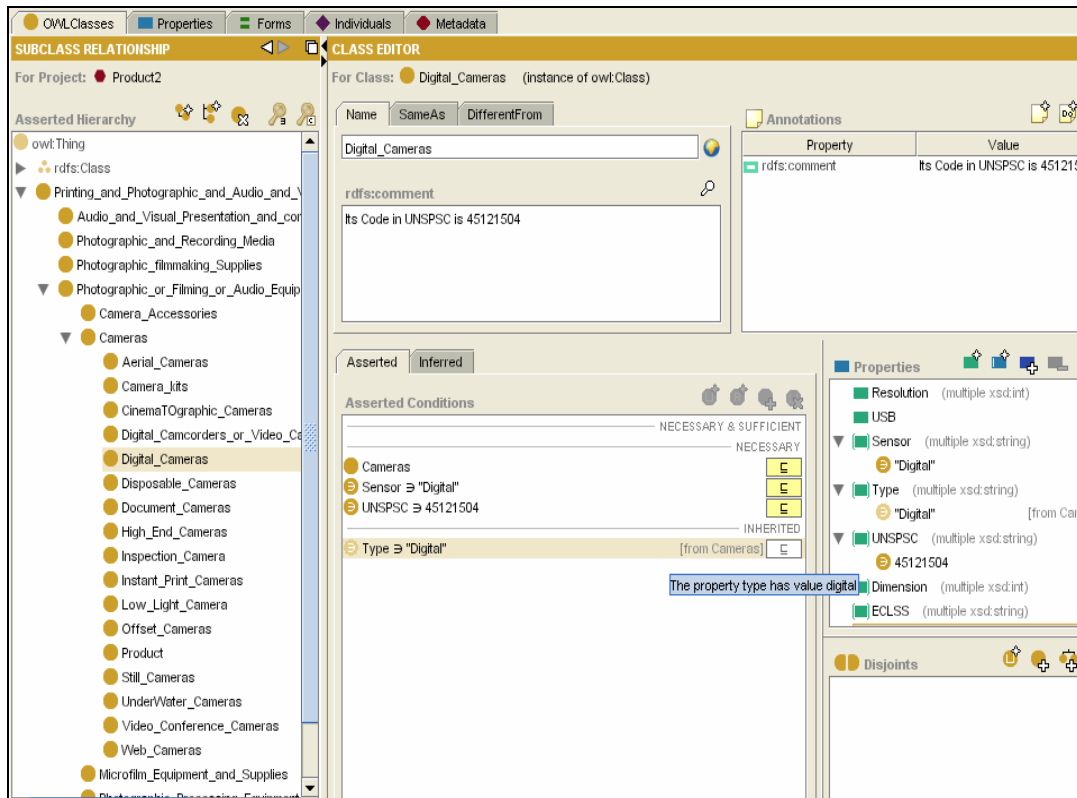


Fig. 7: Definition of catalog classes by Protégé

5.0 SEARCHING ONTOLOGY

Ontology languages and corresponding query languages play key roles for representing and processing information about products. Efforts have been made to develop various ontology languages. Each ontology language provides different expressive power and also computational complexity. Currently, ontology query languages have been developed. SPARQL, RDQL, OWL-QL are some of these query languages [36, 37].

We use SPARQL query language for searching on catalog [38]. The SPARQL query language consists of the syntax and semantics for asking and answering queries against RDF graphs. SPARQL contains capabilities for querying by triple patterns, conjunctions, disjunctions, and optional patterns. It also supports constraining queries by triple RDF graph and extensible value testing. Results of SPARQL queries can be ordered, limited and offset in number, and presented in several different forms.

Electronic catalog should support three types of search. Hierarchical, parametric and free keyword search are types of navigation in catalog. Hierarchical searching is used when customers do not know what they are looking for. Customer selects one class from the upper level classes. Several other classes are derived from the selected upper level class and customer again chooses one of the derived classes. This process continues until customer found her/his favorite product. In addition, user interface of the site must support this navigation.

Parametric search is another type of searching mechanism. If you are searching for a particular product by parametric values, simply select a product family from the product classes. The resulting page will contain a menu with a list of attributes which you can constrain your search by adding constraints on the attributes of this product class. The catalog management system searches the catalog and selects the products which fit your parametric-based query. Another type of search mechanism is free keyword search. In this type of search, customer enters keywords that describe his/her product. Catalog management system must search the schema and data for retrieving the desired product.

We implement the above three types of search for our catalog management system. In our system, customer selects one type of our searching mechanisms and enters her/his data. The system translates this data to SPARQL query

language. We use TopBraidComposer plug-in [39] to evaluate some query on the catalog files. The result show that query precision has been improved.

Example 1: Select all products which derived from 'Digital Camera'.

```
PREFIX dc: <http://localhost/Ontology/>
SELECT ?x
  WHERE { ?x dc:SubClassOf "Digital Camera" }
```

Example 2: Select all products with black color and price is less than 30 dollar.

```
PREFIX dc: <http://localhost/Ontology/>
SELECT ?x
  WHERE { ?x dc:Color "Black" ,
          ?x dc:Price ?Price,
          FILTER(?Price <30) }
```

Example 3: retrieve UNSPSC and ECL@SS code of Camera.

```
PREFIX dc: <http://localhost/Ontology/>
SELECT ?unspscCode , ?eclssCode
  WHERE {?product dc: name "Camera",
         ?product dc: UNCPSC ?unspscCode
         ?product dc: ecl@ss elcssCode }
```

6.0 RECOMMENDATION PROCEDURE

This paper suggests a methodology for personalized recommendations in an e-commerce environment. The methodology consists of five phases as shown in Fig. 8. The recommendation problem should be initiated by classifying the products and customers to improve the analysis process of similar customers and products. In phase II, active customers will be selected by considering a number of previous recommendations. The system does not recommend to a customer if the number of earlier recommendations to him/her has not exceeded a dedicated amount of threshold. Class of customer is been determined in phase III. The next two steps contribute in the process of performing recommendation and utilizing the proposed rating matrix.

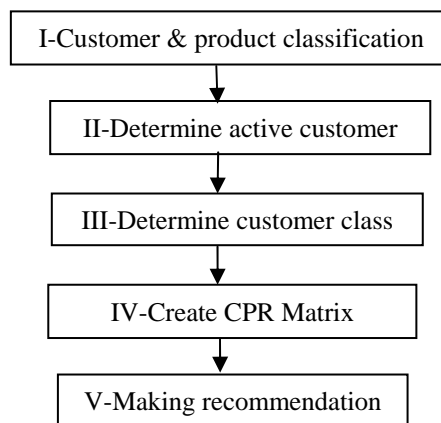


Fig. 8: Recommendation procedure

6.1 Identify Active Customers

In making recommendation, only the customers who are likely to buy the recommended products are considered. Thus, the false positives of a poor recommendation will be avoided. This phase performs the role of selecting such customers based on previous Top-N Recommendations.

The recommendation system maintains previous Top-N Recommendation in the profile of each customer. The profile of a customer includes two parts (Fig. 9). First part is used for presenting personal information such as

instanceOf, numRecommend, name, age, location and etc. InstanceOf is used to indicate to which class the customer belongs. A number of previous recommendations are maintained by numRecommend.

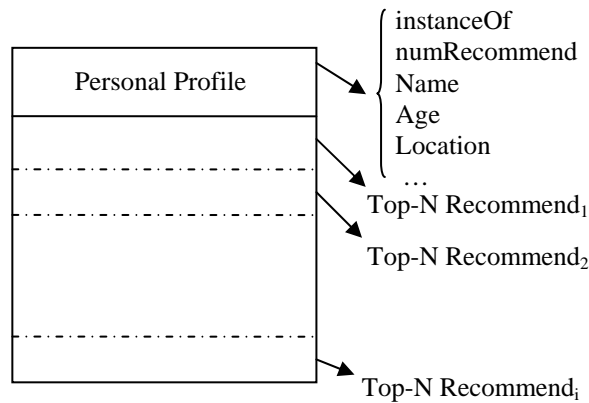


Fig. 9: Customer profile

If the number of previous recommendations of one customer exceeded maxRecommend (which is maintained in customer class), the system deactivates this customer and in the future does not recommend any product to him/her. The customer will become active when he/she purchases some of the products. The numRecommend variable will be reset to zero to let the system recommends products to this customer again.

6.2 Customer-Product Rating Matrix

Most of the model-based recommender systems use the product-product matrix which represents the relationships between products. Customer-Product rating matrix is another choice for these recommender systems where the preferences of each customer to each product are determined by this matrix. Computation complexity of these matrixes grows linearly with the number of customers and products. So, computation of these matrixes requires a lot of time and space.

Our proposed solution to this scalability issue is by utilizing the classification of products and customers. The proposed Customer-Product Rating (CPR) matrix involves the preferences of customers in class i to products in class j and will be shown as $CPR [i, j]$ as you can see in Fig. 10.

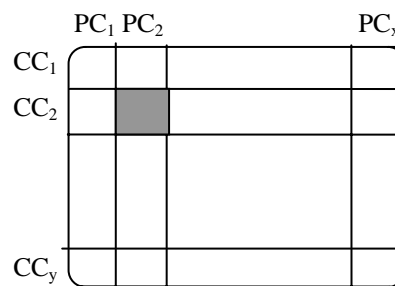


Fig. 10: Customer-Product Rating Matrix (CPR)

The value of $CPR [i, j]$ is between zero and one that is updated dynamically at each purchasing process. When the customer enters the electronic market, it is supposed that he/she belongs to class a ($CC[a]$) and has purchased these products:

$$Buy-CID-Date = \{(i_1, n_1), (i_2, n_2), (i_3, n_3) \dots (i_x, n_x)\}.$$

The above shows that one purchase has been made by a customer with CID code at $Date$. t indicates the total number of purchasing items which is obtained by calculating $n_1+n_2+ \dots+n_x$. In CPR matrix, row a contains these values with respect to the purchase of the current customer where these values must be refined as below:

$$CPR[a][i] = \left[\frac{(\alpha_1 + n_1)}{(100 + t)}, \frac{(\alpha_2 + n_2)}{(100 + t)}, \dots \right]$$

6.3 Customer Class Discovery

A formal definition of classification is not attempted; for our purposes it is sufficient to think of classification as describing the process by which a classificatory system is constructed. The word *classification* is also used to describe the result of such a process. Although indexing is often thought of as classification we specifically exclude this meaning. A further distinction to be made is between classification and *diagnosis*. Every language is very ambiguous on this point:

- How would you classify (identify) this?
- How are these best classified (grouped)?

The first example refers to diagnosis whereas the second talks about proper classification [40]. In this section we study how to diagnose the class of one object.

VSM (Vector Space Model) has been selected as the best approach to diagnose the class of one object [41, 42]. It is standard technique in information retrieval. The VSM allows decisions to be made about to which class each object belongs. It follows these steps:

- Users select attributes of the current object from among all the attributes and assign values to them.
- Equivalents of each attribute will be determined using WordNet ontology and one set is created for each attribute.
- Classes of catalog that have the same attributes as current object is determined.
- The cosine similarity between selected classes and current object is calculated.
- User selects the best class from among the output classes.

The next subsection describes the proposed recommendation algorithm.

6.4 Making Recommendation

One of the most remarkable characteristics of a successful algorithm is the range of products that is recommended by it. Hence, in the proposed algorithm, we divide Top-N-Recommendation between overall product classes. This algorithm recommends some of the products in each class of product classes based on the rating amounts in CPR matrix. Suppose that row a of CPR matrix contains these values,

$$CPR[a][i] = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_x]$$

The algorithm recommends $\alpha_1 * 100$ percent of Top-N-Recommendation from PC_1 class. This means that $\alpha_1 * N$ percent of recommendations in Top-N-Recommendation belong to products in class PC_1 . Similarly, $\alpha_2 * 100$ percent of Top-N-Recommendation belong to products in PC_2 class and so on. The customer sequence in the proposed recommendation system is shown in Fig. 11. In the next section, our proposed method is investigated by doing some evaluation tests.

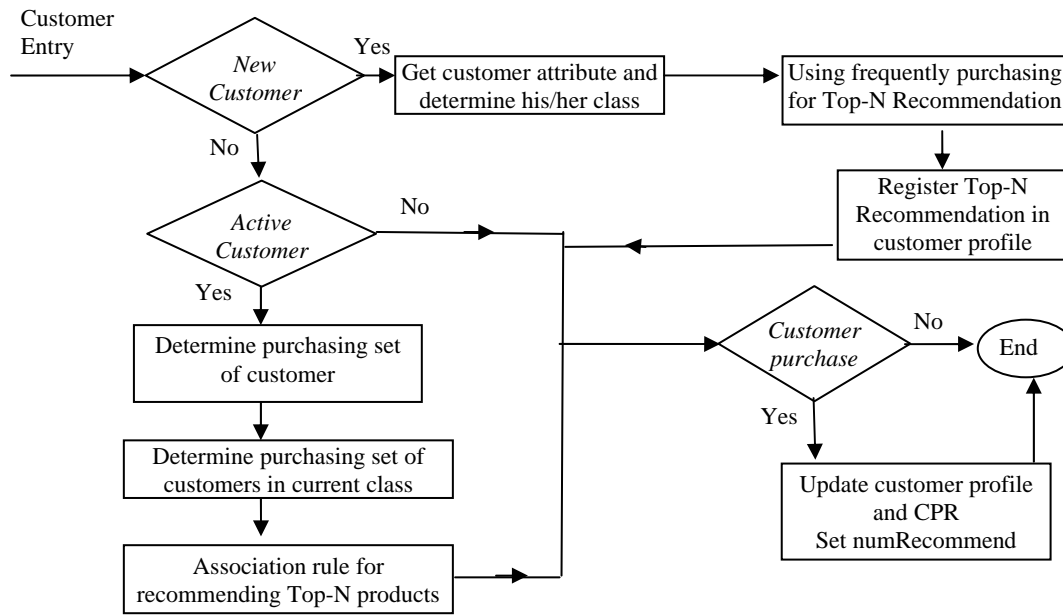


Fig. 11: Customer sequence in e-market

7.0 EVALUATION

7.1 Dataset

We used data from our MovieLens recommender system, MovieLens is a web-based research recommender system that debuted in fall 2003 [42]. Each week hundreds of users visit MovieLens to rate and receive recommendations for movies. The site now has over 6000 users who have expressed opinions on 4000 movies. We randomly select enough users to obtain 200,000 ratings from the database. We divided the database into 80% training set and 20% test set. We performed our procedure on the training set and compared its results with test set.

7.2 Evaluation Metric

To evaluate top-N recommendation, a widely used metric in information retrieval (IR) community, namely *precision* [44] has been utilized. But, we have slightly modified the definition of precision in our experiment in the sense that we have a fixed number of recommended items. In the first step, the algorithm works on the training portion of data where *top-N set* will be generated as the set of recommendations (for each customer). Our main goal is to look into the test set and match the newly recommended products with previously generated top-N set. Products that appear in both sets are members of a special set, which is called the *hit set*. We define precision metrics in our context as below:

- *Precision*. Which is defined as the ratio of hit set size to the top-N set size, i.e., $precision = \text{size of hit set} / \text{size of top-N set}$.
- *Recall*. We define recall as the ratio of hit set size to the test set size, i.e., $recall = \text{size of hit set} / \text{size of test set}$.

These two measures are, often conflicting in nature. For instance, increasing the number N tends to increase recall but decrease precision. The fact that both are critical for the quality judgment leads us to use a combination of them. In particular, we use the standard *F1 Metric* [43] that gives equal weight to both of them and is computed as

$$F1 = 2 * Recall * Precision / (Recall + Precision)$$

We compute F1 for each individual customer and calculate the average value as our metric.

Coverage measures the percentage of the universe of items that the recommendation system is capable of recommending. For the prediction task, it is calculated as the percentage of unrated items, a rating for which can be

predicted by the system. An alternative is to calculate coverage as the percentage of items of interest to a user rather than considering the complete universe of items.

7.3 Experimental Results

In this section we present and discuss the evaluation results for this recommender algorithm. The algorithm utilizes the data set described above as input and the produced recommendations were evaluated using the useful recommendations identified by two types of customers. First, we evaluate our algorithm in recommending to new customers. Later on, it is evaluated for customers that have made purchase earlier.

Considering the coverage metric, the result of evaluating our algorithm for two types of customers is shown in Fig. 12. We calculate coverage metric for the number of the classes of this dataset. Coverage metric changes in between 0.7 to 0.2 and that means our algorithm is able to recommend 20 to 70 percent of all the products from 21 product classes.

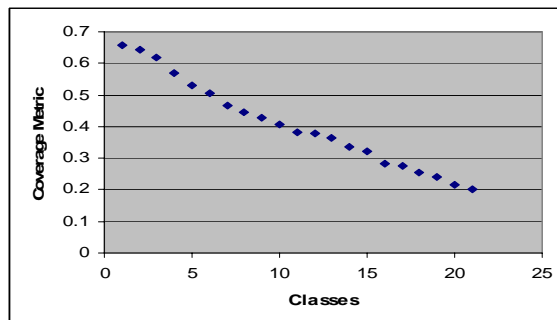


Fig. 12: Coverage metric for all customers

Considering F1 metric, the results of the custom implementation of our algorithm are presented in Fig. 13. These results have been acquired when the customer is new and had not made any purchase in the past. F1 metric is optimal when the number of recommendation is around 30.

This algorithm can recommend better product to customer when it knows the previous actions of the customer. It determines the purchasing set of the customer and customers in his/her class. The algorithm uses association rules for specifying recommended products. We apply Apriori algorithm for mining association rules and its result are recommended to customer. Fig. 14 shows that recommending products to earlier customers has better results, based on F1 metric. F1 metric increases as the number of recommendations increases.

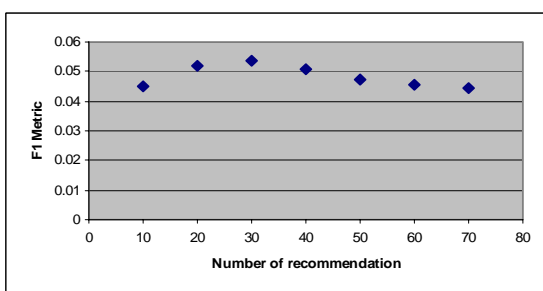


Fig. 13: F1 Metric for new customers

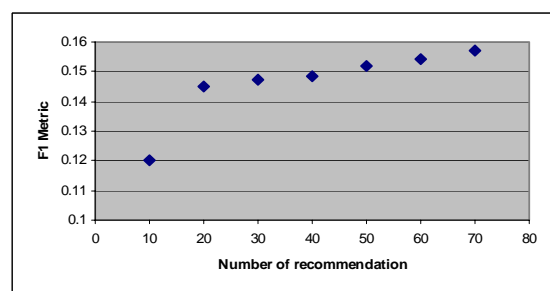


Fig. 14: F1 Metric for Earlier customers

8.0 CONCLUSIONS

The motivation for this research came from the limitations of the current electronic catalog management systems. Designing electronic catalog management system (ECMS) is complicated by a number of factors. Presenting product information, designing classification system, searching and navigation aspects, diagnosing class of one product and customization or recommendation procedure of these systems are the challenges for ECMSs.

We presented a brief history of electronic catalogs. Definitions, functions, and structures of electronic catalogs were described and some of the classification systems and their limitations were analyzed. The authors believe that

ontology is the best way for modeling and presenting electronic catalogs. OWL ontology language was selected among the current ontology languages. Electronic product catalogs were modeled and implemented using the primitives of this language. The proposed model for presenting product catalog maps the concepts to WordNet ontology. So, the classes of this model are more precise than earlier standard classification systems. It can define attributes for each of the concepts in electronic catalogs. In this model, it is possible to define class hierarchy and also to map the concepts of this electronic catalog to the equivalent concepts in the standard classification systems.

Searching and navigation mechanisms of the ECMSs are another challenge for these systems, that we solved using three types of navigation which are hierarchical, parametric and free keyword search. We also used VSM to diagnose class of one product.

Another ability of ECMS is recommender system. Recommender systems are personalized information filtering technology used to predict whether a particular user will like a particular item (prediction problem) or to identify a set of N items that will be of more interest to a certain user (top-N-recommendation problem). In this paper, we introduce a procedure for recommending products to customers. We model products and users information using OWL ontology language. Using this approach, sellers recommend semantic object to customers. Our algorithm does not have 'New Item' problem since we classify the products and the users based on their property. The experimental results showed superior performance in terms of coverage and precision.

Each product may be assigned to several class groups. So, using fuzzy logic to assign product instances to product groups is a very interesting approach to enrich this model. Customizing web page for each person may be able to improve the efficiency of this system.

ACKNOWLEDGMENT

The authors wish to thank Mr. Behrouz Shahgholi for his help in preparing this paper.

REFERENCES

- [1] Zona Research, How Shoppers Search Online: A Preliminary Typology, <http://www.zonaresearch.com/promotion/samples/zaps/zap06.htm>
- [2] K. Stanoevska et al., "Internet electronic product catalogs: an approach beyond simple keywords and multimedia", *The International Journal of Computer Networks*, 2000, pp. 701-715.
- [3] A. Segev et al., *Designing electronic catalogs for business value: Results the commerceNet Pilot*, CITM working Paper Citm-WP-1005, October 1995.
- [4] U. Tim et al., "Electronic sales assistance for product configuration", in *Proceeding of the 11th International Bled Electronic Commerce Conference*, Bled, Slovenia, June 1998, pp. 8-10.
- [5] A. M. Keller et al., "Using infomaster to create housewares virtual catalogs", *The International Journal of Electronic Commerce and Business Media*, 1997, pp. 41-44.
- [6] Z. Cui et al., "an Ontology-Based Approach to eCatalogue Management", *BT Technology Journal*, Springer Netherlands, Vol. 21, No. 4, October 2003, pp. 76 - 83.
- [7] UNSPSC, *Why Coding and Classifying Products is Critical to Success in Electronic Commerce*, White Paper, 1998.
- [8] E. Schulten et al., "The e-commerce product classification challenge", *IEEE Intelligent Systems*, USA, 2001, pp. 86-93.
- [9] J. Leukel, "Standardization of product ontologies in B2B relations-on the role of ISO 13548", in *Proceeding of the Americas Conference on Information Systems*, 2004.

- [10] D. Kim et al., “Catalog Management in E-Commerce Systems”, in *Proceeding of Comp. Sci. & Technology*, 2003.
- [11] D. Kim et al., “A Semantic Classification Model for E-Catalogs”, *IEEE International Conference on E-Commerce Technology (CEC'04)*, 2004, pp. 85-92.
- [12] L. Lee et al., “Building and operational product ontology system”, *The International Journal of Electronic Commerce Research and Applications*, 2005, pp. 1-13.
- [13] O. Cocho et al., “Solving Integration Problems of E-Commerce Standards and initiatives through Ontological mapping”, in *Proceeding IJCAL 2001 Workshop on E-Business & the Intelligent*, 2001.
- [14] D. Fensel et al., “Product Data Integration in B2B E-commerce”, *IEEE Intelligent Systems*, 2001, pp. 54-59.
- [15] D. Beneventano, “A framework for the classification and the reclassification of electronic catalogs”, in *proceeding of SAC 2004*, pp. 784-788.
- [16] H. Khosravi Farsani, M. A. Nematbakhsh, “Modeling Electronic Product Catalog using OWL Ontology Language”, in *Proceeding of IADIS Conference*, 2006.
- [17] J. B. Schafer et al., “Recommender systems in e-commerce“. in *Proceedings of the 1st ACM Conference on Electronic Commerce*, 1999, pages 158-166.
- [18] P. Resnick, H. R. Varian, “Recommender Systems”. *Special issue of Communications of the ACM*, Vol. 40 No.3, 1997.
- [19] Y. Benjamin et al., “Personalization of information access for electronic catalogs on the web”, *Journal of Electronic Commerce Research and Applications*, 2003.
- [20] M. Balabanovic et al., “Fab: Content-Based, Collaborative Recommendation”, *Communications of the ACM*, vol. 40, 1997, pp. 66-72.
- [21] B. Mobasher et al., “Automatic personalization based on Web usage mining”, *Communications of the ACM*, 2000, pp.142-151.
- [22] B.M. Sarwar et al., “Item-based Collaborative Filtering Recommendation Algorithms, in *Proceedings of WWW 10, 10th International World Wide Web Conference*, 2001, pp. 285-295.
- [23] M. Deshpande et al., “Item-based top-n recommendation algorithms”, *ACM Trans.*, 2004, pp.143-177.
- [24] B. Sarwar et al., “Item-based collaborative filtering recommendation algorithms”, in *WWW10*, 2001.
- [25] B. M. Sarwar et al., “Analysis of recommendation algorithms for e-commerce”, in *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, Minneapolis, MN, USA, October 2000, pp. 158-167.
- [26] Y. Yang et al., “Evaluation of online personalization systems: a survey of evaluation schema and a knowledge-based approach”, *Journal of electronic commerce*, 2005.
- [27] G. Karypis, “Experimental evaluation of item-based top-n recommendation algorithms”, in *Proceedings of the ACM conference on information and knowledge management*, ACM, New York, 2001.
- [28] B. Mobasher et al., “Intelligent Techniques for Web Personalization”, *IJCAI 2003 Workshop, ITWP 2003*, Acapulco, Mexico, August 11, 2003, Revised Selected Papers Springer, 2005.
- [29] H. Khosravi Farsani, M. A. Nematbakhsh, “A Semantic Recommendation Procedure for Electronic Product Catalog”, *Transactions On Engineering, Computing and Technology*, Vol. 16, pp. 350-355.

- [30] J. K. Kim et al., “A personalized recommendation procedure for internet shopping support”, *Journal of Electronic Commerce Research and Applications*, 2002, pp. 301-313.
- [31] Y. Zhang et al., “An Associative Classification-based Recommendation System for Personalization in B2C E-Commerce Applications”, *Expert Systems with Applications*, Vol. 33, No. 2, 2007, Forthcoming.
- [32] C. H. Lee et al., “Web personalization expert with combining collaborative filtering and association rule mining technique”, *Journal of Expert Systems with Applications*, Elsevier, 2001.
- [33] B. Mobasher, “Web Usage Mining and Personalization”, in *Practical Handbook of Internet Computing*, M. P. Singh(eds), CRC Press, 2004.
- [34] M. K. Smith et al. ,”*OWL Web Ontology Language Guide*”, <http://www.w3.org/TR/owl-guide/>
- [35] G. Antoniou et al., “Web Ontology Language: OWL”, in *S. Staab and R. Studer (Eds)*, Handbook on Ontologies in Information Systems, Springer 2003.
- [36] R. Fikes et al., “OWL-QL - a Language for Deductive Query Answering on the Semantic Web, *Technical Report KSL-03-14*, (Knowledge Systems Lab, Stanford University, CA, USA).
- [37] A. Seaborne. *Rdql - a query language for rdf*, w3c member submission 9 january 2004. <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>.
- [38] *SPARQL: Query Language for RDF*, <http://www.w3.org/TR/rdf-sparql-query/>
- [39] TopBraid Composer, <http://www.topbraidcomposer.com/download.html>
- [40] *Three Automatic Classification*, www.dcs.gla.ac.uk/Keith/Chapter.3/Ch.3.html
- [41] C. Aasheim et al., “Scanning World Wide Web Documents with the Vector Space Model”, *forthcoming Decision Support Systems*.
- [42] G. Salton, et al., “A Vector Space Model for Automatic Indexing”, *Communications of the ACM*, Vol. 18, No. 11, 1975, pp. 613–620.
- [43] MovieLens 2003, <http://www.groupLens.org/data>
- [44] G. Kowalski, *Information Retrieval Systems: Theory and implementation*, Kluwer Academic publisher, 1997.

BIOGRAPHY

Hadi Khosravi Farsani is a MSc student at Department of Computer Engineering, University of Esfahan. His research areas include Electronic Catalogs, Information Management, Knowledge Management and Ontology-based Information management.

Mohammad Ali Nematbakhsh received his B.Sc from Louisiana Tech University, M.Sc and Ph.D. from University of Arizona, in Electrical and Computer Engineering in 1981, 1983, and 1987 respectively. He worked for Toshiba Co. in USA and Japan for several years before joining the University Of Esfahan, Iran. He has published four USA registered patents, a book on Database systems and more than 50 papers. He was chairman of 7th CSI/IEEE conference in 2000, Graduate Coordinator of Computer Engineering Department between 2004 and 2006 and consultant to Central Bank of Iran for Totally Integrated Electronic banking System (CIBA) 1993-2004. He was selected the best Engineering graduate supervisor in Iran in 2004 and the best-awarded researcher in the University of Esfahan in 2003. His current research is in E-Commerce, Ontology based solutions, Multi-Agent Systems, Data Engineering and Protocols.