Chapter 4

Applications of ABPSWIR Architecture

In order to evaluate the effectiveness of the proposed methods, it is quite essential to apply in different domain and check whether they are suitable for these applications. The agent based personalized semantic search discussed in chapter 3 is extended to different area like education domain and e-catalog service systems. Section 4.1.1 explain how Agent based personalization is used in E-learning approach. Section 4.1.2 explains the agent based approach is used to improve personalization in e-catalog services system.

4.1 Application I : An Effective Personalized E-Learning System Using Agent Technology in Semantic Web

4.1.1 Introduction

As today the amount of accessible information is overwhelming, the intelligent and personalized filtering of available information is a main challenge. A personal information agent that is delivering the right
information at the right time by accessing, filtering and presenting information in a situation aware matter is needed. Applying Agent-technology is promising, because the inherent capabilities of agents like autonomy, pro- and reactivity offer an adequate approach. An agent-based personal information system is developed for collecting, filtering, and integrating information from specific domain.

Among different types of agents, the personal assistant agents are particularly interesting to this research. This type of agents operates at the user interface level and actively assists users by offering information and advice to the users. These agents usually apply a kind of intelligent learning algorithm so that they can intercept the user input, examine it and take action that are more specific to those particular users needs at that moment. These are also called learning or adaptive agents.

Personalization of web search is to carry out retrieval for each user incorporating his/her interests. The user profiles are then used to improve retrieval effectiveness in web search. A user profile are learned from the user’s search history and a domain hierarchy respectively which are combined to map a user query into a set of categories, which represent the user’s search intention and serve as a context to disambiguate the words in the user’s query. This chapter explores how agent based personalization can help individuals take advantage of their unique past information interactions when searching. An agent based personalized approach is developed and implement in education domain, to cope with currently existing challenges of Information Retrieval over the web.

4.1.2 Related work

The keyword based search engine fail to represents the complete semantics in the query. In the semantic web framework the semantics in the user query are identified and these are summarized according to the context. Then the results are classified into possible domains or groups and displayed to user
according to his choice. Domain specific knowledge needed for retrieval is stored using ontology. Ontology has become one of the main components in knowledge management [44], [103], e-learning [20], medical models [29], knowledge in diagnostic systems [37] and the semantic web. Ontology aware systems provide the possibility to perform semantic search. The user can search the destinations using several criteria related to subject.

The main focus of this research is to develop a knowledge-based by constructing education ontology for any Undergraduate students. Many papers deal with SWS that uses the OWL language for constructing ontologies. DySE System (Dynamic Semantic Engine) [77] implements a context-driven approach in which the keywords are processed in the context of the information in which they are retrieved, in order to solve semantic ambiguity and to give a more accurate retrieval based on user interests. This system splits the user query into subject keywords and the domain specific keywords. It uses a dynamic system that constructs ontology dynamically and uses that as a knowledge base. Ontology Construction in education Domain [61] deals with the construction of Ontology for specific University constructing instances specifically. Here the usage of Protg tool for constructing the ontology is illustrated. It states the various issues which play a key role in realizing the vision of semantic web such as XML (Extensible Markup Language) and XML Schema, RDF (Resource Description Framework and RDF Schema), URI (Uniform Resource Identifier), Unicode and SPARQL (Standard Protocol for RDF Query Language), Search Engines and Agents, and Ontology etc. Ontology development is the objective of the above system and it has provided the guidelines to work in it with education domain as example. Query sentences as semantic networks [62] paper describes procedure for representing the queries in natural language as semantic networks.

Here a syntactic analysis of the query is done by parsing the query using Stanford parser to tag each and every word with their corresponding parts of speech. Candidate set generation is an important method used here. A plain
text based and word net based comparisons are done to match the related concepts in the ontology. Semantic Information Retrieval System [101] is mainly concerned with retrieving information from a sports ontology using the SPARQL query language. Here specific information is retrieved from the ontology. The sports related information is queried from the ontology and it is done using SPARQL language. It provides a basement for any further research to achieve intelligent fuzzy retrieval of sport information through fuzzy ontology. The pages retrieved from web search needs to be ranked for getting more relevant links.

A Relation-Based Page Rank Algorithm for Semantic Web Search Engines [46] proves that relations among concepts embedded into semantic annotations can be effectively exploited to define a ranking strategy for Semantic Web search engines. This sort of ranking behaves at an inner level (that is, it exploits more precise information that can be made available within a WebPage) and can be used in conjunction with other established ranking strategies to further improve the accuracy of query results. With respect to other ranking strategies for the Semantic Web, this approach only relies on the knowledge of the user query, the WebPages to be ranked, and the underlying ontology. Thus, it allows one to effectively manage the search space and to reduce the complexity associated with the ranking task.

In [3] ontology model is built for retrieving Tamilnadu tourism Information. In [48], user profiles are created using ontology model based on the knowledge collected from world knowledge database and local repositories. However, up to this point, ontologies modeling user profiles are application specific, with each one having been created specifically for a particular domain. In [73] the author, develop and implement a semantic search engine confined to the university domain. In this work, ontology model is constructed for education system based on the knowledge from local repositories. The semantics for query parsing is framed using world knowledge base. User profile is personalized using vertical data format mining techniques. This personalization improves the accurate identification of user
context for search.

The overview of the existing systems gives multitude approaches for semantic information extraction. Though these above systems perform a semantic analysis, it has been implemented in a more generic way. Hence in order to further enrich this process to retrieve more promising results a system has been proposed for queries relating to meticulous domain. In this proposed system, in combination with some of the above said methodologies, some more procedures have also been added to perform semantic information extraction in a better way.

In order to overcome these critical issues, the proposed system Agent Based Personalized E-learning System (ABPES) was designed. The system ABPES retrieves the semantically relevant results for the user query by considering the semantics and context of the query. The Semantics of the query is analysed by means of the following procedures:

- The user query is initially analysed grammatically and syntactically by parsing.
- The domain related keywords in the ontology are retrieved to form the refined query.

The results obtained are more relevant by adopting the following procedure

- The refined queries that serve as the input for the search engine is formed based on the semantic analysis of the user query.
- The web links retrieved for all the newly formed refined queries are re-ranked based on the domain specific information.

In this way the proposed system provides a semantic search that retrieves the appropriate results for the user query. The Agent Based Personalized E-learning System (ABPES) Figure 4.1 is discussed here based on their important components. The proposed system consists of different components
like User Agent, Expansion Agent, Semantic Agent, Personalized Agent and Student profile & Knowledge Base (ontology). All agents are monitored entirely to fulfill proprietary system functions, including Information Retrieval and Knowledge Base update.

![Figure 4.1: Architecture of Agent Based Personalized E-Learning System](image)

### 4.1.3 User Agent

User Agent is act as interface between user and system, and provides a friendly platform to the student. It can build user interest model according to student browsing narration record and registration data. Users browsing or evaluating behaviour can be stored and learned by User Agent, so user interest may be updated and improved in any time.

In order to update the profile, user agent need feedback from current search result, the browsing history and the expanded query. The user feedback is the
degree of satisfaction on search period, keyword matching accuracy and overall performance of the system. The system will compare these satisfaction degrees with historical records, which consists of both the satisfaction degree of last feedback and last strategy. The Comparing results are defined by the following fuzzy relationship:

\[ s(x) = \begin{cases} 
  \text{Satisfied}, & \text{if } x > \text{new}x \\
  \text{Good}, & \text{if } x = \text{new}x \\
  \text{Unsatisfied}, & \text{if } x < \text{new}x 
\end{cases} \]  

(4.1)

In this function, \( x \) represent user feedback (both implicit and explicit) of updated content of URL. After getting the comparing result, the system use the inference method to decide the modifying strategy of the current profile. If the overall performance is Satisfied or Good then

\{ 
  Modify search period, search source url respectively 
\}

Else

\{ 
  Restore the existing one 
\}

4.1.4 Expansion Agent

Ontology provides a common understanding of a term and also its relationship with other terms. Thus a hierarchy can be formed with the related terms. Thus considering our domain, different author represent each term in a subject will express its purposes and functionalities in different terms. The user query should be parsed so that the stop words that are not needed can be removed from the query by parsing the query the nouns, verbs and other parts of speech can be used separately if needed. The nouns can be used to get their related terms and properties from the constructed ontology. Semantic Agent aims to find the semantic features in the users queries. It will make use of agent technologies, ontology technologies to analyse the association relation in the users queries and document to extract semantic features.
<table>
<thead>
<tr>
<th>Query</th>
<th>Expansion</th>
<th>Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input:</td>
<td>Expression E in the form of query</td>
<td></td>
</tr>
<tr>
<td>Output:</td>
<td>semantic expanded Query q</td>
<td></td>
</tr>
<tr>
<td>Parameter used</td>
<td>$t_1, t_2, t_3...$ - terms in input query</td>
<td></td>
</tr>
</tbody>
</table>

```java
QuerySemantics( )
{
    if (E=true) then
    {
        For each term in Expression
        {
            QUERYEXPAND(E)
        }
    }
}
```

User can automatically add or update the relations and concepts to domain ontology Ontology model. Part of the ontology for teaching and learning strategies is defined in Figure 4.2.

![Figure 4.2: Part of the ontology for teaching and learning strategies](image-url)
4.1.4.1 Algorithm for creating dynamic ontology

Input:
key - term to add
$D_0$ - domain ontology
$P_0$ - possible ontology
$R_0$ - related ontology

Output: creating ontology model

Parameter used
$k, p$ - related and possible terms

userontology(key) {
    if (key isRelatedTo k in $D_0$ by p) then {
        db() ← add key p, k to db
        // where K is associated with Domain
    }
    if (key isRelatedTo k in $P_0$ by p) then {
        db() ← Add key P k
    }
}

4.1.5 Semantic Agent

Searching Agent can employ diversified self-governed searching engine according to user demand, and converts user request to the syntax format that adapts to called searching engine. Searching Agent collects all data from initiative Searching Agent takes out invalid links, deleting excrecent information, and finally processed data are transmitted Personalized Agent.
4.1.5.1 Personalized Semantic Information Retrieval (PSIR) Algorithm

input:
uid, semantic query

output:
list of URLs related to user query

Parameter used:
uid - user id
uinterest - user interest used
equery - semantic query related to education domain

PSIR (uid)
{
If uid exits then {
Re-ranking(CP, uid, equery, interest)
}
else
{
For each user entered
{
userProfiledb ← uid, equery, uinterest
For each search
{
Usersearchdb() ← uid, squery, interest
Apply Assarlg(uid, squery, interest)
UP() ← squery, interest
}
}
}}

In this algorithm the proposed system first generates ID for each user that visits Website and stores his activities in this site. Second the proposeer needs anything from Website the system takes the request and rates the WebPages that related to his interest and also WebPages that are recommend from web using the Response Equation.

\[ newResponse = c \times systemScore + (1 - c) \times RequestScore \]  \hspace{1cm} (4.2)

where \( c \) has a value between 0 and 1. When \( c \) has a value of 0, conceptual rank is not given any weight, and it is equivalent to pure request based ranking. If
c has a value of 1, request based ranking is ignored and pure conceptual rank is considered. Both the conceptual and request based rankings can be blended by varying the values of c.

4.1.6 Personalized Agent (PA)

Personalized Agent use Re-ranking algorithm to find the new score based on user interest.

4.1.7 Knowledge Base

Knowledge Base is used for storing every user interest model, user record, and rules or parameters that serve for ensuring system well-balanced circulation.

4.1.7.1 Dynamic User Profile

In this method, context is implicitly defined through the ontology based user profiles, which are updated over time to reflect changes in user interests/needs by using the users past search history. User Profile is defined as $(U_i, (query_j, context_j, ranking_j))$, where $1 \leq i \leq N$, N denotes number of users and $1 \leq j \leq M$, M denotes number of query topics for the user. Query topic is defined as set of query terms. Context of $j^{th}$ Query topic extracted using algorithm. Weight $j$ is weight given to the $j^{th}$ query topic for the user. When the user poses the new query topic, the context retrieved from ontology is added in query topic in order to expand the query. Subsequently, system send expanded query topic to search engine for retrieving documents. The system also monitors the users browsing history and updates the users profile whenever the users relevant retrieved page changes, in terms of recent context of query terms. Whilst users older interest / context gets deviated over a period of time and it is also updated in user profile.

4.1.8 Implementation and experimentation

Our experimental evaluation is designed to address the following particular questions:
• Can the query expansion with ontology improve accuracy of retrieval, satisfying the users need?

• Does the user profile created dynamically by observing the users behaviour helpful in improving personalization?

• Can re-ranking of the documents retrieved from search engine on the basis of users behaviour help to give personalized result?

In this section experiments carried out to evaluate the performance of proposed system will be discussed from a quantitative point of view by running different experiments to evaluate the precision of the results. The proposed system is implemented in C#.Net as Web-based system. The number of stored documents is more than 5 lakhs documents. These Web documents are about various education domain such as Computer Science, Physics, Mathematics domain.

4.1.8.1 Performance Analysis

The data set consists of real world data provided by actual users. Experiments were carried out in two steps. In the first step the system was initially launched and invitations were sent out to people asking them to sign up. People were asked to login to the system after signing up and conduct certain specific search queries on the system. Users were given specific topics like deadlock, process management, exception, java input and asked to make searches specific on these topics. Users were also asked to give feedback those links which they thought were relevant to the query. This was simulated by initial click on the link and then by providing much more information about the link by specifying certain tags and a small description for each of the link clicked. This way the initial part of the experiments included collecting user queries and building up the query log.

After the end of the first part of the experiments the system had sufficient users, queries, URLs and query log for the algorithms to be run. In
the second part of the experiments the algorithms were included in the system and users were asked to search for something which they had already searched or something to similar to their previous query. The results were displayed in two columns where in the left column had the original Google results for the issued query and the right column had the results obtained from proposed agent based systems. The results also display the score that was obtained for each of the URL. Currently only the first 10 results from Google and a combination of Google and reranked results are displayed.

Users were also provided a check box where in they could give out their opinion regarding the results obtained. For each query issued and the results obtained users could compare the results obtained by Google and proposed system give out their satisfaction by checking on the checkbox based on whether they are satisfied or not. So for each of the queries a survey was conducted as to whether the user was satisfied with the proposed search results compared to the Google results.

When the proposed system ABPES was tested, a marked improvement in performance was observed for most of the queries as a result of semantic analysis, although a small fraction of them had negative and similar performance with a generic search engine. Table 4.1 shows the samples of the performance levels of new system comparing it with Google.

From the table 4.1 it is depicted that the precision value of proposed system ABPES is higher than the values obtained in Google search engines. The relative recall values estimates the retrieval effectiveness between Google and new system. The more relative recall values of proposed ABPES system shows that ABPES is more effective in retrieval than Google search engine.

4.1.8.2 Precision and Recall

Figure 4.3 shows the precision Vs recall graph for proposed system and GOOGLE system. The graph drawn taking the first 10 queries into
<table>
<thead>
<tr>
<th>SAMPLE QUERIES</th>
<th>GOOGLE</th>
<th>ABPRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyword</td>
<td>precision</td>
<td>recall</td>
</tr>
<tr>
<td>polymorphism</td>
<td>0.67</td>
<td>0.55</td>
</tr>
<tr>
<td>memory compaction</td>
<td>0.52</td>
<td>0.45</td>
</tr>
<tr>
<td>encapsulation</td>
<td>0.78</td>
<td>0.67</td>
</tr>
<tr>
<td>disk space management</td>
<td>0.45</td>
<td>0.4</td>
</tr>
<tr>
<td>abstract classes</td>
<td>0.89</td>
<td>0.78</td>
</tr>
<tr>
<td>looping statement</td>
<td>0.59</td>
<td>0.53</td>
</tr>
<tr>
<td>deadlock</td>
<td>0.69</td>
<td>0.61</td>
</tr>
<tr>
<td>abstract class</td>
<td>0.79</td>
<td>0.65</td>
</tr>
<tr>
<td>interface in java</td>
<td>0.63</td>
<td>0.54</td>
</tr>
<tr>
<td>input output</td>
<td>0.51</td>
<td>0.48</td>
</tr>
<tr>
<td>interface</td>
<td>0.82</td>
<td>0.69</td>
</tr>
<tr>
<td>physical model</td>
<td>0.6</td>
<td>0.51</td>
</tr>
<tr>
<td>Erdiagram</td>
<td>0.45</td>
<td>0.4</td>
</tr>
<tr>
<td>overloading</td>
<td>0.56</td>
<td>0.49</td>
</tr>
<tr>
<td>looping statement</td>
<td>0.74</td>
<td>0.64</td>
</tr>
<tr>
<td>Inter process communication</td>
<td>0.58</td>
<td>0.41</td>
</tr>
<tr>
<td>process control block</td>
<td>0.72</td>
<td>0.68</td>
</tr>
<tr>
<td>semaphore variable</td>
<td>0.59</td>
<td>0.43</td>
</tr>
<tr>
<td>functions in c</td>
<td>0.71</td>
<td>0.53</td>
</tr>
<tr>
<td>operator overloading</td>
<td>0.6</td>
<td>0.4</td>
</tr>
<tr>
<td>iostream</td>
<td>0.64</td>
<td>0.55</td>
</tr>
</tbody>
</table>

Table 4.1: Performance levels
consideration, their corresponding precision and recall values are plotted for both Google and ABPES systems. The precision recall curve in the graph clearly depicts that the proposed system retrieves the accurate links for the user query based on semantic relatedness.

![Figure 4.3: Precision Vs Recall graph for proposed system and GOOGLE system](image)

**4.1.8.3 Personalization time:**

Time to retrieve any information depends on the type of search engine, size of data set, relevancy between query and documents, user history and re-ranking algorithm used. The personalization performance can be expressed:

\[
\text{Personalization performance} = \sum_{i=1}^{n} \text{Grank} + \text{UserRank} \quad (4.3)
\]

For each page find

\[
\text{UserRank} = \sum_{i=1}^{n} UR + VT + FC \quad (4.4)
\]

Where \( UR \) user rating \( VT \) page view time and \( FC \)-frequency count and \( n \) represents threshold value.
This Figure 4.4 discussed the performance efficiency of the system when the system uses to retrieve the result. It is observed that 80% users, out of 20 users in our data set, have found improved precision with the proposed approach in comparison to the standard search engine (Google) results. While 20% users have achieved equal precision with both approaches. It has been observed that users who posed Queries in unpopular context than well liked context got better performance. In addition, when the system can extract the exact context of users need, the Precision and recall is found better than other search engine results.

4.1.9 Discussion

Compare the result of retrieval documents between agents based system and existing one based on multiple query with single user is tested in Experiment-1. This experiment shows the average precision that is based on retrieving results. The precision is improved 80% over keyword based systems. Experiment 2- Measuring performance efficiency of the both systems over the different user. The experiment shows that agent based personalized search is better. The time to retrieve semantic results by very low than keyword based one. In sum, the overall precision is improved 40%-50%.
4.2 Application II: Agent Based Personalized E-Catalog Service System

4.2.1 Introduction

With the emergence of the e-Catalog, there has been an increasingly wide application of commodities query in distributed environment in the field of e-commerce. But e-Catalog is often autonomous and heterogeneous, effectively integrating and querying them is a delicate and time-consuming task. Electronic catalog contains rich semantics associated with products, and serves as a challenging domain for ontology application. Ontology is concerned with the nature and relations of being. It can play a crucial role in e-commerce as a formalization of e-Catalog. User personalized catalog ontology aims at capturing the users' interests in a working domain, which forms the basis of providing personalized e-Catalog services. This paper describes a prototype of an ontology-based Information Retrieval agent. User personalized catalog ontology aims at capturing the users' interests in a working domain, which forms the basis of providing personalized e-Catalog services. This paper, present an ontological model of e-Catalogs, and design an Agent Based Personalized E-Catalog Service System (ABPECSS), which achieves match user personalized catalog ontology and domain e-Catalog ontology based on ontology integrated.

As Internet technologies develop rapidly, companies are shifting their business activities to e-Business on the Internet. Worldwide competition among corporations accelerates the reorganization of corporate sections and partner groups, resulting in a break of the conventional steady business relationships. For instance, a marketplace would lower the barriers of industries and business categories, and then connect their enterprise systems. Electronic catalogs contain the data of parts and products...
information used in the heavy electric machinery industry. They contain not only the commercial specifications for parts (manufacturer name, price, etc.), but also the technical specifications (physical size, performance, quality, etc.). Clearly defined product information is a necessary foundation for collaborative business processes. Furthermore, semantically enriched product information may enhance the quality and effectiveness of business transactions. As a multifunctional applied system, it serves for advertisement, marketing, selling and client support, and at the same time it is a retail channel.

As the number of Internet users and the number of accessible WebPages grow, it is becoming more and more difficult for users to find documents among e-Catalogs that are relevant to their particular needs. Users can search with a search engine which allows users to enter keywords to retrieve e-Catalogs that contain these keywords. The navigation policy and search have their own problems. Indeed, approximately one half of all retrieved documents have been reported to be irrelevant. The main reasons for obtaining poor search results are that (i) many words have multiple meanings (ii) key words are not enough to express the rich concepts and the natural semantics of customers’ queries. (iii) The property query lacks of semantic support, and is difficult to search for knowledge, and has other problems of mechanisms. (iv) Related merchandises cannot be returned. What is needed is a solution that will personalize the e-Catalog selection and be presented to each user. A semantically rich user model and an efficient way of processing semantics are the keys to provide personalized e-Catalog services. In view of the existing limitations, we develop a personalized ontology based on user model, called user personalized catalog ontology, which has the same level of semantics as domain ontology.

The rest of this section is structured as follows: Section 4.2.1 give an introduction about e-catalog system. Section 4.2.2, explains the theory of propose system. Section 4.2.3 explain new modelling methodology for generating user personalized catalog and product domain ontology. Then in
Section 4.2.4 explain the implementation of the system and its evaluation. Conclusion and future work are drawn in Section 4.2.5

4.2.2 Related work

E-catalogues play a critical role in e-procurement marketplaces. They can be used in both the tendering (pre-award) and the purchasing (post-award) processes[98]. Companies use e-catalogues to exchange product information with business partners. Suppliers use e-catalogues to describe goods or services that they offer for sale. Meanwhile buyers may use e-catalogues to specify the items that they want to buy. Matching a product request from a buyer with products e-catalogs that have been provided by the suppliers, helps companies to reduce the efforts needed to find partners in e-marketplaces.

Researches in recent years show that applying ontology to e-commerce scenarios would bring benefits such as solving the interoperability problems between different e-commerce systems [19][33]. Especially, e-Catalog, which is a key component of e-commerce systems, seems to be the most adequate domain within e-commerce scenarios where ontology can realize the expression of e-Catalog on a semantic level. It is possible for e-business systems to offer diverse interoperable services by sharing well-defined e-Catalog model containing rich semantics,[66] described in principle how ontologys can support the integration of heterogeneous and distributed information in ecommerce scenarios which is mainly based on product catalogs, and what tasks are needed to be mastered. E-Catalog ontology model is defined as ECO (concepts, relationship, properties, axioms and individuals). The traditional key-based retrieval method cannot satisfy massive heterogeneous personalized catalog service, then [8] introduce meta search engines, but this method is passive service.
Within unstructured data a keyword search engine can do a very valuable job like in Google.com or Baidu.com. But this technique does not utilize the semantics available in structured data. Moreover, it has lots of problems with the syntax, despite the semantics of typed e-Catalogs is clear. E.g., in the ranked keyword search XRANK[?] would cause a lot of problems. Also exploiting structured and typed data, parametric search aims to find the right alternatives in case there is no perfect match [15]. Iteratively, the user can soften or skip some search conditions. The most problematic deficit in this technology is that there is no deterministic way, indeed no confirmed way at all to find the best alternative. The user never knows when it is best to terminate the search and with which result. That is, traditional key-based retrieval method can not satisfy massive heterogeneous personalized catalog service, then [26]introduce metasearch engines, but this method is passive service.

[48] Provided an intelligent catalog recommend method using customer requirements mapping with product categories.[49] Brought forward personalized e-Catalog model based on customer interests and [50] is a personalized catalog service community, WebCatalog [51] designed enterprise e-Catalog based on customer behavior. The knowledge representation and acquisition of client catalog turns into the key problems. In order to reach an effective method, K-clustering algorithm and e-Catalog segmentation approach are described in [52] and [53] described the customer segmentation method based on brand and product, price. In [54] the author researched personalized catalog service with one-to-one market by association rules and CART. In recent years, personalized ontologys (also known as private ontology, such as [48] are introduced into e-Catalog service, Peter Haase put forward personalized ontology learning theory based on user access and interest coordination[55]. In distributed system, there are sharing concepts of domain ontologys and personalized knowledge ontologys [62]. Therefore, it has important theoretical and practical significance to apply personalized ontologys to personalized e-Catalog service. The Architecture of Agent based personalized E-Catalog service system is illustrated in Figure 4.5.
4.2.3 Basic function of ABPECSS

To implement agent based catalog service first of all, personalized user catalog ontologys are customized according to consumers(PCO). Secondly, we need to build Semantic e-Catalog ontologys(SCO). Thirdly, we match the two kinds of ontologys by match algorithm through semantic reasoning and expanding agent which generates match result sets. The basic the theory of
distributed semantic query based on e-Catalog ontology is users input key words, phrases, sentences or paragraphs (users’ queries, Uq) in user querying interface. Query generator agent translates Uq to ontology descriptor. Query reasoning and Expanding Agent is responsible for reasoning and expanding the descriptor using the semantic match result set is, then outputs semantic queries (Sq) in forms of Sparql and finally extract data from distributed e-Catalog database. Searching and Filtering Agent combines the distributed results and filters repetitive and invalid results. Personalized ranking agent rearrange the result sets and recommended to the user.

4.2.4 Methodology

4.2.4.1 Method of Designing User Catalog ontology

In order to satisfy customer’s personalized requirement, we should master more information of the customers. Sometimes customers also cannot describe their own thought, to understand their potential mind, we need user e-Catalog ontology based on consumer behaviour, we propose a personalized approach to build personalized catalog ontology (PCO). PCO supposed to be formed by

- First, build user personal ontology (PCO) based on users’ personal information and preferences
- Second, extract user catalog information from user purchase history, user searching keywords, user browsing catalog, user feedback information
- Third, web resource according to user catalog ontology information

Agent based e-catalog organizes a group of keywords expressing users’ interest through PCO, when users puts semantic query, it is no longer a simple keywords match, but considering users’ personal preference and information, and tightly integrates the users and products, so that the system can improve the semantic query precision rate and recall rate, as well as be conducive to sort query results.
Figure 4.6: Framework of User Personalized Catalog Ontology

Figure 4.6 shows a user catalog ontology framework, in which the proposed system describe user interest information, user preference and product concepts, properties and individuals that users are interested in, including product area, brand and quality authentication. Users associated with the product by property hasPreference, and we set aside a weight interface in property "has Preference", indicating the fact users’ different observation extent about different of a product.

4.2.5 Generating Semantic Catalog ontology (SCO)

Generation domain e-Catalog ontology is divided into three steps:

- Extraction of the core concepts and properties for domain e-Catalog ontologys, according to the UNSPSC standards, wordNet standards and semantic catalog dictionary.
• Construction of a SCO model.

• Acquisition standardized DECO by e-Catalog ontology pruning subsystem, combining WorldNet and semantic catalog dictionary.

4.2.6 Semantic Match Based on Ontology

One critical step of semantic match is that calculation semantic match degree between the terms of ontology concepts. There have been many methods to calculate conceptual semantic match in e-commerce scenarios. Common calculation methods and models are: (i) Identifier-based method [30], which uses word-building to find the semantic match degree between the concepts, and primarily reflects the linguistic similarity of the two concepts; (ii) Synonym dictionary-based method [33], which organizes all concepts to a tree hierarchy structure according to synonym dictionary where there is only one path between any two nodes and this path length is taken as a measure of semantic distance of the two concepts; (iii) Feature Match-based model[33], which calculates semantic match of concepts by the collection of properties; and ((iv) Semantic relationship-based model[40], also known as the semantic distance-based model, which calculates semantic match of concepts based on hierarchy information and is mainly used in the same ontology.

4.2.6.1 Property-based Semantic Match

Property-based semantic match method respectively calculates the semantic match degree of data type and objective properties (smd(P1, P2) and smo(P1, P2)) and then sets weight for the semantic match degree of the two kinds of properties, and at last, integrates them to gain the semantic match based on property.

To query user preferences product, we should get the product similar with user preferences, namely calculating the instance similarity between SCO individual and PCO individual. We calculate the semantic match of the
individuals by the property value-based method by applying k-medoids algorithm. Calculate the semantic match method based on linguistics, when we calculate semantic match degree of the property values.

\[
\text{sim}(C_1, C_2) = \frac{\text{ed}(C_1, C_2)}{\|C_1\| + \|C_2\|}
\]  

(4.5)

\(\|C_1\|\) is the length of the string \(C_1\), \(\|C_2\|\) the length of the string \(C_2\), \(\text{ed}(A,C_2)\) is the same number of characters in \(C_1\) and \(C_2\). String \(C_1\) and \(C_2\) are input parameters, in the process, which are the properties values of two products calculate the individual semantic match of the two products through comparing several groups property semantic match degree.

### 4.2.6.2 Results Personalization

The personalization helps in getting relevant results for the users query. As shown in the query-processing steps, the personalization starts with the query enrichment step, where utilize the user profile to expand the query and to fill in the incomplete query templates. Here, the system go into more detail with the results of personalization steps and show how to capture the users feedback.

#### 4.2.6.3 Results of personalization steps

Personalizing the results involves presenting the results in the most effective way possible through several steps. The first step is answering the users query in the same language he asks it in, regardless of the language of the ontology and the knowledge base, which has the annotated data. The second step is answering the users query in appropriate syntax based on the question type; a confirmation question is different than a subjective question, as the user expects a yes or no answer in the first type, while s/he expects a list of items in the second type. So, an answer is personalized to express the understanding of the query and to be familiar to the user. The third step is ranking the results based on the users preferences and interests. Finally, it
filters the non-relevant food or health information based on the user profile.

### 4.2.6.4 Users feedback

Continuous feedback collection is required to sharpen the users experiences. Feedback is not only explicit, but also implicit, as it can be collected through different measures. Many measures could help in reflecting the implicit feedback, such as time spent in browsing the results, clicks on the data sources, clicks on the result facets related to the search results, etc. All interactions and feedback are recorded and logged in the usage log which is analyzed after each query to know how effective the results are and how it can improve the future recommendations. This is reflected in the user profile ontology.

### 4.2.7 Implementation and experimentation

The system was evaluated by having 20 users implement the system to create personal ontologies. The user was given a query interface to input his or her query parameters and view each one of their concepts and every concept from the SCO (Semantic Catalog ontology) that had been matched to the personalize catalog concept. Also, the user was able to decide which concept or property was not needed when reasoned and expanded the query. In the experiment, the new system take different electronic items as an example. The user was asked to compare the semantic query result and that from the keyword-based search engines and decide if ABPECSS was the better. Therefore, proposed system manually create the domain Semantic Catalog ontology (SCO) and user personalized catalog ontology (PCO) and calculate semantic match degree in the system. Table 4.2 shows the sample of the performance levels of new systems.

The system is evaluated the system with two measures, precision and
<table>
<thead>
<tr>
<th>concepts</th>
<th>Total concepts</th>
<th>Found correct</th>
<th>Correct manually</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dell Inspiron 15R i3531-1200BK</td>
<td>89</td>
<td>71</td>
<td>74</td>
<td>91.36%</td>
<td>80.43%</td>
</tr>
<tr>
<td>Dell Alienware 18 Gaming Laptop</td>
<td>89</td>
<td>71</td>
<td>93</td>
<td>71%</td>
<td>5.54%</td>
</tr>
<tr>
<td>Canon EOS 6D Black SLR Digital</td>
<td>50</td>
<td>16</td>
<td>56</td>
<td>78.00%</td>
<td>84.21%</td>
</tr>
<tr>
<td>Nikon D810 DSLR Camera (Body Only)</td>
<td>90</td>
<td>53</td>
<td>78</td>
<td>90.00%</td>
<td>81.54%</td>
</tr>
<tr>
<td>Bargains Depot USB Cable Lead Cord</td>
<td>45</td>
<td>19</td>
<td>39</td>
<td>93.00%</td>
<td>8.95%</td>
</tr>
<tr>
<td>Nikon 1 AW1 14.2MP Waterproof</td>
<td>90</td>
<td>53</td>
<td>78</td>
<td>90.00%</td>
<td>81.54%</td>
</tr>
</tbody>
</table>

Table 4.2: Different level of query relevance, shown in Figure 4.7 Precision measures the number of relevant pages that were seen against the total number of pages that were seen. Relevance measures the number of relevant pages seen plus the number irrelevant pages not seen vs. The total number queried.

Figure 4.7: Precision Vs Recall graph for proposed system Vs GOOGLE

The next experiment aims at determining the importance of personalization by using generated dynamic user model during using the system. The user model is used to re-rank the retrieved documents to match the user interest.
4.2.7.1 Personalization time

Time to retrieve any information depends on the type of search engine, size of data set, relevancy between query and doc. User history and re-ranking algorithm used. Figure 4.8 discusses the performance efficiency of the system when the system uses to retrieve the result. It is observed that 80% users, out of 30 users in our data set, have found improved precision with the proposed approach in comparison to the standard search engine (Google) results, while 34% users have achieved equal precision with both approaches. It has been observed that users who posed Queries in unpopular context than well liked context got better performance. In addition, when the system can extract the exact context of users need, the Precision and recall is found better than other search engine results.

4.2.8 Discussion

Experiment 1- compare the result of retrieval documents between agents based system and existing one based on multiple query with single user. This experiments shows the average precision that is based on retrieving results. The precision is improved 60% over keyword based
systems. Experiment 2- Measuring performance efficiency of the both systems over the different user. The experiment shows that agent based personalized search is better. The time to retrieve semantic results is very low than keyword based one. In sum, the overall precision is improved 40%-50%.