WONJUNG KIM
KNOWLEDGE BASED PERSONALIZATION
(Under the Direction of DR. AMIT SHETH)

Semantic Web aims at bringing order to the information chaos on the web by associating semantics and context with content, thereby improving its value. It focuses on making content usable by computers for answering questions and performing tasks on behalf of users. Knowledge sharing using ontologies is critical for realizing this vision. Users would create and manage ontologies and share them on a network, thus creating a personalized view of the knowledge and still interoperating with other ontologies in the network. Searching for ontologies that best fit a user's perception is non-trivial. To help him do this, his personalized knowledge base, profile and query history can be used to deploy personalization techniques. We discuss this knowledge based personalization approach as used in the InfoQuilt system. We describe the use of multiple personalization techniques used in combination and particularly, query relationship to predict related ontologies based on current context. Query relationships use InfoQuilt's framework for inter-ontological relationships to do this by describing relationships between queries on two domains.

INDEX WORDS: InfoQuilt, Ontology, Knowledge Base, Personalization

KNOWLEDGE BASED PERSONALIZATION

by

WONJUNG KIM

B.S. Sookmyung Women's University, Korea, 1998

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2001

KNOWLEDGE BASED PERSONALIZATION

by

WONJUNG KIM

Approved:

Major Professor: Amit Sheth

Committee:I. Budak Arpinar
Hamid Arabnia

Electronic Version Approved:

Gordhan L. Patel
Dean of the Graduate School
The University of Georgia
December 2001

ACKNOWLEDGEMENTS

I wish to thank my advisor Dr. Amit Sheth for his guidance and encouragement. I also take this opportunity to thank Dr. Ismailcem Budak Arpinar and Dr. Hamid Arabnia for consenting to serve on my committee. I thank Shuchi Patel and Kemafor Anyanwu for their help. I also thank Madhan Arumugam, my colleague. I also thank my parents for their support and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The universality and easy accessibility of the World Wide Web has made it a very popular medium for sharing information globally. However, the information shared is primarily meant for humans. The Hypertext Markup Language (HTML) used to create web pages only focuses on the presentation and formatting of the page. Semantic Web is proposed as the next generation web where information is given well-defined meaning. This will allow computers to interpret the information and its context, and reason about it, thereby improving the value of the information [1, 2, 3]. This will also eventually make it possible for users to ask questions such as "*Is this a good site for a landfill?*". Further, the answer to this question would be computed using information available from a number of independently developed heterogeneous sources. The user will also not have to worry about the differing formats, structures, and media of data.

A number of challenges exist in achieving this vision. The first challenge is the need to support semantics, and context. Use of ontologies is emerging as a widely accepted basis for this. They provide a basis for capturing domain knowledge, the meaning and context of data as well as information requests. For example, they can be used to model the knowledge "*What are the characteristics of a good landfill site?*". A lot of research has focused on the representation of ontologies. RDF [4] and DAML + OIL [5] are some of the widely-used representations. Another important issue is that of knowledge sharing. Imposing a certain perception or model of domains on all users to simplify the task of correlating information does not seem to be a practical approach since coming up with such a model that all users can agree on seems impractical. This is because different users tend to have different perceptions of domains and these views are largely based on the components or aspects of the domains that are of more interest to them personally. Instead, standards like DAML+OIL [5] that allow users to independently create, manage, interoperate and share their knowledge in the form of ontologies seems a much more practical and useful approach. This approach allows users to create and maintain their own "views" of knowledge and still interoperate with other users. Users or communities can define their own ontologies, extending basic ontologies. This allows the bottom up design of meaning and also sharing of high-level concepts [6]. There is also a need to support techniques for ontology interoperation [7, 8] and inter-ontological relationships. The ontologies can then be used for annotating content so as to associate a well-defined meaning and context. This helps a user identify information relevant to his information request. Finally, there is a need for tools that allow users to collaborate with other users and formulate and execute information requests.

A peer-to-peer architecture is in line with the approach of knowledge sharing. It allows users to register themselves with a peer-to-peer network, maintain their own set of ontologies, and share them with the rest of the users. The knowledge sharing capability also allows users a semantic and personalized choice of using ontologies created or maintained by other users in the network to construct IScapes. Our previous work in InfoQuilt project [9, 10] used a multi-agent information brokering architecture with no support for knowledge sharing. It focused on support for complex inter-ontological relationships, a computing paradigm to support complex information requests called *Information Scapes* or *IScapes* and use of IScapes to support human assisted knowledge discovery (HAND). We have extended the InfoQuilt project to support a peer-to-peer architecture. This work is presented in [11]. A peer (user) in such a system can create a *personalized knowledge base* and still be connected to the *global knowledge base*. The *global knowledge base* is the entire shared network of ontologies available through the peer-to-peer network. The *personalized knowledge base* is essentially the set of ontologies that the user finds to be most relevant for describing his domains of interest according to his perception. The user may create these ontologies himself (by extending other ontologies on the network or using components developed by other users on the network) or search for them in the global knowledge base and then download them from the network. The personalized knowledge base of a user is essentially his view of the knowledge.

Right now, there are about 150 ontologies listed on daml.org. But there are only few people working in this area right now. In the future, if the semantic web becomes a reality, there will be many more users creating and sharing thousands of ontologies. Then, it will be difficult for a user to search for relevant ontologies to create queries. The task of searching for relevant ontologies becomes non-trivial. Knowledge about a user's interests can be used by the system to help him do this by using a simple keyword search to first identify a list of all related ontologies and then using personalization techniques to re-rank the ontologies in the list. In InfoQuilt, we use the user's personalized knowledge base (ontologies of the user's interest) as a basis for personalization. This thesis presents our knowledge based personalization approach.

Most research in the area of personalization has been done in the context of e-commerce web sites. Although there are a set of general techniques available, these are usually adapted according to the characteristics of the web site, expected user navigation pattern, and sometimes even a fixed set of domains of interest. Pretschner and Gauch [12] present summaries of a number of approaches and commercial products. We combine a number of personalization techniques and adapt them to use the personalized knowledge base. Particularly, we describe use of InfoQuilt's framework to support complex inter-ontological relationships to support query relationships. A query relationship is conceptually similar to market basket association rules [13]. It defines possible relationships between query domains. For example, a query for flight details may cause a query for hotel reservations. Suppose a user queried about a flight to a certain location and then queried for "Ramada". The term "Ramada" may apply to different domains. It can be the name of a company, for example. To simplify our example, assume that the system finds two ontologies relevant to the keyword "*Ramada": company, hotel*. Use of a market basket association rule can indicate that when a user searches for flight information, it is more likely that the user is interested in the domain of "hotels". Consider another example. Assume that the user has created market basket rules (Airline → UGAFootball), and (Airline → UGABasketball) to indicate that he would like to attend football and basketball games if UGA bulldogs are playing in the game. So it is likely that he will look for airline tickets when such a game is scheduled. In this case, if the user uses the keyword "bulldog[1]" to search for relevant ontologies, the market basket rules only indicate that the domains UGAFootball and UGABasketball are equally relevant. A query relationship goes a step further by using InfoQuilt's framework for complex inter-ontological relationship to precisely define (Airline → UGAFootball) and (Airline → UGABasketball). For example, it is possible to construct a query relationship that indicates that if a UGAFootball game is scheduled some time after the arrival date of the flight and in a location near the destination airport, then the user may be looking for flights to attend that game. The system can now use this to evaluate whether such a game is scheduled or not. Assume that the user's previous IScape (query) for flights inquired about flights that reached Atlanta airport on November 15, 2001 and there is a home game scheduled for the UGA football team in Athens on November 16, 2001. However, there is no basketball game scheduled around that time for the UGA basketball team. This suggests that UGAFootball is likely to be more interesting than UGABasketball.

The rest of the paper is organized as follows. In chapter 2, we briefly describe the architecture of the InfoQuilt system (the peer-to-peer architecture) and its support for complex inter-ontological relationships as well as the IScape paradigm. InfoQuilt uses a number of techniques for personalization. In chapter 3, we describe these techniques and

[1] Bulldog is the mascot of University of Georgia.

the overall algorithm that shows how all the techniques are used together. In chapter 4 we discuss related research in the area of personalization. Finally, we note our conclusions and directions for future work in chapter 5.

CHAPTER 2

BACKGROUND – THE INFOQUILT SYSTEM

In this chapter, we describe the InfoQuilt system. We discuss the peer-to-peer architecture that allows users to share their knowledge in the form of ontologies, InfoQuilt's support for complex user-defined inter-ontological relationships, and the powerful information request and knowledge discovery paradigm called IScape.

2.1 Overall Architecture

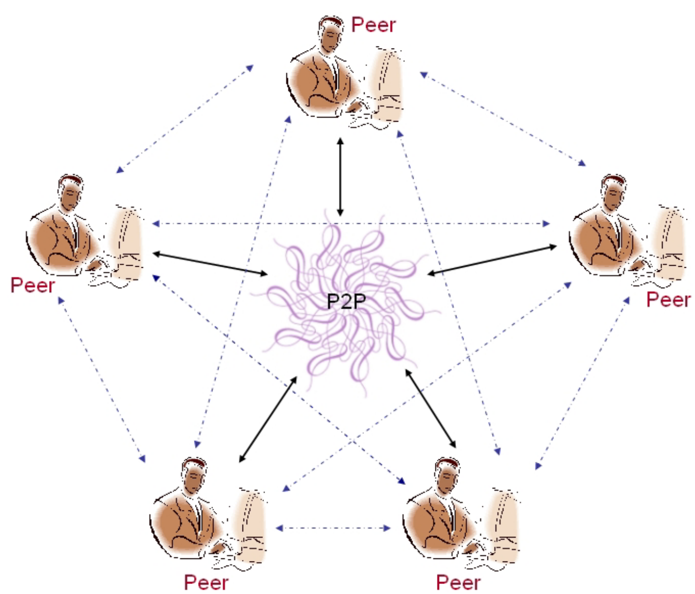

Figure 1: InfoQuilt System Peer-to-Peer Network

InfoQuilt has a user-centered peer-to-peer network with which users are registered. The network provides a shared directory of information and services. [11] presents a detailed discussion of the peer-to-peer network architecture used in the InfoQuilt system. As shown in Fig. 1, each user corresponds to a peer. A peer can register itself with the network, thereby gaining access to the global knowledge base (the shared ontologies). A user can then search for relevant ontologies from the network. Once he finds an ontology that is of interest, the peer-to-peer network can provide him with contact information to connect to the user that owns the ontology (i.e. the user that actually created it or maintains it). The user (peer) can use this information to establish a connection with the user that owns the ontology and download it.

2.2 Architecture of a Peer

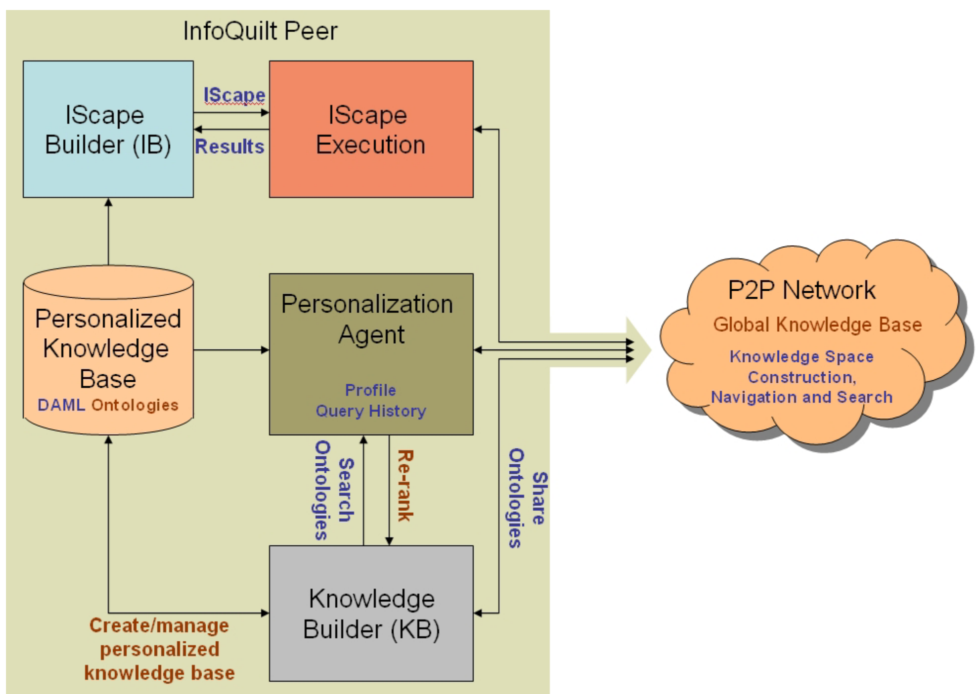

Figure 2: Architecture of a Peer in InfoQuilt Peer-to-Peer Network

Figure 2 shows the architecture of a peer connected to the network. Each peer creates and maintains its own personalized knowledge base which reflects the corresponding user's perspective or view of the knowledge. The next section describes this personalized knowledge base in more detail. The user creates, manages, and shares the personalized knowledge base using the Knowledge Builder (KB) tool. The personalization agent uses the user's profile, query history and personalized knowledge base to help him search for relevant ontologies from the network. Another tool called the IScape Builder (IB) is used to construct complex information requests, referred to as *Information Scapes* or *IScapes* in InfoQuilt.

2.3 Personalized Knowledge Base

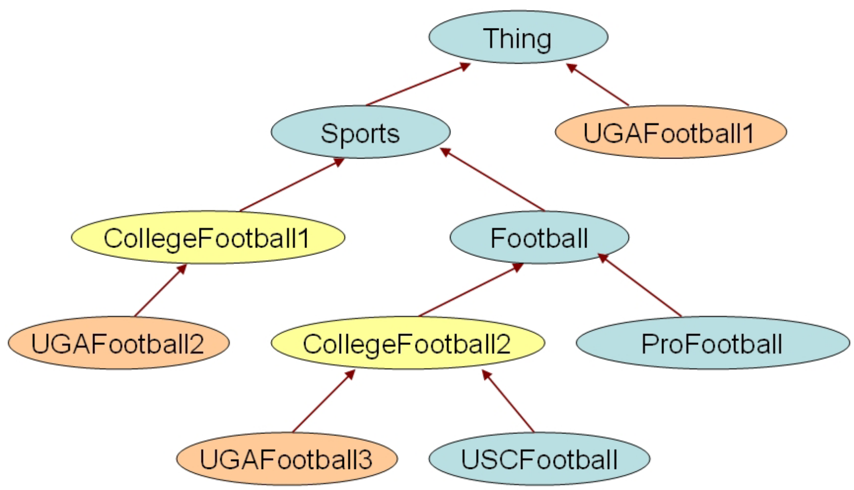

Figure 3: Example of Ontologies Supported by DAML + OIL

As mentioned earlier, each user can create his own personalized knowledge base, i.e. his personal view of the knowledge. The DAML+OIL framework is flexible enough to allow several users to create different ontologies describing the same domain. It does not enforce a strict domain hierarchy in that sense. For example, consider the ontologies in Fig. 3. There are two different ontologies for the domain of college football. They represent two different views of the domain. Similarly, there are 3 different ontologies for the domain of UGA football. The ontology *UGAFootball1* is shown to extend the ontology *Thing*, which is the root of all ontologies in the DAML+OIL standard. Although this kind of knowledge modeling may seem inappropriate, it is not incorrect. More importantly, it is not always possible to evaluate the semantic similarity of two ontologies. Most personalization techniques based on some kind of domain knowledge do not work well with this type of a hierarchy because they can only indicate user's *ontologies* of interest (as opposed to domains of interest).

In InfoQuilt, a user's personalized knowledge base is therefore required to have a strict domain hierarchy. For each domain in the hierarchy, the user can specify only one ontology. This ontology can be downloaded from the shared global knowledge base or created by the user using the Knowledge Builder tool. We believe this is a reasonable constraint since the personalized knowledge base should have ontologies that describe user's view of the domain. We can therefore assume that the user either finds such an ontology for each domain from the shared global knowledge base or creates one. Note that the domain hierarchy does not impose any constraints on the ontologies themselves. That is, if domain B is defined to be a subdomain of domain A, it does not imply that the ontology associated with domain B is related to that associated with domain A with a *subClassOf* relationship. The domain hierarchy is simply a way of organizing the user's domains of interest (and not ontologies) into a hierarchy.

Figure 4 shows an example of personalized knowledge base in the context of ontologies in Fig. 3. The *Root* node is a special case in that it may or may not have an ontology associated with it. For all other nodes, the label in black denotes the domain name and the name of the associated ontology is displayed in brown. Note that although according to its definition, the ontology *UGAFootball1* extends the ontology *Thing*, it is still associated with the domain UGAFootball, a sub-domain of CollegeFootball, which is associated with the ontology *CollegeFootball2*. It is not necessary for UGAFootball1 to be a *subClassOf* the ontology *CollegeFottball2*.

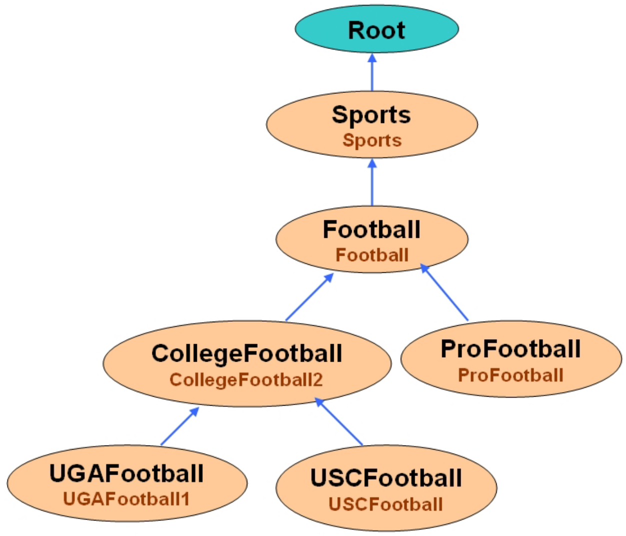

Figure 4: Example of Personalized Knowledge Base

The Knowledge Builder (KB) tool allows users to manage their personalized knowledge base. This includes creating new ontologies, downloading ontologies from the shared knowledge base and sharing the ontologies in the personalized knowledge base (i.e. adding them to the global knowledge base).

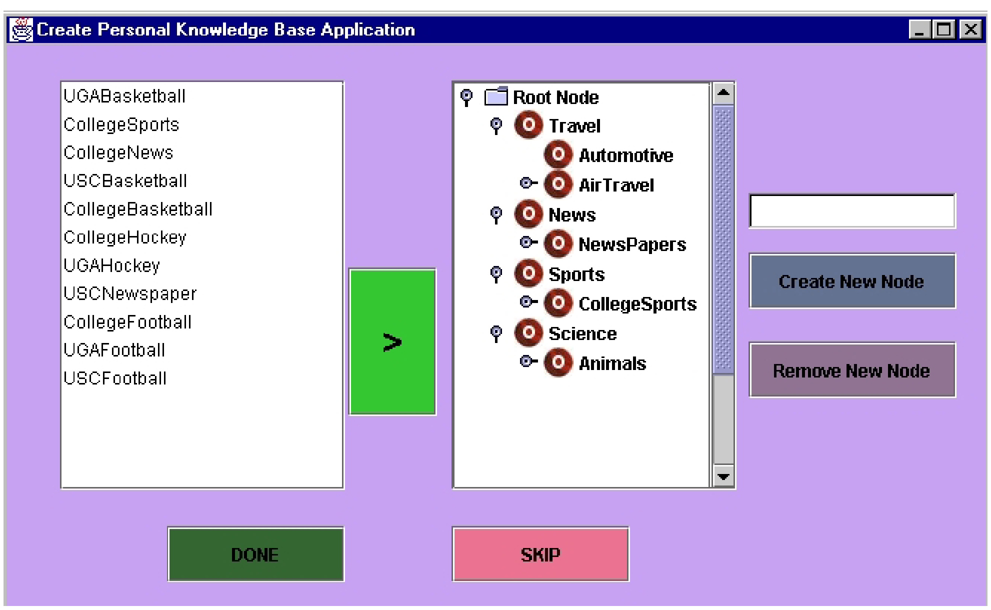

Figure 5: Personalized Knowledge Base Creation and Maintenance

Figure 5 shows a screen shot of the part of the tool that allows users to set up and manage their personalized knowledge base. The list on the left shows the available ontologies (that user created or downloaded from the network) and the domain hierarchy is shown to the right. The user can associate the ontologies with the nodes in the hierarchy.

2.4 Inter-Ontological Relationships and User-Defined Functions

In [9], we argued the need for inter-ontological relationships more complex than those supported by the DAML+OIL framework. For example, consider the following query:

"*Find all earthquakes with epicenter in a 5000 mile radius area of the location at latitude 60.790 North and longitude 97.570 East and find all tsunamis that they might have caused.*"

The relationship that an earthquake can cause a tsunami is a complex relationship between the ontologies earthquake and tsunami. The semantics of such relationships cannot be modeled using DAML+OIL. For example, this particular relationship has a spatial and temporal aspect. We can say that a tsunami may have occurred due to an earthquake if it occurred *some time after* the earthquake and in a *near-by region*. This example uses a fuzzy or approximate match of event dates and locations. These are very context-specific. For example, the temporal margin in the context of this relationship may be that of few days. However, that in the context of some other relationship (e.g. effects of administering a certain drug) may be that of few minutes. InfoQuilt supports an infrastructure to model such relationships and use of user-defined functions to compute such fuzzy or approximate matches.

Since such relationships are not supported by DAML+OIL, we continue to provide them as a feature available at the peer. In the future, we would like to provide the capability of sharing these relationships as well. It is this framework for complex relationships that provides the basis for use of query relationships for personalization as described in section 3.2.5.

2.5 Information Scapes and Human-Assisted Knowledge Discovery (HAND)

InfoQuilt supports specification of complex information requests. This paradigm is referred to as *Information Scapes* or *IScapes* in InfoQuilt. An IScape is defined as "*a computing paradigm that allows users to query and analyze the data available from diverse autonomous sources, gain better understanding of the domains and their interactions as well as discover and study relationships*." [9] An IScape is specified using components of the ontologies in the personalized knowledge base, inter-ontological relationships and user-defined functions. InfoQuilt supports human-assisted knowledge discovery (HAND) by allowing users to use IScapes to construct hypothesis that can then be substantiated or unsubstantiated by querying, integrating, and analyzing information available from diverse autonomous information sources. Refer to [9] for further details and detailed example.

CHAPTER 3

PERSONALIZATION IN INFOQUILT

As mentioned earlier, the task of searching relevant ontologies that best describe user's perspective of a domain becomes non-trivial in the context of a system with a peer-to-peer network that supports sharing of independently created ontologies. InfoQuilt therefore uses a number of personalization techniques or their adaptations that use user's profile, query history, personalized knowledge base, and query relationships to help the user locate useful ontologies. This, in turn, helps the user in building his personalized knowledge base. The peer uses a search service available from the peer-to-peer network to search for the ontologies using a simple keyword search. In the rest of the paper, we will use the term *query* to imply a keyword search unless explicitly specified. The service returns a list of ontologies that match the keywords. The personalization module then re-ranks this list. We describe the representation of user profiles in section 3.1. In section 3.2, we describe various techniques and indicators used to suggest user's interest. In section 3.3 we show how all the techniques described in section 3.2 are put together to re-rank a list of ontologies returned from the peer-to-peer network. Finally, we present several examples in section 3.4 to demonstrate our approach.

3.1 Representation of User Profiles

Most personalization techniques rely heavily on user profiles that keep track of users' domains of interest. Typically, a domain hierarchy is used as a basis for this. Data is classified according to this domain hierarchy. It is also possible to have multiple classifications for a data item. For a given query (keyword query), the retrieved results can easily be sorted by assigning a higher rank to a data item that is classified under one or more classifications that are known to be of interest to the user.

Apart from the domains of interest, keywords themselves can also be good indicators of user's interests. For example, for a given keyword query, the system can monitor which results out of those returned did the user find relevant. This can be done by monitoring user's clickstream, for example. This can give an indication of the domains that the user finds more relevant, given a keyword. This technique helps system learn the terms that the user is used to using more often in the context of each domain of his interest.

Another crude indicator of user's interests is the frequency of queries in each domain. This can also be monitored automatically as mentioned earlier. Given a query, using knowledge of user's domains of interest only indicates that the data items classified under the domains listed in user's profile may be more relevant than others. Using frequency of queries can be used to sort the results at a higher granularity.

The approaches described above are some of the most common approaches used in most personalization techniques. The Personalization Agent in InfoQuilt also uses these approaches. It maintains a profile for the user. A user profile consists of a set of tuples, each of which is of the following format:

<*keyword, ontology, frequency, latest interest, IScape*>

A tuple <*k, o, f, l, i*> can be interpreted as follows. *k* is the term used to query, *o* is the ontology that the user selected when the results were returned[2], *l* is a Boolean value, which if true, indicates that when the term *k* was last used in a query, the user found the ontology *o* to be relevant, and *i* is the name of the IScape that was constructed from the

[2] Note that the result returned is a list of ontologies.

selected ontology[3]. The next section describes how these values are used by the Personalization Agent. Although user profiles are maintained automatically by observing user queries, InfoQuilt also provides a tool that allows users to explicitly manage their profiles. Figure 6 shows a screen shot.

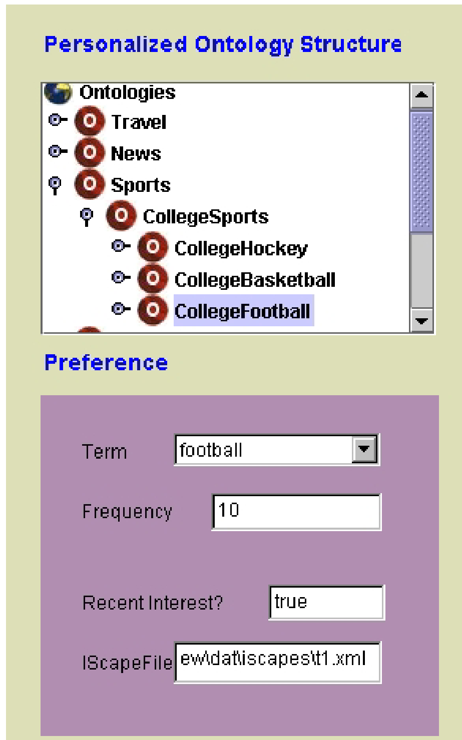

Figure 6: Personalized Knowledge Based User Profiles

3.2 Personalization Techniques

This section describes the techniques and indicators used by the Personalization Agent in the InfoQuilt system. When a user submits a keyword search, the Personalization Agent forwards it to the peer-to-peer network's search service. The service returns a list of relevant ontologies. For each ontology in the list, a score can be computed according to

---

[3] IScapes have to be stored on the user's machine before they are executed. They can therefore be referenced using their name.

each applicable technique, described the next sections. Note that these scores are on a scale of 0 to 1.

3.2.1. Keywords matched

This technique is that on which keyword search is based. A result with more number of keywords (out of those specified) matching is likely to be more relevant. The score is computed as $M_1 = m/n$, where m = no. of keywords matched and n = total number of keywords. Note that this is not the same as checking the number of times a keyword appears in a data item. Using a larger number of keywords relevant to a domain will help in identifying the relevant ontologies more precisely. Note that an ontology can consist of definitional as well as assertional components. Definitional components define the domain itself whereas assertional components are analogous to entities on the domain. This allows the user to search using the terms “CoopIS” and still locate the ontology “Conference” if it has an assertional component for the CoopIS conference.

3.2.2. Profiles matched

This technique takes into account the knowledge of user’s domains of interest with respect to the specified keywords. It is based on the assumption that if the user has already used some of the keywords (from the current query) in some queries in the past, the ontologies from those tuples, if in the result list, can be considered to be more relevant. Note that this technique tries to exploit the situation where several of the keywords in the current query indicate a common domain of interest. The keywords are first matched against the tuples in the user profile. Suppose n tuples were found for those keywords (i.e. the keyword in the tuple was used in the query). Assume that an ontology O_i appeared in m out of those n tuples. The score for O_i according to this technique would then be $M_2 = m/n$, where m = no. of profiles matched and n = total number of profiles matched given set of keywords. This technique seems similar to the first technique. However, there is a subtle difference. The first technique uses keyword match as a score.

On the other hand, this technique uses the knowledge about what users find relevant, given a set of keywords. That is, it is based on user's profile.

For example, assume that the user is very interested in UGA football team. He has queried several times in the past about the team and its schedule. The user profile contains the following tuples.

T_1 : *<bulldog, UGAFootball, f1, true, IScape*$_1$*>*

T_2 : *<schedule, UGAFootball, f2, true, IScape*$_1$*>*

Suppose the user queries for the schedule of the UGA Football team using the keywords "bulldog schedule". The ontologies returned are UGAFootball, UGABasketball and UGAHockey. Using the first technique, both the keywords would match all the three ontologies[4]. Their scores would therefore be the same. However, the profile matching technique examines the use of these keywords in the past and the ontologies that the user found relevant (in this case, UGAFootball). The scores for the ontologies will be:

UGAFootball : 2/2 = 1

UGABasketball : 0/2 = 0

UGAHockey : 0/2 = 0

3.2.3. Knowledge about latest context (domain)

This technique takes into account knowledge about the latest context. For a given ontology O_i, if there exists a tuple in the user profile such that first item (keyword) is one of the keywords used in the current query, second item (ontology) is the ontology O_i, and the third item (latest interest) is *true*, then M_3 = 1, else M_3 = 0. This technique takes

[4] We are assuming that the keywords "bulldog" and "schedule" match all the three ontologies. This is a reasonable assumption as both the keywords are relevant to all the three ontologies.

advantage of the fact that a term in the current query, when it was last used in another query, a particular ontology (out of several other relevant ones) was found to be more relevant. For example, the keyword "bulldog" is relevant in several domains (UGAFootball, UGABasketball, UGAHockey, EnglishBulldog, and so on). If there exists a tuple <*bulldog, o, f, true, i*> in the profile such that *o* is an ontology in the list returned, then it can be inferred that ontology *o* is more likely to be of interest to the user compared to other ontologies related to the keyword. So, if *o* is UGAFootball, then it is more likely that user is interested in UGAFootball as compared to UGABasketball, UGAHockey, etc.

3.2.4. Frequency of querying a domain

The profile of the user keeps track of the frequency of user's queries for a given domain. This technique weighs the ontologies in the result list by this frequency[5]. So, if an ontology in the list is queried more often than another, it will be ranked higher. This technique uses all tuples from the user's profile that have one of the keywords used in the current query. That is, we exclude the tuples that use keywords other than those used in the current query. Note that a keyword can be relevant to multiple domains or ontologies. Therefore, this technique allows the system to use the frequency of using one ontology over others that are relevant to the keyword. This technique is similar to the second technique (profile matching). The difference is that profile matching takes into account the number of keywords that point to the same ontology as that of interest, whereas this technique uses the query frequency. For example, if a user's profile indicates that a user is interested both in UGAFootball as well as UGABasketball and he has used the query term "bulldog" to query these ontologies in the past, then there will be following tuples in the user's profile.

---

[5] Note that although we use the term frequency here, it implies the number of times a user used the keyword in the tuple and selected the ontology in the tuple.

T_1 : <*bulldog, UGAFootball,* f_1*, true, IScape*$_1$>

T_2: <*bulldog, UGABasketball,* f_2*, false, IScape*$_2$>

Assume that the user's current query is "bulldog schedule" and the search service in the network returns UGAFootball and UGABasketball ontologies[6]. The score computed for an ontology O_i according to this technique is its normalized frequency $M_4 = f / F$, where f is the sum of frequencies f_i from all tuples such that the ontology in those tuples is O_i and F is the sum of frequencies f_i from all tuples (that match the keywords in the current query). So for our example, the score for UGAFootball would be f_1 / F and that for UGABasketball would be f_2 / F. Here $F = f_1 + f_2$. If $f_1 > f_2$, then UGAFootball would be ranked higher than UGABasketball[7]. That is, given the keyword "bulldog", if the user is known to be more interested in UGAFootball than UGABasketball, then UGAFootball is ranked higher. Note that in this example, the profiles use a single keyword. Consider another example:

T_1 : <*bulldog, UGAFootball, 10, true, IScape1*>

T_2: <*bulldog, UGABasketball, 8, false, IScape2*>

T_3: <*football, UGAFootball, 12, true, IScape1*>

The query is "bulldog football schedule". The ontologies that match are UGAFootball and UGABasketball. In this case, the scores will be:

UGAFootball : (10 + 12) / (10 + 8 + 12) = 0.73

UGABasketball : (8) / (10 + 8 + 12) = 0.27

[6] We have included only 2 result ontologies to simplify the example. Ideally, there would be several other ontologies identified.

[7] This is true if this technique is used by itself. The overall personalization algorithm combines all the techniques and the re-ranking may be different based on other factors.

Note that the term "football" also contributes to increase the score for UGAFootball since the user has used it in the past in the context of UGAFootball ontology. Note that if the frequency in T_2 is greater than that in T_1 and T_3 combined, then this technique would rank UGABasketball higher than UGAFootball. This is because this technique purely relies on frequency of finding an ontology more relevant than others, given a keyword.

3.2.5. Query Relationships

Several e-commerce sites create databases of user clickstreams and actions. Data mining techniques are then applied to extract user navigation and browsing patterns. A lot of research has been done in this area [e.g. 13, 14, 15, 16]. Such rules help in predicting user's next action or probable interest based on his current navigation. For example, consider the rule A, B, C → D. Assume that A, B, C and D are actions such as viewing an item's details, buying it, etc. If a user's current usage pattern matches this rule, the system can predict that his next action will be D with a certain confidence [13]. Similarly, A, B, C, and D can also be items. Since certain items tend to be purchased in groups, the fact that a user purchased A, B, and C can help the system predict that he may also purchase D.

Query relationships in InfoQuilt are similar rules but they are more concrete. For example, assume that a rule indicates that if the bulldog football team has a game scheduled, then it is likely that the user will query about air travel and vice versa. Similarly, another rule indicates that if there is a basketball game scheduled in which the UGA Basketball team is playing, then it is likely that the user will query about air travel and vice versa. Suppose the user queried for air travel on a certain date to a certain location. He then queries for "bulldog schedule". If the ontologies UGAFootball and UGABasketball are found, then the rules only indicate that both UGAFootball and UGABasketball are relevant with confidences *x* and *y*. However, this confidence is based on the number of times the domain is found to be relevant. This is similar to the previous technique based on frequency in that sense. However, query relationship in InfoQuilt make these rules more concrete by specifying the relationship between these two queries. We use the framework for modeling inter-ontological relationships as a basis to support this. For example, the first rule will correspond to the following query relationship:

```
spatiallyNear( UGAFootball.gameVenue, Flight.arrivalCity) &&
temporallyNear( UGAFootball.gameDate, Flight.arrivalDate)
```

InfoQuilt supports the use of user-defined functions as specialized operators to model complex relationships. We describe this work in detail in [9]. Here, the functions spatiallyNear and temporallyNear compute the spatial and temporal proximity of the football game and the flight. That is it specifies that if there is a football game scheduled in a place *close by* the arrival city and at a date *soon after* the arrival date, then the user is quite likely traveling for the game.

The query relationships are used as follows. First, a list of all applicable query relationships is constructed. This includes all query relationships that involve one ontology from the last executed IScape (available from the user's profile) and one ontology from the result list. For each ontology in the result list, the query relationships in which it is involved are evaluated one by one until one is found to be true. If at least one query relationship is found to be true, the score for that ontology $M_5 = 1$, else $M_5 = 0$. Note that if more than one query relationship for an ontology evaluate to true, it does not indicate that it is more relevant than another ontology for which only one query relationship is found to evaluate to true.

Consider the following example. The user's last query was about a flight to Atlanta on November 16, 2001. Next, the user queries for "bulldog schedule". Note that this example does not use the user's profile for any information other than for knowing the last executed IScape. We therefore do not describe the user's profile in detail. The following table shows the games scheduled for UGA football and basketball teams.

Table 1: Sample Data for Example

Team	**Flight Query**	**Basketball**	**Football**	**Football**
Date	Nov. 16, 2001	Nov. 17, 2001	Nov. 19, 2001	Dec. 1, 2001
Location	Atlanta, GA	Athens, GA	Springfield, MA	Athens, GA

It is clear from the above table that the query relationship between Flight and UGAFootball will not return any results since the game on Nov. 19 is not near Atlanta and the other game is on Dec. 1. The query relationship between Flight and UGABasketball however finds one game scheduled for Nov. 17 in Athens, which is very near Atlanta. So the score for UGABasketball would be 1 and that for UGAFootball would be 0.

3.2.6. Distance from a domain of interest

This technique evaluates the smallest distance of an ontology from another ontology that is known to be of interest to the user based on the domain hierarchy in the personalized knowledge base. The smaller the distance, the more relevant it is likely to be. For example, the domain UGAFootball is known to be of interest to the user. If the user now queries for "gamecock schedule", the ontology USCFootball is found to be relevant by keyword match. However, the user's profile knows nothing about user's interest in this ontology. It therefore evaluates the distance between USCFootball and all ontologies in the user's profile. The smallest distance is then considered as the score. The distance is computed as follows. Suppose we are computing the distance of ontology A to another distance B known to be in the user profile. The initial score is 1. From ontology A, traverse to the common ancestor of ontologies A and B by moving higher in the hierarchy. Each time you traverse to a parent domain, multiply the score by 0.5. From the common ancestor traverse down in the hierarchy to reach ontology B. Each time you traverse to a child domain, multiply the score by 0.25. For our example of computing the distance of USCFootball from UGAFootball, the score would be $M_6 = 1*0.5*0.25 = 0.125$. Note that if the ontology is in the user's profile, its score would be 1. Also note that this technique can be applied only to compute scores for ontologies that are in the personalized knowledge base because they are organized into a domain hierarchy as described in section 2.3.

3.3 Personalization Algorithm

The techniques described in the section 3.2 take advantage of different characteristics and indicators. InfoQuilt's Personalization Agent uses all these techniques and combines their scores to compute a weighted sum that is used for re-ranking the ontologies. Not all techniques can be applied in all situations since some of them use information from the user's profile while some do not. The following table shows the techniques used for each case. Case 1 is when the user's profile contains at least one tuple that matches the keywords in the query and case 2 is when such tuples are not found in the profile.

Table 2: Personalization Techniques Applicable to Query Cases

Technique	**Case 1**	**Case 2**
Keywords Matched	✓	✓
Profiles Matched	✓	×
Knowledge of Latest Context (Domain)	✓	✓
Frequency of Querying a Domain	✓	×

Query Relationships	✓	✓
Distance from a Domain of Interest	✓	✓

Next we arranged the techniques in an ascending order so that more powerful techniques appear earlier. The following table shows the techniques in this order.

Table 3: Order of Personalization Techniques

1	Keywords Matched
2	Profiles Matched
3	Query Relationships
4	Frequency of Querying a Domain
5	Knowledge of Latest Context (Domain)
6	Distance from a Domain of Interest

Here are some points of rationale for selecting this order. The keyword match is obviously a much stronger indicator than other techniques since it relies on the fact that the keywords specified by the user were found in the ontology. The profile matching technique relies on the knowledge of user's past context and use of the keywords in the query. It is however ranked second to the keyword matching technique because it works best when the keywords were used by the user in the past. For a new set of keywords, this technique does not perform well. On the other hand, if they keywords were used in the past, it adds to the score as computed by keyword matching technique. Query relationships do not rely on user's profiles. They depend on actual information available. They can strongly agree or deny whether the fact that a query in one domain can lead to a related query in another domain applies in the current context of user's query. This score is better than a crude score based on average query patterns (frequency). Knowledge about latest context is not a very strong measure of interest because of the fact that it is usually difficult to detect abrupt change in the user's query context unless established otherwise due to the use of query relationships. We use this order to assign a default set of weights to these techniques for the two cases. These weights are however configurable. The user can experiment with these weights to adjust them to work best with his personalized knowledge view and query patterns[8]. The default weights assigned are shown in the table below.

Table 4: Weight Model for Personalization Techniques

	Technique	**Case 1**	**Case 2**
1	Keywords Matched	0.4	0.5
2	Profiles Matched	0.2	-
3	Query Relationships	0.15	0.35
4	Frequency of Querying a Domain	0.1	-
5	Knowledge of Latest Context (Domain)	0.1	0.1
6	Distance from a Domain of Interest	0.05	0.05

3.4 Examples

In this section, we present a number of example scenarios to demonstrate how the above algorithm applies. Figure 7 shows the personalized knowledge base of a user. We will use it for all the examples in this section. The name of the domain is shown in black and the name of the ontology associated with the domain is shown in brown. All ontologies in the personalized knowledge base have the same name as the domain itself. This is not

[8] The weights specified here were arbitrarily assigned. However, by conducting a series of experiments, we can try to come up with a best weight assignment. We list this as one of the tasks to be undertaken next.

necessary. It is done only to simplify the example. The black arcs indicate query relationships. They indicate that a user may be interested in attending football and basketball games in which UGA bulldogs are playing. Similarly, an air flight reservation may also cause a hotel reservation. The query relationships between UGAFootball and Flight and UGABasketball and Flight indicate that the game must be scheduled at the most 3 days after the date of arrival and at a location that is at the most 150 miles from the arrival city. Similarly, the query relationship between Flight and Hotel indicates that the travel date and hotel stay period should match exactly.

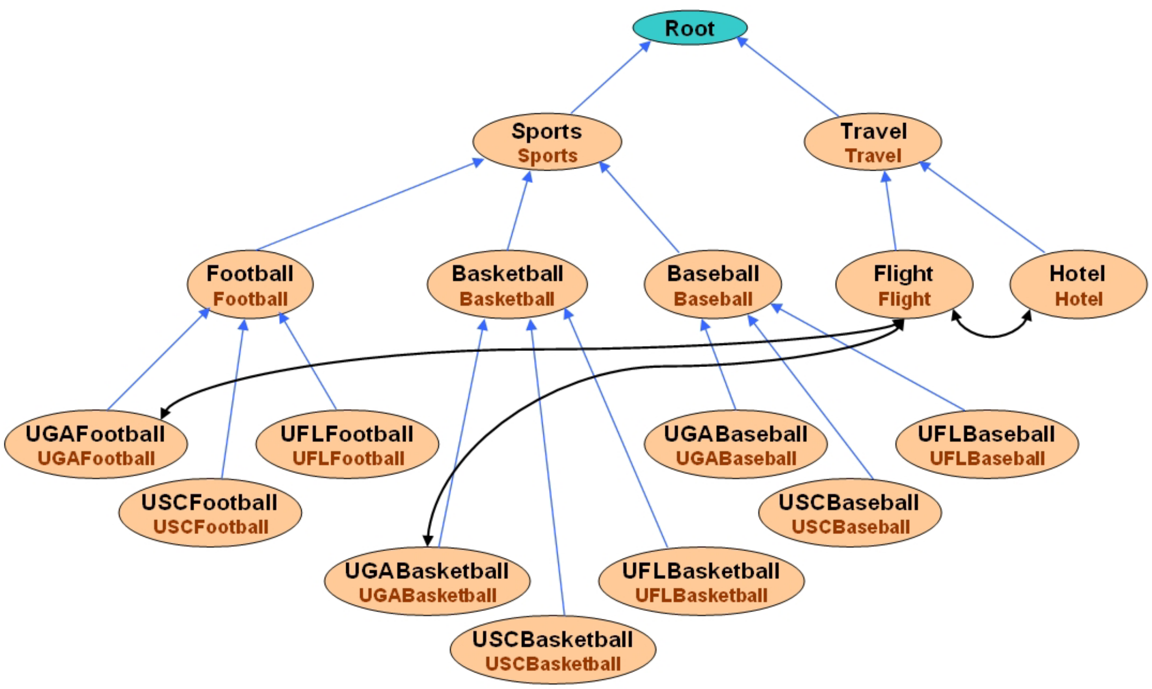

Figure 7: Personalized Knowledge Base for Examples

Example 1

Assume that the user's profile contains the following tuples:

T_1 : <bulldog, UGAFootball, 10, false, $IScape_1$>

T_2 : <bulldog, UGABasketball, 12, true, $IScape_2$>

Suppose the user queries for “bulldog schedule”. Assume that the last IScape executed by the user was *$IScape_2$* involving the domain UGABasketball. Note that this query is a case 1 query. The following table lists the ontologies returned from the peer-to-peer network by the search service and shows how a score is calculated for each one of them. Each row in the table corresponds to and ontology returned by the search service. The columns indicate the techniques applied and the weighted scores computed by applying the corresponding technique. We indicate the techniques by their numbers as used in table 4. Refer to table 4 for weight assignments. The final column in the table shows the final score computed for the ontology by combining all the scores.

Table 5: Score Calculations for Example 1

Ontology	**1**	**2**	**3**	**4**	**5**	**6**	**Sum**
UGAFootball	1*0.4	0.5*0.2	0*0.15	(10 / 22) * 0.1	0*0.1	1*0.05	0.595
UGABasketball	1*0.4	0.5*0.2	0*0.15	(12 / 22) * 0.1	1*0.1	1*0.05	0.705
UGABaseball	1*0.4	0*0.2	0*0.15	0*0.1	0*0.1	0.5*0.5* 0.25*0.25 *0.05	0.401
BulldogsFootball	1*0.4	0*0.2	0*0.15	0*0.1	0*0.1	0*0.05	0.4
BulldogsBaseball	1*0.4	0*0.2	0*0.15	0*0.1	0*0.1	0*0.05	0.4
EnglishBulldogs	0.5*0.4	0*0.2	0*0.15	0*0.1	0*0.1	0*0.05	0.2
Bulldogs	0.5*0.4	0*0.2	0*0.15	0*0.1	0*0.1	0*0.05	0.2

Note that UGAFootball and BulldogsFootball are two ontologies for the same domain. They are however different views of the same domain. The personalized knowledge base of the user however indicates that the ontology UGAFootball better reflects the user's perspective of the domain. The ontologies BulldogsFootball, BulldogsBaseball, EnglishBulldogs and Bulldogs are not in the user's personalized knowledge base and profile. As a result they are assumed to be of less interest to the user compared to others. Further, the ontologies EnglishBulldogs and Bulldogs are found to be less relevant by the keyword matching technique since they match fewer keywords (in this case, one) than other ontologies. Their scores are therefore even lower than BulldogsFootball and BulldogsBaseball ontologies. Although the ontology UGABaseball is in the user's personalized knowledge base, it has a lower score than UGAFootball and UGABasketball due to the reason that the user's profile does not indicate anything about user's interest that ontology. Among UGAFootball and UGABasketball, the latter has a higher score due to two reasons. First, the term "bulldog" has been used more often to query about UGABasketball than to query UGAFootball and that it is also the latest context for that term. That is, in the latest query using the term bulldog, the user found UGABasketball more relevant.

Example 2

We modify the above example slightly to show how the results change. Suppose UGAFootball is the latest context for the keyword "bulldog". That is, the user profile contains the following tuples.

T_1 : *<bulldog, UGAFootball, 10, true,* $IScape_1$*>*

T_2 : *<bulldog, UGABasketball, 12, false,* $IScape_2$*>*

Consider the same query again ("bulldog schedule"). The following table shows the score calculation. We show the scores for only UGAFootball and UGABasketball ontologies. The scores for the rest of the ontologies would remain the same as in table 5. The columns that change are highlighted.

Table 6: Score Calculations for Example 2

Ontology	**1**	**2**	**3**	**4**	**5**	**6**	**Sum**
UGAFootball	1*0.4	0.5*0.2	0*0.15	(10 / 22) * 0.1	**1*0.1**	1*0.05	**0.695**
UGABasketball	1*0.4	0.5*0.2	0*0.15	(12 / 22) * 0.1	**0*0.1**	1*0.05	**0.605**

Thus, using the keyword "bulldog", although the ontology UGABasketball has been queried more often than UGAFootball, since UGAFootball is the context used in the last query that used the term "bulldog", it is scored higher.

Example 3

We modify the above example again as follows. Assume that the last query execute by the user was to locate the Flight ontology and he executed an IScape querying for details about flights to Atlanta on November 16, 2001. The user profile has the following tuples:

T_1 : *<bulldog, UGAFootball, 10, true,* $IScape_1$*>*

T_2 : *<bulldog, UGABasketball, 12, false,* $IScape_2$*>*

T_3 : *<airline, Flight, 1, true,* $IScape_3$*>*

Consider the same query again ("bulldog schedule"). This example demonstrates the effect of using query relationships. Assume the game schedules as shown in table 1. The following table shows the scores. Again, we show the scores for only UGAFootball and UGABasketball ontologies since these are the only ontologies that have any query relationships with the ontology Flight. The columns that change are highlighted. The scores of the remaining ontologies will be the same as shown in table 5.

Table 7: Score Calculations for Example 3

Ontology	**1**	**2**	**3**	**4**	**5**	**6**	**Sum**
UGAFootball	1*0.4	0.5*0.2	0*0.15	(10 / 22) * 0.1	1*0.1	1*0.05	0.695
UGABasketball	1*0.4	0.5*0.2	**1*0.15**	(12 / 22) * 0.1	0*0.1	1*0.05	**0.755**

The query relationship between UGAFootball and Flight does not return any results since although there are two football games scheduled for UGA football team, they do not match the requirements specified by the query relationship. One game is to be played in Springfield, MA. That is not within 150 miles from Atlanta (the destination city of the flight). The other game is scheduled for December 1, 2001. The user however queried for airline tickets for November 16, 2001. The score in column 3 for UGAFootball therefore does not change. The query relationship between UGABasketball and Flight returns one result since there is a basketball game scheduled in Athens (close to Atlanta) on November 17, 2001. The score for UGABasketball is therefore higher than that for UGAFootball.

Example 4

Our last example is a case 2 query. Assume the same profile as in the earlier example. The query is “gators schedule”. There is no information in the profile that matches the terms in the query. The following table shows the ontologies returned and their scores calculated according to the weight assignments for case 2 type queries (refer to table 4).

Table 8: Score Calculations for Example 4

Ontology	**1**	**3**	**5**	**6**	**Sum**
UFLFootball	1*0.5	0*0.35	0*0.1	0.5*0.25 *0.05	0.506
UFLBasketball	1*0.5	0*0.35	0*0.1	0.5*0.25 *0.05	0.506
UFLBaseball	1*0.5	0*0.35	0*0.1	0.5*0.5*0.25*0.25 *0.05	0.501
GatorFootball	1*0.5	0*0.35	0*0.1	0*0.05	0.5
Alligator	0.5*0.5	0*0.35	0*0.1	0*0.05	0.25

Again, since the ontologies GatorFootball and Alligator are not in the user’s personalized knowledge base, they have a lower score. Further, the ontology Alligator matches fewer keywords (only “gators”) than other ontologies and therefore has a score even lower than GatorFootball. The ontologies UFLFootball, UFLBasketball and UFLBaseball are all in the user’s personalized knowledge base. However, the user’s profile does not indicate any knowledge about the user’s interest in these ontologies. Their minimum distance from an ontology that is in the user’s profile is therefore useful. These distances for UFLFootball (from UGAFootball) and UFLBasketball (from UGABasketball) are relatively smaller compared to that of UFLBaseball (from UGAFootball or UGABasketball). Therefore, they have higher scores. Thus, since the user profile can indicate that although nothing is known about the ontologies returned from the search service, since the profile indicates user's interest in UGAFootball and UGABasketball, the user is more interested in Football and Basketball as compared to Baseball.

CHAPTER 4

RELATED RESEARCH

The area of Personalized Search and Content Delivery is very broad. It has primarily been used in the context of e-commerce web sites. Personalization techniques have evolved from simple techniques that store user's interests by requiring the users to register with their site and then asking for an explicit profile (for example, by asking them to fill out a survey) to techniques that automatically log user's activities such as his clickstream, action sequences, etc into a log and then applying data mining techniques to extract navigation, browsing and usage patterns. There are broadly two ways of using these patterns. These are known as filtering algorithms. One type of algorithm tries to classify users into a set of broad groups (a community of users). It then tries to fit a user into one of these groups based on his navigation pattern, clickstream, etc. This is known as collaborative filtering. The other technique identifies each user individually and maintains individual profiles. This technique is known as individual filtering. Several personalization approaches also use a domain classification or concept hierarchies [17, 18] to track user's interests.

The personalized knowledge base created by a user in InfoQuilt is similar to a domain classification. However, there is one key difference. The ontologies associated with the nodes in this classification additionally provide knowledge about these domains. They provide the following two key advantages:

- Keywords, terms and concepts relevant to domains are usually specified in the ontologies. These are useful for locating relevant domains (and ontologies).

- The inter-ontological relationships that provide a basis for constructing query relationships can be useful for identifying domains that the user's profile provides no information for, but are related to other domains in the profile that are known to be of interest to the user.

It is difficult to directly compare personalization approaches. This attributes to the fact that application of personalization techniques largely depend on the actual application. Additionally, they have mostly been applied to search some content (documents, stories, emails, web pages, etc.). However, in InfoQuilt, we are employing them to search for relevant ontologies. A part of research in other systems also focuses on matching content to user's interests and search criteria based on techniques such as similarity [19], keyword matching, use of "cue words" and "cue phrases" [20], etc.

OBIWAN [17] uses a vector space model to classify documents according to an ontology. They then use length of the document, time spent by users on a document, and the strength of match between the content of a document and a category (i.e. $\gamma(d, c_i)$, where d is a document and c_i is a category) to track user's interests. They however conclude that the length of documents can be ignored. The user profiles used in InfoQuilt keep track of querying frequency for each domain. This measure is similar to the time spent on a document. Their work focuses on use of the strength of match between a document and a category. In InfoQuilt, however, we use personalization to search for ontologies, whose categories and context are already known.

Datta et al. [13] describe a collaborative filtering approach. They use a model of an e-commerce site and user's interactions with the site. This model is described in [21]. It serves both as a product catalog and as a basis for tracking user navigation. The profiles consist of action rules and market basket rules. If the user's actions match some action rule, that rule can then be used to predict his next action with a certain confidence. Similarly, since items tend to be bought in groups (e.g. a user that buys cereal will probably also purchase milk), market basket rules are used to model these groups. They are used to recommend products that the user might be interested in, considering the products he bought previously. This technique works well with user's changing interests. It is therefore called a dynamic profile. For example, user first logs in to purchase some computer science book for his son. Next he logs in to buy something else. The system does not keep recommending computer science books to the user. The tuples in user profiles used in InfoQuilt have a Boolean value which indicates the latest context (domain) with respect to a given keyword. This serves a similar purpose. The query relationships supported in InfoQuilt are similar to market basket rules. They are however more concrete and personalized because they exactly specify the condition when the relationship is true.

myPlanet [20] is an ontology-driven personalized news publishing service. They use ontologies to allow users to track user's interests explicitly. Simple relationships in the ontology are used to deliver content that may be of interest to the user. For example, if the user says he is interested in the research area "Genetic Algorithms", apart from stories on the genetic algorithms, they also deliver stories about research projects that have genetic algorithms as one of their research areas. This is similar to market basket rules in [13].

CHAPTER 5

CONCLUSION AND FUTURE WORK

The vision of semantic web is aimed at bringing order to the chaos of information on the web by associating semantics and context with content and thereby improving value of the information and making it more usable by computers (as well as people) for answering questions and performing tasks (for example, "*book a flight ticket to Atlanta next week according to my schedule.*"). Knowledge sharing in the form of ontologies is a critical feature for realizing this vision. We find the peer-to-peer architecture in line with this. [11] describes how we extended InfoQuilt to support knowledge sharing using this architecture. It is very likely for users to disagree on their perspectives of a given domain. The peer-to-peer architecture therefore allows users to create ontologies according to their own views, share them and use ontologies created by other users using the network. They can create a personalized knowledge base, which reflects their view of the knowledge. To help users locate relevant and interesting ontologies from this potentially huge repository, InfoQuilt deploys personalization techniques using the personalized knowledge base, query history and user's profile. In this paper, we described this approach to personalization used in InfoQuilt.

The personalization agent uses an individual filtering algorithm that combines a number of techniques. The following are some of the key advantages of our approach:

- Ontologies in the personalized knowledge base reflect the user's perception of the domain.
- The keywords and terms that are relevant to a domain are specified by the ontology. Additionally, it is also possible that since a certain ontology is chosen to

be more in line with user's perception of a domain (because it is in his personalized knowledge base), the terms used in the ontology are those more intuitive to the user as compared to others that may be missing from the ontology. These keywords and terms are useful for identifying relevant ontologies.

- A number of techniques are used in combination to help the users locate relevant ontologies that are more likely to be of their interest.
- InfoQuilt's framework to support complex relationships provides a good foundation for defining concrete query relationships that can identify related domains of interest in the current context of user's query with more confidence as compared to some other techniques such as navigation and usage patterns extracted from web logs. This is because the query relationships can define the meaning and context of the association between the domains that causes the user to be interested in one of them, if he is interested in the other.

Following are some directions for future work in this area.

1. For each domain, it is possible to identify a set of terms or phrases that very strongly indicate the context. We plan to use these in the future for automatic classification of content. These can also be used to search for relevant ontologies instead of simple keyword search.
2. The only type of relationships in the ontologies used for identifying domains that may be of interest to the user is "is-a" (as implied by the domain hierarchy). We can explore the user of other types of relationships supported by ontologies for this as well.
3. Evaluating query relationships using the IScape that the user executed last requires work equivalent to at least evaluating one IScape. Instead, the results from the previous IScape can be cached.

4. Keyword matching can be further given weights depending on which component of ontology the keyword matched. For example, if a keyword matches the name of a class as opposed to description, it should have higher value.
5. Experimenting with large amount of users and ontologies can help in identifying a reasonable weight assignment for the techniques.

BIBLIOGRAPHY

1. T. Berners-Lee (with Mark Fischetti), Weaving the Web, The original design and ultimate destiny of the World Wide Web, Harper, 1999.
2. T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities,” Scientific American, May 2001.
3. D. Fensel and M. Musen, Eds. “The Semantic Web: A Brain for Humankind,” IEEE Intelligent Systems, March/April 2001
4. http://www.w3c.org/RDF
5. http://www.daml.org/2001/03/daml+oil-index.html
6. R.Cover, DARPA Agent Mark Up Language (DAML) (http://xml.coverpages.org/daml.html)
7. Prasenjit Mitra, Gio Wiederhold and Stefan Decker: A Scalable Framework for Interoperation of Information Sources. Proceedings of the 1st International Semantic Web Working Symposium (SWWS `01), Stanford University, Stanford, CA, July 29-Aug 1, 2001, Jul. 2001
8. E. Mena, A. Illarramendi, V. Kashyap, and A. P. Sheth. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. International Journal on Distributed and Parallel Databases, Vol. 8, No. 2, pp. 223-271, April 2000.
9. A. Sheth, S. Thacker, and S. Patel, Complex relationships and Knowledge discovery support in the InfoQuilt System. Technical Report, LSDIS Lab, Univ. of Georgia, September 2001 (submitted for publication).
10. S. Patel, and A. Sheth. Planning and optimizing Semantic Information Requests using domain modeling and resource characteristics, Sixth International Conference on Cooperative Information Systems (CoopIS 2001)2001.

11. M. Arumugam, A. Sheth, and I. Arpinar. The Peer-to-Peer Semantic Web: A Distributed Environment for Sharing Semantic Knowledge on the Web. Technical Report, LSDIS Lab, Univ. of Georgia, November 2001. (submitted for publication)
12. A. Pretschner, and S. Gauch. Personalization on the Web. Technical Report. Information and Telecommunication Technology Center, Department of Electrical Engineering and Computer Science, University of Kansas, December 1999.
13. A. Datta, K. Dutta, D. VanderMeer, K. Ramamritham, and S. Navathe. An Architecture to Support Scalable Online Personalization on the Web. The VLDB Journal 10: 104-117 (2001).
14. AG Buchner, M Baumgarten, SS Anand, MD Mulvenna, and JG Hughes. Navigation Pattern Discovery from Internet Data. In Proceedings of WEBKDD (1999). Workshop on Web Usage Analysis and User Profiling, 1999.
15. R. Cooley, B. Mobasher, J. Srivastava. Data Preparation for Mining World Wide Web Browsing Patterns. Knowl Inf Syst 1(1):5-32.
16. M. Spiliopoulou, LS Faulstich, K. Winkler. A Data Miner Analyzing the Navigational Behavior of Web Users. In International Conference of ACAI (1999): Workshop on Machine Learning in User Modeling, 1999.
17. A. Pretschner, S. Gauch. Ontology Based Personalized Search. In Proceedings of the 11th IEEE Conference on Tools with Artificial Intelligence, 00. 391-398, Chicago, November, 1999.
18. T. Kurki, S. Jokela, R. Sulonen, and M. Turpeinen. Agents in Delivering Personalized Content Based on Semantic Metadata. In Proceedings of the AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace, pp. 84-93, Stanford, USA, 1999.
19. K. Bradley, R. Rafter, and B. Smyth. Case-Based User Profiling for Content Personalization. In Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Trento, Italy, August, 2000.

20. Y. Kalfoglou, J. Domingue, E. Motta, M. Vargas-Vera, SB Shum. myPlanet: An Ontology-driven Web-Based Personalized News Service. In Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing, Seattle, USA, August 2001.
21. A. Datta, D. VanderMeer, K. Ramamritham, and S. Navathe. Toward a Comprehensive Model of the Content and Structure, and User Interaction of a Web Site. In Proceedings of the Workshop on Technologies for E-Services (in cooperation with VLDB 2000), 2000.