# Data Mining for Personal Navigation

Gurushyam Hariharan[a],Pasi Franti[b], Sandeep Mehta[c]

[a]Bachelor of Engineering, University of Delhi
[b]Professor,Dept of Computer Science, University of Joensuu
[c]Bachelor of Technology, Indian Institute of Technology

## ABSTRACT

Relevance is the key in defining what data is to be extracted from the Internet. Traditionally, relevance has been defined mainly by keywords and user profiles. In this paper we discuss a fairly untouched dimension to relevance: location. Any navigational information sought by a user at large on earth is evidently governed by his location. We believe that task oriented data mining of the web amalgamated with location information is the key to providing relevant information for personal navigation. We explore the existential hurdles and propose novel approaches to tackle them. We also present naive, task-oriented data mining based approaches and their implementations in Java, to extract location based information. Ad-hoc pairing of data with coordinates (x, y) is very rare on the web. But if the same co-ordinates are converted to a logical address (state/city/street), a wide spectrum of location-based information base opens up. Hence, given the coordinates (x, y) on the earth, the scheme points to the logical address of the user. Location based information could either be picked up from fixed and known service providers (e.g. Yellow Pages) or from any arbitrary website on the Web. Once the web servers providing information relevant to the logical address are located, task oriented data mining is performed over these sites keeping in mind what information is interesting to the contemporary user. After all this, a simple data stream is provided to the user with information scaled to his convenience. The scheme has been implemented for cities of Finland.

**Keywords:** Location, location based information, data mining, personal navigation

## 1. INTRODUCTION

Joe wakes up to a beautiful Sunday morning. Without much planning, he sets out for a holiday. As he disembarks, his Jini wishes him: "Good Morning Sir. Welcome to Helsinki square. Can I help you?" Joe commands: "breakfast". Within seconds, the Jini informs,"Pizzeria Italiano, A-10, Central Market. Cheese pizza + large cola today only 5 euros." He is supplied with a map that guides him to the nearby pizzeria where the Jini has already booked a seat for him. He next queries for the nearest cinema hall. Upon indication of a choice, the Jini guides him to the Cinema hall after having booked his favorite corner seat in the box. He comes out of the hall, and commands the Jini to help him locating a cab service within five hundred meters. Within two minutes, the cab company identifies Joes location and agrees to pick him up in five minutes.

No we are not in the medieval ages and Joe is not the son of Aladdin. Joe's Jini is nothing but his PDA hooked up to a futuristic service provider. Implementation of such system for travelers, historians and other mobile users would be a Herculean task combining the contemporary technologies of global positioning, dynamic map handling, hypertext mining, and web-link analyses.

Plenty of solutions already exist for mobile data communication, global positioning and information retrieval to implement many of the features needed by Joe and other mobile users. Nevertheless, it is expected that such idiot-proof systems will not be widely available for the users for a long time. One way to do is to sit and wait for others to implement and provide the services of your desire, which may or may not appear in the near future. Another way is to utilize the existing information by the means of data mining [5,6], and to provide the user with the relevant information on the demand.

Further author information:
Gurushyam Hariharan: E-mail: gurushyam@acm.org, Telephone: 91 11 5089695

The Internet is an ocean of information and the key point for utilizing this information is the definition of what is relevant. Working of search engines [3] is often viewed as being quite similar to traditional IR schemes. The process of Information Retrieval was restricted to static data present either in the form of a table or a file or at the most a collection of data files. Data was nothing more than a collection of records; each of them associated with a unique key. A search algorithm would accept an argument 'a' and try to find a record whose key is 'a'. A successful search is more often referred to as 'retrieval'. Translating this scheme to the WWW, when a user submits a search query, in the form of a phrase, a software working on the engine's algorithm goes through the index to find web pages with keyword matches and ranks the pages in terms of relevance. What differentiates traditional schemes to WWW searches is the scale of data being considered. The drawback of such approach is the explicit need of the user for inputting key words and for relevance feedback. In the present paper, we will discuss a fairly untouched dimension to relevance: location. The location is typically represented by the pair of coordinates (x,y), such as latitude and longitude, and it can be obtained automatically by GPS or by GSM network[14]. The location is particularly of importance in personal navigation, where it can be used for the following:

1) Plotting the location on the map,

2) Giving navigational information to the user,

3) Providing information regarding the contemporary location of the user.

We consider the third case, which is basically a problem of information retrieval with the new attribute, location, included. The applications of personal navigation have also higher demand of automated processing of the information. The user does not have the time to browse the web, and explicitly define what is relevant for him at the particular moment. Instead, the location can be used as relevance feedback. The location can be obtained automatically and should be processed without interaction with the user.

The overall scheme of using information retrieval in personal navigation is shown in Fig. 1. The abstraction involves several issues that will be addressed in this paper:

1) Finding the location information from the data,

2) Task-oriented data extraction from the web,

3) User profiling (parameters)

Ad-hoc pairing of data with coordinates (x, y) is very rare on the web. On the other hand, many web pages include logical address (State/City/Street), which can be converted to coordinates via simple address coordinate database. The location-based information can then be picked up either from fixed and known service providers (e.g. Yellow Pages) or from arbitrary website on Internet. Once the web servers providing information relevant to the logical address are located, task oriented data mining is performed over these sites keeping in mind what information is relevant to the contemporary user. After all this, a simple data stream is provided to the user with information scaled to his convenience. We will also discuss the role of user profiling in personal navigation. User profiles are defined as individual properties of the users that allow us to define the relevance of the data for each person individually. User profiling aims at minimizing the user input during the retrieval process by automating the relevance feedback. This can be achieved either by creating predefined user profiles, or by monitoring the actions of the user and then adaptively learn his profile. The latter approach is essentially a data mining problem, and it can be attacked by learning the behavior of a single user, or by modeling the user as a part of the population of all users. The rest of the paper is organized as follows. The use of location-based information in personal navigation is outlined in Section 2. The problem of finding location information in web pages is studied in Section 3, and the issues in task-oriented data extraction are addressed in Section 4. The possibilities of user profiling are then discussed in Section 5. In Section 6, we present a prototype of address coordinate database that has been successfully implemented for the cities of Finland. Conclusions are drawn and potential future research problems are outlined in Section 7.

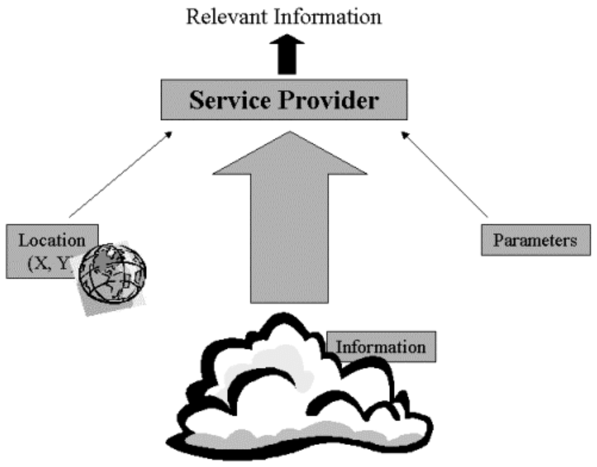

**Figure 1**: Outline of the overall schema.

## 2. DATA PROCESSING FOR PERSONAL NAVIGATION

Any navigational information sought by a user at large on earth is evidently governed by his location. The coordinates and the location themselves are not very interesting but they are the keys for accessing location-related information. The relevance of the information is thus defined by the preferences of the user along with his location. A number of location-based information providers do exist in the market. Such services response with information regarding the location for which the user queries the service provider. But such a service usually depends on a very limited, static database, resulting into an unintelligent solution. The needs of users, on the other hand, are ad-hoc at most times; hence a fixed database would not suffice. Another option for a mobile user is to browse and retrieve information on the move from the Internet. The problem with such a system is the immensity of information on the Internet. The user would be required to browse page after page manually. A mobile user is usually constrained in time and bandwidth domain. Expecting him to access web sites on the move is a tall asking. Automatic information retrieval from a dynamic data bank, such as World Wide Web, cannot be addressed by traditional queries. At the same time, opening a mobile user to the breadth of the web is not a good option either. Hence, scaled versions of the WEB (e.g. WAP) have not worked out to be very impressive so far. An evident solution to provide automatic location based information for personal navigation is to apply data mining in all steps of the process: information retrieval, defining relevance, and processing the data the client device. We could relate this to agent based systems on the Internet [7,8,9,10,11]. The procedure of providing information to the mobile user can be divided into the following steps:

1) Locate the user on the globe,

2) Find useful information-sources,

2a. Known information providers (e.g. yellow pages) 2b. Independent web sites,

3) Extract relevant information,

4) Process the data for the client device.

The task of providing a user with location specific information starts with locating the user on the globe. The user carries a hand-held device that can locate itself using a Global Positioning System (GPS) or a like-wise network such as GSM. The coordinates of a client device provided by the network can be in the form of waypoints or a latitude-longitude system. The next step is to convert the given coordinates (x, y) to a more useful description of the location, such as street, town, or country. This is because ad-hoc pairing of data with coordinates is very rare on the web but many web pages are attached with address. The location based information can then be searched from fixed and known service providers (e.g. Yellow Pages) or from any arbitrary website on the Web. The retrieved data is then processed for extracting the relevant information using the location and other relevance parameters as input. The location can be efficient factor for filtering out irrelevant information but it is not sufficient alone, and other factors of relevance should also be considered. For example, our leisure traveler, Joe, was interested in movie theatres and pizzerias. What would not interest him are the city council documents, history of the city, and so on. Only the acute information is relevant when he is on the move. Another traveler, on the other hand, might be a historian, and the scheme should identify him accordingly. In this case, the system would venture into the depths of history of a place and provide it to him on demand, on the scale supported by his device. After all this, a simple data stream should be provided to the user with information scaled to his convenience. Whether the information is pull-up or push-up could very well be left to the discretion of the user and the service provider.

## 3. ATTACHING LOCATION TO DATA

Relevance is traditionally defined by using keywords, or by using query-by-example approach, in which a sample input data is given and the system then searches data similar to the input. Additional information can be collected by monitoring the users behavior and adapt to his habits in the due course of time. Overall, task oriented data mining must be performed over the web sites keeping in mind what information is interesting to the contemporary user. Thus, traditionally the definition of relevance has been provided through

1) Keywords
2) Query-by-example
3) User profiling (monitoring user's history)

In what follows we will discuss the eventualities of using location as an extra touchstone for defining relevance.

### 3.1. How to use the location in the query

Given location (x,y) we determine the city under purview. The most basic of task would be to provide information about that city. Now consider a user's interest in finding out the nearest music stores. Hence, given location (x,y) we use coordinate-address conversion to know addresses of nearby music stores. The user query consists of the location (x,y) and some search parameters, apart from keywords. For example, any data within a given diameter from the current location can be accepted. The accuracy and tolerance in the distance depends on the movement of the user. Furthermore, the distance can also be used in the ranking of the search results so that information located nearer will have higher ranking value, and vice versa.

### 3.2. How to find the location information of arbitrary data

Ideally, we formalize the search queries using the current location (x,y), and the location of the candidate data (x2,y2) as the parameters. The location parameter can then be used by matching the coordinates using distance as a criterion. However, in many cases, the candidate data does not have the exact location (x2,y2), but only something less accurate such as street address and/or city. In such cases, we convert the location of the input and the query data to the same level of accuracy. This can be achieved in two ways:

(1) Converting the input coordinates (x,y) to a higher level corresponding to the comparative data ; e.g. (x,y)  nearest city. Then perform search by matching the city names.

(2) Vice versa; converting the input locations to the lower level or the coordinates. Same database applies in both cases but now the search is performed by matching the coordinates, which is an ideal case.

### 3.3. How to use coordinates in a query

Location can usually be used as a direct input for information mining, in a query. But in many a practical situations, we could face some practical anomalies. Use of distance is not always obvious to measure, especially if we use higher scale location as input. For example, suppose that we are moving from London to a small nearby city, e.g. Watford. Both cities have geographical centers but while we have covered half the distance from the geographical center of London, we still are in London and not in Watford. The point that emerges is that Euclidean scales might not hold good for practical situations. Hence, the less accurate location should be dealt accordingly and the distance depends on the size and shape on the object location. One way is to use a circle (x,y,diameter) for representing the location, and accordingly, revise the distance calculation. In general, the problem is one related to the more general classification problem, which is in itself is also a problem in data mining!

## 3.4. How to build the database

Building a database, or an index of web pages considering location issues is a challenging task in itself. It is quite different from the use of the database, and has its own complications, which need to be addressed.

First and foremost, we begin with populating a database that lists cities or addresses, along with their corresponding coordinates. Evidently, it can assume Herculean proportions if we need to cover the potential application area (the globe!) extensively. A solution, which we adopted, was to use service providers for building the database. Fixed address-coordinates services on web (e.g. National Land Survey of Finland) can be used efficiently if we do away with the issues of compatibility and costs.

A more effective solution would be to build a database dynamically using the web as a resource! (e.g. by crawling the web, or interactively searching existing services and manually processing the good ones). As an instance, one could carry out task oriented hypertext mining that could mine out the surface address of various targets (restaurants, hotels, places of historical importance) from the WWW.

As a matter of wishful thinking, it would be ideal, if the cult of specifying location coordinates along with surface addresses becomes ubiquitous. Applications akin to those for extracting surface addresses from websites can again be written for extracting coordinates to populate the database.

For experimental purposes, we implemented a piece code in Java that queries a thesaurus on the web [http://www.getty.edu/research/tools/vocabulary/tgn/index.html] and extracts the location coordinates of a number of cities in UK and Finland from the Internet. It takes as input a file containing the names of cities and outputs a file containing the latitude and longitude of the cities.

## 3.5. Practical problems

The effectiveness of the system would be dependent on the extensiveness and accuracy of the database for locationcoordinate conversion. Hence, the population of the database with accurate information is very important. Higher the precision of the latitude and longitude, greater is the accuracy[14]. Spiders can be created that could lazily crawl the web and extract addresses and keep expanding the database. Once the database is constructed to a considerable accuracy, not much maintenance is needed.

We discuss some of the practical issues that emerge in creating and maintaining such a database. If the dynamic building of the database is to be automated, it becomes a demanding data mining task in itself. Apart from finding the relevant pages and analyzing their content, we encounter the reliability issues. How can we know for sure that any found coordinates are accurate and true? The only solution is to compare the information regarding the location under scrutiny repeatedly from independent sources. If there are several coordinates of the same town with only small difference between each other, we can conclude that the coordinates are reliable and we can take the average or median. On the other hand if we have two coordinates with great difference, we infer that there is some anomaly. Another practical issue is the presence of a number of coordinate systems. While populating a database, we should make sure that the coordinate system is ubiquitous.

The intricacy of locating the user in the real world is directly proportional to the intricacy of the information targeted at the user. Let us suppose our user is flying above Pakistan on his way to London from Delhi. He might be just interested in the country as a whole. Hence, locating him to be above Pakistan suffices our job. On the other hand, if the user is with his device in a market place in Helsinki, and we would like to provide him with the intricate details about the sale on his favorite music album, we need to locate him with greater accuracy. The adjoining location hierarchy of Fig. 2 symbolizes such a scalability of the location.

After locating the client on the globe one needs to translate the coordinates into a real world location. If the system were built on a very extensive database, the translation would result to a precise location of the user. This would help the system in providing truly location specific information.

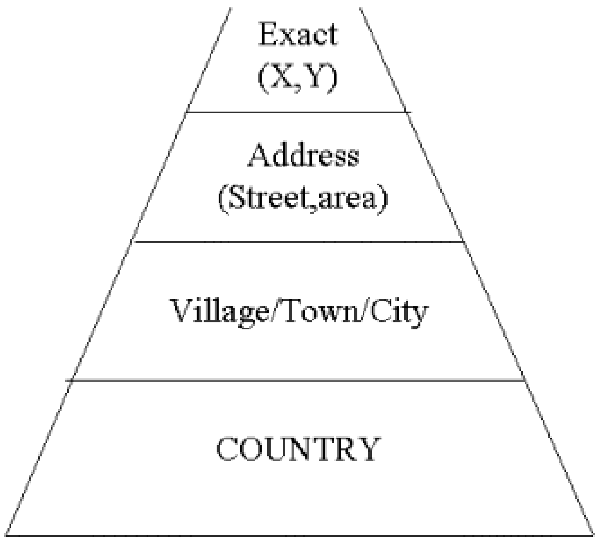

**Figure 2**: Location Triangle.

## 4. DATA EXTRACTION: TASK ORIENTED SEARCH OF WEB

The effectiveness of the system depends upon the capability of the system to effectively mine information from the World Wide Web to solace the needs of the users. When a web user queries a search service for Indian restaurants in Helsinki, the search engine lists links of the pages of interest by various methodologies [3]. It is not our purpose to improve on any of the present search technology but the goal is to extract scalable information from any given search services. A mobile user is restricted in time and bandwidth domain. Hence, the system should do the (location based) back end search of the information, and provide the user with just the required stream of data and not the entire web page or links to it. Hence our system leverages the capabilities of a search engine and picks up required information from the web pages linked by the search engine, as a result of well-formed search words.

### 4.1. Meta service provider

The goal of the data extraction is to provide the user information service that is not restricted to any of the existing service, but that is still easy to use. The main ideology of such service is denoted as Meta Service Provider, as described in Figure 3. The core of the service can be a fixed service or a set of services, but the framework is widened by the inclusion of the location, user profile, and the free Internet.

It is a big challenge to implement such meta-service provider and we merely try to define the concept and outline possible solutions. The data extraction from the World Wide Web itself is a demanding data mining task. Once that is achieved, we need to provide the information to the mobile user in a form easy to use. Data should be formatted so that it can be readily passed to a mobile user. The result should be offered to the user without lengthy feedback-query loops.

To sum up, the system should find the relevance of the given information on the basis of limited feedback. Evidently we need highly sophisticated data mining techniques, but it is likely not to be sufficient alone. Thus, we expect some semi-automatic preprocessing of the information and dynamic build of semi-fixed databanks.

### 4.2. Fixed providers

Reliability of the system greatly depends on the information it provides. There could be umpteen approaches for mining information from the Internet. One approach would be to identify a service provider on the web (for instance, a yellow pages web site like www.yell.com) and build an application that queries and mines the database as and when required.

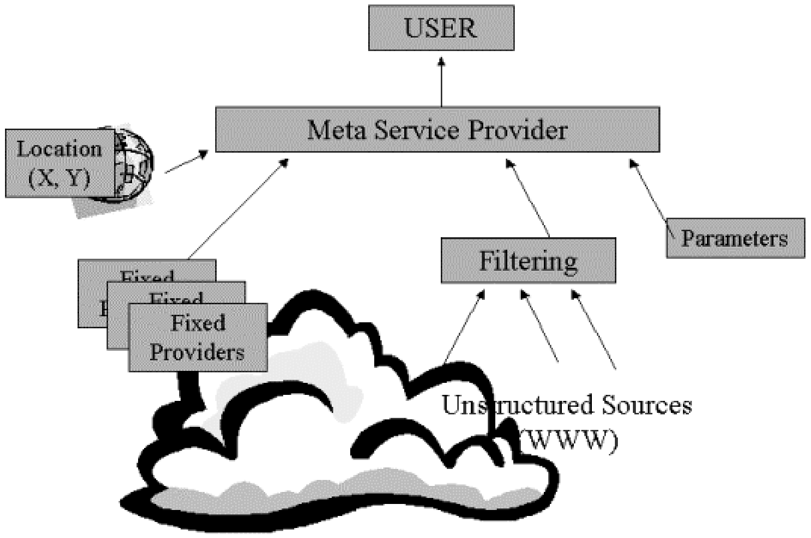

**Figure 3**: Summing up the entire Schema.

Since the structural properties of a particular website is uniform, programs can be implemented to parse the web pages in the site and extract necessary information. Varied approaches can be used to do the same. One method would be to code agents that would fill up a form on the website and acquire necessary information. Another approach, which we have used, is to query the server by constructing strings based on the contemporary requirements, and then parse the target web page for information.

We implemented a code that takes coordinates of the user as input and extracts the information from a web based yellow page, http://www.yell.com. The web site is a compilation of all major hotels, restaurants, pubs, hospitals, pizzerias etc in the UK. The site also provides maps with four zoom levels, to reach the target place. Implementation details are in section 6.

Thus, with just takes the location coordinates of a client device, in the UK we extract

1) Addresses and telephone numbers of nearest pizzerias, hotels, restaurants.

2) Corresponding maps with 4 zoom levels.

Such an approach confines the domain of the client to the database of the fixed service provider. The limited amount of information restricts ad-hoc querying. What if the client happens to be a historian and wants to find out more about the ancients monuments near his contemporary location? His ad-hoc query would fail, since YELL does not provide such information. Hence, to improve the intelligence of the scheme, we cannot just restrict ourselves to fixed providers.

## 4.3. Unstructured sources

Data mining provides a solution to our fix. The web can be mined to address the queries and needs of the user in a more reliable manner. A data mining approach can evidently support ad-hoc querying since the underlying extensiveness of the WWW removes all restrictions. A meta-service-provider of the kind discussed earlier in the section, queries the database on fixed service providers and stores information. Apart from this it also looks up the web for answering ad-hoc queries.

We have implemented a scheme that mines the web for addresses of places according to the arbitrary requirements of a mobile user. With such a scheme in place, it is evident that the users need not be restricted. Now, with the domain as big and wide as the web, a historian on the move can find out more about the forts near his contemporary location.

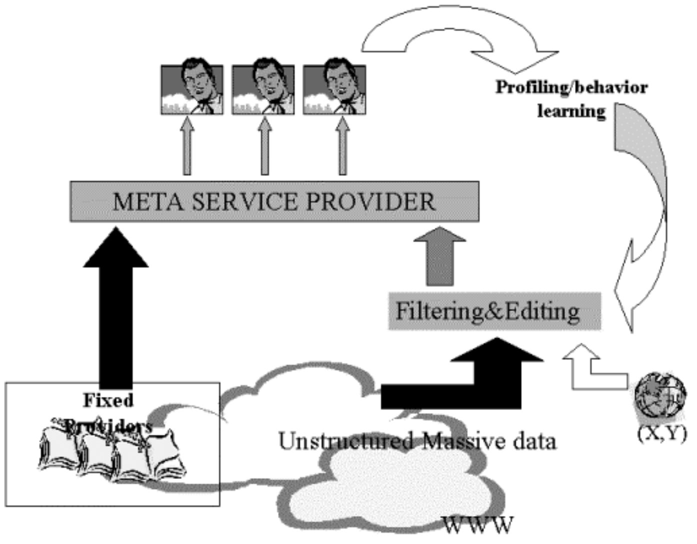

**Figure 4**: User Profiling for Personalization.

Since a mobile user is usually restricted in the time domain, we need a system that does not take a lot of time in mining information after the requirements have been specified to it.

A scheme could use such a program with certain modifications to lazily crawl the web for information and store the Knowledge-concentrates. When a user requests for some information, first the system looks into its database for the availability of data. If there is a hit, the information is provided to the user as it is. If there is a miss, on-the-fly mining of data takes place. With the disk to cost ratio ever increasing, we suggest that the mined information (now onwards called, knowledge concentrate) be permanently stored. Storage of all mined information is inevitable for faster access to information.

In a perfect situation, the user is provided with information from the knowledge-concentrate from the disk. The time taken to address the needs are minimal in this case. Hence it would not be ideal if we have some lazy data mining being carried on by the server. Some human annotation could possibly give the required direction to the server to carry on the lazy mining.

Evidently, if the request is from a client in a heavily habited place from where there are frequent requests, the value of x would be more and information is easily passed to the client. But in cases where requests come from unexplored or un-ventured areas, on-the-fly mining has to be done.

## 5. USER PROFILING

User Profiling could be defined as individual properties of the users that allow the service provider to define the relevance tailored for each person as an individual. In this section we discuss the issues involved in the same.

### 5.1. Background

User Profiling in the field of Personal Navigation has to additionally define properties of the user that can define relevance of information for an individual. User Profiling here is a data mining process aiming at intelligent user data segmentation, i.e., clustering of the data. The emerging models (customer profiles) have to be represented in a way, which helps in the understanding of the customer domain better by using various advanced visualization techniques. Such segmentation results can be exploited for example in the planning of new services and interfaces, modifying and personalizing existing services, and in direct, personalized marketing.

Creation of user profiles can be achieved in three levels in increasing order of complexity: Keywords manually typed by user: The most basic level is where the user specifies the query string and we generate response accordingly. Form-based static identification: The second option is that we have a form, which stores users' preferences and we use that information. Automatic learning: The third and most complex option would be to use a non-invasive learning approach to building user profiles.

### 5.2. Keyword based profiling

The first approach, keyword-based profiling, bases itself completely on the user and does not store any user profile or preferences. As an example, the user would have to specify that he wants addresses of restaurants in downtown Helsinki.

### 5.3. Static profiling based on forms

The second approach, static form-based profiling, is an invasive way for handling profiles based on information provided by the user on his preferences. The user fills out a form at the time of registration, which gets stored as his profile. This profile can have users areas of interest (like discos, pubs, sales etc) age, profession, whether he is/is not a frequent flier etc. Then the user can be given information unique to him or his group of users (say historians). The snag in this type of profiling is that it is static. Obtaining the user profile in the beginning of his session, leads to a static environment. One can never predict the mutability of ones behavior.

### 5.4. Automatic learning

The third would be the ideal choice in case we want to provide value added services to the user. The user creates a basic profile and then on, the system automatically learns from the user behavior. This kind of automation can be approached in two ways: By keeping track of user's past behavior and drawing inferences from that behavior. By clustering the people according to their usage of the service and defining content for each cluster.

#### 5.4.1. Individual user profiling

This is the layer in which individual interests and information are stored. The knowledge concentrate that exists in this layer is the result of mining of data that is generated by the individual experiences of the users. As an instance, consider a historian who visits different cities to explore their historical importance. His profile should reflect the same. Similarly, the profile of a world traveler should reflect his habits, viz. preferred class of hotel, etc. A home user should have a profile, which reflects the same. It would be of general importance to maintain useful information regarding the frequently visited areas, device used and others too.

Mining of user behavior could be used for added functionality. If a user visits a particular store every weekend, we can provide him with information about special offers etc in that particular store. Or, if a user takes a particular route to office everyday, we could target more on the shops lying in the area he passes through. Such knowledge concentrates could be used to build the confidence of the user in the system.

#### 5.4.2. Clustering of users

This layer maintains the knowledge concentrate that reflects clusters of users based on various factors. For example a cluster could be formed of home users residing in Helsinki area using PDA's, another cluster might be of people who query for clothes' sale on a regular basis. We can then provide value added services to the user by performing a look-ahead search to usual queries made by them. Additionally we can provide them with information about the area of their interest (a PDA user in Helsinki would like to know about add-on products for his PDA).

## 6. IMPLEMENTATION

In this section we discuss the implementation carried on by us based on the theory and design goals discussed earlier in the paper.

### 6.1. Coding the information extractor for a known provider

The program calls a method to find the city/village nearest to the given coordinates. The method uses a text file that lists the city names and their coordinates (latitude and longitude).

### 6.1.1. Information extraction

A query string is then constructed according to the information that has to be extracted.

For example if HOTELS AND INNS in Berkshire have to be found and their maps and addressed extracted, the query string would look like.

http://search.yell.com/search/....&companyName=&businessType=HOTELS

The query string returns the page with the address of hotels and links to the site providing their maps.

Since the pattern of organizing information is ubiquitous for the web site, a generic code was written to extract the names, address and telephone numbers of hotels and the maps to reach the hotels.

The public class implements the HTMLEditorKit.ParserCallback class. We override certain methods (viz. handleStartTag, handleEndTag, handleComment, handleText) according to the properties of the pages at www.yell.com. Flags indata, inrow, intable and inA are set if certain conditions are satisfied. Eg. inTable is set if a ¡table¿ tag is encountered in the stream from the web page.

Thus, by studying the uniform structural properties of the web pages of a site (here, of YELL), we extract information when certain conditions are fulfilled.

### 6.1.2. Map extraction

Attempts have been made to dynamically pass maps to a mobile user [12, 13]. We have attempted nave approaches to obtain maps from a fixed provider such as YELL.

Maps on YELL correspond to the pin-code of the area the hotel is located in and are present in zoom level asked for. The naming of the files is also uniform.By constructing the name of the corresponding map file with the pin-code information and the zoom level required, we directly access and download the map file from the directory which hosts these files (http://maps.yellowpages.co.uk/static/)

## 6.2. Ad-hoc querying for reliability-coding the information extractor for the web

We found no need for re-inventing new search techniques for mining the web. Research in hypertext content analyses has led to standardized web mining tools [1][3][4]. In our implementation, we leverage on the capabilities of Google [1], a standard search engine. We construct queries on the fly and pick up the first 80 links to find out information. For implementation purposes, we demonstrate the extraction of addresses of hotels near to the location of the user on the globe. As done earlier, we call a method to find out more about the clients contemporary location (village/city), from his coordinates. The locality name is passed to another method that generates a search string for Google, of the form:

http://www.google.com/search?asq=hotels+"+city+"&num=100&btnG=Google+Search&asepq.......

where city has the value of the city under consideration.

First 80 links given by Google are taken one after the other, and the web pages are searched for addresses. Addresses are located by their structural properties.

In our implementation, we store the addresses thus mined out of the web in a database. Such a scheme can thus lazily crawl the web for location specific information. Once the need arises, data is streamed to the user according to his convenience. Alternatively, we could mine the web in real time and provide a stream of data to a user on demand.

Methods can be written on the same lines to extract other information from the web pages.

## 7. CONCLUSION

In this paper, we have discussed our research initiation towards location-based information retrieval. Location is recognized as a new dimension to the relevance of information on the World Wide Web. We have discussed various issues in task oriented data mining of the Web with relevance feedback and have suggested solutions to the same.

## ACKNOWLEDGMENTS

We are highly grateful to Dr. SK Gupta, Department of CSE, IIT Delhi, for his valuable guidance and discussions on the topic. We acknowledge National Technology Agency of Finland (TEKES), Kata Electronics, Benefon Ltd, Tikka Communications and Arbonaut Ltd, FINLAND, for supporting the work with the project "Dynamic use of maps in mobile environment".

## REFERENCES

1. S. Brin and L. Page, *The Anatomy of a Large Scale Hypertextual Web Search Engine*, Proc. 7th World Wide Web Conference, Elsevier Science, Amsterdam, 1998.
2. S. Chakrabarti, B.E. Dom, S.R.Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg, *Mining the Web's Link Structure*, IEEE Computer, vol. 32, no. 8.
3. S. Chakrabarti and G. Hariharan, *Web-Search gets Personalized*, Vivek, National Center for Science and Technology (NCST), India. pp3-16, Vol 13, No.2
4. S. Chakrabarti, M. van den Berg and B. Dom. *Focused crawling: a new approach to topic-specific web resource discovery* Computer Networks, 31:1623-1640, 1999. First appeared in the 8th International World Wide Web Conference, Toronto, May 1999.
5. Rakesh Agrawal. *Data Mining: Crossing the Chasm.* Invited talk at the 5th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining (KDD-99), San Diego, California, August 1999.
6. A. Nalli, *Software Agents in E-business*, BSc Thesis, Department of Computer Science, University of Joensuu, Finland
7. P.Dasgupta, N.Narasimhan, L.E.Moser and P.M.Melliar-Smith, (1999). *MAgNET: Mobile Agents for Networked Electronic Trading*, IEEE Transactions on knowledge and data engineering, 11 (4):509-525. July/August 1999.
8. D.Friel, (1999). *Electronic commerce on the Web: digital drivers for 1999*, Business Economics, January 1999.
9. C.C.Hayes, (1999). *Agents in a Nutshell–A Very Brief Introduction.* IEEE Transactions on knowledge and data engineering, 11 (1):127-132. January/February 1999.
10. M.P.Singh and M.N.Huhns, (1997). *Internet-based agents: Applications and Infrastructure.* IEEE Internet Computing, 1 (4):8-9. July/August 1997.
11. E.I. Ageenko, P. Kopylov and P. Franti, *Optimizing context template for compression of multi-component map images*, GraphiCon00, Moscow, Russia, 151-156, 2000.
12. E.I. Ageenko, P. Kopylov and P. Franti, *On the size and shape of multi-level context templates for compression of map images*, IEEE Int. Conf. on Image Processing (ICIP'01), Thessaloniki, Greece, October 2001. (to appear)
13. J.Syrjarinne, *Studies of Modern Techniques for Personal Positioning*, Thesis for the degree of Doctor of Technology, Tampere University of Technology, Finland
14. G.Hariharan and S.Mehta, *Location Based Information Retrieval Framework*, Thesis for the degree of Bachelor of Technology, Indian Institute of Technology, Delhi.
15. P.Franti, *Dynamic use of maps in mobile envirnment*, Project supported by TEKES at University of Joensuu, Finland.( http://cs.joensuu.fi//pages/franti/dynamap/ )