SUPPORTING SEARCHING FOR MOBILE CLIENTS

CHAN LUI

M.PHIL.

THE HONG KONG POLYTECHNIC UNIVERSITY

2005

The Hong Kong Polytechnic University

Department of Computing

Supporting Searching for Mobile Clients

Chan Lui

A thesis submitted in partial fulfilment of the requirements for the Degree of Master of Philosophy

November 2003

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written nor material which has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

__________________________ (Signed)

Chan Lui (Name of Student)

Abstract

In the near future, mobile devices (i.e. mobile clients) will become indispensable tools in all walks of life. As Personal Digital Assistant (PDA) and mobile phones converge and there is an increase in the number of wireless stations for computer network connections, searching using a mobile device will be one of the most popular modes of usage. There has been substantial research in effective means of entering queries using mobile devices for searching information. However, there has not been much research about the presentation and organization of information in the retrieval list. Therefore, we propose the use of hierarchical access structure to support searching using mobile devices and this study examines the use of hierarchical access structure to assist the searching of information in the retrieved documents using a mobile device, as well as identifying the preferred user interface feature sets. Since effective hierarchical access depends on the effective user interface elements (i.e. the access structure and user interface features), a survey was conducted to examine (i) whether users (i.e. the subjects of the survey) prefer the hierarchical or list access structure when they have information overloading situations like accessing

the retrieval list and (ii) what user interface feature sets are preferred by the users. It was found that the users prefer the use of hierarchical access structure for accessing information using mobile device (PDA). Also, the relationships between the user preference of the overall user interface and the user preference of those individual features of the corresponding user interface were found. Through the survey, six preferred feature sets were identified. In addition to the survey, a Contextual Inquiry was carried out to find out the user requirements of using mobile devices (specifically PDA) for searching. After running the experiments for Contextual Inquiry, an affinity diagram was used to collect and organize the qualitative observations and feedback from the subjects. A search behavior flow chart was also obtained as a summary of user searching behaviors. Based on the results of the survey and the Contextual Inquiry, a hierarchical access interface was designed. An experiment was carried out to measure whether the list access structure based on Google's PDA version is more effective than our hierarchical access by using TREC-6 queries for the English information retrieval. We found that hierarchical access is preferred and is more effective under specific circumstances over list based access. In practice, we believe that hierarchical access can be a complimentary rather than a competitive alternative to list access, so that the navigation of the retrieval list can be improved as a whole.

Publications

1. Chan, D.L., Luk, R.W.P., Mak, W.K., Leong, H.V., Ho, E.K.S. and Lu, Q., Multiple Related Document Summary and Navigation Using Concept Hierarchies for Mobile Clients, *Proc. ACM SAC 2002,* pp. 627-633, 2002.

2. Chan, D.L., Luk, R.W.P., Leong, H.V., Ho, E.K.S. and Lu, Q., A Preliminary Study of User Interface Design Issues for Mobile Web Access, *Proc. WWW11 Mobile Search Workshop,* pp. 4-14, 2002.

3. Chan, D.L., Luk, R.W.P., Ho, E.K.S. and Lu, Q., A Preliminary Study on Multiple Documents Access via Mobile Devices, *Proc. The Second Human.Society@Internet Conference*, pp. 116-127, 2003.

Acknowledgements

It is my pleasure to acknowledge those who have given me support and advice for the accomplishment of this research.

I am grateful to my chief supervisor, Dr. Robert Luk, for his constant advice and support throughout the research. I must also thank my co-supervisor, Dr. H.V. Leong, for commenting on and supporting my research. Their professional comments and advice guided the direction of my research and this thesis.

Special acknowledgment also goes to the subjects and volunteers of this research. Most of the experiments in this research were done with their help and contribution.

Table of Contents

Chapter 1

Introduction

1.1 Mobile Information Access

The use of mobile equipment and the Internet is becoming popular. Thus, their integration is inevitable. According to a survey, which was about mobile phone usage, conducted in Tokyo in February 2001, approximately 30% of mobile phone users used the browser function of their mobile phone at least once per week (Figure 1.1).

In several years' time, it is believed that there will be more people accessing the Internet using mobile devices (e.g. PDAs and phones) than conventional personal computers (PC) [1, 2]. In a wireless e-commerce workshop held at Dalhousie University in April 2000 [1], it was shown that the number of mobile subscribers in 1999 (i.e. 428 million) already greatly exceeded that of Internet users (i.e. 241 million) and the market of mobile data services will explode to $110 billion by 2010. In Asia-Pacific, NTT DoCoMo has announced that 21.75 million of their users had subscribed to the mobile Internet service as of March 2001 [3]. As for the global market, by 2005, wireless analyst firms predict anywhere from 500 million to 1.4 billion people who will use mobile devices to access the Web [3]. Rather than just pointing to the future, this wireless explosion happened in the first quarter of 2001. The number of users of the mobile Internet surpassed 40 million (about 34 million users in Japan, 4 million in Europe, and 2 million in the USA) [3]. According to another survey [4], in December 2000 - February 2001, conducted by Taylor Nelson Sofres focusing on Asia-Pacific areas, 57% of the adult population have a mobile phone (where Hong Kong has the highest mobile phone ownership: 83%) (Figure 1.2) and 14% of adult population have a WAP [5] phone (i.e. mobile phone supporting Wireless Application Protocol) (where Japan has the highest WAP phone ownership: 44%) (Figure 1.2) and 34% of WAP phone owners access the Internet using their phones (where Japan leads this at 62% - followed by Hong Kong - 25% and Korea - 17%) (Figure 1.3)

New technologies, such as WAP, enable users of mobile devices to access Internet for information and allow mobile knowledge seekers to access information on the Internet during a meeting or on the road, while away from their desktop [2,4,6]. Clearly, current and upcoming use of mobile devices and services can provide information anytime and anywhere via Internet as the number of people who access the Internet using mobile devices (PDAs, phones etc) nowadays is increasing (Figure 1.4). One of the most important features of the Internet is the availability of the huge amount of information and Internet search engines are important tools for finding relevant information. However, most of the traditional search engines were designed for desktop computers while mobile devices have very different interaction characteristics from desktop computers. Owing to inherent constraints on mobile devices, traditional search engines are not suitable for deployment on mobile devices, which, in turn, cause difficulties for mobile Internet access. Moreover, wireless connections of mobile devices may be charged at a higher rate than standard Internet connections. Traditional information access and presentation structures (rich but not tidy in layout of information) use large amount of data for representing small amount of useful information. Such access and presentation structures assume the availability of communication channels of high bandwidth, which, in turn, results in costly use of mobile network resources. In order to meet the needs of the hundreds millions of users who access the Internet for searching information using their mobile devices, the ability to deliver timely, accurate, effective, affordable and concise mobile information access is very important. Therefore, a good information access structure is needed for supporting mobile searching (i.e. search for information using mobile devices) and this research was carried out by reviewing existing work on mobile searching and experimentally investigating the feasibility of our proposed "Hierarchical Access Structure" in supporting searching for mobile devices.

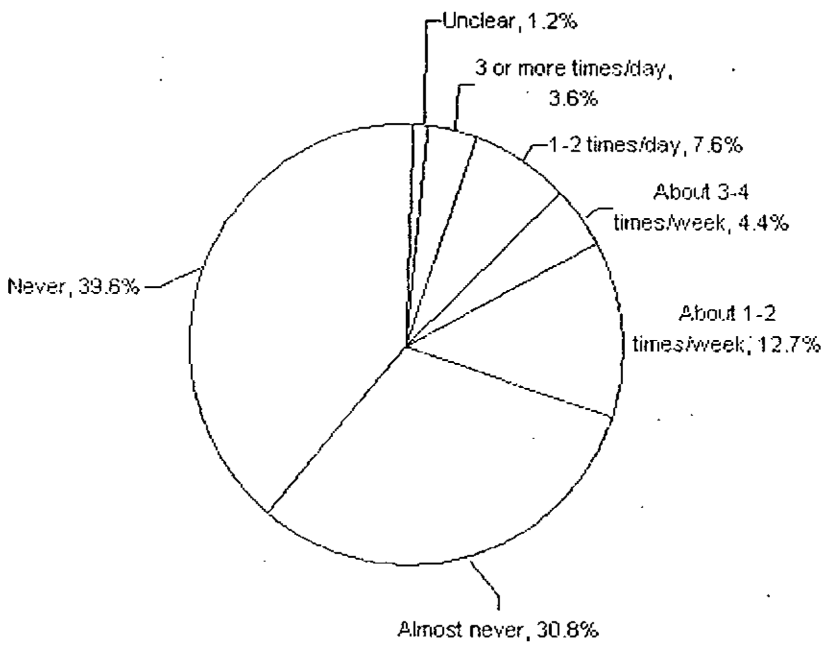


Figure 1.1 Percentage of usage of browser function of mobile phones [6]

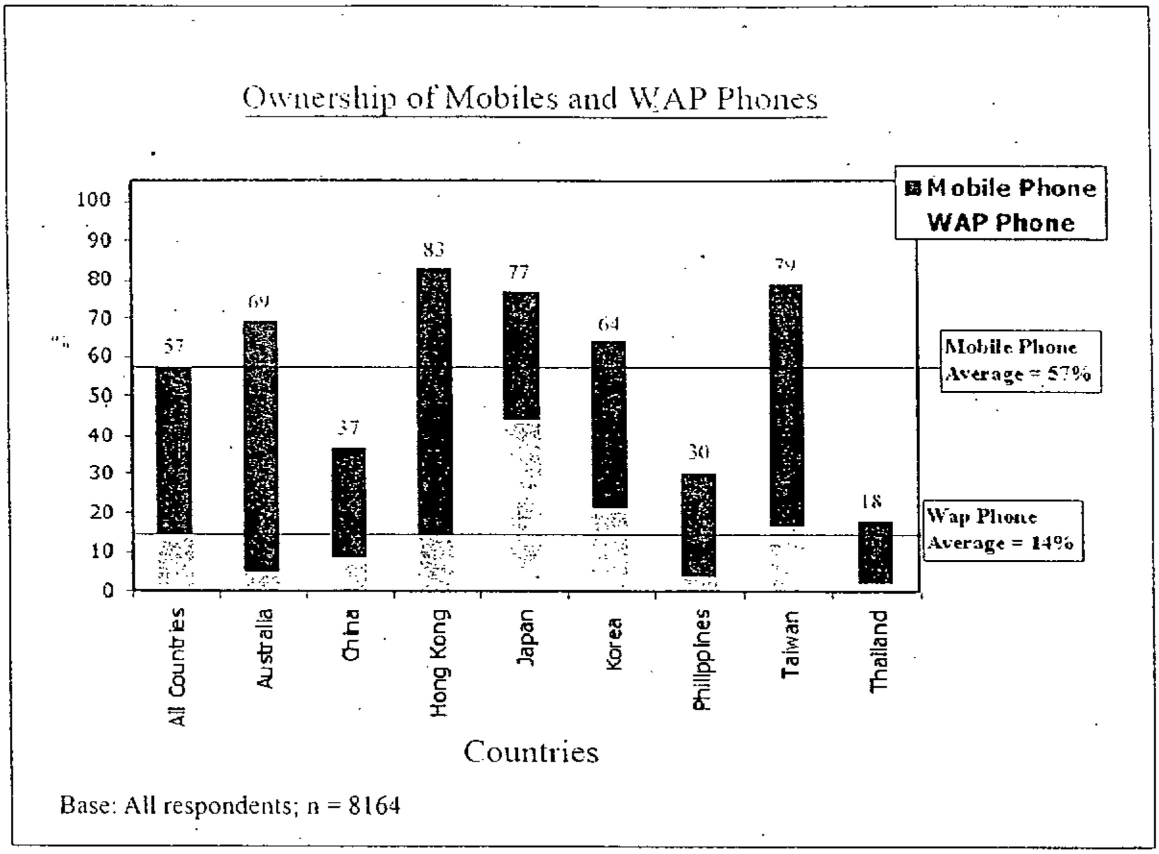


Figure 1.2 Ownership of mobiles and WAP phones [4]

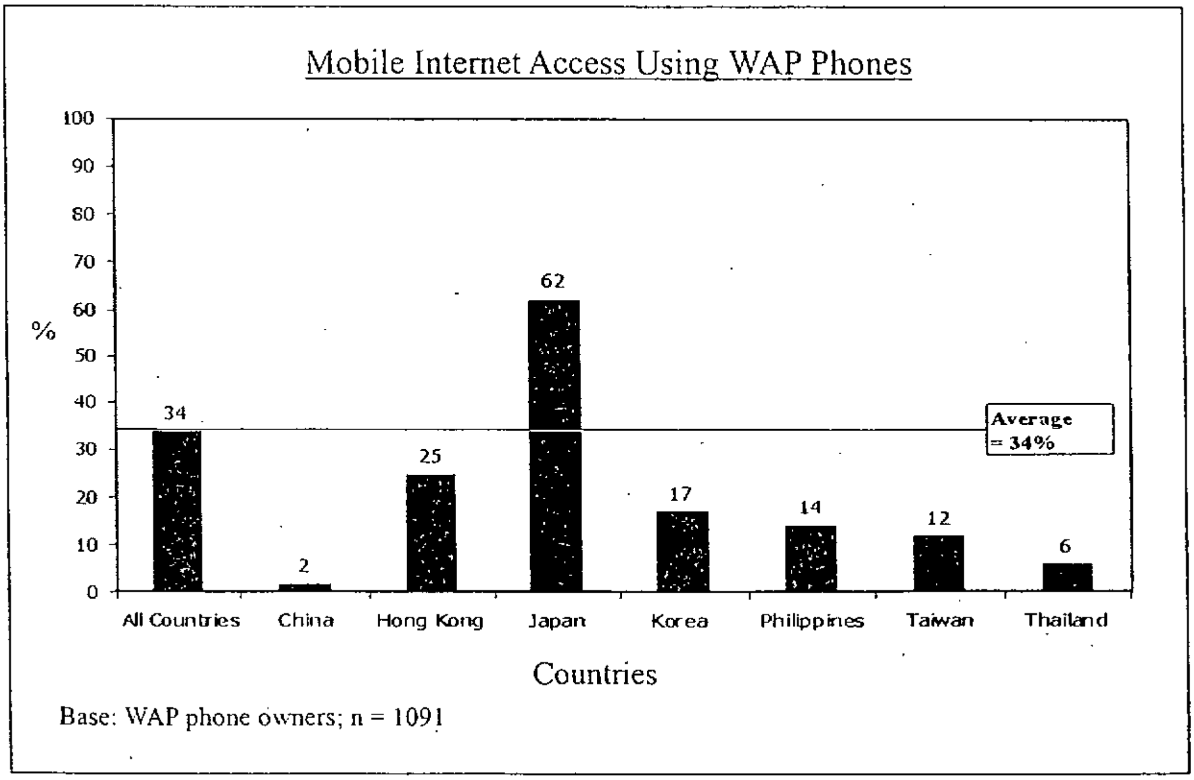
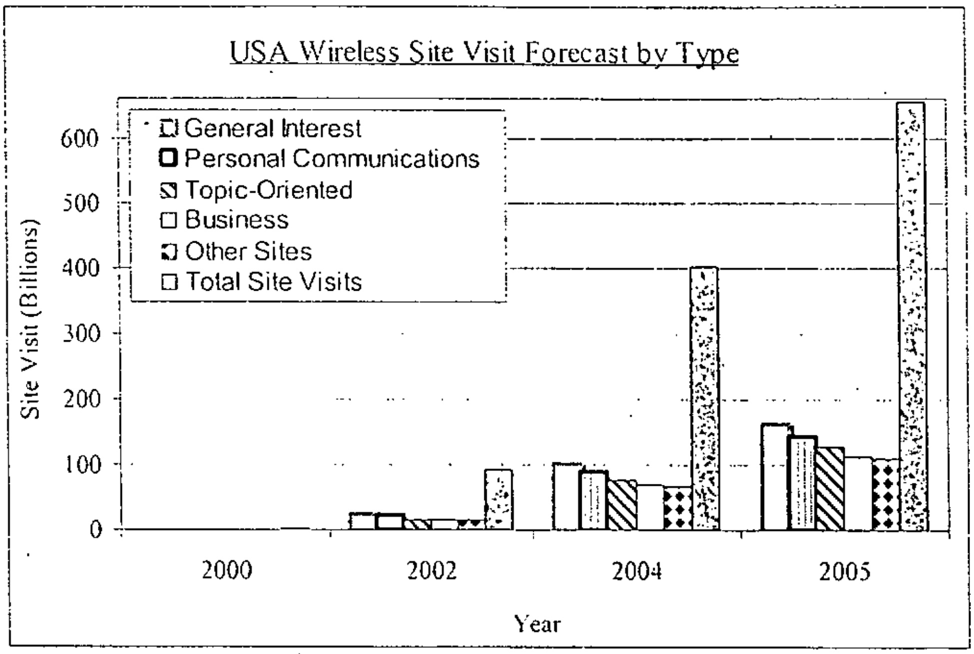


Figure 1.3 Mobile Internet access using WAP phones [4]



	Year			
Type of Site	2000	2002	2004	2005
General Interest	1.2	25	101.6	162
Personal Communications	0.8	22.3	90.2	143.8
Topic-Oriented	0.1	16.3	77.4	128.6
Business	0.1	15.1	68	111.5
Other Sites	0.5	14.2	66.2	109.3
Total Site Visits	2.7	92.9	403.4	655.2

Figure 1.4 USA wireless site visit forecast by type [3]

In the past few years, there were substantial interests in supporting mobile search to overcome the limitations of mobile devices. These limitations include small screen, limited bandwidth, limited computational resources and limited battery life. Much research work focused on the formulation of queries, inventing more intelligent or sophisticated input mechanisms for mobile devices to overcome those limitations. We, however, focused on the less investigated area of accessing retrieval results using mobile devices. Typically, there are hundreds, if not thousands, of retrieved documents that users can browse. These documents are essentially organized in a list access structure, with paging to jump to different parts of the list (e.g. Figure 1.5). The list access structure is highly effective when the retrieval effectiveness (i.e. precision of retrieved document set with respect to the query) is high but the list access structure is not effective when the retrieval effectiveness is low. This is particularly troublesome for mobile search because many Internet users pose short queries that are more likely to have low retrieval effectiveness and because the re-formulation of a query is much more difficult using a mobile device than using a desktop computer with a keyboard. Therefore, there is a need of a more effective access structure than the list for browsing search results.

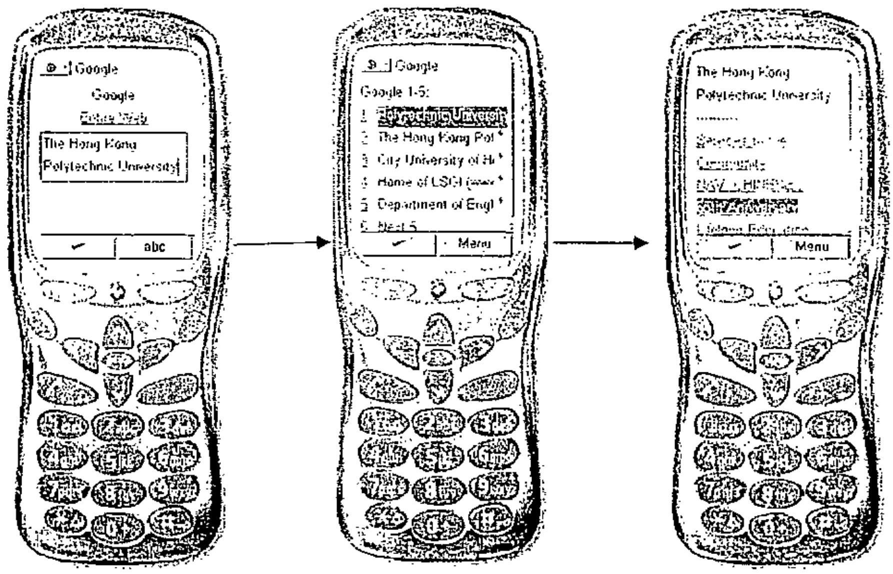


Figure 1.5 Sample search from Google mobile

The underlying problem of browsing the set of retrieval results is essentially the problem of information overloading, which is particularly pronounced when users are using mobile devices. In the past, hierarchical access structure was used to mitigate the problem of information overloading. For example, the Yahoo! Web site uses a manually-created hierarchy to guide Internet users to find the relevant Web sites of their interests. We believe that the use of hierarchy can also help in retrieval results organization. In particular, we are interested in the application of organizing the retrieved results into a hierarchy so that the mobile device users can effectively access the retrieved results using the hierarchical access structure.

1.2 Research Objective and Scope

The objective of this thesis is to explore the feasibility of applying hierarchical access structure for supporting mobile information access based on the following factors:

1. User preference
2. User requirements
3. Information access performance

The focus of this research is as follows:

1. Our analysis concentrates on the issues about PDA. This is because PDA is one of the most common types of mobile devices nowadays and it was designed for organizing information as well accessing to information.
2. We focus on addressing issues of accessing search result. This is because retrieval of information is mostly achieved via search engines which deliver services to desktop computers and mobile devices in a similar way. However, effectively accessing search results from a mobile device and a desktop computer may be very different in nature.

1.3 Organization

In this thesis, we introduce our research topic in Chapter 1 and present a literature review about this research in Chapter 2. Then, we explore issues on human preference and requirements on supporting mobile searching with hierarchical access structures in Chapter 3. We further analyse the information access performance of hierarchical access structure for supporting mobile searching in Chapter 4. Finally, we conclude this thesis in Chapter 5.

Chapter 2

Literature Review

In this chapter, we review various approaches on providing mobile searching services. These approaches can be roughly classified as pull-based searching approach and push-based searching approach. We further explore issues on different approaches of mobile information access and explore works related to our proposed hierarchical access structure for mobile information access.

2.1 Searching Approaches

2.1.1 Pull-based Search Engines

With the rapid increase in demand of mobile information searching facilities, search engines are extending their Web search services to mobile devices. Google [7] has a good feature that allows those using WAP browsers to use a special version of the search engine where the search results are formatted for small screens. In addition, when a user visits any link, Google continues to convert HTML into a WAP format on the fly [8], making it easier to view the Web while the user is using a mobile device. However, the HTML to WAP (i.e. HTML to WML) mechanism does not seem working well with some complicated HTML pages.

Google mobile is a typical pull-based retrieval system, which is always waiting for a request (i.e. query) from users and searching for "answers" to the query. Similar to Google, many search engines, which are designed for mobile devices, like Waply [9], Excite Mobile [10], Fast Search [11] and WannaWAP[12] are pull-based. One of the advantages of such search engines is that users can specify their request by posting a query. However, the disadvantage is that users have to input a query. Studies on query expansion [13, 14] shown that a richer query would likely be able to retrieve a better result that has less irrelevant information. This is important for mobile devices as it is difficult to browse a bulky retrieval result set with a small display, while the limited input capability of common mobile devices makes the long and detailed query input process annoying.

2.1.2 Push-based Web Directories

Other than pull-based search, there is another major approach on information searching – push-based search, which is different from the pull-based search in that requests are not issued by a user directly. Instead, information is well-organized and available for users to search. A typical example is news portals for mobile devices [15, 16, 17, 18, 19] where news is organized with a directory and users can search for their interested news in the directory. Another example is mobile directories like mobile MSN from Microsoft [20] and mobile Yahoo! [21]. Clearly, push-based search does not suffer from problems with query input while the central problem of this approach is the access user interface and the organization of the delivered information [22, 23, 24, 25, 26, 27, 28, 29]. Also, such Web directories are usually built manually. The creation of manually-created hierarchy consumes large amount of manpower and it is difficult to adopt for the huge amount of information on the Internet.

2.1.3 Comparison of Pull-based Search Engines and Push-based Web Directories

By comparison, information in the push-based approach is often organized in a hierarchical access structure, while a search result from the pull-based approach is organized as a list of indexes.

A recent study shows [30] that only few users (8%) are very confident in what they have to search. It is likely that browsing information delivered from a push-based server is an excellent way to begin a search for most users as they do not need to know the "right" keywords to find what they are looking for. However, browsing is inefficient and time-consuming if users know exactly what they are searching for. A pull-based system, such as search engines, is a far better choice for precise search.

It is necessary to point out that the list structures (i.e. search result) of pull-based systems are usually created automatically whereas the hierarchical structures of push-based systems are painstakingly assembled by human beings. The former is benefited from the full power of the computer resources used to create their indexes of the information, and therefore are generally orders of magnitude larger in scope than push-based systems created by human beings. That means a pull-based system will be a better choice if one is looking for comprehensive and exhaustive results. A directory (hierarchical access structure), on the other hand, will serve users better if they want a selection guide, limiting the results to only the best or most useful Web resources. However, the push-based approach is inherently more biased than the pull-based approach, simply because they are usually compiled by human. A pull-based system such as search engines, on the other hand, tends to be less biased. It might be highly desirable to have a mobile search and access mechanism that comes with the advantages of both pull and push-based searching.

2.2 Supporting Pull-based Mobile Search

We can identify four major components for supporting information searching, namely, querying, retrieval, access and feedback (Figure 2.2.1).

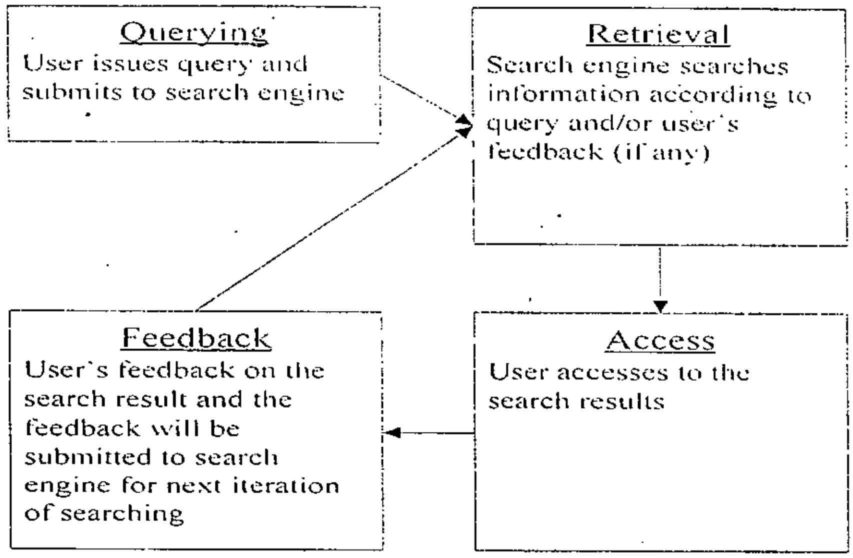


Figure 2.2.1: Four major components in supporting information searching

2.2.1 Querying

A query is a string of words that characterizes the information that the users seek. It is the first step to start a search, which is issued by the users. It is possible that query processing can significantly improve retrieval efficiency. Much work has been done in this area.

According to a study on queries logged by Alta Vista [31], the common length of query is one to two words. Clearly, information provided by the initial query is not enough for accurate information retrieval. Query expansion has long been suggested as a technique for query processing in information retrieval. One of the earlier studies was carried out by Jinxi Xu and W. Bruce Croft [32]. Their approach is to use the corpus to discover word relationships (global information) and document retrieved by initial query (local information) in order to reformulate the query for the next retrieval. Another approach on the topic of query expansion where the users take part in the process of query expansion is known as relevance feedback. Mark Senderson's CiQuest project [33] and Efthimis N. Efthimiadis's Interactive query expansion system [30] are examples of this approach.

To be more specific to address the query issues on mobile searching, a query processing model for mobile computing using concept hierarchies and summary databases was proposed by Sanjay Kumar Madria, Mukesh Mohania and John F. Roddick [34]. There are also two similar approaches: an agent-based query expansion model by using concept hierarchies proposed by Scott Parent , Bamshad Mobasher and Steve Lytinen [35] and a query transformer from Sang Goo Lee [36]. Owing to the inherent limitation of input capability of common mobile devices, Motorola proposed an Intelligent Keypad Text Entry Technology for Embedded Systems – iTAP [37] which comes with an intelligent keypad text entry method to allow a user to press only one key per letter. Input speed, accuracy, and overall usability can be increased as the system predicts the most likely word associated with a particular sequence of key presses. However, the above input methodology predicts the user input at word level only. IBM Research Lab 's Yariv Aridor, David Carmel, Yoelle S. Maarek and Aya Soffer enhanced the above approaches to the entire query level by suggesting optional terms for completing and disambiguating queries [38] while users still have to input by using annoying key press. No doubt, poor input capabilities are still troubling the submission of searching request. Further enhancements of technologies are necessary.

2.2.2 Retrieval

All information retrieval (IR) systems search for relevant documents. It is excellent, if an IR system is able to retrieve all the relevant documents and exclude all those irrelevant. Clearly, one problem to such an ideal target is the issue of predicting which documents are relevant and which are not.

One possible way is the use of Boolean retrieval model. A Boolean based search strategy retrieves those documents which are 'true' for the query. This formulation only makes sense if the queries are expressed in terms of index terms or keywords and combined by the Boolean operators. Unfortunately, the Boolean based search suffers from a series of drawbacks. Firstly, its retrieval is based on a binary decision criterion without a notion of grading scale, which usually results in a bad retrieval performance. Secondly, substantial training is needed to enable a user to formulate good Boolean queries [39]. Owing to such drawbacks, Salton and Wu [40] proposed an extended Boolean search model in 1983 that combined the idea of similarity measure in the vector space model and the Boolean logical implications in combining terms for retrieval. Despite the inherent drawbacks of Boolean based search, it is a search technique for the expert users who can write good queries to perform accurate search.

Other than the traditional Boolean retrieval model, another commonly used model is the vector space retrieval model. One of the reasons that the vector space model [39] was introduced is that the categorical decision of relevance of the Boolean retrieval model is too limiting. Different from the Boolean retrieval model, the vector space model measures relevance between a query and documents by assigning non-binary weights to index terms in query and documents. The similarity between the term weight vector of a query and documents are then compared.

In terms of mobile searching, Yariv Aridor, David Carmel, Yoelle S. Maarek and Aya Soffer proposed searching by anchor text [38]. Other than searching with full text, the size of the index of anchor text is relatively small and can be sent to mobile device for disconnected searching (i.e. users can perform local search with their mobile device and there is no need to connect to the remote server). In this way, the connection time and the costs can be reduced. However, updated information cannot be retrieved unless the client is connected to the server again. In addition, the storage of the indexes may be too large for mobile devices.

2.2.3 Access

Recent research in information access by mobile devices focuses on a number of diverse aspects. One basic aspect is effective browsing of individual Web pages using mobile devices. Buyukkokten *et al.* [41] tried to improve PDA navigation more effectively by deploying specialized link ordering. When browsing a Web page in detail, Buyukkokten *et al.* [41] also suggested the use of Accordion Summarisation, which consists of summarising a Web page into an expandable tree format similar to the directory display in Microsoft Windows Explorer, with other supporting facilities, e.g. automated view transitions. Other aspects to improve the use of mobile devices include reducing the size of pictures to fit into smaller screens of mobile devices [42] or presenting information in smaller chunks (e.g. [43]).

Given the recent interest of mobile clients in information access, there are relatively few works in summarising and navigating related or similar documents for mobile devices. There are some ways to summarise and navigate through multiple documents. The simplest way is the use of a list of documents [42] with a machine-generated abstract/extract using conventional summarisation techniques (e.g. [20]), as in many search result pages of Web search engines. The list format is manageable if there are, say, less than 10 documents for the users to browse. However, the typical number of documents in search results and the typical amount of news articles returned from multiple news portals, are much larger than 10.

Although it is possible to page through the list, relations between documents and relations between concepts are not explicit in the list-structure.

Another approach is to identify similarities and differences at the document or subdocument levels. Clustering techniques [44, 45, 46, 47] can be applied to identify similar documents. Recent work on summarisation (e.g. [48]) of multiple documents can delineate the differences and similarities between document fragments. While this is useful to the users, there are very little navigational facilities and navigational information available for the users to select the desired similar documents and similar document fragments. For clustering, it is possible that some hierarchical structures are used, but there is no specific navigational information in the hierarchy.

A more Web-oriented approach is to use the hyperlink structures to build a hierarchy of Web page relations (e.g. Navigational View Finder [49], WebToc [50] and WebCutter [51]). By means of some summary link data visualization, users can see where the concentration of Web pages is for a particular Web site. However, this hierarchy is based on link relations and some additional attribute values (e.g. author), which may or may not reflect the concepts that are found in the related documents. Instead of hyperlinks, other methods [52, 53] used text classifiers to classify Web pages into pre-defined categories. Chen and Dumais [54] have shown that human subjects liked this kind of interface and were able to find information faster (i.e. about 50%). However, this approach requires a set of training data (with category labels) which may not be available in some (Internet) applications. If there is a large amount of documents grouped in the same category, there is no navigational facility for these documents, apart from a ranked list of titles, which is not very suitable for mobile devices. Although it is possible to incorporate more and finer categories, this will affect categorization performance and it is difficult to know beforehand whether the set of pre-defined categories are large or small enough for a specific query.

Another similar approach is simply building concept hierarchies from document collections. Research work has evaluated different methods to build concept hierarchies, including those based on lexical properties [55] and subsumption [56]. As for our research, we choose to extend the subsumption technique for summarising multiple related documents because it is relatively simple to implement and it's performance can be improved (refer to Chapter 4, Section 4.2.5). Since the concept hierarchy in [56] was designed specifically for browsing the search results of queries from a mono-server, certain information available about the collection in a mono-server may not be available from meta-search engines or push-based information providers. Some adjustments to the current method of constructing concept hierarchy are needed.

2.2.4 Feedback

The word "feedback" is normally used to describe the mechanism by which a system can improve its performance on a task by taking into account the past performance. In other words, a simple input-output system feeds back the information from the output so that this may be used to improve the performance of the next input [57]. Rocchio [58] started the discussion of an implementation of feedback – the relevance feedback. Some other researchers, such as Hideo Joho and Mark Sanderson, investigated the use of concept hierarchies for relevance feedback where the concept hierarchies were generated from the retrieved document set and presented to users.

Users then select query expansion terms from the hierarchy and the system uses such expansion terms (i.e. feedback) for retrieving relevant documents [59]. Nowadays, relevance feedback is widely used for query expansion [30, 33, 34] in information retrieval.

2.3 Hierarchical Access for Pull-based Search

We propose a hierarchical access structure where information is presented with an automatically-generated hierarchical structure, called concept hierarchy, which is derived from the data of the retrieved results generated by search engines [56]. Concept hierarchies (Figure 2.3.1) are similar to Yahoo! categories and have the capability to summarise multiple (related) documents as well as supporting effective searching using mobile device by taking the advantages of pull and push-based searching approaches.

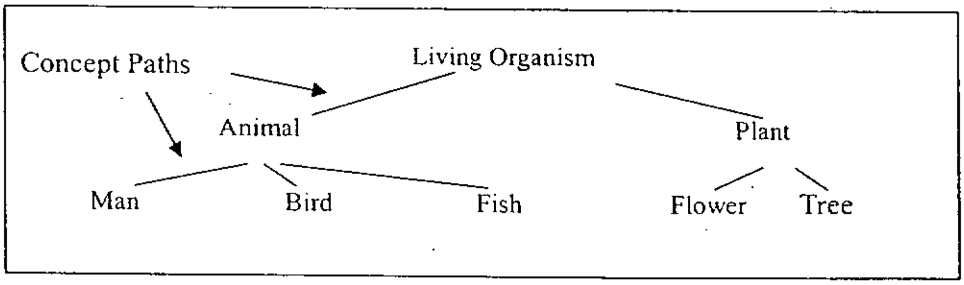


Figure 2.3.1 A typical concept hierarchy, where "Living Organism" is a general term for describing two more specialized terms "Animal" and "Plant". Similarly, "Animal" and "Plant" are general terms for describing the more specialized terms "Man", "Bird", "Fish", "Flower" and "Tree".

2.3.1. Concept Hierarchy

A concept hierarchy is a hierarchical representation of concepts from the most general to the most specific. For a typical concept hierarchy constructed from a set of documents, the hierarchy's member concepts are generally representing topical concepts of the given document set and thus the hierarchy summarises the documents with a set of concepts. Due to such inherent characteristics, concept hierarchies have the capability to support navigation and summarisation of documents.

2.3.2 Why Hierarchical Access?

The reasons of proposing the use of hierarchical access structure for supporting mobile searching are as follows:

Firstly, there are new challenges in information access by mobile devices, due to limited display area, limited navigation capability, limited battery life, high wireless communication cost and relatively slow CPU processing speed [60] of mobile devices. It is known that the formulation of searching queries with small display is difficult [23] and hierarchical access structure allows mobile device users to select the required information without submitting queries.

Secondly, problems with searching also occur when many documents are returned. Unlike interactive Internet browsing, users of mobile devices are unlikely to be willing to pay for the communication cost for interactive search. Instead, the users desire some kind of summary of the returned documents in order to help them to decide which documents are relevant before downloading the documents or traversing the hyperlinks. The investigation on the effective access of information using mobile devices has been an interesting topic [61, 62, 63, 64]. Since searching using mobile devices is not interactive, the search programs residing in the mobile support station [65] will try to produce a comprehensive search, typically by posting the queries to multiple search engines. This increases the number and the diversity of documents returned and a study [66] has shown that meta-searching can increase search coverage by as much as 16%. Faced with information diversity and overloading, summarising multiple documents with concept hierarchy can help users of mobile devices to select which subsets of the retrieved documents are relevant.

Thirdly, apart from user initiated (pull-based) search, news portals provide a large amount of up-to-date daily news, which represents a form of push-based [67] information search. However, not every news portal has the same coverage and with the same level of details. If mobile service providers can effectively summarise multiple related news articles, users of mobile devices can quickly assimilate a vast amount of related and similar news without the necessity to browse through all the news portals. Hence, making summary of multiple (related) documents with concept hierarchy appears to be a good value-added service to users of mobile devices.

Fourthly, the concept hierarchy fulfills the needs for selecting subset(s) of related documents based on the information from the concept path (Figure 2.3.1) of the hierarchy. Also, it is relatively simple to navigate through the concept hierarchy by selecting the child of each node or by pressing the backward key to reach the parent of the current node and humans are familiar with this type of information access without much training since concept hierarchies are commonly used for browsing and searching for documents (e.g. in Yahoo!, in library or in product catalogs).

Fifthly, the proposed hierarchical access structure is automatically generated from the search results of the search engines. It is benefited from the full power of the computer resources and therefore is able to handle large amount of information, like pull-based search engines, instead of the push-based directories created by human beings. That means the proposed hierarchical access structure will probably be a suitable choice if the user is looking for comprehensive and exhaustive results. Meanwhile, the proposed hierarchical structure serves users with selected guidance and navigation capacities. The hierarchical structure, ideally, limits searching results to only the best or most useful Web resources for user's information need. Also, the hierarchical structure is generated by computers automatically, which, in turn, reduces the bias of human compiled push-based Web directories.

Lastly, users can interact with the hierarchical access structure in order to reduce the scope of information search. As the concept hierarchy represents the retrieved document set, users can search the hierarchy for defining the scope of the desired document in the document set. As different selection at the parent level can result in different concepts shown to the users (because different parent concepts come with different child concepts), the concepts at the current level are affected by user's selection at the previous level. In this way, the system keeps interacting with the users. Such an interactive mechanism for accessing the document set can help users to identify the desired documents more effectively.

Chapter 3

User Preference, Requirements and Behavior Analysis

It is well known that mobile users are one of the major components of mobile information accessing systems. Thus, one of the major functions of mobile information accessing system is to serve human beings. Therefore, we believe that, human factors, such as user preference, requirements and behavior must be taken into consideration when we are analysing mobile information accessing issues.

3.1 User Preference

In this section, the human preferences on hierarchical access and the user interface issues of mobile information access are explored. The aim of examining the preferences and interface issues is to determine whether user interface designs have impact on hierarchical access and the feasibility of applying hierarchical access on supporting mobile searching based on users' preference and requirements. If the user interface designs have some impact on the preference of hierarchical access, then this study can gain some insight into the design of user interface for hierarchical access. The analysis of user preferences is carried out by using existing user interfaces of mobile content providers. This is because many mobile content providers are loaded with too much information to display within a single screen of a mobile device and some mobile content providers use the hierarchical information access structure to manage information overloading, similar to our proposed use of hierarchical access to access to retrieval results. Although the hierarchical access structures of mobile content providers are built manually and are typically static, they are the starting point and the solely available resources, which are close to our daily life, for our investigation into the suitability of hierarchical access on supporting mobile searching.

3.1.1 Experimental Setup for User Preference Survey

The experiment (i.e. survey) was carried out with a Palm IIIc emulator (Figure 3.1.1.1), running on a PC. We believe that such a substitution of a physical mobile device is acceptable in this case, as the experiment involved only the use of simple user interface features for item selection. Seventy-seven user interfaces (i.e. samples), including list and directory, were collected from sixty-four (offline) and thirteen (online) English news content providers respectively. These user interfaces were evaluated by 27 subjects (Appendix A.1, Subject 1 - 27) – 9 post graduate students, 17 undergraduate students and one research assistant. There were males and females aged between 20 – 26 and most of them came with some background knowledge of human computer interface design, and therefore the subjects can make appropriate judgments for answering questions about user interfaces. In addition, we need subjects who had experience with PDA and online search experience varying from modest to expert; more than half of them came with less experience on searching with mobile equipment and Internet browsing with PDA. These subjects reflect the most active population using mobile devices (Appendix A.2) and they are expected to continue to use mobile devices in the future.

Moreover, we pre-identified the feature set of each of the user interfaces and the pre-identified set of features was judged by the subjects. If the support and confidence of the subjects' judgment are higher than the minimum requirements, features will be added/removed from the pre-identified feature set (refer to section 3.1.2 for details)

Each subject was given a set of tasks (i.e. evaluate a set of user interfaces). For each of the tasks, the subjects were instructed to navigate throughout a user interface with the emulator and then complete a questionnaire (Figure 3.1.1.2) based on their preference of the user interface. The perceptions were measured with preference values (from 1 to 10) on several aspects, including their general perception on the access structure as well as the user interface and preference on features in the user interface. During navigation, the subjects' actions were automatically logged (Figure 3.1.1.3) for timing those actions (e.g. navigation time and evaluation time) but there was no time limit to complete the task.

Each content provider was evaluated by 20.5 subjects on average. According to the user background information collected, they are likely to have preference towards the use of catalog (directory/hierarchical access structure) on searching information but their attitude towards searching with a catalog and a list was positive in general, and there was no bias on the preference of hierarchical access and list access.

On average, our subjects spent approximately 2 minutes and 24 seconds to complete a task. About 50% (1 minute 11 seconds) of the evaluation time was used for navigating/browsing the given user interface of a particular content provider (Appendix B).

Figure 3.1.1.1: A Palm IIIc emulator

Subjects were asked to assign preference values of features which the subjects were sure that it was available on the user interface. The scope of preference values ranges from 1 to 10 where 1 for the least important and 10 for the most important.

Other related questions, such as the nature of user interface design (List/ Directory), subjects' experiences on content providers, general preference of user interface design and comments.

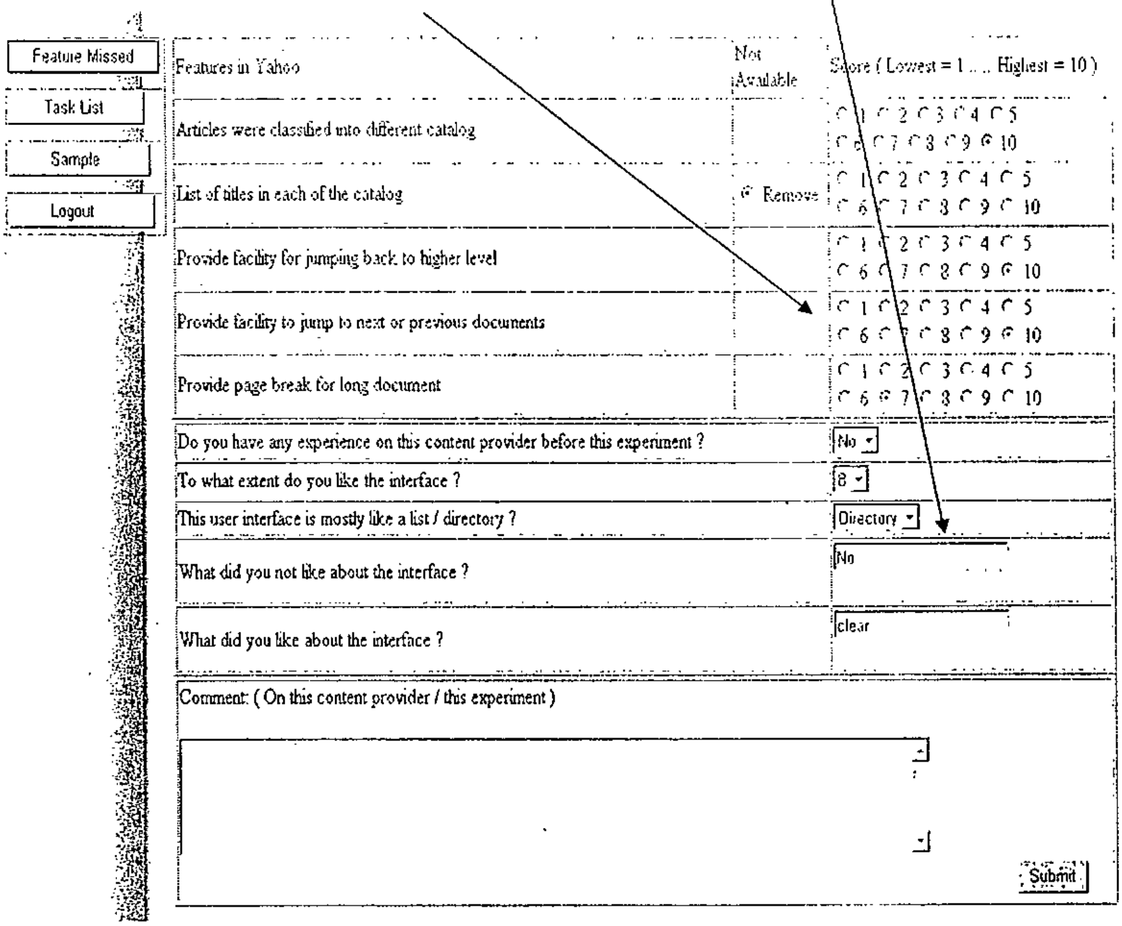
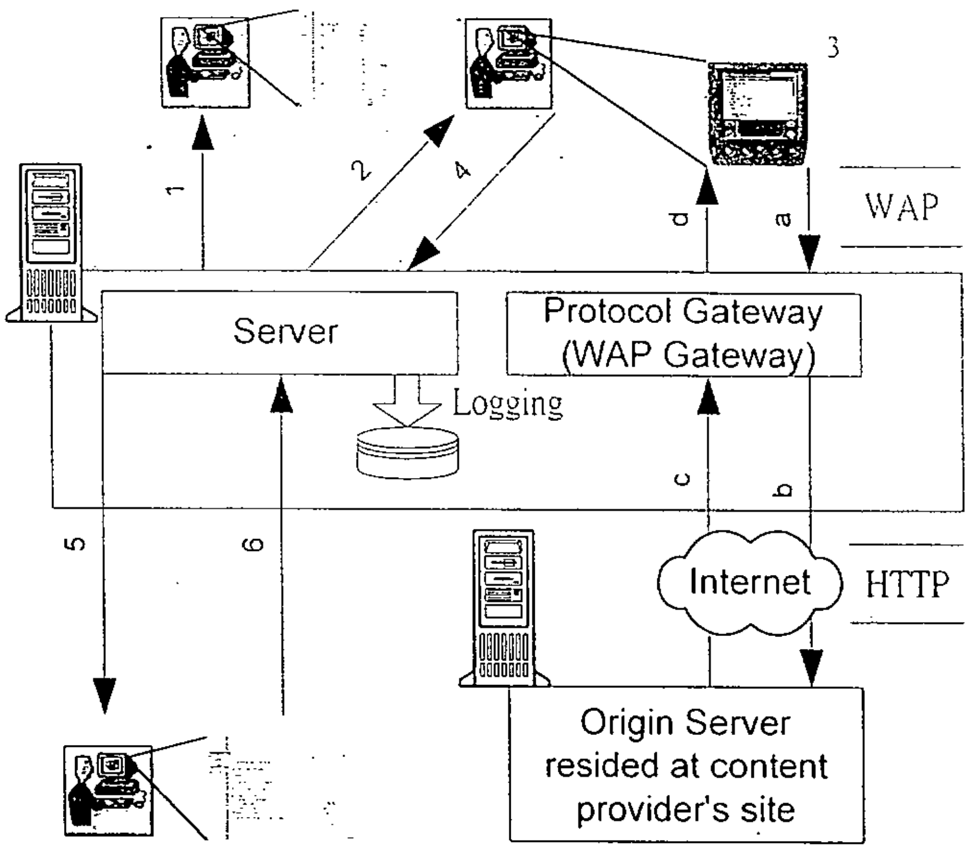


Figure 3.1.1.2: An online questionnaire about the user interface the subjects had browsed.



Keys:

1. Each of the subjects was assigned a list of tasks and they completed the tasks at their workstations.
2. For each task, the subjects were asked to download a PDA emulator from our server and used the emulator to browse a sample of user interface.
3. If the given sample of user interface was in online mode, the emulator communicated with the original server via a protocol gateway (WML Encoder, Protocol Adaptor):
    a. Encoded request
    b. Request
    c. Response (content)
    d. Encoded response
4. The browsing time will be sent to our server automatically and logged in our database immediately.
5. Our server sent a questionnaire about the sample of use interface to the subject.
6. The subjects completed the questionnaire and the questionnaire was sent to our server, where data collected from the questionnaire were saved to our database automatically.

Figure 3.1.1.3: Experimental procedures

3.1.2 Result and Discussion

We employed data mining concepts (i.e. the concepts of support and confidence of data analysis) and data normalization technique to analyse the collected data.

Owing to the biases of individual subjects in assigning preference values, some outliers might dominate very high or very low preference values. As a result, the maximum and minimum preference values were different across different subjects. Therefore, we normalized all the preference values by Z-Score Normalization ($Y = [y - m] / std$), where Y as the normalized value, y as the original value, m as the mean of original values and std as the standard deviation of original values.

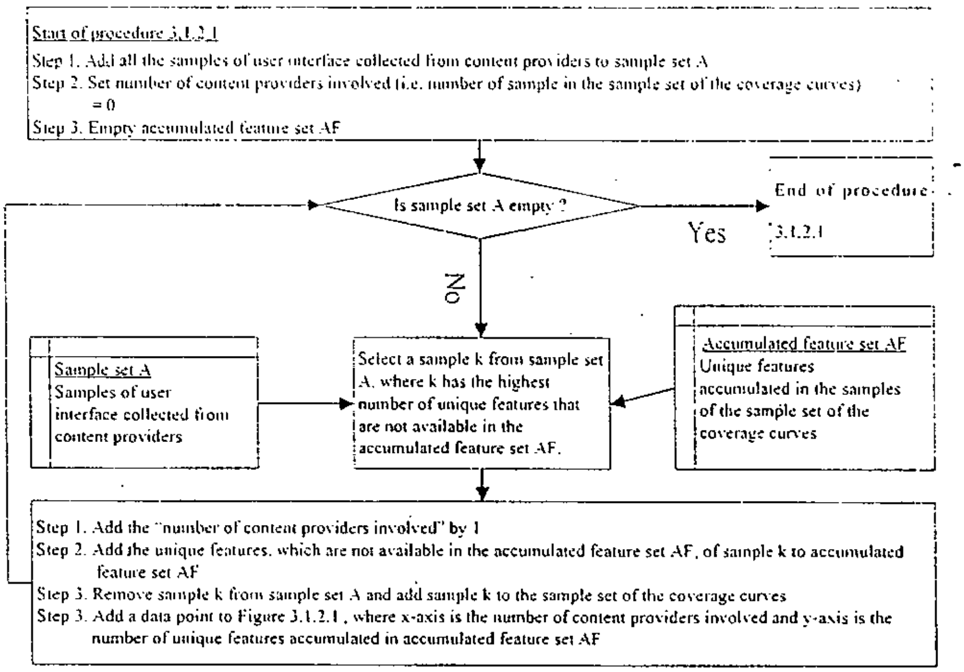
On the other hand, the concepts of support and confidence (Appendix C.1) were used to show the degree of significance of data analysis. According to prior analysis (Appendix C.2), a minimum requirement of support was set to be 70% in order to get the maximum benefit from the analysis of the collected data. Furthermore, there might be no observation if the minimum confidence requirement was too high and the observation might not be representative if the minimum confidence requirement was too low. Therefore, we performed our analysis with various values of minimum confidence requirements (i.e. 50%, 70% and 90%).

The results of analysis on the availability/missing of features at the pre-identified feature set of each samples of user interface show that there were 16 unique features found at offline samples and 9 unique features found at online samples and a total of 18 unique features found at all of the samples when the minimum requirement of confidence level was set to 50%. However, there were 16 unique features found at offline samples and 10 unique features found at online samples and a total of 19 unique features found at all of the samples when the minimum requirement of confidence level was set to 70% or above. This is because one of the pre-identified features of online samples was judged as not available (with confidence 50% and support 74%) but there was not enough confidence to support such a judgment if the minimum confidence requirement was 70% or above. As we were seeking for high confidence in the result, we ignored the dropping of the feature, which was judged as not available with only 50%

of confidence, from the pre-identified feature set. Accordingly, the analysis regarding features was based on 70% (because 90% is too tight) of minimum confidence requirement.

It is necessary to address how well the survey covered those existing user interfaces features. Our survey, approximately, covered enough sample of user interface if the number of accumulated features reached a saturated condition (i.e. when the number of samples was increasing, there were no changes on the number of unique features accumulated or the increase of the number of accumulated unique features was very slow).

Therefore, coverage curves were used to test the coverage level of the survey. We continuously added samples to the sample set of the coverage curves according to procedure 3.1.2.1 and the unique user interface features, which were found in the samples of the sample set, were found and were known as accumulated features. Figure 3.1.2.1 shows that the number of samples was increasing but there were no changes on the number of features accumulated after sample (4) and sample (7) were added to online and offline analysis respectively. The encouraging result was that the survey had already covered enough sample of user interface. This is because the coverage curves were saturated.



Procedure 3.1.2.1 Procedures for adding samples to the sample set of coverage curves

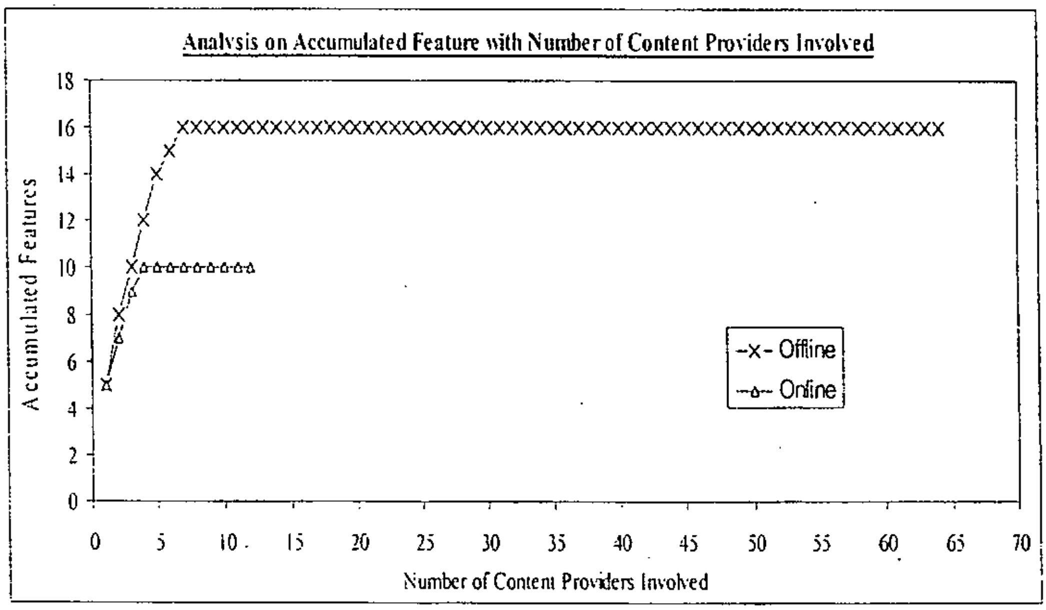


Figure 3.1.2.1 Coverage curves of accumulated features found

Issues of Information Access Structure:

The first issue explored was the population of the list and directory access structure (i.e. interface). The analysis was carried out with 70% minimum support requirement at 50%, 70% and 90% minimum confidence requirements respectively. User interfaces that fail to fulfill the minimum requirement of confidence were considered as hybrid. All the hybrid user interfaces were not considered for the analysis.

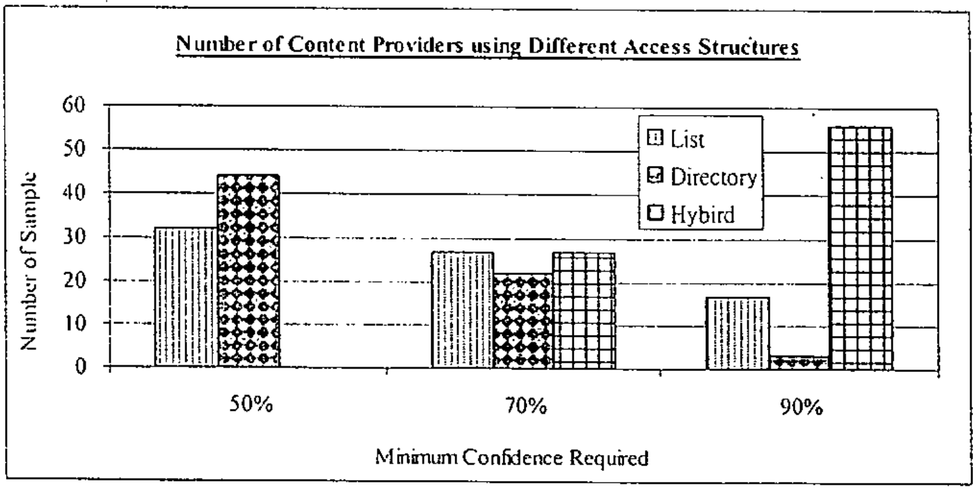


Figure 3.1.2.2 Number of content providers (i.e. sample of user interfaces) using different access structures based on different levels of confidence in the identification of the nature (i.e. list / directory) of access structure by the subjects.

According to Figure 3.1.2.2, there are more content providers providing directory interface than list interface if the minimum requirement of confidence on interface type judgment was set to 50%. However, there are more content providers which provide list interface than directory interface if the minimum requirement of confidence on interface type judgment was set to 70% and it was found that the gap between the population of list and directory was becoming much more significant when minimum confidence requirement was increasing from 70% to 90% (i.e. becoming much more certain). The results suggest that list interface is more common than directory interface. This is because the proportion of list interface in our samples was higher than that of directory interface.

As list was much more commonly used than directory, no hierarchical access structure was needed to be developed if list was also human preferred compared with directory. Normalized preference values of samples were divided into ten intervals and population of samples of list and directory were distributed into the ten intervals accordingly. Analysis was then carried out by comparing the population of list and directory at different intervals. If there were a large population at relatively high preference intervals, it means that the access structure is relatively more preferred by human and vice versa.

Interestingly, it was found that the dominant populations of list were at a lower preference interval comparing with that of the dominant populations of directory (Table 3.1.2.1). It means that the subjects prefer directories over lists. It also indicates that existing and commonly used access structures may not fulfill human's preferences. One of the possible reasons is that a directory can summarise huge amount of information into a small piece of text and can facilitate navigation using only limited display area of mobile devices while list structures fail to do so and require much more scrolling and page jumps. Thus, a directory structure comes with strong summarising and navigating capacity. Further analysis and innovation on mobile information access mechanisms to support mobile searching seem to be necessary, particularly in the development of hierarchical access structure as it is one of the human preferred models for navigation when there is information overloading.

Table 3.1.2.1: Percentages of the Preferences of List, Directory and Hybrid Interfaces

Nature of UI	List						Directory						Hybrid					
Online/Offline	Online			Offline			Online			Offline			Online			Offline		
Minimum Confidence	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9	0.5	0.7	0.9
Preference Level	%																	
0-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2-3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3-4	50	50	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0
4-5	50	50	100	7	8	13	0	0	0	0	0	0	0	0	0	0	0	0
5-6	0	0	0	67	76	75	40	33	0	18	0	0	0	50	36	0	30	31
6-7	0	0	0	27	16	13	60	67	0	76	88	100	0	50	55	0	70	64
7-8	0	0	0	0	0	0	0	0	0	6	13	0	0	0	0	0	0	4
8-9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9-10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total # of Sample	2	2	1	30	25	16	10	6	0	34	16	3	0	4	11	0	23	45

Issues of User Interface Feature:

It was assumed that human preference not only relates to information access structure in general (as mentioned) but also the user interface features. If this assumption is correct, then a preferred access structure can be governed by its nature and also features. It means that we have to develop a hierarchical access model with the consideration of "preferred" features. Therefore, the second issue is to explore the features found in the samples of user interface.

It is assumed by common sense that more features come with higher human preference on the user interface. In order to judge the assumption, the availability of user interface features on each of the samples was judged by users and all the samples were then grouped according to the number of user interface features contained. Accordingly, an analysis was carried out by comparing the average preference on each group of samples of user interface with the same number of interface features contained. If the initial assumption is true, the average preference of a given sample of user interface should increase with increasing number of features contained. However, it is found that the average preference increased with the increase in the number of features and with a diminishing return after a certain critical point is reached (Figure 3.1.2.3). It was different from our assumption which was made based on common sense. The difference may indicate that user interface features can be useful on wireless searching while too many features may degrade the utility of the user interface. This is because the limited display area on mobile devices cannot contain many user interface features, which may cause confusion. In this way, a mobile searching interface should provide only the necessary and effective user interface features.

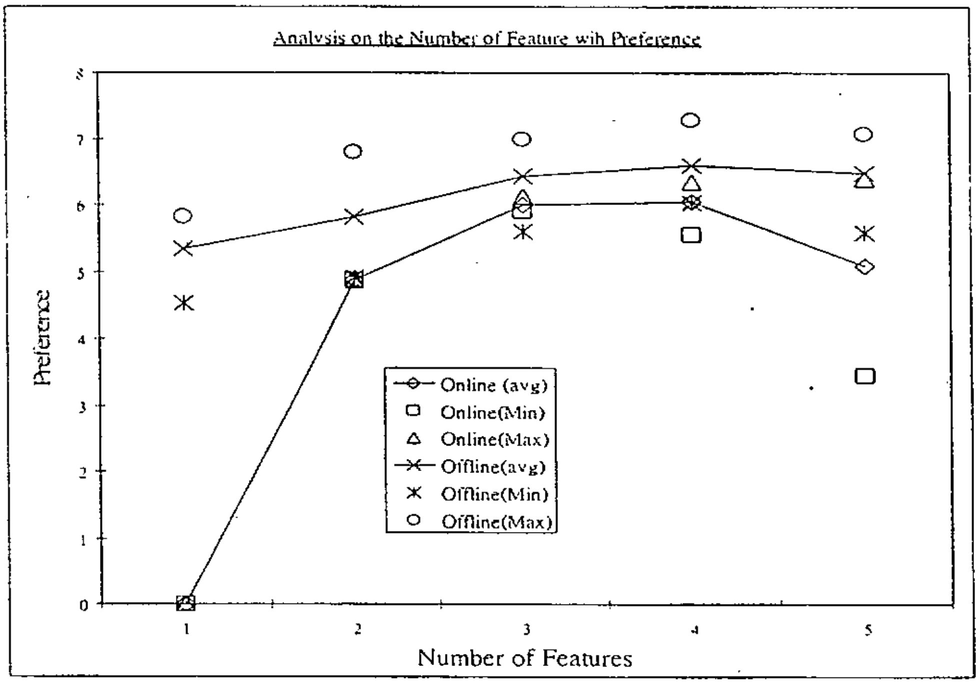
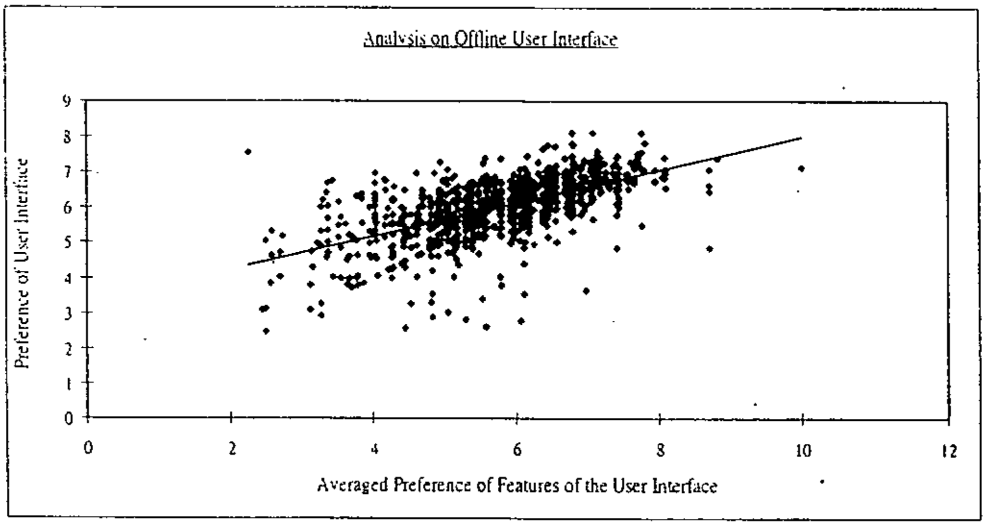


Figure 3.1.2.3 User preference with respect to the number of features

Having established that only a limited number of features can be added to a searching interface, the selection of features becomes an important issue in the design of a search interface for mobile devices. As shown in Figure 3.1.2.3, the critical number of features is likely higher than one. It was assumed that there were group/groups of features that were interacting with each other. In order to provide a preferable searching interface, an analysis was carried out to explore whether preferred combination of features is one of the possible factors governing a preferred overall user interface preference.

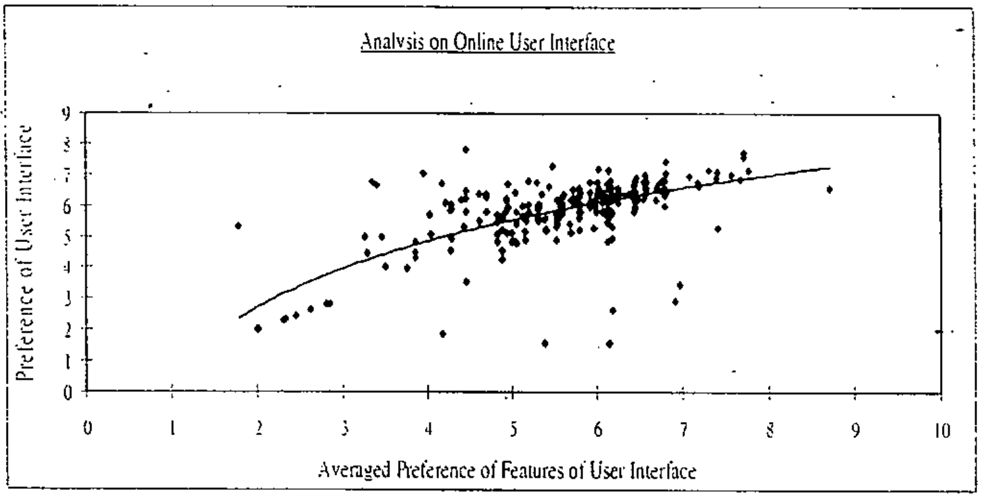


R-Square

Linear	0.3676	was used for analysis
Logarithmic	0.3575	
Power	0.3387	
Exponential	0.3386	

Figure 3.1.2.4 Averaged preference of features with respect to preference of user interface.

Remark: Each of the data points is representing the preference values assigned by a user on a user interface.

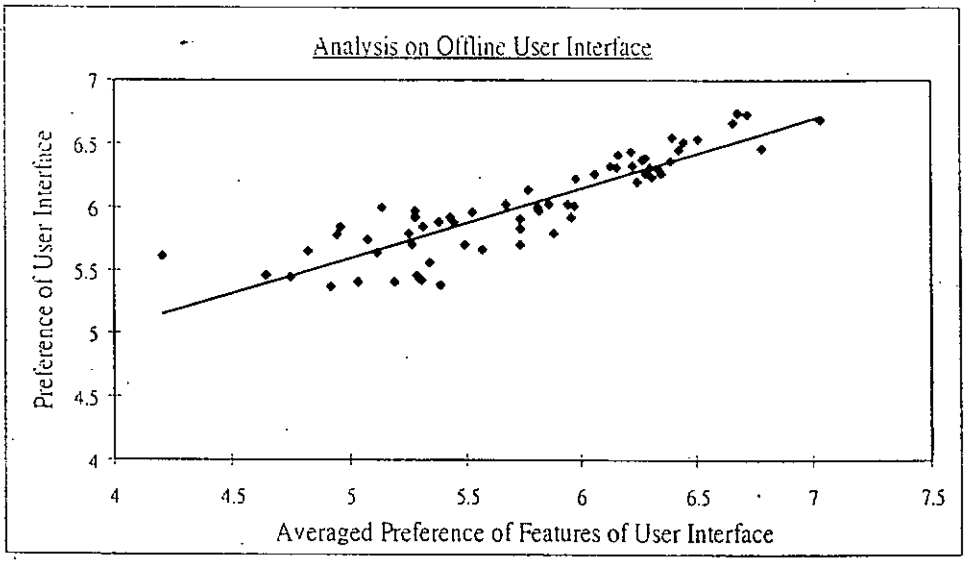


R-Square	
Linear	0.4392
Logarithmic	0.4794 was used for analysis
Power	0.4574
Exponential	0.3965

Figure 3.1.2.5 Averaged preference of features with respect to preference of user interface

Remark: Each of the data points is representing the preference values assigned by a user on a user interface.

Figure 3.1.2.4 and Figure 3.1.2.5 show that there is a positive relationship between the preference of user interface and the average preference of the contained features while there are still a number of exceptional cases (noisy). The problem can be reduced by averaging the preference values assigned by users on each of the samples if the exceptional cases were caused by user variations. Figure 3.1.2.6 and Figure 3.1.2.7 show that preference values generally fall into the trendline and it can be concluded that the exceptional cases were caused by user variations on assigning preference values. Therefore, it is suggested that a preferred combination of features may be one of the possible factors governing a preferred overall user interface.

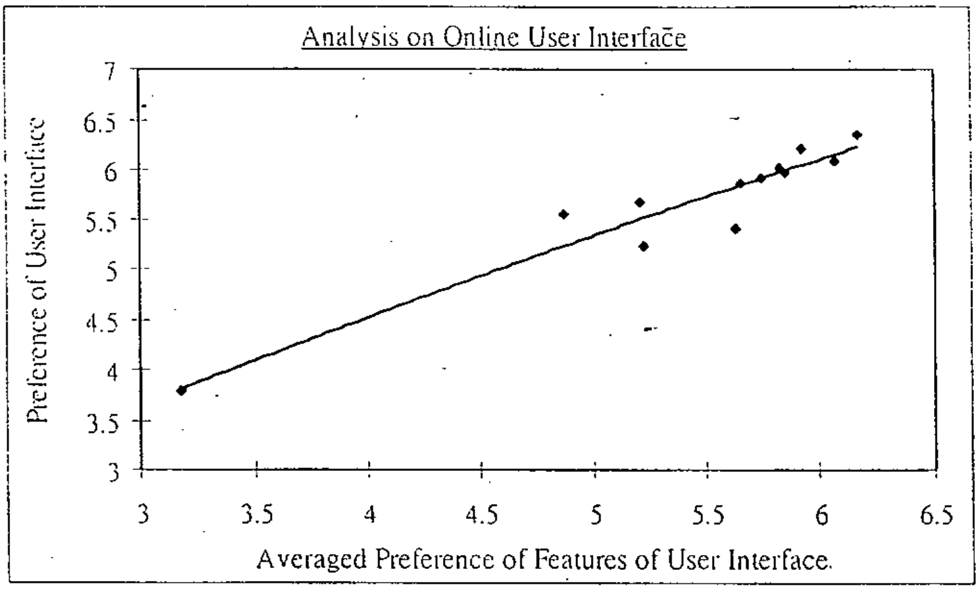


R-Square

Linear	0.7887	was used for analysis
Logarithmic	0.763	
Power	0.7594	
Exponential	0.7832	

Figure 3.1.2.6 Averaged preference of features with respect to averaged preference of user interface

Remark: Each of the data points is representing the averaged preference values assigned by users on a user interface.



R-Square

Linear	0.9118
Logarithmic	0.9108
Power	0.931 was used for analysis
Exponential	0.9191

Figure 3.1.2.7 Averaged preference of features with respect to averaged preference of user interface

Remark: Each of the data points is representing the averaged preference values assigned by users on a user interface.

As it is possible that the preference of a sample of user interface can be governed by its user interface features, 76 samples (that fulfill the 70% minimum support requirements) were divided into 3 groups according to user preferences (top 25 from the highly preferred group; last 26 from a less preferred group). All the subsets of features of each of the samples were considered as a feature set and there were 183 unique feature sets found. In order to select those feature sets, which were relatively common to the preferred user interfaces, any feature set at three (a value higher than one so as to have a higher degree of guarantee on the importance of the feature set) or more than three at the highly preferred group ONLY were considered as the highly preferred features set (Table 3.1.2.2). It was found that grouping information into directory and showing information with list as well as connectivity facilities for providing navigation are the highly preferred patterns for mobile searching interface.

Table 3.1.2.2: Preferred Feature Sets

Feature Sets	User Interface Features
1	Provide facility for jumping back to higher level
	Provide facility for jumping in between different catalogs
	List of titles together with abstract in each of the catalogs
	Articles were classified into different catalogs
2	Articles were classified into different catalogs
	Provide facility to jump to the next or previous document
	List of catalog names together with related titles
3	Provide facility for jumping back to higher level
	Provide facility for jumping in between different catalogs
	List of titles together with abstract in each of the catalogs
4	Provide facility to jump to the next or previous document
	List of catalog names together with related titles
5	Articles were classified into different catalogs
	List of catalog names together with related titles
6	List of catalog names together with related titles

3.2 User Behavior and Requirements

Establishing user requirements is well accepted as an essential step in developing useful and usable systems, such as mobile searching system. Thus, there is a definite requirement to make users' true needs to be understood by system developers [68]. In the previous section, we explored the feasibility of our proposition of using hierarchical access for mobile searching based on user preferences. In this section, we further explore the feasibility of our proposition from practical requirements and user behaviors. However, a major challenge in analysing user requirements as well as behaviors for mobile searching system is the versatility of usage patterns and usage contexts in which the application takes place. Traditional means of requirement survey and usability tests are not enough for capturing users' activities and users' needs in their practical situation. We did our analysis in exploring mobile searching system requirements and user behaviors from user's practical environment, based on a well-known practical analysis methodology of user requirements called Contextual Inquiry which is a component of Contextual Design (Appendix D). The Contextual Inquiry focuses on discussing and analysing practical difficulties as well as functional requirements when mobile knowledge seekers are searching for information with mobile devices. The results represent the needs in the field, the difficulties experienced and are the behaviors of the mobile knowledge seekers when they are searching information using mobile devices.

3.2.1 Similar Work

Research based on Contextual Design focuses on the development of practical systems in order to support real work practices. Williams and McClintock [69] recognized that a closer relationship between users and developers is necessary to support the development of user friendly products better. Cabletron Systems, Inc. [70] also formed two cross-functional teams to conduct Contextual Inquiry studies in 1996 and 1997. Both teams agreed that the methodology exceeded their expectations. Another two research parties ([71] and [72]) also reported their user-centered studies based on Contextual Design in 1997 and 1999 respectively. The former one presented the work of using a sequence of steps to keep an ongoing and evolving understanding of user requirements under consideration of the system designers and developers for designing clinical workstation software. The latter one tried to derive user data from Contextual Inquiry in a large, multi-site product development organization. Their distributed teams collected and analysed data from users in order to get implications on product design. Moreover, the human-computer interaction lab at the University of Maryland developed research based on

Contextual Design to enable users to participate throughout the technology development process. The technique is called cooperative inquiry which makes use of users' practical experiences for developing software [73]. Konstantinos from University of Aegean also presented their investigation on the use of Contextual Design for designing complex and innovative systems. They found that Contextual Inquiry of Contextual Design is an emerging practice for investigating behavioral requirements of users in their working environment. The advantage of Contextual Design over other requirements elicitation methodologies, such as Requirements Engineering and User Task Analysis, is the focus on observation and in-work interviews for extracting requirements and designing customer-centered systems. Researchers have successfully applied and adapted the methodology to a special case of gathering user requirements for a mobile exhibition system [74]. It is not difficult to realize that such a software design process (the Contextual Inquiry) is generally being accepted as one of the practical and useful software design methodologies among different parties.

Mobile technology researchers and developers have also employed Contextual Design for analysing and addressing user requirement issues.

Nokia conducted a user needs study on their Nokia 9000 Communicator [75] based on Contextual Inquiry with six professional users in 1996. Such a user need study has investigated both the required functionality and human computer interaction issues. Despite numerous practical challenges, the study was successfully applied in the development of mobile communication devices.

A more recent research, which was based on Contextual Design, [76] reported a study with a group of novice wireless phone users on wireless communications usability. The researchers of the study closely tracked the users after service acquisition. The study was taking a technology-as-system analytical approach, and described the wireless telephony system as four socio-technical components: hardware, software, netware, and bizware. Their particular organization of the system is intended for the practical application of designing for usability. The research group believes that the analysis for accessing the usability of wireless telephony system should be beyond the device and technology itself to include the context of working environment of users' "real life" so as to improve usability in the most fundamental ways.

Contextual Design was widely and successfully used on designing and testing different software systems and even physical hardware including wireless telephony systems and hardware. It is believed that Contextual Inquiry can be one of the most suitable methodologies for analysing the feasibility of our proposed hierarchical access structure for supporting mobile searching.

3.2.2 Experimental Setup

The analysis was carried out by situating an interactive experiment in the real activities of mobile device users in order to cover a wide range of naturally occurring user experiences, requirements and behaviors. Naturalistic study is difficult to accomplish, while direct observation by shadowing users is possible for very small numbers of participants [76].

The data collection and analysis approaches were primarily qualitative. We tried not to limit our observations to a pre-defined issue space as Palen L. and Salzman M. [76], and therefore tried to collect wide range of information about mobile searching systems. Our analysis is based on observation as well as user provided information, such that the analysis is user based.

Users:

Fifteen experienced PDA users were invited to participate in our study – ten males and five females (Appendix A.1 Subject 12 – 17, 19 – 27). They were the subgroup of the subjects who participated in the user preference survey as described in section 3.1. This is because we need to have a better understanding and comparison of the users' preference, behavior and practical requirements. In order to address user interface issues, all the users come with human computer interaction training experiences. They are undergraduate students, aged 20 – 23. The users had experience from modest to expert with mobile phone and Internet. They have on average at least four years of online search experience and with the experience to access the Internet as well as mobile devices. These subjects reflect the most active population using mobile devices (Appendix A.2) and these subjects are expected to continue to use mobile devices in the future.

Experimental Tools:

A PDA (Pocket PC - Toshiba e740) (Figure 3.2.2.1) was connected to a laptop computer, which, in turn, was connected to the Internet for accessing Internet resources (Figure 3.2.2.2). An audio recorder and a video capturing tool (Remote Display Control) were used for recording the experiment. Meanwhile, data traffic and user interactions were recorded by proxy server in the laptop computer. Two Internet browsers, namely, Internet Explorer and Klondike WAP browser (Figure 3.2.2.3, 3.2.2.4), were employed for accessing Web and WAP based information respectively. As the browsers' interfaces and functionalities are similar, we can minimize the biases of interface effect caused by the different Web and WAP content.

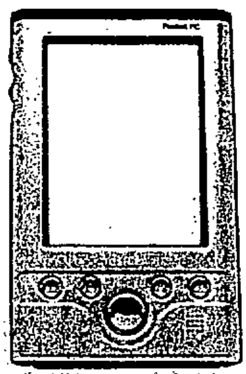


Figure 3.2.2.1:
Toshiba e740

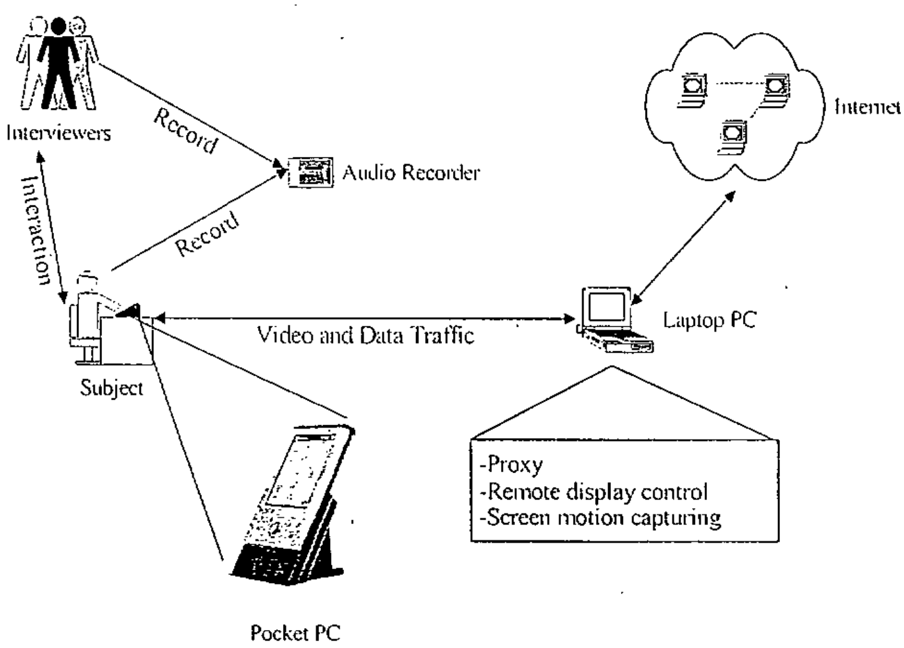


Figure 3.2.2.2 Experimental setup for the Contextual Inquiry of user behavior in mobile searching

Figure 3.2.2.3
Internet Explorer

Figure 3.2.2.4
Klondike WAP
Browser

Search Tasks:

In this experiment, the search tasks were designed for observing and analysing practical user requirements, opinions, and behaviors when the users were searching with mobile devices. Users were asked to perform various searching tasks (Appendix E) on Web based and WAP based information by using keyword based or hierarchical based searching tools/engines. The tasks were categorized into four sections which were further divided into eight sub-sections so as to make sure that the users performed searching on various tasks for mobile information access.

Data Collection Channels:

Four interviews, each comes with two sub-sections lasting for one hour, were conducted with each user. The interviews were open-ended and conducted while users were searching with a mobile device for the given tasks such that requirement issues were discussed with the users at the same time.

Interview:

While the interview was in progress, an interviewer observed and discussed with users on their requirements and difficulties as well as opinions on mobile searching system. Such information, together with user's actions and interviewer's observations, were recorded during the interview. A short discussion was conducted with the users in order to discover and wrap up problems and ambiguities after each interview.

Video and Audio Tapes:

All interviews were video taped and audio taped as supplementary information.

Questionnaires:

Additional information, such as user background, was collected with post experimental questionnaires in order to take a better understanding of users' behaviors.

Experimental Procedures:

1. The experimental requirements, rundown and tools were briefed.

2. Users were requested to try the experimental tools and perform searching in order to familiarize with mobile searching situations.
3. Searching tasks of section I were briefed and each of the users was asked to make sure that s/he knew exactly what was going to search.
4. Section IA was performed and followed with a short break.
5. Section IB was performed and followed with a short break.
6. A short discussion was conducted for summarising and wrapping up findings.
7. A short break was given to the users and the interviewer for relaxation.
8. Searching tasks of section II were briefed and each of the users was asked to make sure that s/he knew exactly what was going to search.
9. Section IIA was performed and followed with a short break.
10. Section IIB was performed and followed with a short break.
11. A short discussion was conducted for summarising and wrapping up findings.
12. A short break was given to the users and the interviewer for relaxation.
13. Searching tasks of section III were briefed and each of the users was asked to make sure that s/he knew exactly what was going to search.
14. Section IIIA was performed and followed with a short break.

15. Section IIIB was performed and followed with a short break.

16. A short discussion was conducted for summarising and wrapping up findings.

17. A short break was given to the users and the interviewer for relaxation.

18. Searching tasks of section IV were briefed and each of the users was asked to make sure that s/he knew exactly what was going to search.

19. Section IVA was performed and followed with a short break.

20. Section IVB was performed and followed with a short break.

21. A short discussion was conducted for summarising and wrapping up findings.

3.2.3 Result and Discussion

In the following sections, we further explore the feasibility of hierarchical access and factors governing the design of hierarchical access based on user requirements and behaviors. It is possible that some of the focus and scope of user requirements may not be expected before the experiment. If we fix the focus and scope of data collection, then we may miss many useful user requirements that we did not expected. Therefore, the collection of data was based on open-ended interviews, discussion and observations. As a result, users can raise their requirements, opinions and difficulties freely while they were searching via mobile device. However, the collected data are normally non-structured and do not have a fixed focus on representing the users' requirements, opinions and difficulties. It is difficult to make conclusion from such data directly. An affinity diagram was, therefore, used to consolidate such data for analysis. An affinity diagram is a hierarchical structure which represents information from general to specific in which each of the parent nodes is representing a general idea concluded from the associated child nodes. In our case, the collected data were represented by leaf nodes of the affinity diagram and those similar nodes belonging to the same layer were grouped and consolidated in order to find a general idea of each group and formed an upper layer with the general ideas of the groups. As a result, the affinity diagram represents grouped and concluded ideas from the collected data. Therefore, we are able to explore the requirements, opinions and difficulties of users by referring to those consolidated conclusions (i.e. ideas) in the affinity diagram. The result of the analysis was summarised using affinity diagrams and the behavior was summarised using flow charts constructed from the data collected in the experiments mentioned in section 3.2.2.

Analysis of Affinity Diagrams (AD) (Appendix F):

Requirements, opinions and observations, which were collected from users during open-ended interview in the experiment, were all regarded as requirements. The requirements were consolidated and complemented by video and audio records in order to avoid losing information. The consolidated requirements were then re-constructed into affinity diagrams which represent a hierarchical relationship among different requirements and provide a summary on similar requirements. The following analysis was based on the affinity diagrams.

Feasibility of Hierarchical Access:

We assumed that hierarchical access was not only a human preferred access structure but also a suitable access structure for supporting mobile searching in practical situations. If our assumption was accepted, hierarchical access should fulfill those practical requirements mentioned in the affinity diagrams. Here, we try to judge our assumption from the following dimensions.

Firstly, owing to the limited input capacity of mobile devices, selection based input is much more suitable than typing based input [AD – 1.1.1.3.6], which favors the use of hierarchical access as all of the input using hierarchal access is selection based. This finding was further confirmed by [AD – 1.1.1.3.7] that a local catalog (hierarchy) was much more preferred than local keyword search for accessing a Web site.

Secondly, [AD – 2.1.1.3.1] indicated that much information was not grouped or organized, resulting in ambiguous layout, slower accessing speed and waste of display area. The use of hierarchical structure is one of the ways for grouping and organizing disordered information. Thus, users also agree that hierarchical structure can provide a tidy layout [AD – 2.1.1.6, AD – 2.1.1.6.5.2] and they prefer using hierarchical structure for grouping information [AD – 2.1.1.6.6.3]. This is because, according to their practical experience, hierarchical structure can help them to identify relevant information quickly (We will try to further validate this judgment in Chapter 4) by providing a tidy, organized layout and summary of information [AD – 2.1.2.3.1.1.5].

Thirdly, it would be much more convenient to use hierarchical access if a user knows that the searching item belongs to which catalog in the hierarchy [AD – 2.1.2.3.1.1.7.3]. In some cases, users need to use a hierarchy when they cannot specify queries or do not have a clear objective for searching [AD – 2.1.2.3.1.1.7.1, AD – 2.1.2.3.1.1.7.2].

Finally, hierarchical access could probably be a good summariser for concentrating important information so as to fulfill the requirements mentioned in [AD – 2.1.2.3.1.2.1], which, in turn, allows users to anticipate the relevance of a group of documents by browsing only a small and organized hierarchical summary. Potentially, users can have a faster accessing speed to information (In Chapter 4, we will further prove this judgment by an experiment of information access performance) as mentioned in [AD – 1.1.1.2.2] and make a decision on whether to wait for the loading of detailed content or not as mentioned in [AD –1.1.1.2.1.2, AD – 1.1.1.2.1.3].

Firstly, users require a simple, clear, tidy and sharp layout with a common or static form, in which components are indicated clearly and fitted to a single screen that fully utilize the screen [AD – 1.1.2.1]. This is because a complicated and information-rich layout causes ambiguity when it is used to access information with a mobile device that has a limited display. Therefore, simple text-based links, and well-contrasted text (black) and background (white) were considered to be the most suitable for showing information in mobile devices [AD – 2.1.1.4, AD – 1.1.2.2].

Secondly, users reported that summarisation, concentrated information, heading, indexing and local hierarchy (catalog) could be useful for them to understand, search and identify useful content quickly and conveniently in mobile devices. This is because it is difficult for them to browse a huge amount of information with a small display area in a mobile device while they may be able to do so when they are using desktop computers [AD – 2.1.1.7.1]. Moreover, they also suggested that only abstract and heading are necessary content in the search result while additional informative information such as the total number of search result and the total number of pages could be useful [AD – 2.1.1.8.2, AD – 2.1.1.8.3].

Thirdly, users indicated that they will not be able to search for a single purpose for more than 15 minutes if they use mobile devices to search for information in real life [AD – 1.1.1.2.1.1]. Also, they cannot accept slow loading and accessing speed when they are searching via mobile devices [AD – 1.1.1.2.2]. They suggested removing pictures and information that do not fit onto a single page so that the loading speed can be increased and the amount of scrolling can be reduced (slower accessing speed) [AD – 1.1.1.2.2]. As scrolling made users difficult to locate, capture and access the information [AD -2.1.1.3.3], they suggested splitting pages, physical control keys and hierarchical summarisation to deal with or even replace scrolling pages. Thus communication cost (data traffic rate), accessing speed and loading speed as well as tidy layout have higher priorities than other functions such as pictures and scroll bars [AD – 1.1.1.3.9].

Fourthly, users suggested sorting search results by relevant ranking or alphabetical order or user interest if possible [AD – 2.1.2.3.2]. However, they did not prefer having more than 20 items in a single page [AD – 2.1.2.3.3.1, AD – 2.1.2.3.3.1.2].

Fifthly, it was required that function keys should be unique and user familiar, with descriptions and clearly shown. For drawing users' attention effectively and for users' convenience, function keys should be grouped together and put at the top and bottom of a long page (scrolling is required) or at the bottom of a single screen page [AD – 2.1.1.1]. In addition to seeking attention, highlighting of important keywords in the search results, abstracts and headings could be useful hints to help users to identify useful information. This is because users indicated that they usually pay attention to such items. Also, highlighting visited links can also help users to quickly access useful information as they can skip those visited but useless links quickly [AD – 1.1.1.1].

Finally, users are seeking for a feasible mechanism to group information, and hierarchy is one of the possible methods helping users to identify information quickly by providing a tidy display and a summary of information. The users use hierarchical searching when they are unable to specify queries or do not have a clear objective for searching or when they know exactly the search items that belong to certain parts of the hierarchy. Users accepted changing on hierarchy's component based on dynamic data (i.e. dynamic hierarchy) if necessary, while information provided by hierarchy has to relate to human's common sense and comes with a static style. Also, they preferred detailed hierarchy and accepted multiple paths that matched the same concept which were either related to human's common sense or to the underlying content of the information units. As for displaying, catalog (hierarchy) and headings should not be put together and names of nodes of catalog should be well separated in a single screen [AD – 2.1.1.6]. Furthermore, users required status tracing functions such as forward, backward and direct jumping functions to go to the different status or levels of the hierarchy. This could be useful when users change their ideas on the selection of hierarchical items [AD – 3.1.2.2].

Analysis on Behavior Flow Charts:

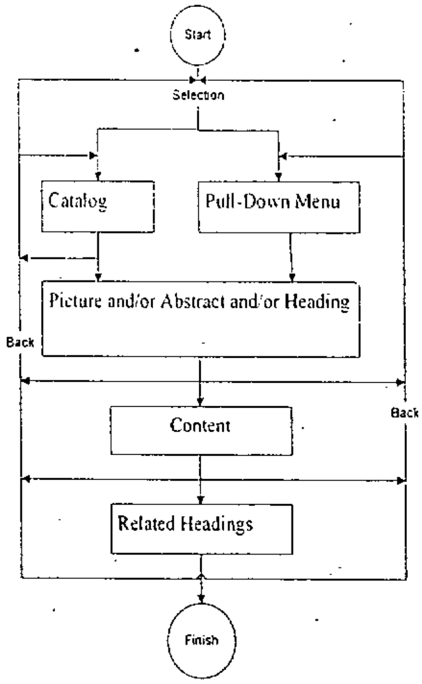


Figure 3.2.3.1 User search behavior flow chart

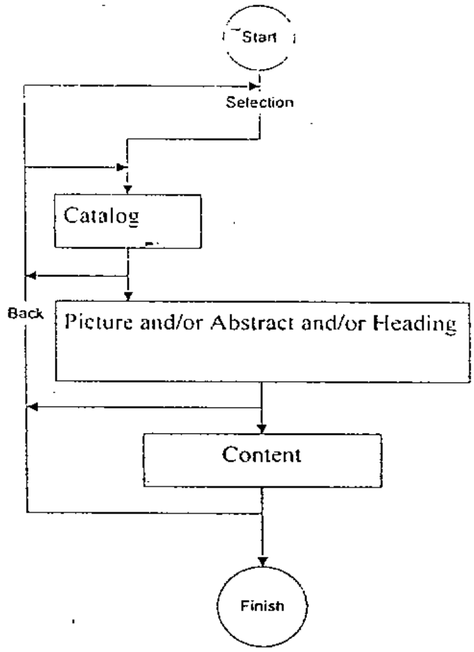


Figure 3.2.3.2 Simplified user search behavior flow chart

Each user behavior was recorded using video (refer to section 3.2.2) and the behavior was summarised using a flow chart. The final analysis integrated all the users' search behaviors by combining the flow charts. As the aim of this section is to explore issues on designing hierarchical access structure, only user behaviors regarding hierarchical access were analysed. In order to make sure that the behavior flow chart (Figure 3.2.3.1) is representing all the users' behaviors regarding hierarchical access, the behavior flow chart contained all the paths collected. In order to simplify the analysis and focus on hierarchical access to search result only, paths via pull-down menus were ignored because pull-down menus are not the part of the hierarchical access structure. Also, paths via related headings were ignored because it was a particular feature of the news portals but not related to the search results. After discarding the irrelevant user behavior, the final analysis of the user search behavior related to searching is shown in Figure 3.2.3.2.

It was found that user behavior of hierarchical access on mobile device is quite simple and straight forward. There are only three major components involved, namely, catalog, picture/abstract/headings and content. We may further ignore the issue of picture because pictures are not preferred [AD – 1.1.1.3.8, AD – 1.1.1.3.9]. Instead, it is found that the linkage in between the above three components was important. In order to fulfill users' behavior, there should be paths for linking each of the three components of the hierarchical access structure.

3.3 Comparison and Contrast on User Preference Survey and Contextual Inquiry

3.3.1 Feasibility of Hierarchical Access Structure

According to the user preference survey, hierarchical access structure was highly preferred by users. This can be explained by the user requirement analysis of Contextual Inquiry. Owing to the limited display area and input capacity of mobile device, users require a tidy, organized and summarised selection based access structure. The access structure should provide fast access and good prediction on the relevance of underlying content. Hierarchical access structure is one of the access structures which can fulfill such requirements.

3.3.2 Similarities and Contrast on User Interface Features

As the results of the user preference survey were based on user's perception of the user interfaces, it may not fully represent the practical requirements of mobile users when they are performing mobile searching. Therefore, the Contextual Inquiry lets the users to raise their difficulties, opinions and requirements while they are actually searching for information using mobile devices. The user preference survey could be considered as a prior study which raises general issues that we have to take into consideration.

The Contextual Inquiry further confirms and complements those issues in the survey. Table 3.3.2.1 summarises the important user interface features indicated by the survey and Contextual Inquiry respectively. Some of the features found in the survey are similar to features found in the Contextual Inquiry. However, some of the features were only indicated as important in either the user preference survey or the Contextual Inquiry.

Particularly, the importance of categorization of information (a similar important feature in the survey is "Articles were classified into different catalogs") supports the use of hierarchical access structure because hierarchical access structure is similar to catalog.

Table 3.3.2.1 Summary of Important Features Identified

Important Features Identified	
User Preference Survey	**Contextual Inquiry**
No similar features	Bold or underline or highlight those keywords and visited links
No similar features	Always fit information on a single screen (i.e. avoid scrolling)
Provide facility for jumping back to higher level	Backward and forward functions
No similar features	Selection based input
No similar features	Local catalog, menu and abstract to provide a summary of information
No similar features	Description on function keys
No similar features	Grouping on function keys
No similar features	Text-based links
No similar features	Black and white background and fonts
No similar features	Page break/split long pages for avoiding scrolling
No similar features	Physical function buttons
No similar features	Pull-down menu
Provide facility for jumping in between different catalogs	Function for directly jumping in between different status, e.g. Page number list <1,2,3,4,5>
No similar features	Sorting items by relevant ranking, alphabetic order or user interests
No similar features	Show total number of search results or available documents or pages
No similar features	Summary of information by heading, abstract or local catalog/hierarchy
Articles were classified into different catalogs	Categorization of information
List of titles together with abstract in each of the catalog	Abstract + heading would enough for browsing search result
Provide facility to jump to the next or previous document	No similar features
List of catalog names together with related titles	No similar features

3.4 Proposed Interface Design

With reference to the user preference survey and Contextual Inquiry, a preferred combination of user interface features contains all of the features which are known as preferred features (i.e. the features that were only found at those user interfaces which attained high overall user preference) in user preference survey and required features in Contextual Inquiry. This is because such features would probably fulfill both user preference and practical requirements in comparison with others (Table 3.3.2.1). The combination can also contain some other features which may either be known as preferred features at user preference survey or required features at Contextual Inquiry subject to practical requirement and functions of the interface (Table 3.3.2.1).

3.5 Summary

In this chapter, the issues regarding user preference of user interfaces, features of user interfaces and access structures for mobile searching were addressed. A survey was conducted in order to explore various user preferences on user interface, features of user interface and access structures of the mobile search system.

It was found that list is much common than hierarchical structures but not preferred by the users (i.e. the subjects of the survey). Instead, the users prefer using those hierarchical information access structures for browsing with mobile devices, which, in turn, suggest the need and the use of our proposed hierarchical access structure.

It was also found that a preferred combination of user interface features may be one of the possible factors governing the overall user interface design. In addition, the quality of interface will be improved with an increase in features contained while there will probably be a critical number of features (i.e. interface quality may be degraded if too many features were added). We also discovered a number of feature sets that were commonly found at the highly preferred interfaces.

We further analysed practical requirements and behaviors on access structure of the mobile searching system. It was found that hierarchical access structure is not only preferred by users but it can also fulfill the practical requirements.

It was also found that hierarchical access structure based on analysing user behaviors could be as simple as only containing hierarchically organized catalogs, functions for navigating in between different catalogs, title list of items contained in the hierarchy and related contents.

Although hierarchical access is preferred by users and practically required, it is not commonly deployed. One of the reasons is probably that hierarchical structures are financially costly because they are constructed manually for the mobile content providers, similar to the hierarchies of Yahoo! and Open Directory. In the following chapter, we will further explore the feasibility of automatically constructing concept hierarchy and the information access performance using concept hierarchy to support the searching of documents in search results using mobile devices. We will also use the findings of this chapter for designing a sample hierarchical access structure in order to test the information access performance issues of hierarchical access structure.

Chapter 4

Hierarchical Access Structure

In the last chapter, we explored the feasibility of hierarchical access on supporting mobile searching based on subject preference and requirements. It was found that hierarchical access is much more preferred and required by human subjects than the commonly used access structure, i.e. list. In this chapter, we further explore the feasibility of automatically constructing concept hierarchies and the performance of the concept hierarchy as a hierarchical access structure to support the access of search results using mobile devices.

4.1 Types of Hierarchies

4.1.1 Manually-created Hierarchies

Most of the existing hierarchies are created manually; typical examples are Yahoo! [21], Opendirectory Project [77] and electronic library catalog. As manually-created hierarchies are created by human, their structure as well as hierarchical terms are easily interpreted and understood by human beings.

However, manually-created hierarchies cost huge amount of manual effort to construct, e.g. Yahoo! employs hundreds of experts for updating the Yahoo! hierarchy and Opendirectory involves thousands of people for creating the directory. It is costly for manually-created hierarchies to support mobile Internet search because of the huge amount of information on the Internet, e.g. more than 3 billions of Web pages have been indexed by Google [7]. Moreover, human-created hierarchies are very general because they are required to fulfill the needs of generic subjects. Therefore, automatically-generated hierarchy is needed for supporting mobile searching. This is because automatically-generated hierarchies can provide intricate details of hierarchical structures at a more affordable cost than manually-created hierarchies.

4.1.2 Pre-defined Hierarchies

Work has been done on finding ways to deal with the problems of manually-created hierarchies. One of the aspects is to automatically classify information into manually pre-defined hierarchies (e.g., [78, 79]). Such methodologies can, no doubt, save lots of manual work but such hierarchies are not adaptable to varying search contexts and the rapidly changing amount of information on the Internet. It would be desirable if there is an automatically-generated hierarchy that adapts to individual subject search contexts and search results (e.g. concept hierarchy which is generated from search result directly).

4.1.3 Lexical Hierarchies

Another way to generate hierarchies automatically is the use of hierarchical structure of phrases that appear frequently in the set of documents. Such hierarchies of phrases are called lexical hierarchies and much work has been carried out in the generation of lexical hierarchies (e.g., [80, 81]). The work depends on frequently occurring words within phrases or noun compounds of a document set to discover the major ideas/concepts of that document in order to construct the hierarchy by making the more frequent terms as the parent concepts and less frequent terms as the child concepts. Accordingly, the nature and structure of lexical hierarchies are highly dependent on phrase discovery and the ranking of candidate terms. However, a long document that uses a term frequently could make the term be considered as a much better candidate term than it actually is, which, in turn, degrades the quality of the hierarchy and affects subject's selection of hierarchical terms.

4.1.4 Subsumption Hierarchies

Other than pre-defined and lexical hierarchies, another way to generate hierarchies automatically is the use of subsumption relationships between terms and the generated hierarchies are known as subsumption hierarchies. There are some terms that broadly occur among documents and define a general topic across documents. Some other terms co-occur with the terms of general topics while only occur in a few documents. Such terms represent the details or aspects of a general topic in a more specific manner. Subsumption hierarchies harness the power of "general to specific relationship" for generating hierarchies. A subsumption hierarchy [56] comes with the following characteristics:

1. It reflects the topics covered within the documents by associating terms selected from documents.
2. It is composed of cohesive association of terms. For any given association of subsumption hierarchy, a parent term is more general than its child terms.

3. For any given association of terms in a subsumption hierarchy, the following definition of subsumption relationship "P(X/Y) = 1 and P(Y/X) < 1" must hold if term X subsumes term Y. Thus, term X subsumes term Y if and only if the documents in which Y occurs are a subset of the documents in which X occurs.
4. Any given term could be able to subsume all of its descendents and a child may have more than one parent.
5. Subsumptions are transitive and therefore, a hierarchy can be built by recursively subsuming terms until there are no more subsumptions.

4.1.5 Justification for Using Subsumption Hierarchies

Subsumption was selected for generating our hierarchical access structure. Firstly, manual hierarchies cost much manual work and thus, it is impossible for handling much information on Internet (billions of Web pages). Secondly, pre-defined hierarchies cannot adapt to varying subject search contexts and changing information environment. It is clear that lexical and subsumption methodologies are relatively suitable for constructing our hierarchical access structure because they are automatically generated and can adapt to varying and changing of subject search contexts and search results. However, the two methodologies employ very different techniques for generating hierarchies. Lexical methodology relies on occurrence frequency of terms and subsumption looks at both terms and co-occurrence of terms among documents. Both methodologies work fairly well for homogeneously small document sets. However, lexical hierarchy may perform poorer than subsumption when it is applied to heterogeneous and large document sets [83]. This result may be due to the fact that choosing any most frequent subset from a heterogeneous and large document set can produce a large number of key terms and important relationships can be left out. As a result, subsumption was selected for generating our hierarchical access structure.

4.2 Pilot Study

The aim of this pilot study is to examine the feasibility of automatic generation of a concept hierarchy and how we could select candidate terms for constructing hierarchies.

4.2.1 Concept Hierarchy Construction

The concept hierarchy is built based on the subsumption relations between concepts. Assuming that each concept is represented by one term, which may have multiple words, term x is said to subsume term y (i.e. $x \supseteq y$) if $x \supseteq y$ *when* $p(x|y) \geq 0.8$ *and* $p(y|x) < 1$ [56]. According to Sanderson and Croft [56], the threshold value of 0.8 was derived after some experimental studies. Here, we use the same threshold without adjustment. Before the construction of a concept hierarchy from a document set, stop word filtering and stemming were carried out. The remaining issues are how terms were selected and whether terms can contain multiple words.

4.2.2 Term Selection

We use two terms weighting schemes: one is based on the well-known term frequency and inverse document frequency product (Equation 1) and another one is based on the ratio of term frequency and inverse document frequency (Equation 2)

$$w_{i,j} = t_{i,j} \times \log_2\left(\frac{N_j}{df_{i,j}}\right) \quad (1) \qquad w_{i,j} = \frac{t_{i,j}}{1 + \log_2\left(\frac{N_j}{df_{i,j}}\right)} \quad (2)$$

$w_{i,j}$ is the weight of term i in cluster j, N_j is the number of documents in j and $df_{i,j}$ is the number of documents in j that has term i, $t_{i,j}$ is the number of occurrence of term i in j. The rationale for Equation 2 is to select terms that occur in different documents, instead of penalizing these terms as in Equation 1. Another method of term extraction is based on words in document titles. We want to compare the performance using words in document titles with those selected by term weighting schemes. If the performance using words in document titles are significantly better, words in document titles will outweigh words in document content for term selection.

4.2.3 Multiple Word Terms

Due to the fact that some words are polysemous, the concept hierarchy may be grouping information in a variety of different contexts though, to a certain extent, clustering provides the (topical) contexts. For example, *Jordan* could refer to *Michael Jordan* and the country *Jordan*. Hence, the concept hierarchy is required to contain a sub-hierarchy to indicate the difference between *Michael Jordan* and the country *Jordan*. However, the ancestors of the node for *Jordan* in the hierarchy may not have anything in common to both Michael *Jordan* and the country *Jordan*. Thus, adding one or more additional word(s) to distinguish them in different contexts can reduce this type of semantic overloading. If we use a dictionary to store multiple word terms, there are problems related to the dictionary coverage and neologism in compound words. Instead, we propose to identify multiple word terms, using the association score [82], i.e. $association(x,y) = \log_2 \frac{p(x,y)}{p(x)p(y)}$ for the association of x before y, where $p(.)$ is the probability of the argument in all the documents of the same cluster. Since the accuracy of the estimation of the availability of a multiple word term depends on the number of the multiple word term in the document collection, multiple word terms with an occurrence lower than some threshold (i.e. 5 as in [82]), are discarded. Since the association score of every two consecutive words, x and y, is compared with the confidence level, only two word terms are discovered. For multiple word terms extracted from titles, typically their occurrence frequency in only the titles is very low. Hence, the association for these terms extracted from titles is based on the frequency derived from the body of the documents.

4.2.4 Preliminary Evaluation

We have evaluated our concept hierarchies in the same way as Sanderson and Croft [56], so that we can roughly compare our performance with theirs. In this preliminary evaluation, 30 Web documents were used for generating concept hierarchies. Five subjects (evaluated concept hierarchies derived from terms from *term frequency-inversed document frequency*) and three subjects (evaluated concept hierarchies derived from title terms) were presented with the parent concept, child concept and grand parent concept if there is one. The subjects were asked to judge whether the child concept was (i) an *aspect of* the parent, (ii) a *type of* its parent, (iii) *same* as its parent, (iv) *opposite* to its parent, or (v) *unrelated* to the parent. For comparative purposes, a concept hierarchy based on the selected terms but randomly subsumed (i.e. randomly grouped as parent and child conceptual pairs) together was built and evaluated. Table 4.2.4.1 shows the performance of the random hierarchy and our concept hierarchy. All performance measures were macro-averaged across different subjects, which are typically lower than the micro-averaged results.

The desirable parent-child relations are *aspect of* and *type of* relations. If we combine both performance measures (i.e. 71%), it is similar to the performance observed (i.e. 72%) by Sanderson and Croft [56]. The difference in the distribution lies in that fact that our concept hierarchy contains more *aspect of* relations instead of *type of* relations. For the *same* and *opposite* relations, the performance of our concept hierarchies is about the same as the random case. Compared with Sanderson and Croft [56], we have noticeably fewer (about 5%) of the *same* relations. However, we have just about the same amount of increase in *don't know* (or *unrelated*) relations compared with Sanderson and Croft [56]. This may be an indication that the term selection performance needs improvement, as the amount of *don't know* relations in the random case is also substantially higher than that of Sanderson and Croft [56]. If terms extracted from titles are used to build concept hierarchies, these terms appeared to have less *don't know relations* and much more *aspect-of* relations. The overall performance can be improved to 82%.

Table 4.2.4.1 Performance of Concept Hierarchies. Key: M: for concept hierarchy based on random (R) or subsumption (S) method. E: for terms extracted from the document body (D) or from the title (T). A, T, S, O and D are the relation types: aspect of, type of, same as, opposite and don't know, respectively. Column P is the overall performance (A+T).

Techniques		Classified Relation Types					
M	E	A	T	S	O	D	P
R	D	23%	11%	2%	4%	60%	34%
S	D	58%	13%	3%	2%	24%	71%
	T	80.71%	1.62%	0.24%	0%	17.43%	82.33%

4.2.5 Summary

We found that our derived concept hierarchy from extracted terms using *tf-idf (term frequency-inversed document frequency)* score yields a similar performance (71% of relationships are "aspect of" and "type of") compared with that of Sanderson and Croft (72% of relationships are "aspect of" and "type of"). One method to reduce the amount of *don't know* relations is to use more relevant terms. Using terms in titles is a viable alternative, where the desirable *parent-child* relations can be increased to about 82%. It would probably suggest that title terms and human-created summary terms outweigh other terms in term selection for constructing concept hierarchies.

4.3 Experimental Setup

4.3.1 Testing Data

The aim of this experiment is to test the performance of our hierarchical access structure in supporting access to search result using mobile devices. Fifty queries (i.e., 301-350) in the TREC-6 test collection were used.

According to Chapter 3, the subjects required tidy and textual information for supporting mobile search. They do not prefer having pictures because pictures occupy too much display area. TREC-6 consists of only textual data that comes without pictures and, therefore, it is suitable for testing our proposed hierarchical access structure.

The search engine (IRE) in this experiment performs similarly as the AT&T's search engine (Table 4.3.1.1). IRE generated fifty search result sets based on vector space model with pivoted document length normalization and TDN (Title (T) plus Description (D) plus Narrative (N)) queries without relevance feedback. TDN queries were used because TDN contain full description from TREC evaluators on the queries and have a better spread of query performance than other query types (Although TN had the best_spread of query performance, TN was not used in the experiment because it did not contain full description from TREC evaluators on the queries) (Table 4.3.1.2, Figure 4.3.1.1). Therefore, our subjects were able to have a closer understanding on the queries as TREC evaluators, which may increase the level of consistency of our subjects and TREC evaluators on the judgment of relevance of search results.

Table 4.3.1.1 Precision of AT&T and IRE on TREC-6 to TREC-8 Based on Vector Space Model with Pivoted Document Length Normalization and TDN Queries without Relevant Feedback.

Data	AT&T	IRE
TREC-6	21.79%	23.01%
TREC-7	21.82%	23.56%
TREC-8	27.28%	25.92%

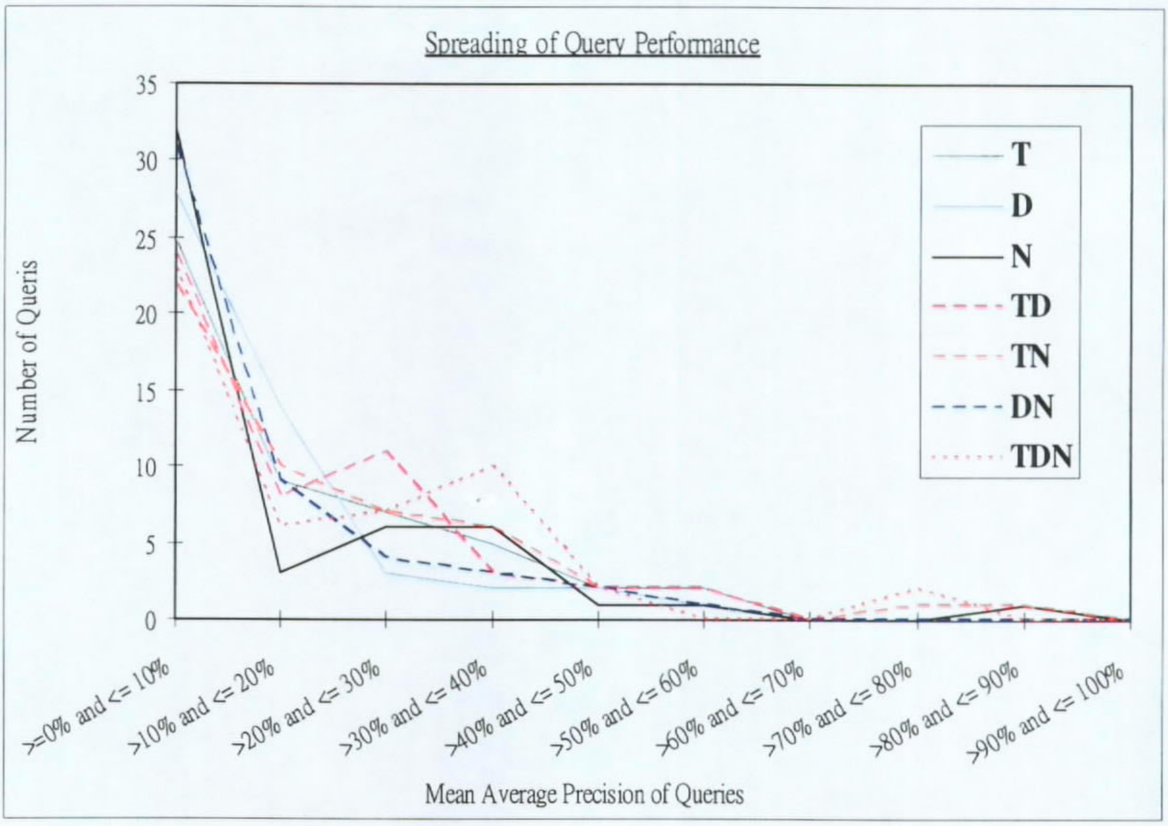


Figure 4.3.1.1: Query performance distribution.

Key: T, D and N are nature of queries where T means title, D means description and N means narrative.

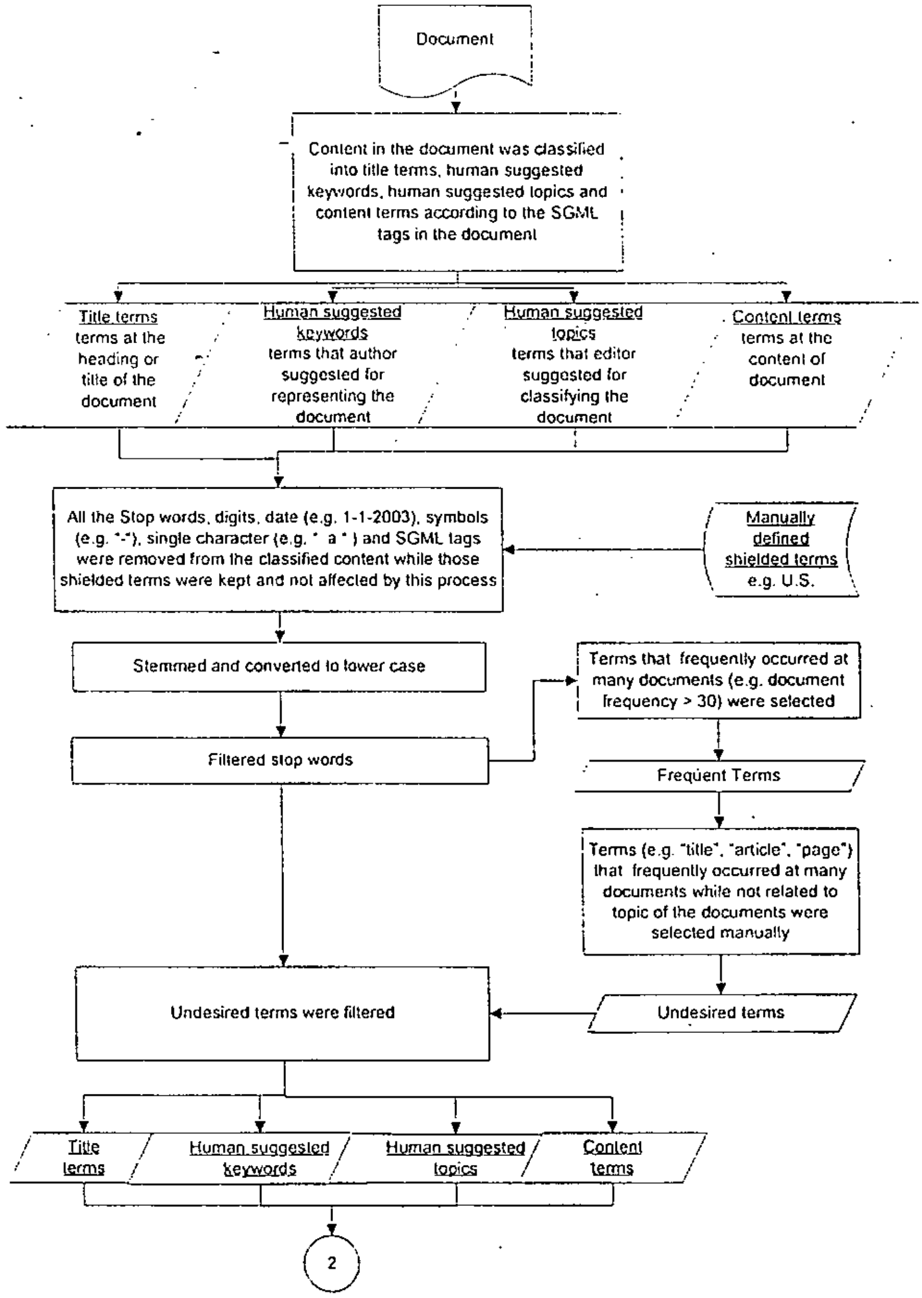
Table 4.3.1.2 Spreading of Query Performance

Key: T, D and N are nature of queries where T means title, D means description and N means narrative.

Precision Interval	T	D	N	TD	TN	DN	TDN
	Number of Queries						
>= 0% and <=10%	25	28	32	24	22	31	23
> 10% and <=20%	9	14	3	8	10	9	6
> 20% and <=30%	7	3	6	11	7	4	7
> 30% and <=40%	5	2	6	3	6	3	10
> 40% and <=50%	2	2	1	2	2	2	2
> 50% and <=60%	2	1	1	2	1	1	0
> 60% and <=70%	0	0	0	0	0	0	0
> 70% and <=80%	0	0	0	0	1	0	2
> 80% and <=90%	0	0	1	0	1	0	0
>90% and <=100%	0	0	0	0	0	0	0
Variance	60	83	95	59	47	91	52

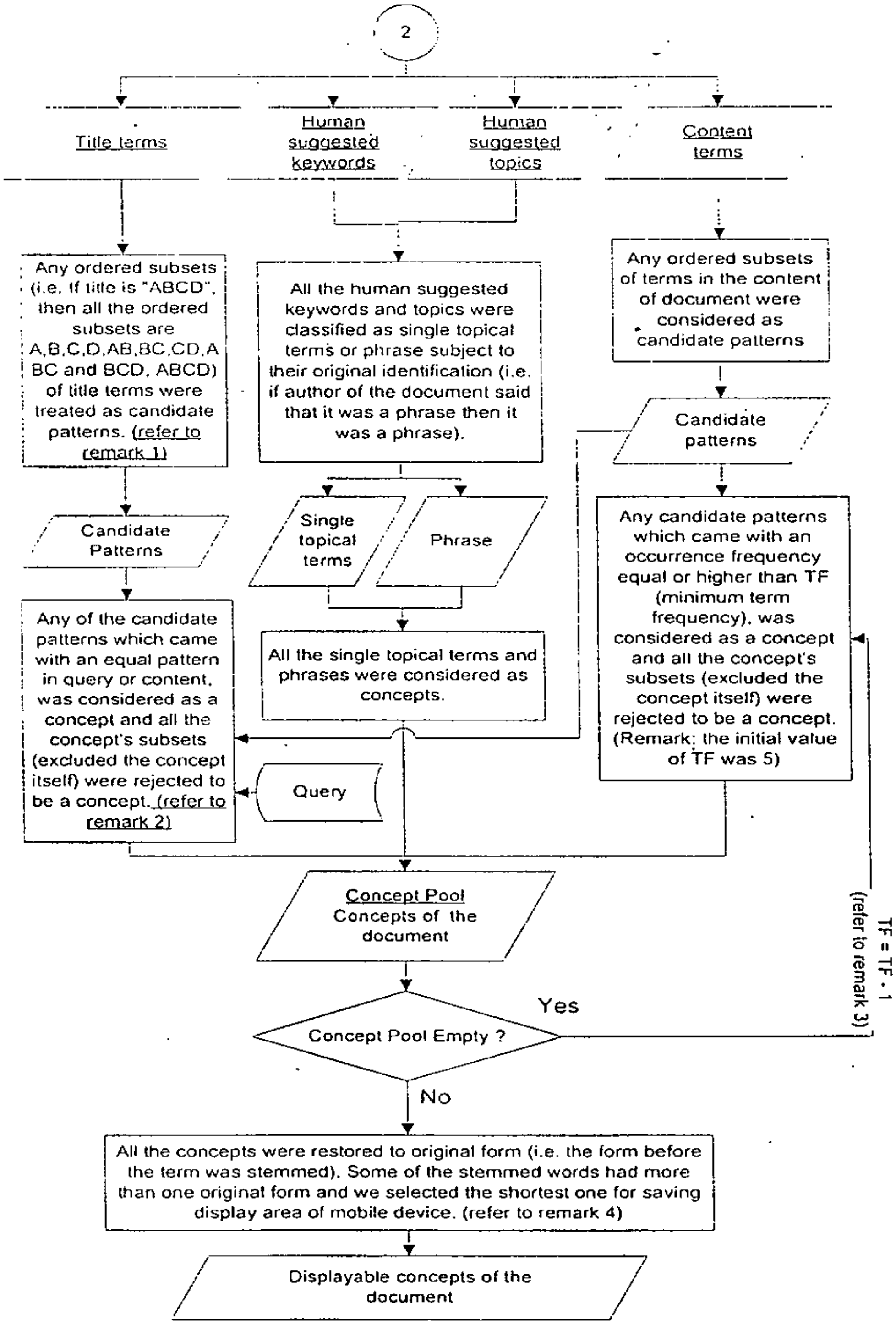
4.3.2 Construction of Concept Hierarchy

First of all, for each document in the retrieval list, words were selected as candidate terms according to the following flow chart (Flow chart 4.3.2.1):



Flow Chat 4.3.2.1: Candidate terms selection

The candidate terms were then processed with the procedures based on the following flow chat (Flow Chat 4.3.2.2):

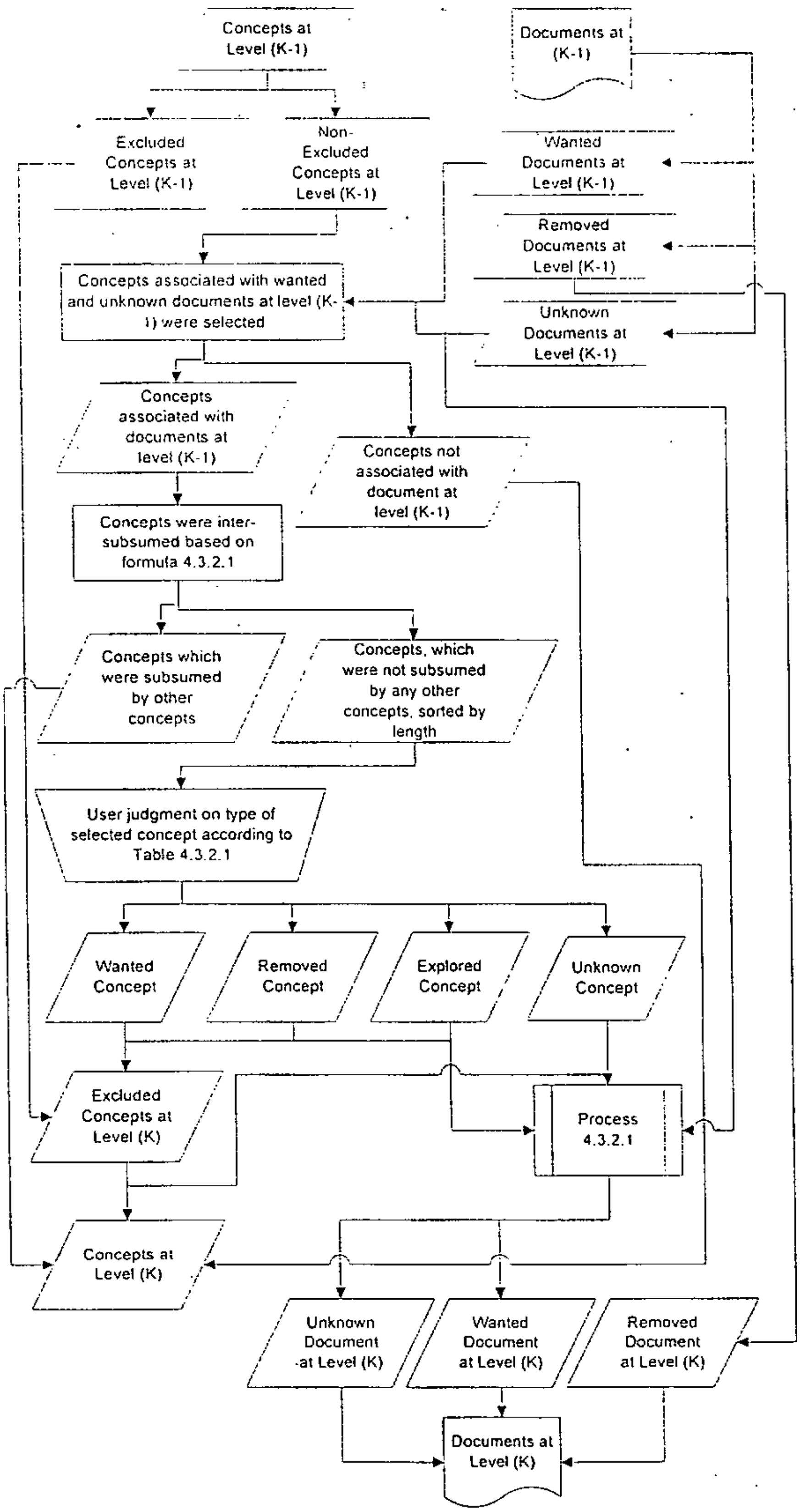


Flow Chat 4.3.2.2: Candidate terms pre-processing

Remarks on Flow Chat 4.3.2.2

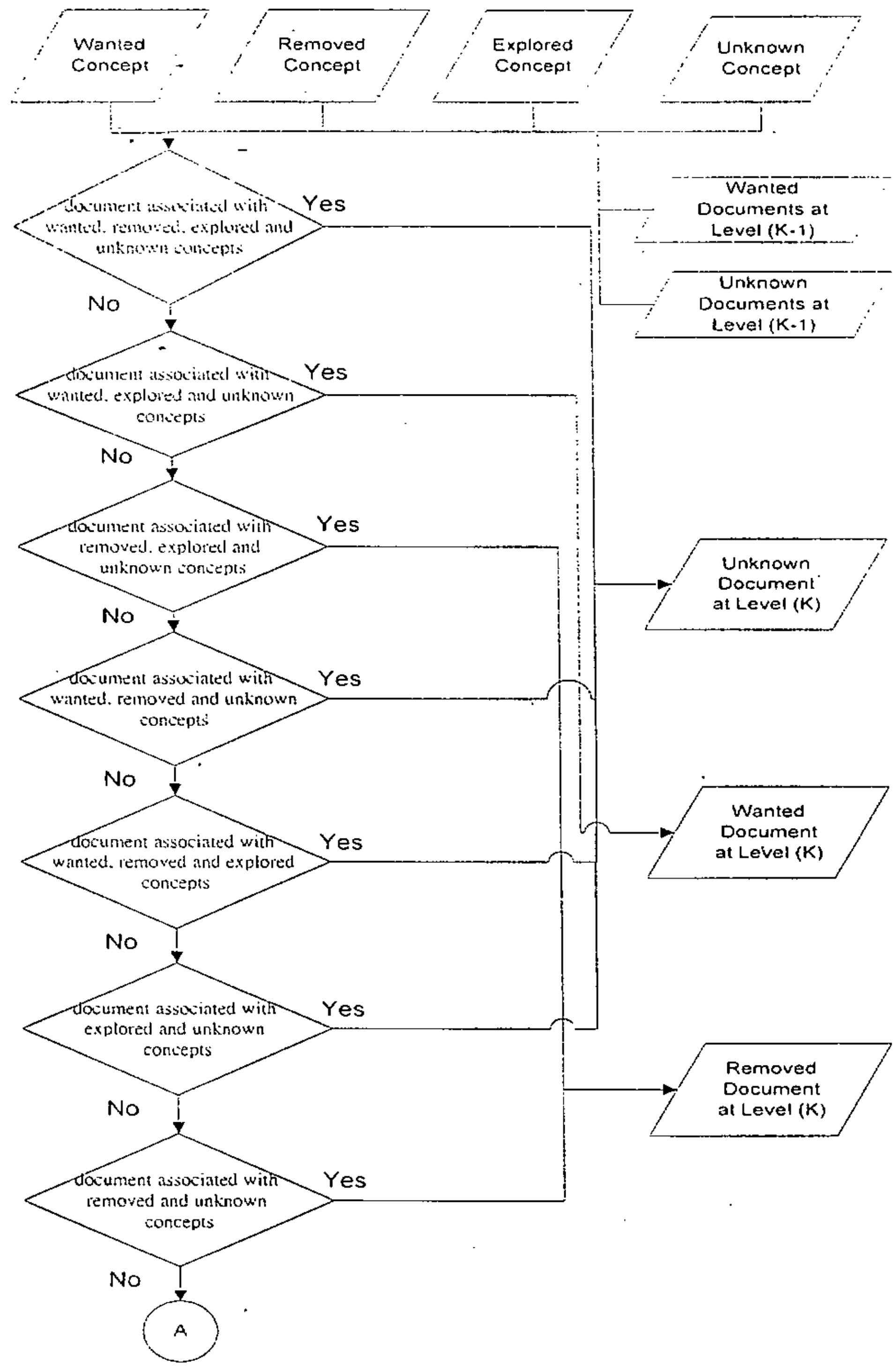
1. All the title terms were treated as topical terms of the document, as suggested by our pilot study.
2. Because there were only a few numbers of terms in a title; it was difficult to extract phrases (i.e. pattern with multiple terms) from title terms based on phrase extraction mechanism such as tf-idf. By matching candidate patterns with possible patterns, a query and a content not only confirmed the availability of the patterns but also reduced ambiguity in between query, title and content on understanding of concepts (e.g. the pattern words ABC in a title was treated as concept A and concept BC and the same pattern in a query was treated as concept ABC).
3. Certain concepts must be found when TF was relaxed to 1 because there was no empty document.
4. Because the stemmed words may be different from normal English words, it may cause understanding difficulties.

Concept Hierarchy Construction: The hierarchical information of concept hierarchy was generated based on $P(X/Y) = 1$, $P(Y/X) < 1$ (Formula 4.3.2.1). Each time when a subject requested to browse a given level of the concept hierarchy, concepts at the level are generated by a procedure based on the following flow chart (Flow chart 4.3.2.3), where K = requested level.

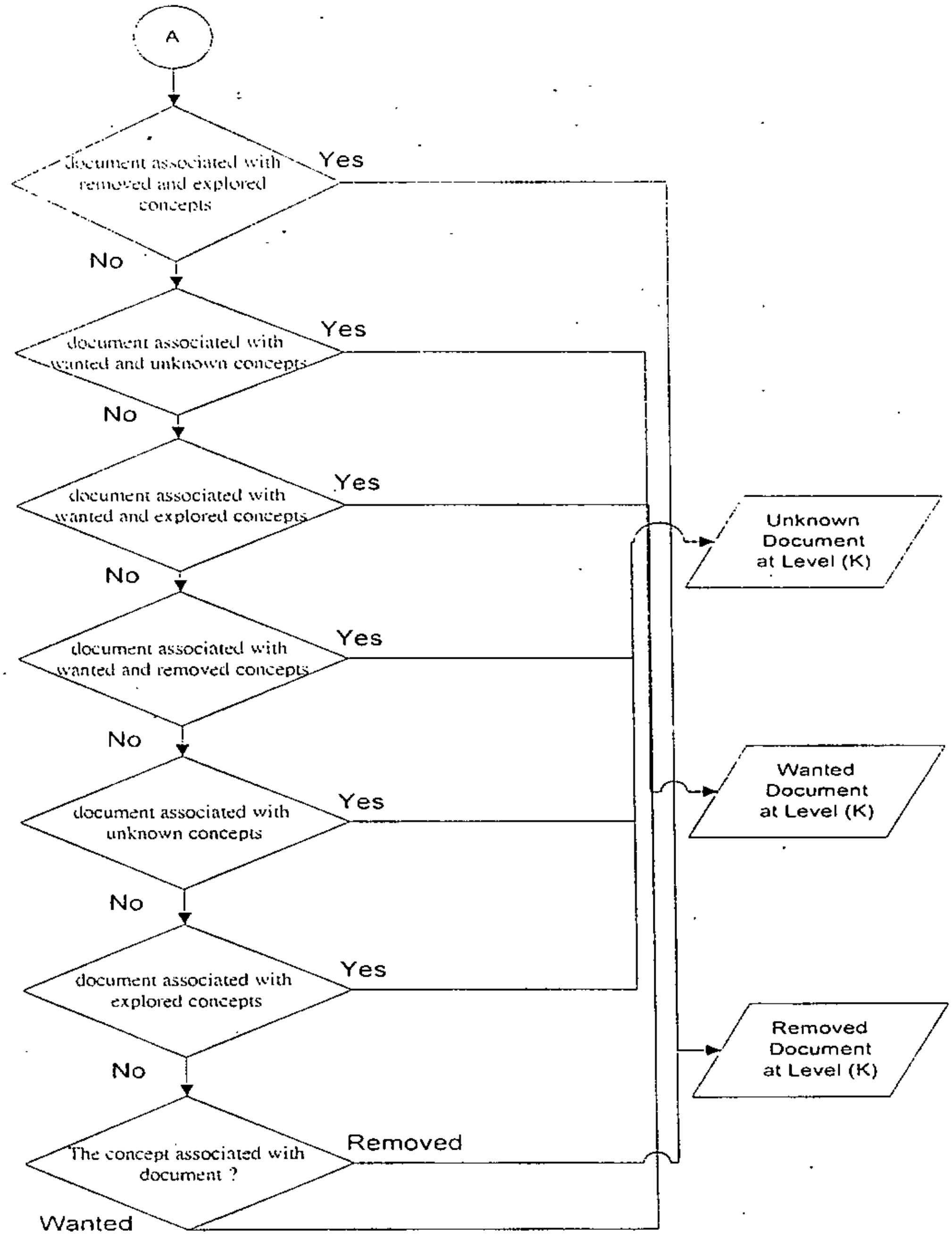


Remark: Type of document (Table 4.3.2.2)

Flow Chart 4.3.2.3: Concept hierarchy construction



Process 4.3.2.1: Process for judging the type of document (Part I)



Process 4.3.2.1: Process for judging the type of document (Part II)

Table 4.3.2.1 Possible Judgments on the Concept

Types of judgment on the concept	Remark
Wanted	All the documents, which contain these concepts and available at this layer, are relevant
Removed	All the documents, which contain these concepts and available at this layer, are irrelevant
Explored	Some of the documents, which contain these concepts and available at this layer, are relevant and others are irrelevant
Unknown	None of the above

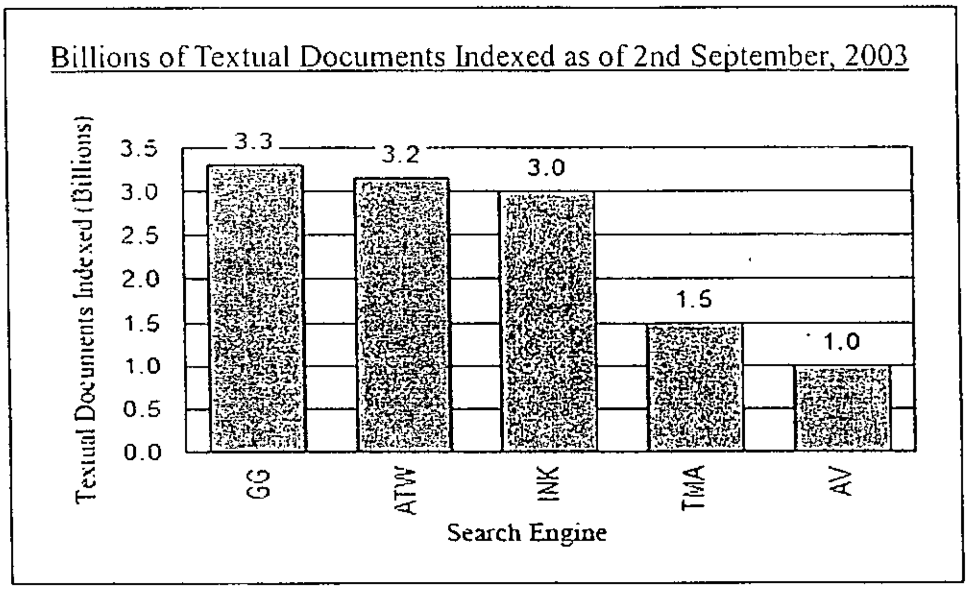
Table 4.3.2.2 Possible Types of Document

Type of document	Remark
Wanted	Subject is sure that the document is relevant
Removed	Subject is sure that the document is irrelevant
Unknown	Subject do not sure whether the document is relevant or not

4.3.3 Information Accessing Structures

According to Chapter 3, list access structures were more common than hierarchical access structures. Thus, many searching systems present their search results through a list. In order to examine the feasibility of our hierarchical access structure on supporting mobile information access in daily life, a list access structure was chosen to compare with our hierarchical access structure from existing searching systems used by the general public.

According to figure of Search Engine Watch on 2nd September, 2003 (Figure 4.3.3.1), Google indexed most textual documents when compared with other search engines, and in turn, Google had the biggest information based on textual documents compared with other search engines. Also, such type of textual information was similar to the type of information (i.e. text documents from TREC-6) used in this experiment.



Key: GG=Google, ATW=AllTheWeb, INK=Inktomi, TMA=Teoma, AV=AltaVista.

Figure 4.3.3.1 Amount of textual documents indexed, which include HTML files, text documents, PDF files, Microsoft Office documents and other similar files. Image and multimedia files are not included [84].

Other than the consideration of type and size of information, the likelihood that the general public will use the searching system and public's preference on the services of that system were also taken into consideration. According to a survey of Nielsen//NetRating in January 2003, more than

60,000 home and work surfers were measured and most of the people preferred searching information using Google and Google also has the longest total search hours. Users probably did more searching via Google than other search engines (Figure 4.3.3.2 and 4.3.3.3)

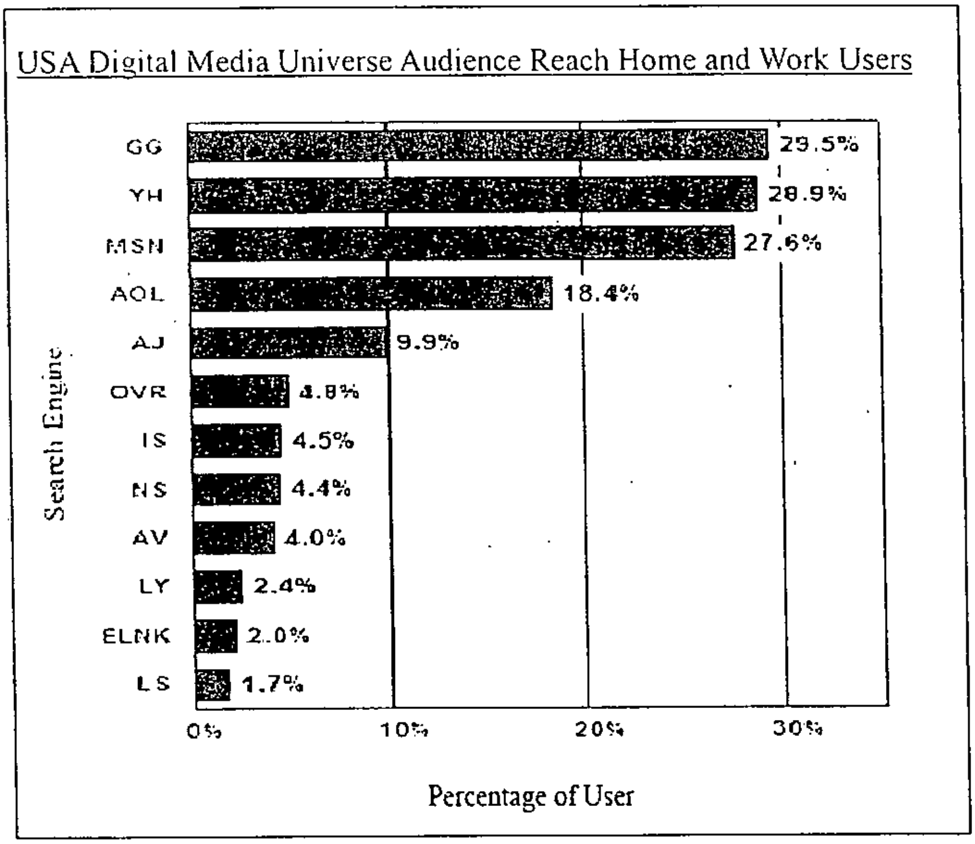
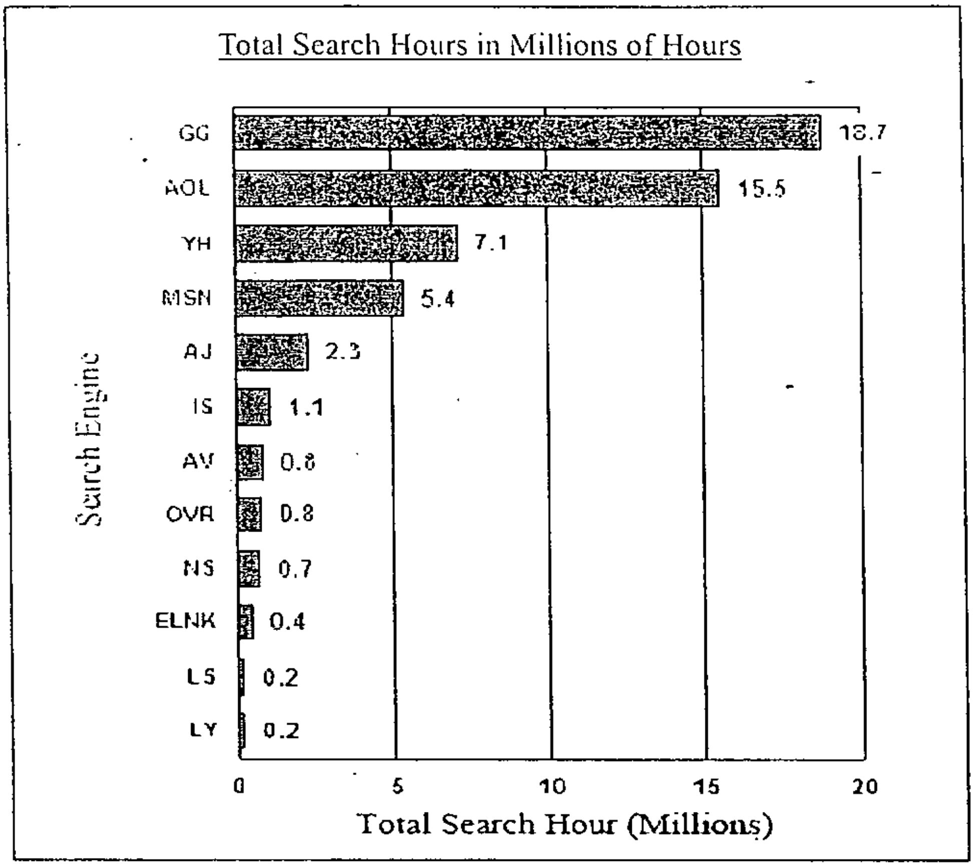


Figure: 4.3.3.2 USA digital media universe audience reach home and work users, January 2003 [85]



Key: GG=Google, YH=Yahoo, MSN=MSN, AOL=AOL, AJ=Ask Jeeves, OVR=Overture (GoTo), IS=InfoSpace, NS=Netscape, AV=AltaVista, LY=Lycos, ELINK=EarthLink.com, LS=LookSmart

Figure: 4.3.3.3 Total search hours in millions of hours, January 2003 [85]

Search Engine Watch also invited public to nominate search engines in January 2003. Winners of the nomination round were then voted. People were allowed to name both a "winner" (Table 4.3.3.1) and a second place "runner-up." (Table 4.3.3.2). A total of 556 valid votes were received and Google voted as the most preferred search engine.

Table 4.3.3.1: "Winner" or First Choice Search Engine Vote [84]

Search Engine	Vote (%)
Google	65%
AllTheWeb(Fast)	13%
Teoma	7%
Yahoo	7%
AltaVista	6%
MSN	1%
Lycos	1%

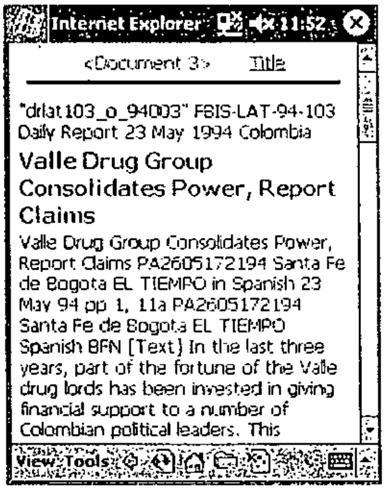
Table 4.3.3.2: "Second Place" or Second Choice Search Engine Vote [84]

Search Engine	Vote (%)
AllTheWeb(Fast)	29%
Yahoo	19%
Google	19%
AltaVista	13%
Lycos	8%
Teoma	8%
MSN	5%

Accordingly, we tried to explore the interfaces of the PDA versions of the nominated search engines. It was found that only Google provided list access structure for PDA (Table 4.3.3.3). Therefore, in our case, Google's access structure for PDA was treated as the most suitable candidate for comparing with ours if the access structure of Google's PDA version can also be adopted according to the findings in Chapter 3.

Table: 4.3.3.3: Search Engine Status on Mobile Services

Search Engines	Service for Mobile Devices
Google	WAP and PDA
AllTheWeb(Fast)	Only found the version for WAP
Teoma	Can't be found and we assumed no such services
Yahoo	Only catalog found at PDA version and we assumed no list structure
AltaVista	Only found a testing version for WAP but it did not open to public when we were doing this analysis
MSN	Only provide catalog at PDA version and we assumed no list structure
Lycos	Can't found PDA version and we assumed no such services



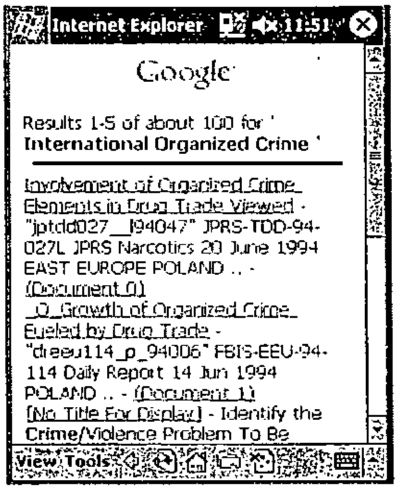


Figure 4.3.3.4 List access structure

The PDA version of Google access structure (Figure 4.3.3.4) comes with titles (headings) and abstracts that were identified, in Chapter 3, as necessary features for list access structures because such features fulfill both user preference and requirements (Table 3.3.2.1). The Google access structure also comes with other features which fulfill users' practical requirements, namely, "Keywords are bolded", "Using text-based links", "Black and white in color of background and fonts" and "Provided total number of search results". However, there are no features at the Google access structure for "Jumping back to higher level" and "Go to the next or previous page" while such features were important according to the findings reported in Chapter 3. Therefore, such features were added and the final access structure was used for comparing with our hierarchical access structure.

As for our hierarchical access structure (Figure 4.3.3.5), the design comes with all the necessary features found in both the user preference survey and the Contextual Inquiry (refer to section 3.4 and Table 3.3.2.1), and some of the important features found in Contextual Inquiry (Table 3.3.2.1). Those necessary features are "Provide facility for jumping back to higher level (backward and forward functions)", "Provide facility for jumping in between different catalogs (functions for directly jumping in between different status, e.g. page number list <1,2,3,4,5>)", "Articles were classified into different catalogs (categorize information)" and "List of titles together with abstract in each of the catalog (abstract and heading for browsing search result)". Those important features are "Always fit information on a single screen (i.e. avoid scrolling)" , "Grouping on function keys" , "Black and white background and fonts" , "Page break/split long pages for avoiding scrolling" , "Sorting items in alphabetic order" and "Show total number of search result or available document or pages". Therefore, our hierarchical access structure can be treated as fulfilling findings in Chapter 3 in general. In the following section, our hierarchical access structure is compared with the modified list access structure from Google's PDA version in order to demonstrate the feasibility of our hierarchical access on supporting mobile information access.

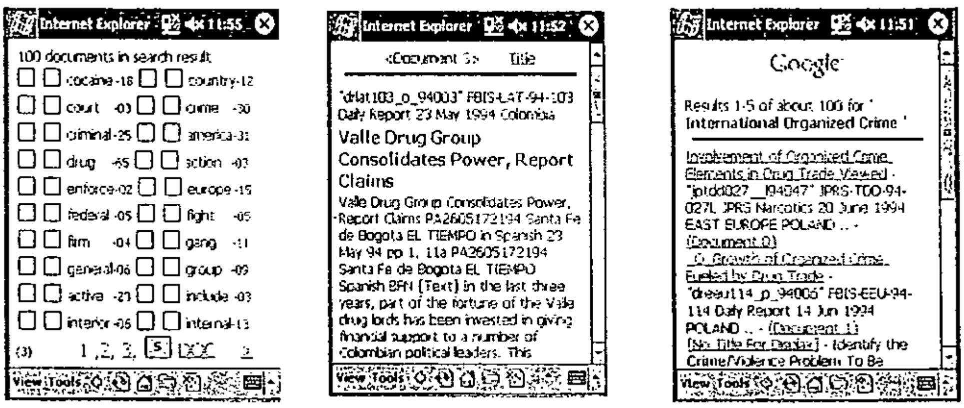


Figure 4.3.3.5 Hierarchical access structure

4.3.4 Experimental Context

Subjects:

Four experienced PDA users (Appendix A.1 – Subject 28 - 31) were invited to evaluate the performance of hierarchical access structure and the list structure as a baseline.

Equipment:

A Toshiba Pocket PC (e740) came with Internet Explorer (Browser) was connected to the server which allows subjects to do searching using two given access structures (List and Hierarchical).

Tasks:

Fifty queries (TREC–6 query 301 to 350) were sorted according to their precision of top 100 search results collected from IRE. Each subject was assigned with twenty-five queries for accessing with the list structure and twenty-five queries for accessing with hierarchical structure (Table 4.3.4.1). As queries were ranked based on their precision and they were assigned to the list and hierarchical structures alternately, there were similar number of good queries and poor queries for accessing with list and hierarchical structure.

Table 4.3.4.1 Tasks Assignment Based on Query Performance

Query ID	IRE Precision	Subject 1 and 4	Subject 2 and 3
301	0.8	Access via hierarchical structure	Access via list structure
324	0.74	Access via list structure	Access via hierarchical structure
343	0.49	Access via hierarchical structure	Access via list structure
337	0.41	Access via list structure	Access via hierarchical structure
311	0.4	Access via hierarchical structure	Access via list structure
335	0.4	Access via list structure	Access via hierarchical structure
302	0.38	Access via hierarchical structure	Access via list structure
333	0.38	Access via list structure	Access via hierarchical structure
313	0.37	Access via hierarchical structure	Access via list structure
341	0.35	Access via list structure	Access via hierarchical structure
332	0.35	Access via hierarchical structure	Access via list structure
306	0.34	Access via list structure	Access via hierarchical structure
321	0.33	Access via hierarchical structure	Access via list structure
326	0.33	Access via list structure	Access via hierarchical structure
331	0.29	Access via hierarchical structure	Access via list structure
349	0.27	Access via list structure	Access via hierarchical structure
304	0.27	Access via hierarchical structure	Access via list structure
323	0.27	Access via list structure	Access via hierarchical structure
319	0.26	Access via hierarchical structure	Access via list structure
307	0.22	Access via list structure	Access via hierarchical structure
316	0.22	Access via hierarchical structure	Access via list structure
329	0.16	Access via list structure	Access via hierarchical structure
330	0.14	Access via hierarchical structure	Access via list structure

346	0.14 Access via list structure	Access via hierarchical structure
350	0.14 Access via hierarchical structure	Access via list structure
334	0.13 Access via list structure	Access via hierarchical structure
317	0.11 Access via hierarchical structure	Access via list structure
322	0.1 Access via list structure	Access via hierarchical structure
342	0.1 Access via hierarchical structure	Access via list structure
327	0.1 Access via list structure	Access via hierarchical structure
303	0.1 Access via hierarchical structure	Access via list structure
345	0.09 Access via list structure	Access via hierarchical structure
325	0.09 Access via hierarchical structure	Access via list structure
339	0.08 Access via list structure	Access via hierarchical structure
315	0.07 Access via hierarchical structure	Access via list structure
314	0.07 Access via list structure	Access via hierarchical structure
340	0.07 Access via hierarchical structure	Access via list structure
347	0.06 Access via list structure	Access via hierarchical structure
336	0.06 Access via hierarchical structure	Access via list structure
310	0.06 Access via list structure	Access via hierarchical structure
338	0.05 Access via hierarchical structure	Access via list structure
305	0.05 Access via list structure	Access via hierarchical structure
328	0.05 Access via hierarchical structure	Access via list structure
320	0.05 Access via list structure	Access via hierarchical structure
312	0.05 Access via hierarchical structure	Access via list structure
308	0.04 Access via list structure	Access via hierarchical structure
318	0.04 Access via hierarchical structure	Access via list structure
344	0.04 Access via list structure	Access via hierarchical structure
348	0.02 Access via hierarchical structure	Access via list structure
309	0.01 Access via list structure	Access via hierarchical structure

4.3.5 Experimental Procedure

List Access Structure:

1. A subject was allowed to go into details of the query and requested to make sure that s/he understood his/her tasks (Figure 4.3.5.1)
2. List (Figure 4.3.5.2) access structure was shown in order to display titles and abstracts of the documents.
3. Each of the titles was evaluated (relevant/irrelevant) by the subject sequentially (Figure 4.3.5.2). The subject was allowed to go into details (Figure 4.3.5.4) of the document if s/he required more information for judgment.

Hierarchical Access Structure:

1. A subject was allowed to go into details of the query and requested to make sure that s/he understood his/her tasks (Figure 4.3.5.1)
2. Hierarchical (Figure 4.3.5.3) access structure, which was generated by Procedure 4.3.2.1, was shown.

3. Each of the concepts was evaluated, according to Table 4.3.2.1 and irrelevant documents were eliminated according to subject's judgment of the concepts (refer to Table 4.3.2.2 and Process 4.3.2.1)

4. Subjects visited the list of titles (Figure 4.3.5.2) of the remaining documents when s/he believed that the remaining set of documents should better to be judged by using the title list (Figure 4.3.5.2) and even by the details of the content (Figure 4.3.5.4) or there were no concepts available at the visited layer of the hierarchy (Figure 4.3.5.5).

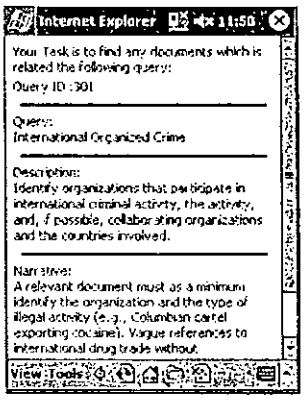


Figure 4.3.5.1 Query

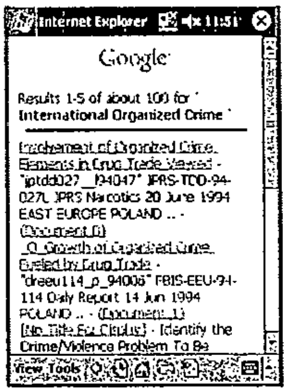


Figure 4.3.5.2 Title list

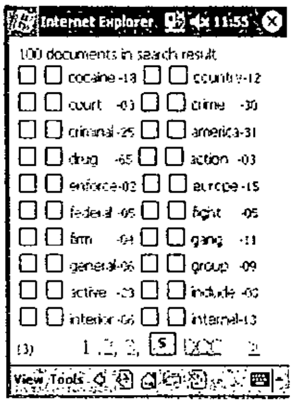


Figure 4.3.5.3 Concepts

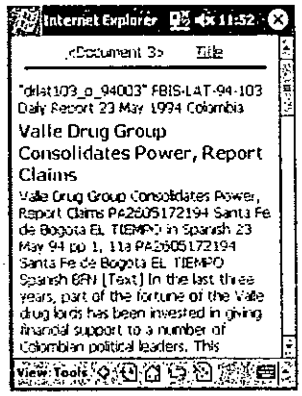


Figure 4.3.5.4 Document

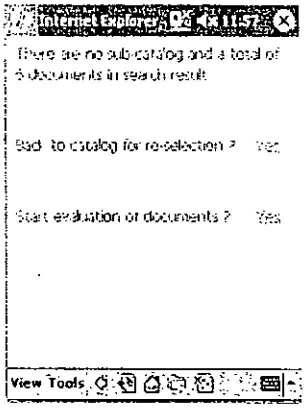


Figure 4.3.5.5 No concept at the requested layer can be found

4.4 Result and Discussion

We further explored various issues on the feasibility of using hierarchical access structure for mobile searching.

4.4.1 Access Time

The first issue explored was the timing issue. According to Contextual Inquiry in Chapter 3 [AD – 1.1.1.2.2], fast accessing to information using mobile devices was one of the important issues that was raised by mobile users.

Thus, a faster accessing speed to information can directly save users' treasurable time, while they were on the road or out of the office. A faster accessing speed to information can also help mobile users to save expenses on mobile communication. This is because mobile operators usually charge their clients based on the usage time of mobile services. A faster accessing speed can reduce the time of communication, which, in turn, saves the cost of communication. Therefore, we explore the feasibility of hierarchical access for supporting mobile search with referring to the timing issues.

How much time does a mobile user need for reaching a document? Reaching means to read any portion of the document, such as a document's title, body, or any concepts from the document where the users can have a perception on the document. It was assumed that accessing information using concept hierarchy was faster than using list. It was because concept hierarchy organized information into hierarchical structure where subjects browsed the general idea of a group of documents or child concepts via a single parent concept quickly. However, subjects have to browse titles of each of the document sequentially via lists. The analysis was carried out by calculating the average time used for reaching a single document (Figure 4.4.1.1) and the details of calculation are as follows:

Let:

1. AvgAccessTime-N be the average time for accessing search result of query N among the subjects
2. AvgReachedDoc-N be the average number of documents reached in the search result of query N among the subjects
3. TimePerDoc-N be AvgAccessTime-N / AvgReachedDoc-N
4. So, the average time for reaching a document is the average value of TimePerDoc-N across query 301 to 350 (i.e. N = 301 to 350)

According to the average time for reaching a document, it was found that subjects required much more time for reaching a document by using a list structure (the average time for reaching a document = 20.84 seconds/document) than via hierarchical access structure (the average time for reaching a document = 8.24 seconds/document). As a result, hierarchical access structure can be roughly considered as a possible access structure that can allow mobile users to reach to documents quickly using mobile devices. The faster accessing speed of hierarchical access structure in comparison with the slower accessing speed of list access structure is because each of the concepts at the concept hierarchy is relating to one or usually more than one document. As a result, a number of documents can be considered as reached when a concept is reached. As for list access structure, only one document can be reached for each of the titles in the list. Thus, the hierarchical access structure provides a summary on the underling document set where users can reach certain concepts from a number of documents by browsing only a small amount of information and use a small period of time.

However, hierarchical access structure cannot be considered as feasible for mobile information access based on timing factors if the faster accessing speed to documents resulted in inefficiency of useful document retrieval (i.e. those user desired documents / related documents). It is possible for the above problem because conceptual terms provided by hierarchical access structure may not be able to provide information as rich as those provided by titles and abstracts in the list structures. As a result, mobile users may not be able to select as many desired / related documents via our hierarchical access structure even they can be faster in reaching certain concepts of the documents. As a result, we further explored the feasibility of hierarchical access by analysing the efficiency of finding the desired /

related documents based on timing factor and the details of calculation are as follows:

Let:

1. AvgAccessTime-N be the average time for accessing to search result of query N among the subjects
2. AvgDesiredDoc-N be the average number of desired documents found in the search result of query N among the subjects
3. AvgRelatedDoc-N be the average number of related documents found in the search result of query N among the subjects
4. TimePerDesiredDoc-N be AvgAccessTime-N / AvgDesiredDoc-N
5. TimePerRelatedDoc-N be AvgAccessTime-N / AvgRelatedDoc-N
6. So, time required for finding a desired document is the average value of TimePerDesiredDoc-N across query 301 to 350 (i.e. N = 301 to 350)
7. and, time required for finding a related document is the average value of TimePerRelatedDoc-N across query 301 to 350 (i.e. N = 301 to 350)

Generally, the time required for finding a related document via hierarchical access structure (225.88 seconds /document) was shorter than that of list structure on average (272.47 seconds /document). The same situation was found at finding the desired documents where the time via hierarchical access was 126.46 seconds per document and via list was 144.20 seconds per document. The encouraging results suggest that hierarchical access structure is feasible for helping mobile users to identify desired or related documents quickly, which, in turn, saves the cost of communication.

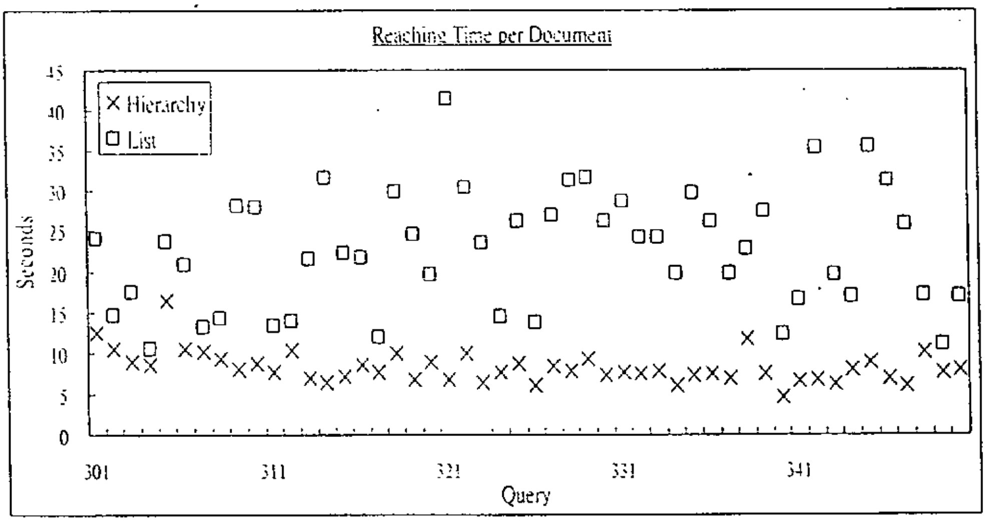


Figure 4.4.1.1: Average time to reaching a document

4.4.2 Efficiency of Data Usage

The second issue explored was efficiency of data usage. It is well-known that the cost to transfer a unit of data via wireless network is more expensive than one via wired network. The inefficiency on data usage not only wastes mobile network resources and increases network loading but also increases the cost of communication of mobile users. If the amount of data transfer can be decreased, then the cost of communication for mobile searching can also be decreased and searching via mobile devices will be much efficient. Our analysis was carried out by analysing the average amount of data used for reaching a document (Figure 4.4.2.1). It was found that subjects, by averaging all the queries, required downloading much more data from the server for reaching a document with a list access structure (1.49KB/document) than a hierarchical access structure (0.48KB/Document). As hierarchical access is a possible way that allows mobile users to reach documents with a small amount of data transfer, it would possibly be one of the proofs that illustrates the feasibility of using hierarchical access structure for mobile information access. In this way, we further explored the efficiency of finding desired / related documents in terms of data transfer. It was found that users need less data for finding a desired /related document via our hierarchical access structure (data for finding a desired document = 7.35KB/document, data for finding a related document = 13.14KB/document), in comparison with list structure (data for finding a desired document = 10.29KB/document, data for finding a related document = 19.45KB/document). Therefore, we conclude that our hierarchical access structure is feasible to be used for mobile information access with regarding the efficiency of data usage because it allows documents to be accessed with a small amount of data and at the same time provides an effective way for mobile users to find the desired or related documents using mobile devices.

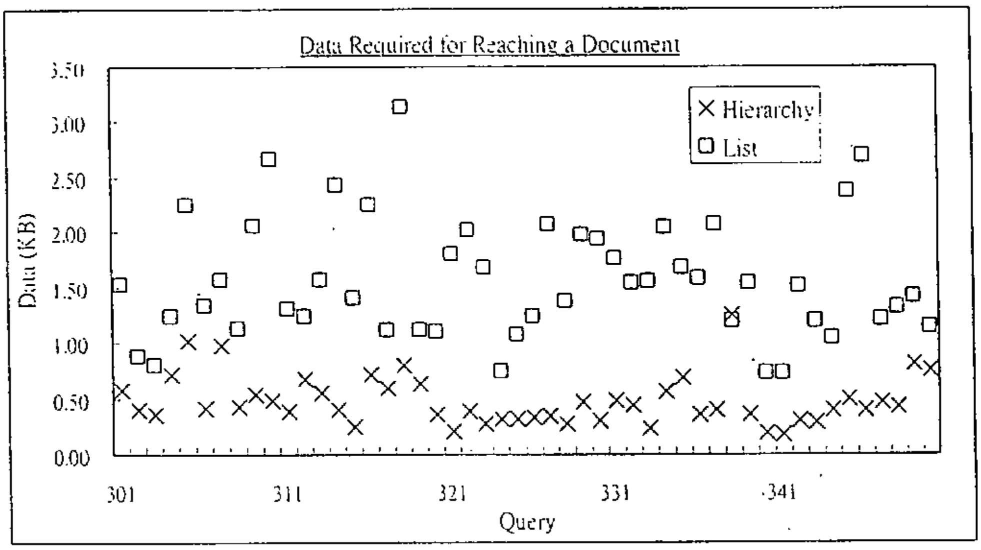


Figure 4.4.2.1 Average amount of data consumed for evaluating one document

4.4.3 Performance of Searching Documents in Search Result

We measured the search performance based on the following dimensions:

Firstly, we explored and compared performance on the remaining document set (non-removed documents after accessing to the hierarchy) and the document set presented at the list structure (i.e. the search results from search engine) based on recall (Figure 4.4.3.1), precision (Figure 4.4.3.2) and F1-measure (Figure 4.4.3.3). It was found that the average precision (20.64%), of the remaining document list from the hierarchical access structure was relatively higher than that of list (average precision = 20.28%). This suggests that it is possible to have a better resulting list of search results via the elimination of undesirable documents using concept hierarchy. However, the average recall (34.90%) and average F1-measure (19.14%) of hierarchical access is worse than that of list (average recall = 39.50% and F1-measure = 19.77%).

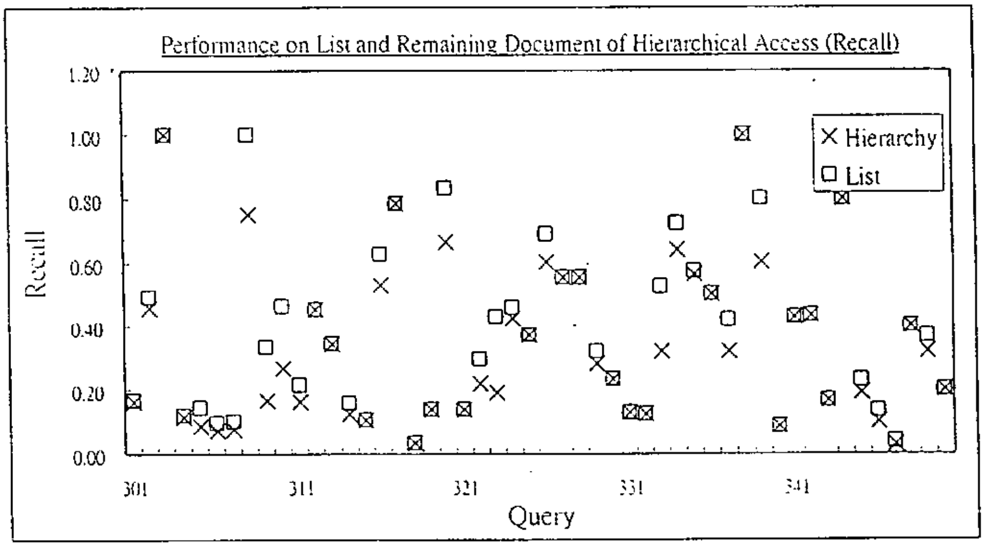


Figure 4.4.3.1 Performance on search result (list) from IRE and remaining document of hierarchical access (recall)

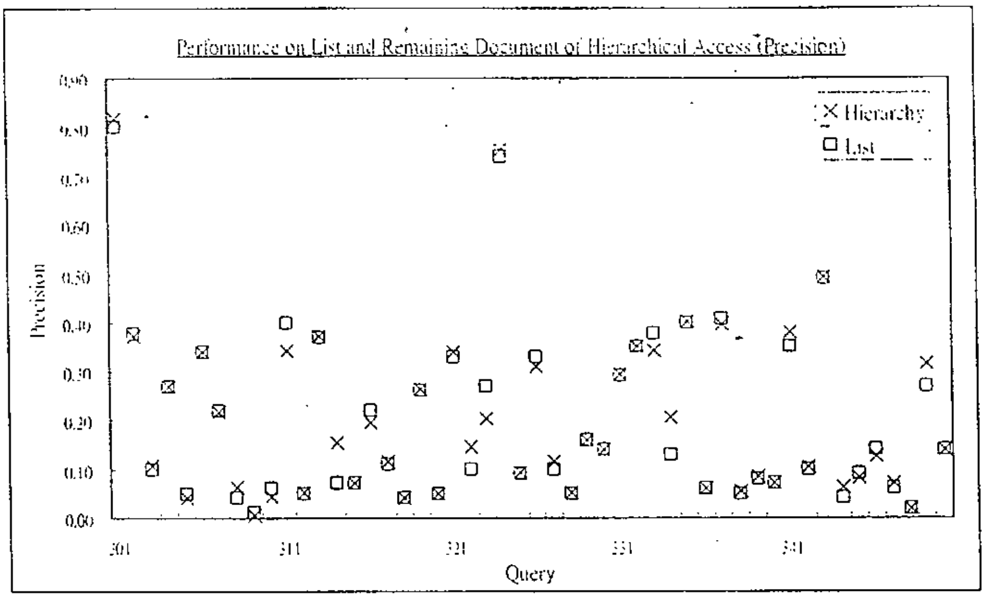


Figure 4.4.3.2 Performance on search result (list) from IRE and remaining document of hierarchical access (precision)

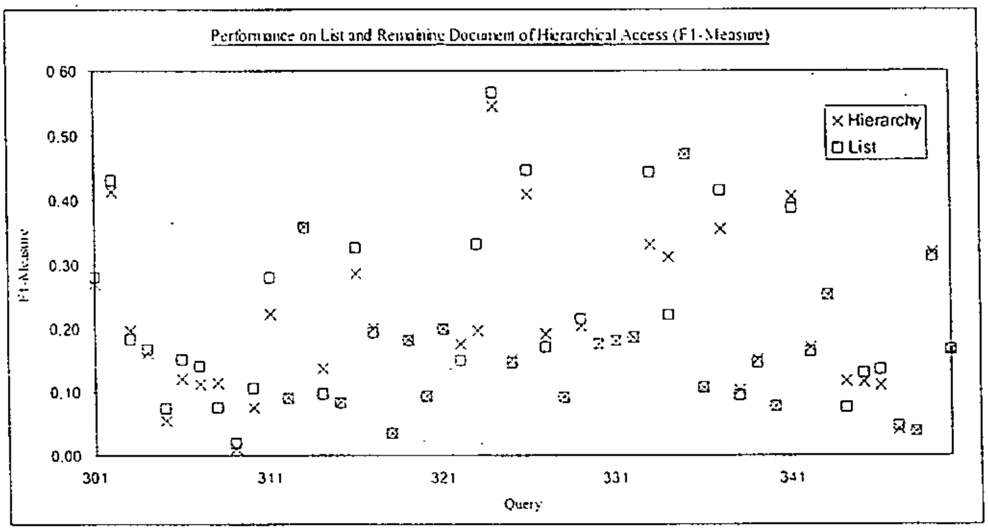


Figure 4.4.3.3 Performance on search result (list) from IRE and remaining document of hierarchical access (F1-measure)

However, such resulting list of documents is just the available document set that mobile users can access for finding relevant / desired documents, and it was possible that mobile users may not be able to access the whole list. Therefore, we further explored the performance of our hierarchical access structure on the attended document set (i.e. the set of documents which the user had visited to the title or content of documents) in order to investigate the performance on the extent that mobile users will actually attend. It was found that even though the average precision on the attended document set of hierarchical structure (32.30%) was slightly higher than that of list (30.31%) (Figure 4.4.3.4), the average recall (22.94%) and the F1-measure (26.83%) of the attended document set of hierarchical structure were lower than that of list structure (Recall = 28.63%, F1-measure = 29.45%) (Figure 4.4.3.5, Figure 4.4.3.6). The lower average recall was probably due to the smaller attended document set of hierarchical structure (34.23 documents in average) than that of list structure (43.80 documents in average). So, fewer related documents found using hierarchical access structure. The fewer attended documents of hierarchical structure might be due to the reason that the users had access to the concept hierarchy before the remaining document set while users can directly access the document set of list when they were starting to search for information. Given the similar base time for searching in both access structures, the time for accessing the remaining document set of hierarchical structure was less than that of list and possibly results in fewer attended document in the remaining document set than that of the document at list.

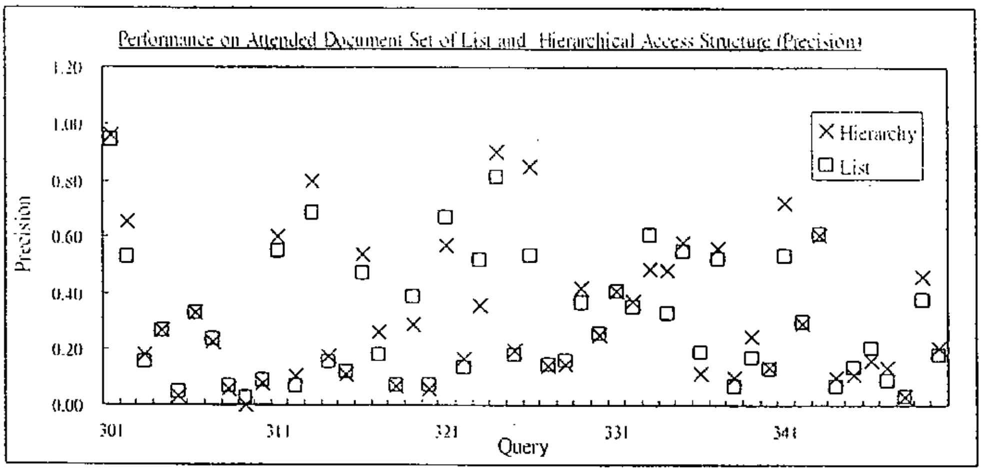


Figure: 4.4.3.4 Performance on attended document of search result (list) from IRE and hierarchical access structure (precision)

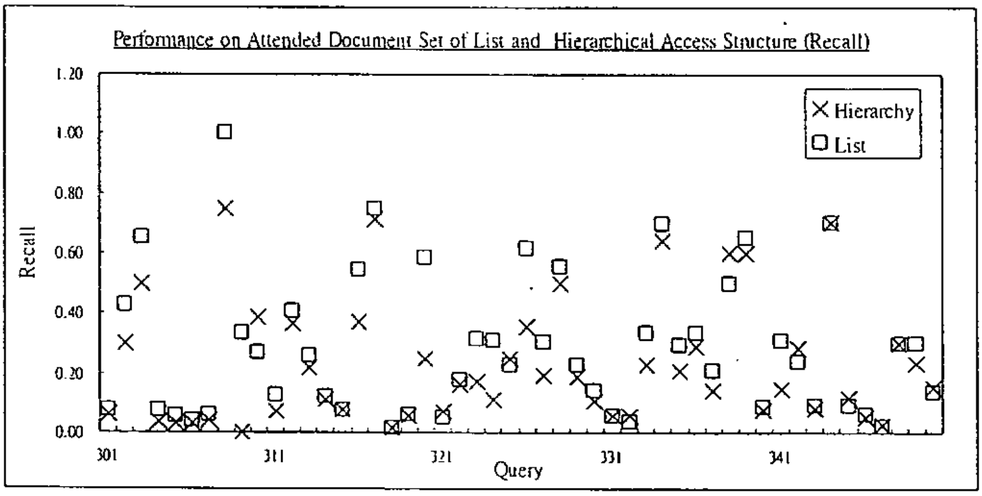


Figure: 4.4.3.5 Performance on attended document of search result (list) from IRE and hierarchical access structure (recall)

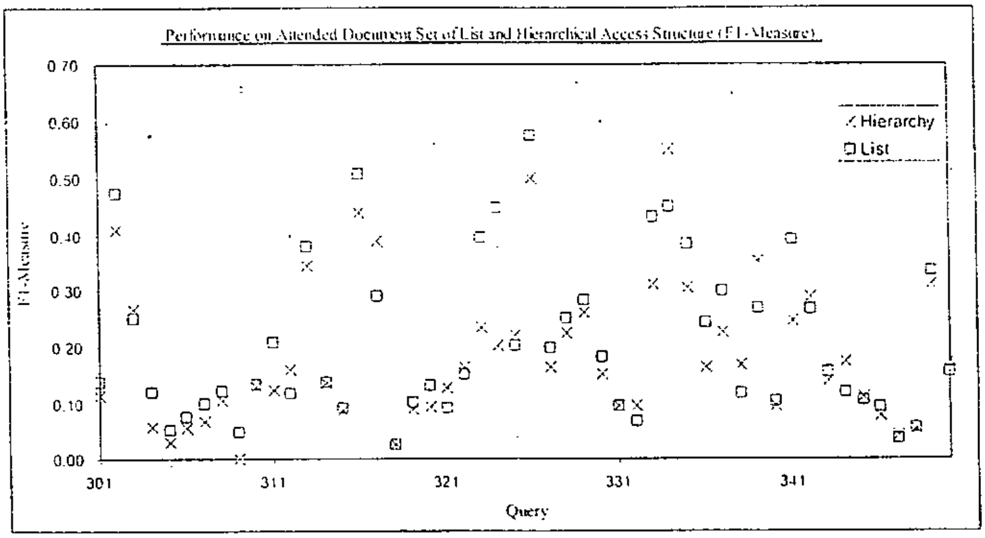


Figure: 4.4.3.6 Performance on attended document of search result (list) from IRE and hierarchical access structure (F1-measure)

Lastly, we explored the feasibility of our hierarchical access structure and the compared list access structure on helping users to judge the relevance of documents and finding the desired documents. As for the first exploration issue, we assumed that answers of TREC-6 queries were the only correct answers and subjects worked as information finder to find out the right (i.e. relevant) documents in the top 100 retrieved documents. Accordingly, accuracy is the percentage of the correctly matched ones (the judgment from subjects on the relevance / irrelevance of a document was the same as TREC) in the given document set. It was found that the accuracy of list and hierarchical structure was fairly close (Figure 4.4.3.7). We averaged the accuracy of 50 queries and found that the accuracy of hierarchical access structure (72.01%) was slightly better than list access structure (71.39%).

However, the measurement could probably be affected by human error (i.e. the subjects). As for the second issue, the documents, which user judged as relevant, were considered as user's desired documents for answering a query. Accordingly, the likelihood of finding a desired document is the percentage of the selected relevant documents in the given attended document set. We averaged the likelihood to find a desired document of the 50 queries and found that it was easier to find a desired document using hierarchical access structure (20.49%) than list access structure (15.27%) (Figure 4.4.3.8).

With comparison to list access structure, although our hierarchical access structure provides slightly better resulting list of document (i.e. higher precision in attended documents), a higher likelihood to find a desired document, a similar consistency in judging the relevance of documents by our users and the TREC evaluators, it provides poorer overall performance of finding relevant documents in the attended document set (i.e. recall and F1-measure). Therefore, we can just conclude that the hierarchical access structure performs fairly well as the list access structure based on the above performance measurements.

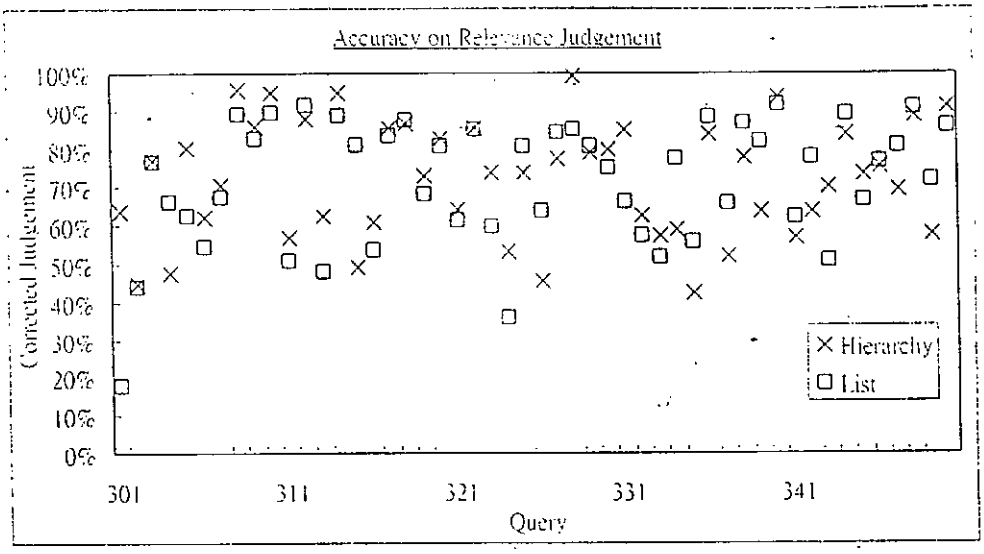


Figure 4.4.3.7 Accuracy of relevance judgment

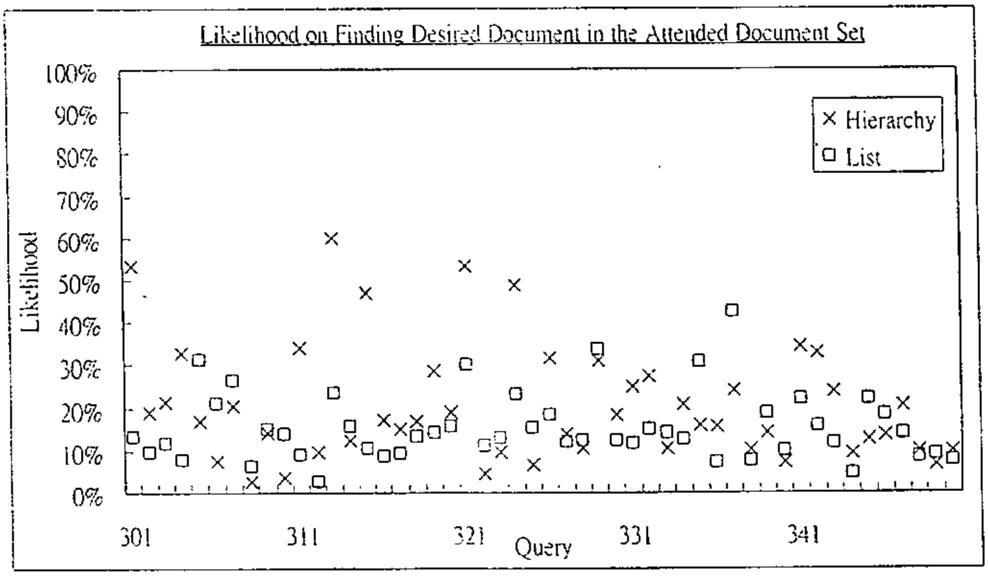


Figure: 4.4.3.8 Likelihood of finding desired document

4.4.4. Analysis of Post Experimental Questionnaire

In the post experimental analysis, we firstly explored the subjects' judgments on the performance of our hierarchical access structure and the list access structure based on their perception (Table: 4.4.4.1).

Table 4.4.4.1 Answers Collected from Questionnaire

	Averaged values across subjects (1 – 10, 1 for worst and 10 for best)	
Questions	Hierarchy	List
The degree of acceptability of using this structure to search information using mobile device	5.50	7.25
The degree of effectiveness of using this structure to search information using mobile device	4.75	5.75
The degree on "how easy" by using this structure to search information using mobile device	5.50	7.00
The degree on "how easy" to select information on the screen	5.75	6.25
The degree on providing a summarisation of information on ALL of the documents. (i.e. summarise all the 100 given documents)	6.25	5.50
Help to identify required information (i.e. select relevant document from the 100 documents)	4.75	5.50
Help to get a general idea of / better understanding on the main topic of the 100 documents	5.50	4.75
Help to predict whether you need to browse the details (the body) of the document.	5.50	6.00
Overall values (average across all of the questions)	5.44	6.00

It was found that our hierarchical access structure performed better than list structure in summarising documents and helping subjects to get the general ideas / understanding on the main topics of the documents. However, there was a poor judgment on the hierarchical access structure at the other questions and a poorer overall value on the hierarchical access structure (5.44) than the list access structure (6.00). We enquired the subjects about the reasons of the poor judgment. All the four subjects reported that the concept hierarchy was not closely related to the queries although it provides a relatively good summary of the document of search results. Therefore, subjects think that the acceptability and efficiency of the hierarchical access structure is poorer than list. The problem may probably be because the precision of search results from IRE was at about 20.28% on average. As the concept hierarchy was constructed from concepts of documents of the search results, the concept hierarchy contains 79.72% of irrelevant material. As a result, the concept hierarchy may appear to be unrelated to the queries. Moreover, the familiarization score (i.e. 1 – 10: 1 for most unfamiliar and 10 for most familiar) of hierarchical access structures / interfaces (5.75 in average) was much lower than that of list (9 in average). This is possible because the list access structure is basically similar to the access structure of Google's PDA version and thus, list access has commonly been used for accessing to search results in daily life. The subjects were already familiar with such type of information access structure. However, subjects are not familiar with our hierarchical access structure. Therefore, the subjects may find it more difficult to use our hierarchical access structure than list.

Secondly, we explored the subjects' preference on features and interface of both types of the access structures. The subjects evaluated both access structures in a similar way as the survey in Chapter 3. It was found that both average preference of features and overall interface of list access structure (preference on feature = 7.13, interface = 8) were higher than those of hierarchical access structure (preference on feature = 6.89, interface = 5.50). This further confirms our findings in Chapter 3 that the preferred user interface features would be a possible factor governing a preferred user interface. The better preference value of list access structure indicates that there was no bias toward hierarchical access structure, as users welcome much more the list access structure than the hierarchical access structure.

Finally, we investigated the importance of the time-related issue (e.g. speed on accessing information), communication cost issue and information relevance issue by asking the subjects to assign an importance score (1-10: 1 for less important, 10 for very important). This was because the measurements of performance in this chapter were mainly based on the above three issues. It was found that the importance of such issues were all at a high level (Timing issue: 8, Communication cost issue: 7.75 and relevance of information issue: 8.5). Therefore, we can regard that the choice of measurements in this chapter was suitable.

4.5 Summary

In this chapter, the feasibility of automatically constructing concept hierarchy and the performance of the concept hierarchy as a hierarchical access structure to support the access of search results using mobile devices were explored.

We explored four possible types of concept hierarchies and it was found that the subsumption hierarchy was the most valuable candidate for supporting mobile search using hierarchical access structure. In our pilot study, title terms were found to be more topical than content terms in representing the topics of a document. Accordingly, all the title terms (stop words excluded) were considered as candidate concepts of concept hierarchy.

With the analysis of experimental results of the information access performance, it was found that the hierarchical access structure supports faster information access using mobile devices and consumes less data transfer compared with the corresponding list structure. Clearly, for accessing, evaluating documents and finding desired documents, using the hierarchical access structure is a faster and more economical way than list structure.

It was also found that more documents could be reached using the hierarchical structure than the list structure with using the similar amount of time. The results suggest that subjects are able to evaluate more documents using the hierarchical structure than the list structure, which, in turn, the subjects are able to reach more search results using the hierarchical structure than the list structure when they are performing mobile search.

With comparison to list access structure, although our hierarchical access structure provides slightly better resulting list of document (i.e. higher precision in attended documents), a higher likelihood to find a desired document and a similar consistency in judging the relevance of documents by our users and the TREC evaluators, it provides poorer overall performance of finding relevant documents in the attended document set (i.e. recall and F1-measure). Therefore, we can just conclude that the hierarchical access structure performs fairly well as the list access structure based on the performance measurements of searching documents in search results.

Chapter 5

Summary and Conclusion

In this thesis, the issue of supporting mobile searching was addressed through a series of experiments. We reviewed and summarised related work, proposed solutions and current available systems on mobile searching. It is believed that our proposed "hierarchical access structure" could be one of the valuable solutions supporting mobile searching. Experiments and survey were performed to explore issues of user preference, practical requirements and information accessing performance. In this chapter, we summarise and conclude our proposition "supporting mobile search with hierarchical access structure", based on the findings and the work in this research.

5.1 Summary of Finding and Conclusion

The objective of this research is to explore the feasibility of applying hierarchical access structure for supporting mobile information access based on:

1. User preference

2. User requirements

3. Information accessing performance

In Chapter 3, the user preference of user interface features and access structures has been addressed. A survey was conducted in order to explore various user preferences of user interfaces and access structures as well as to examine certain assumptions of the access structure and user interface of mobile search systems.

It was found that most of the common existing using interfaces (e.g. list) are not preferred by the users. Instead, the users prefer using those hierarchical information access structures for browsing with mobile devices, which, in turn, suggests the need and the use of our proposed hierarchical access structure.

We further analysed practical requirements and behaviors on access structure of mobile searching systems in Chapter 3. It was found that hierarchical access structure is preferred by users and it also fulfills the practical requirements.

It was also found that hierarchical access structure based on analysing user behaviors could simply contain hierarchically organized catalogs, functions for navigating in between different catalogs, title list of items contained in the catalog and related content.

Although hierarchical access is preferred by users and practically required, it is not commonly deployed. One of the reasons is probably that hierarchical structure is financially costly because they are constructed manually for the mobile content providers, similar to the hierarchies of Yahoo! and Open Directory. Therefore, we further explored the feasibility of automatically constructing concept hierarchy and the information access performance using the concept hierarchy to access to search results using mobile devices.

In our pilot study, title terms were found to be more topical than content terms in representing the topic of a document. Consequently, all the title terms (stop words excluded) were considered as candidate concepts for constructing concept hierarchy automatically.

With the analysis of experimental results of the information access performance, it was found that hierarchical access structure supports faster access to information using a mobile device and consumes less data for browsing the same amount of conceptual information compared with list structure. It was also found that more documents could be reached using hierarchical structure than using list structure for similar amount of time. The results suggest that subjects were able to evaluate more documents using hierarchical structure than using list structure, which, in turn, subjects were able to reach more search results using hierarchical structure than using list structure when they were performing mobile search. Clearly, accessing and evaluating a document using hierarchical access structure is a fast and economic way in terms of timing and data usage issues.

With comparison to list access structure, although our hierarchical access structure provides slightly better resulting list of document (i.e. higher precision in attended documents), a higher likelihood to find a desired document and a similar consistency in judging the relevance of documents by our users and the TREC evaluators, it provides poorer overall performance of finding relevant documents in the attended document set (i.e. recall and F1-measure). Therefore, we can just conclude that the hierarchical access structure performs fairly well as the list access structure based on the performance measurements of searching documents in search results.

5.2 Contribution

The following summarises the major contributions of this research:

A. We explored the following issues in supporting mobile search:

1. user preference issues;
2. user requirement and searching behavior issues;
3. information access performance issues of searching relevant or desired documents in retrieved documents.

B. We discovered a methodology for constructing concept hierarchy in order to support mobile search.

C. We experimentally validated that our hierarchical access structure is a valuable and user preferred structure in supporting mobile searching.

D. Our results could be a valuable reference to other research in supporting mobile searching.

Suggested Future Work

The followings are suggested future work of this research:

1. The documents, which were used in the experiment mentioned in Chapter 4, come with author and/or editor suggested topical terms. However, not all of documents have author/editor suggested topical terms. Therefore, we suggest other researchers to gain some insight into topical term mining if they want to address the issue of accessing the documents that does not have suggested topical terms.
2. When we were working for the survey in Chapter 3, there were only seventy-seven user interface samples from mobile content providers. Therefore, the threshold (i.e. the number of samples in the highly preferred group as mentioned in Chapter 3) for identifying the important feature sets was 3. In the future, other researchers may find more samples for the survey (if any) and use a higher threshold for better reliability in identifying the important feature sets.
3. We believe that the quality of subsumption hierarchies can still be improved, such as 100% ideal parent-child relationships (i.e. child

concept is a type of parent concept and/or child concept is an aspect of parent concept). Therefore, we suggest other researchers to further investigate the algorithm for constructing concept hierarchy.

Reference

[1] Fagan, J., Preparing for the Mobile Net Generation, *Wireless Internet for E-Commerce Workshop*. 2000.

[2] Aridor, Y., Carmel, D., Maarek, Y.S., Soffer, A. and Lempel, R., Knowledge Encapsulation for Focused Search from Pervasive Devices, *ACM Transactions on Information Systems*, 20(1), pp. 25 - 46, 2000.

[3] Harber, E., Wireless Internet: Navigating the Wireless Web, *Wireless Agenda*, 2001.

[4] Taylor Nelson Sofres Interactive, *Asia-Pacific M-Commerce Report*, 2001.

[5] WapForum , (http://www.wapforum.org/).

[6] Video Research Ltd., *Survey on Mobile Phone Usage Situation*, 2000.

[7] Google Mobile, (http://www.google.com/wml).

[8] Kene, M., AT&T Wireless Adding Google to Phones, *CNet News*, 2001.

[9] Waply, (http://www.waply.com/wap/).

[10] Excite Mobile, (http://www.mobile.excite.com/).

[11] Fast Mobile, (http://wap.fast.no/).

[12] Wanna Mobile, (http://www.wannaWAP.com/wap/index.asp).

[13] Charie, C., Query Expansion Using an Interactive Concept Hierarchy, *PhD thesis, The University of Sheffield*, 2000.

[14] Ganguly, S. and Alonso, R., Query Optimization in Mobile Environments, *Technical Report of Rutgers University*, pp. 1 - 17, 1993.

[15] Reuters News, (http://www.reuters.com).

[16] South China Morning Post, (http://www.scmp.com/).

[17] Apple Daily, (http://www.appledaily.com.hk).

[18] CNN News, (http://www.cnn.com).

[19] BBC News, (http://www.bbc.com).

[20] MSN Mobile, (http://mobile.msn.com/).

[21] Yahoo Mobile, (http://mobile.yahoo.com).

[22] Jones, M., Buchanan, G., Thimbleby, H. and Marsden, G., User Interfaces for Mobile Web Devices, *Proc. WWW9 Mobile Workshop Position Paper*, 2000.

[23] Jones, M., Marsden, G., Mohd-Nasir, N., Boone, K. and Buchanan, G., Improving Web Interaction on Small Displays, *Proc. WWW8*

Conference, pp. 51-59, 1999.

[24] Schmidt, A., Schröder, H. and Frick, O., WAP - Designing for Small User Interfaces, *Proc. ACM CHI 2000 Extended Abstracts. Conference on Human Factors in Computing Systems*, pp. 187-188, 2000.

[25] Chan, D.L., Luk, R.W.P., Mak, W.K., Leong, H.V., Ho, E.K.S. and Lu, Q., Multiple Related Document Summary and Navigation Using Concept Hierarchies for Mobile Clients, *Proc. ACM SAC 2002*, pp. 627-633, 2002.

[26] Buyukkokten, O., Garcia-Molina, H. and Paepcke, A., Text Summarisation of Web Pages on Handheld Devices, *Proc. Workshop on Automatic Summarisation*, 2001.

[27] Buyukkokten, O., Garcia-Molina, H. and Paepcke, A., Focused Web Searching with PDAs, *Proc. WWW9 Conference*, pp. 213-230, 2000.

[28] González-Castaño, F.J., Anido-Rifón, L.E., Vales-Alonso, J., Asorey-Cacheda, R., Costa-Montenegro, E. and Conde-Lagoa, D., Hierarchical Atomic Navigation for Small Display Devices, *Proc. WWW10 Conference Poster*, 2001.

[29] Chan, D.L., Luk, R.W.P., Leong, H.V., Ho, E.K.S. and Lu, Q., A

Preliminary Study of User Interface Design Issues for Mobile Web Access, *Proc WWW11 Mobile Search Workshop*, pp. 4-14. 2002..

[30] Efthimiadis, E.N., Interactive Query Expansion: A User-Based Evaluation in a Relevance Feedback Environment, *Journal of the American Society for Information Science (JASIS)*, Vol. 51, pp. 1-25, 2000.

[31] Silverstein, C., Henzinger, M., Marais, H., and Moricz, M., Analysis of a Very Large AltaVista Query Log, *Technical Report SRC*.

[32] Xu, J., and Croft, W.B., Query Expansion Using Local and Global Document Analysis, *Proc. 19th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 4-11, 1996.

[33] Sanderson, M., Concept-Based Interactive Query Expansion Support Tool, *Research Project of The University of Sheffield (http://ir.shef.ac.uk/ciquest/proposal/)*.

[34] Madria, S.K., Mohania, M.K. and Roddick, J.F., A Query Processing Model for Mobile Computing, *Proc. Foundations of Database Organization*, pp. 147-157, 1998.

[35] Scott, P., Bamshad, M. and Steve, L., An Adaptive Agent for Web Exploration Based of Concept Hierarchies, *Proc. 9th International*

Conference on Human Computer Interaction, 2001.

[36] Lee, S.G., A Query Transformer in a Digital-Library Environment, *Proc. INET'97 Conference* (http://www.isoc.org/inet97/proceedings /A7/A7_1.HTM), 1997.

[37] iTAP, Motorola, (http://www.motorola.com/lexicus/html/itap.html).

[38] Aridor, Y., Carmel, D., Maarek, Y.S., Soffer, A. and Lempel, R., Knowledge Encapsulation of Focused Search from Pervasive Devices, *Proc. WWW10 Conference*, pp. 754-763, 2001.

[39] Baeza-Yates, R. and Ribeiro-Neto, B., Modern Information Retrieval, *Addison Wesley*, 1999.

[40] Gerard, S., Edward, A. and Harry, Wu., Extended Boolean Information Retrieval, *Communication of the ACM*, pp. 1022-1036, 1983.

[41] Buyukkokten, O., Garcia-Morlina, H. and Paepcke, A., Accordion Summarisation for End-Game Browsing on PDAs and Cellular Phones, *Proc. CHI 2001*, pp. 213-220, 2001.

[42] Buchanan, G., Farrant, S., Jones, M., Thimbleby, H., Marsden, G. and Pazzani M., Improving Mobile Internet Usability, *Proc. WWW10 Conference*, pp. 673-680, 2001.

[43] Web Clipping Development, Palm Inc., (http://www.palmos.com /dev/tech/webclipping/)

[44] Hatzivassiloglou, V., Gravano, L. and Maganti, A., An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering, *Proc. 23nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 224-231, 2000.

[45] Zamir, O. and Etzioni, O., Grouper: A Dynamic Clustering Interface to Web Search Results, *Proc. WWW8 Conference*, pp. 1361-1374, 1999.

[46] Zamir, O. and Etzioni, O., Web Document Clustering: A Feasibility Demonstration, *Proc. 21st ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 46-54, 1998.

[47] Hearst, M.A. and Jan Pedersen, P.R., The Cluster Hypothesis: Scatter/Gather on Retrieval Results, *Proc. 19nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 76-84, 1996.

[48] Mani, I. and Bloedorn, E., Summarising Similarities and Differences Among Related Documents, *Information Retrieval*, pp. 1-23, 1999.

[49] Mukherjea, S. and Foley, J.D., Visualizing the World-Wide Web with the Navigational View Finder, *Computer Networks and ISDN Systems*, pp. 1075-1087, 1995.

[50] Nation, D.A., Plaisant, C., Marchionini, G. and Komlodi, A., Visualizing Web Sites Using a Hierarchical Table of Contents Browser: WebTOC, *Technical Report of University of Maryland*, 1998.

[51] Maarek, Y. and Shaul, I., WebCutter: A System for Dynamic and Tailorable Site Mapping, *Proc. WWW6 Conference*, pp. 713-722, 1997.

[52] Chakrabarti, S., Dom, B., Agrawal, R. and Raghavan, P., Scalable Feature Selection, Classification and Signature Generation for Organizing Large Text Databases into Hierarchical Topic Taxonomies, *The VLDB Journal 7*, pp. 163-178, 1998.

[53] Mladenic, D., Turning Yahoo into an Automatic Web Page Classifier, *Proc. of the 13th European Conference on Artificial Intelligence*, pp. 473-474, 1998.

[54] Chen, H. and Dumais, S., Bringing Order to the Web: Automatically Categorizing Search Results, *Computer Human Interface Letters*, pp. 145-152, 2000.

[55] Nevill-Manning, C., Witten, I. and Paynter, G., Lexically-Generated Subject Hierarchies for Browsing Large Collections, *International Journal on Digital Libraries*, pp. 111-123, 1999.

[56] Sanderson, M. and Croft, W.B., Deriving Concept Hierarchies from Text, *Proc. 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 206 - 213, 1999.

[57] David, A.G. and Frieder, O., Information Retrieval: Algorithms and Heuristics, *Kluwer Academic Publishers*, 1998.

[58] Rocchio, J.J., The Smart Retrieval System Experiments in Automatic Document Processing, Chapter Relevance Feedback in Information Retrieval, *Prentice Hall*, 1971.

[59] Joho, H., Coverson, C., Sanderson, M., and Beaulieu, M., Hierarchical Presentation of Expansion Terms, *Prco.* ACM SAC 2002, pp. 645-649, 2002.

[60] Jing, J., Helal, H.S. and Elmagarmid, A., Client-Server Computing in Mobile Environments, *ACM Computing Surveys*, pp. 117-157, 1999.

[61] Jones, S., Jones, M. and Deo, S., Using Keyphrases as Search Result Surrogates on Small Screen Devices. *Personal Ubiquitous Computing*. Vol. 8, Issue 1; pp. 55-68, 2004.

[62] Marsden, G., Cherry, R. and Haefele, A., Small Screen Access to Digital Libraries, *Proc. CHI 2002*, pp. 786-787, 2002.

[63] Buyukkokten, O., Kaljuvee, O., Garcia-Molina, H., Paepcke, A. and Winograd, T., Efficient Web Browsing on Handheld Devices Using Page and Form Summarisation, *ACM Transactions on Information Systems*. Vol. 20, No. 1, pp. 82-115, 2002.

[64] Christopher, C. and Wang, F.L., Fractal Summarisation for Mobile Devices to Access Large Documents on the Web, *Proc. WWW2003*, pp. 215-224, 2003.

[65] Imielinski, T. and Badrinath, B.R., Mobile Wireless Computing: Challenges in Data Management, *Communications of ACM*. pp. 18-28, 1994.

[66] Sedlberg, E. and Etzioni, O., The Meta-Crawler Architecture for Resource Aggregation on the Web, *IEEE Expert*. pp. 11-14, 1997.

[67] Acharya, S., Franklin, M. and Zdonik, S., Balancing Push and Pull for Data Broadcast, *Proc. ACM SIGMOD Conference on Management of Data*, pp. 183-194, 1997.

[68] Beyer, H. and Holtzblatt, K., Contextual Design: Defining Customer-Centered Systems, *Morgan Kaufmann*, 1998.

[69] Williams, G. and McClintock, M., Usability at Microsoft, *Proc. Computer Human Interaction 97*, pp. 144-145, 1997.

[70] Cleary, T., Communicating Customer Information at Cabletron Systems Inc., *ACM Interactions*, Vol. 6, Issue 1, pp. 44-49, 1999.

[71] Coble, J.M., Karat, J. and Kahn, M.G., Maintaining a Focus on User Requirements Throughout the Development of Clinical Workstation Software, *Proc. Human Factors in Computing Systems*, pp. 170-177, 1997.

[72] Curtis, P., Heiserman, T., Jobusch, D., Notess, M. and Webb, J., Customer-Focused Design Data in a Large Multi-Site Organization, *Proc. Human Factors in Computing Systems*, pp. 608-615, 1999.

[73] Druin, A., Cooperative Inquiry : Developing New Technologies for Children with Children, *Proc. Human Factors in Computing Systems*, pp. 592-599, 1999.

[74] Konstantinos, G., Adamantia. G., Diomidis, D., and Virola. H., Applying Contextual Inquiry for Capturing End-Users- Behavior Requirements for Mobile Exhibition Services, *Research Project of Athens University.*

[75] Nokia 9000 Communicator, *PC Week*, pp. 39-40, 1996.

[76] Palen, L. and Salzman, M., Beyond the Handset: Designing for Wireless Communications Usability, *ACM Transactions on Computer-Human Interaction*, Vol. 9, No 2, pp. 125-151, 2002.

[77] OpenDirectory Project, (http://www.opendirectory.org).

[78] Koller, D. and Sahami, M., Hierarchically Classifying Documents Using Very Few Words, *Proc. 14th International Conference on Machine Leaning*, pp. 170-178, 1997.

[79] Fuhr, N., Hartmann, S., Lustig, G., Tzeras, K., Knorz, G., and Schwantner, M., Automatic Indexing in Operation: The Rule-Based System AIR/X for Large Subject Fields, *Technical Report (http://www.is.informatik.uni-duisburg.de/bib/docs/Fuhr_etal_93.html.en)*, 1993.

[80] Nevill-Manning, C., Witten, I., and Paynter, G., Lexically-Generated Subject Hierarchies for Browsing Large Collections, *International Journal on Digital Libraries 2*, pp. 111-123, 1999.

[81] Anick, P. and Tipirneni, S., The Paraphrase Search Assistant: Terminological Feedback for Interactive Information Seeking, *Proc. 22nd Annual International ACM/SIGIR Conference*, pp. 153-159, 1999.

[82] Church, K.W. and Hanks, P., Word Association Norms, Mutual Information and Lexicography, *Computational Linguistics*, pp. 22-29, 1990.

[83] Lawrie, D., Discovering and Comparing Topic Hierarchies, *Thesis Submitted to the Graduate School of the University of Massachusetts Amherst*, 1999.

[84] Search Engine Watch, (http://searchenginewatch.com).

[85] Nielsen//NetRatings, (http://www.nielsen-netratings.com).

Appendix:

A. Subjects

A. 1 Subject Profile

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
1	M	26	U	4	2	2	3	2	2	3	2	3	2	4	Y
2	M	25	P	5	2	1	3	2	1	2	1	3	2	3	Y
3	M	25	P	4	1	1	2	1	1	2	1	2	2	3	Y
4	M	23	P	4	2	1	3	1	1	3	1	2	2	3	Y
5	M	23	P	4	2	1	3	1	1	2	2	2	2	4	N
6	M	25	P	4	3	1	3	2	1	3	2	3	3	4	Y
7	M	24	P	5	2	2	3	2	2	3	3	3	2	4	Y
8	F	23	P	5	2	1	3	1	1	2	1	2	1	4	Y
9	M	24	P	5	2	1	3	2	2	3	2	2	2	3	Y
10	M	24	P	5	2	1	3	1	1	3	2	3	3	4	Y
11	F	23	U	4	3	3	3	1	1	3	2	3	2	4	Y
12	M	20	U	1	3	2	2	2	2	3	3	3	3	4	N
13	F	21	U	5	2	2	3	1	2	3	2	3	1	5	Y
14	M	22	U	3	2	2	1	2	2	2	1	3	1	3	N
15	M	22	U	5	3	1	3	1	1	3	2	3	1	4	Y
16	M	22	U	5	3	2	3	2	1	3	3	2	2	5	N
17	M	20	U	5	2	2	3	2	2	3	2	2	1	4	Y
18	M	21	U	5	3	2	2	1	1	2	2	2	3	4	Y
19	F	22	U	5	3	2	3	2	2	3	3	3	3	4	Y
20	F	20	U	5	2	2	3	2	1	2	1	2	1	4	Y
21	M	22	U	5	2	1	3	2	1	3	2	2	2	3	Y
22	F	21	U	5	2	2	3	2	1	3	2	2	1	4	Y
23	M	20	U	5	3	2	3	2	1	3	2	2	3	4	Y
24	M	23	U	5	2	2	3	2	2	3	3	3	3	4	Y
25	M	23	U	5	3	2	3	2	2	3	2	3	2	3	Y
26	M	21	U	4	2	1	2	2	2	1	2	2	2	4	Y
27	F	22	U	4	2	2	3	2	2	2	2	2	1	4	Y
28	M	21	U	5	3	2	3	2	2	2	3	2	1	3	Y
29	F	25	U	5	3	1	3	1	1	3	1	3	3	4	Y
30	F	25	U	5	3	1	3	2	2	2	2	3	3	4	Y
31	M	23	U	3	2	1	2	1	2	3	3	3	2	4	Y

Keys:

A1. Subject ID

A2. Gender (M: Male, F: Female)

A3. Age

A4. Education (P: Postgraduate, U: Undergraduate)

A5. Experience on searching (1– 4: 1 – 4 Years respectively, 5: 5 or more than 5 Years)

A6. Experience on PDA (1: Not at all, 2: Fairly, 3: Extremely)

A7. Experience on browsing with PDA (1: Not at all, 2: Fairly, 3: Extremely)

A8. Experience on reading with PC (1: Not at all, 2: Fairly, 3: Extremely)

A9. Experience on reading with mobile device (1: Not at all, 2: Fairly, 3: Extremely)

A10. Experience on searching with mobile device (1: Not at all, 2: Fairly, 3: Extremely)

A11. Preference on graphic (1: Not at all, 2: Fairly, 3: Extremely)

A12. Preference on access infomation using mobile device (1: Not at all, 2: Fairly, 3: Extremely)

A13. Preference on searching infomation with catalog (1:Not at all,2: Fairly, 3: Extremely)

A14. Preference on searching information with list (1: Not at all, 2: Fairly, 3: Extremely)

A15. Comfortable with experimental tool (1: Extremely Difficult, 2: Difficult, 3: Fairly, 4: Easy, 5: Extremely Easy)

A16. Training with human computer interaction (Y: Yes, N: No)

A.2. Age of Distribution of Subjects and Choice of Subjects

According to a survey [6] in Tokyo and surrounding areas on 16th February, 2001 – 26th February, 2001 on males and females between the ages of 12 to 69 (excluding elementary school students) regarding usage of mobile phone, we believe that the age group, which our subjects belong to, is representing the targeted mobile information seekers of this research.

Firstly, it was found that people aged between 20-34 represent the largest proportion of mobile owners for both males and females (Figure A.2.1). The age group also represents the largest proportion of browser phone (Internet enabled phone) users.

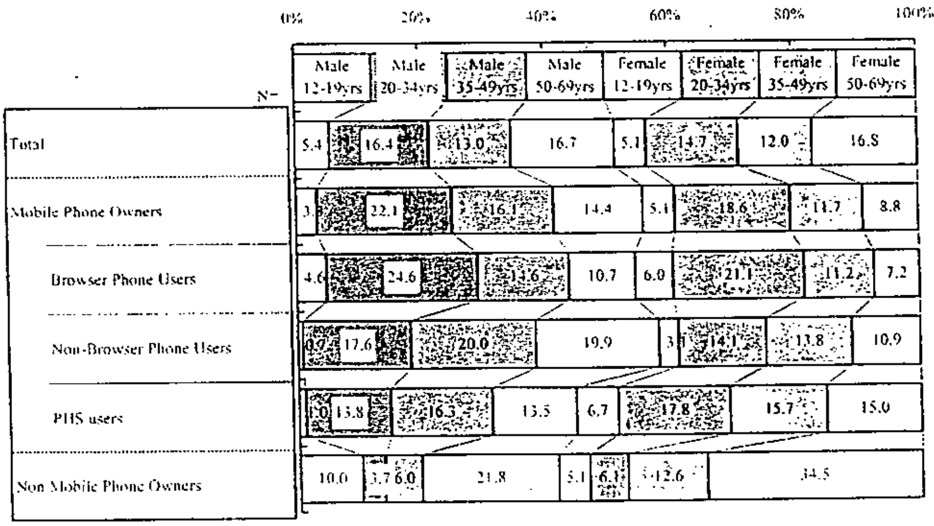


Figure A.2.1: Mobile phone owners by age group [6]

Secondly, it was found that the frequency of Internet access of this group of people was much higher than other age groups (Figure A.2.2).

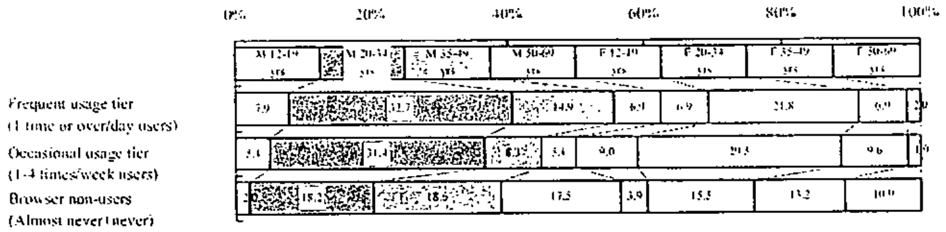


Figure A.2.2: Frequency of Internet access by age group [6]

Thirdly, it was found that they spent more on mobile phone service than the others (Figure A.2.3).

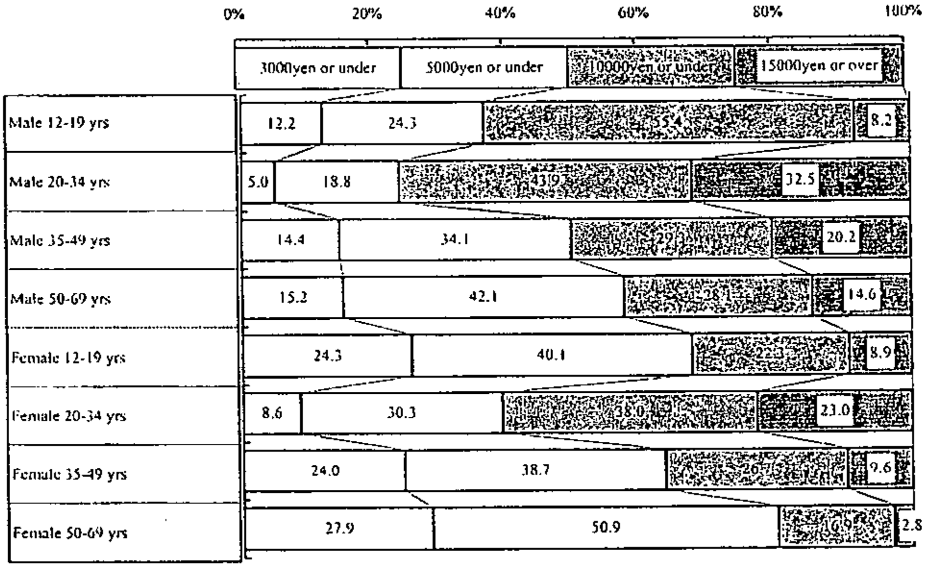


Figure A.2.3: Monthly spending on mobile phone service by age group [6]

In this research, we believe that the target for analysis should be the group which devotees of wireless Internet information seeking and willing to find

Internet information with their mobile devices frequently. According to the mentioned survey in Tokyo and surrounding areas, we know that people in the age group of 20-34 are the most possible target for our research. Therefore, we believe that the age group of our subjects (age from 20-26) (Figure A.2.4) is suitable for our analysis.

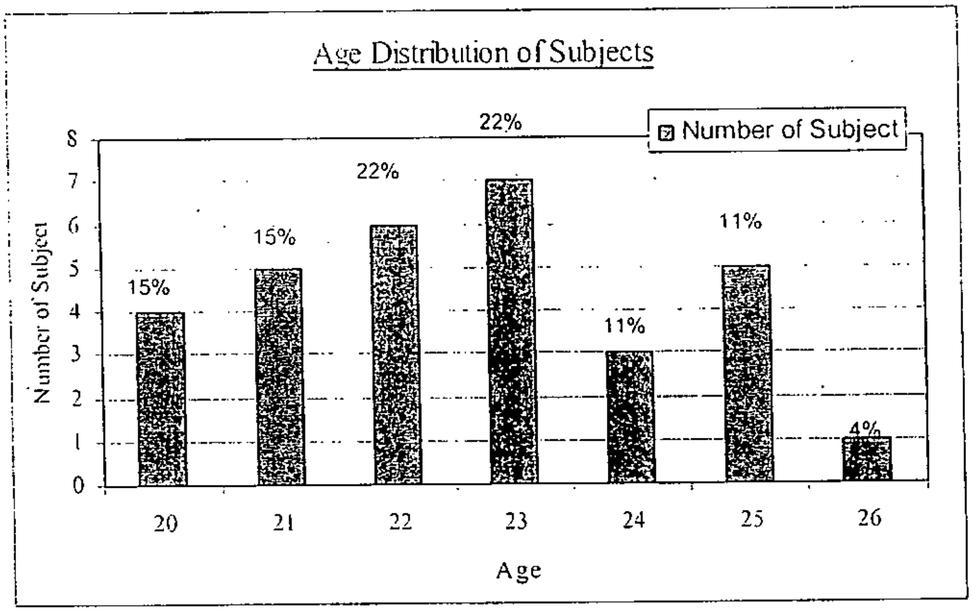


Figure A.2.4: Age distribution of subjects

B. General Experimental Information

Information of the collected results	
Number of content providers evaluated	76
Number of content providers selected	76
Number of users involved	27
Average users per content provider	20.513159
Average evaluation time of each content provider at each evaluation on it	2 minutes:24 seconds
Average navigation time of each content provider at each evolution on it	1 minute:11 seconds
Percentage of navigation to evaluation time	About 50%

C. Definitions and Selection of Minimum Requirement of Support

C.1 Definitions

Definition	Remark
Hybrid user interface	Those user interfaces not fulfill required support and confidence requirements.
Offline	Those content providers allow users to download information from server to mobile device and search / browse when disconnected from Internet.
Online	Those content providers can only allow users to search or browse information with Internet connection.
Support of add / drop feature associated with a content provider	Number of users who evaluated the content provider over total number of users (including those without evaluating the content provider).
Confidence of add/drop feature associated with a content provider	Number of users who evaluated the content provider and wanted to add / drop the feature to/from the content provider over number of users who evaluated the content provider.
Support of identifying nature (list / directory) of a given user interface of a content provider	Number of users who evaluated the content provider over total number of users (including those without evaluating the content provider).
Confidence of identifying nature (list / directory) of a given user interface of a content provider	Number of users who evaluated the content provider and agreed that the user interface is a list/directory over number of users who evaluated the content provider.

C.2 Selection on Minimum Requirement of Support

As for our case, the importance of support and size of samples are equivalent. Also, we want to raise the minimum support requirements (I); meanwhile, we want to make the available sample ratio (J – the ratio of sample that can fulfill the minimum support requirement) as large as possible. With the increasing in I, the number of samples that can fulfill the requirements is decreasing (i.e. decrease in J). Normally, if an increase in I results in an equivalent decrease in J, then the sum (i.e. I + J = P) must be consistent for all the pairs of I and J. Thus, P can be known the total benefit that we can get from the given pairs of I and J. However, it was found that P is not a constant value across all the pairs of I and J. This is because a great increase in I may only cause a small decrease / no change in J and vice versa. Therefore, it is possible that there are some of the I and J pairs which form a higher P than other I, J pairs. If we analyse the collected data set based on the I, J values which donate the highest P value, we can get the greatest benefit from the collected data. In order to explore the possible highest P value, an analysis was carried out by grouping samples according to the number of users who evaluated the samples. According to Figure C.2.1 and Figure C.2.2, it was found that we can get the greatest benefit from the collected data if we set the minimum support requirement (I) as 70%.

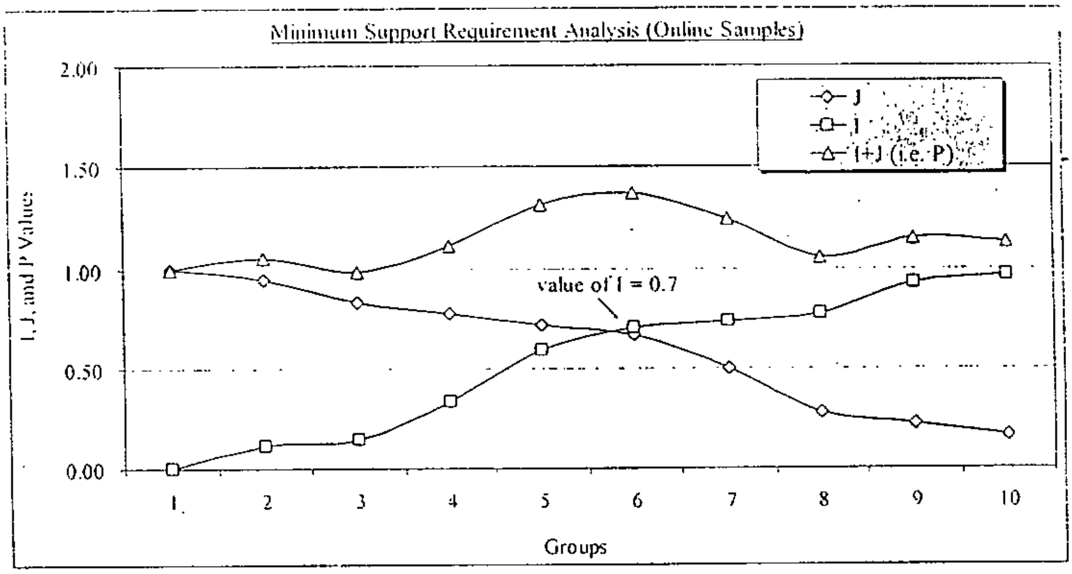


Figure C.2.1: Minimum support requirement analysis (online samples)

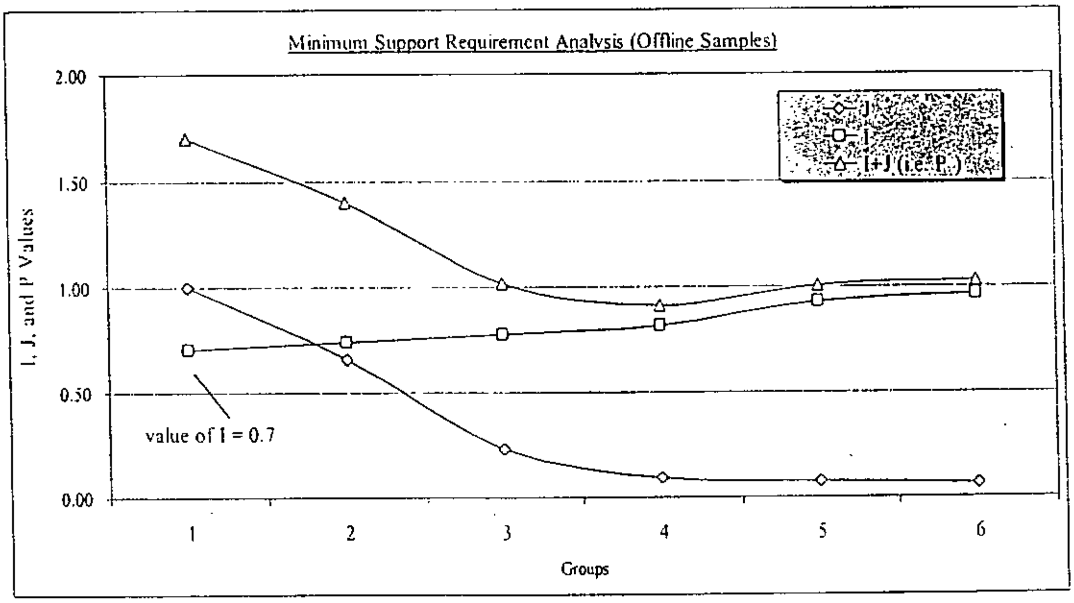


Figure C.2.2: Minimum support requirement analysis (offline samples)

D. Contextual Design

In short, the core premise of Contextual Inquiry is very simple: go where the customer works, observe the customer as he or she works, and talk to the customer about the work. Do that, while can't help but gain a better understanding of the customer. In Contextual Design, it is always try to build on natural human ways of interacting. It is easier to act, not out of long list of rules, but out of a simple, familiar model of relationship. Just like marketing needs to understand what people will buy and how people make buying decisions; designers need to understand what will help people do their work better while fitting into their lives and matching their culture. Contextual Inquiry addresses the first issue of Contextual Design, understanding user requirements, by putting users into their real working situation and collecting data from users' working experiences.

It was found that people with no special background in ethnography learn how to conduct effective interviews much more quickly by acting like an apprentice than by memorizing a list of effective interviewing techniques. It is as simply as a master craftsman teaches while doing. Therefore, a typical Contextual Inquiry interview lasts two or three hours. The designer meets the customer at his or her place of work and, after a brief introduction, watches the customer do work of the sort the designer is interested in. From time to time, the designer interrupts, and inquires details about the customer's work. With 10 to 20 interviews like this, with people who perform widely different roles and work in very different ways, it is usually sufficient to define an area of work, such as mobile searching, because people only come up with a few different ways of approaching a task.

While it is necessary to realize that customers do not think about how they are working, but they can talk about their work as it unfolds. They do not have to develop a way to present it or figure out what their motives are. All they have to do is to explain what they are doing. They are not aware of everything they do or why they do it; they become aware in the doing. Every action they take and every object around them helps them talk about the details of their work. By staying in the context enables us to gather ongoing experience rather than summary experience, and concrete data rather than abstract data. Practically, human love to abstract. If designers start from abstractions, not real experience, and then abstract again to go across all customers, there is little chance the system will actually be useful to real people. Thus, the principle of context tells us to go to the customer's workplace and see the work as it unfolds. This is the first and most basic requirement of Contextual Inquiry.

When we perform a study on Contextual Inquiry, we may ask the following questions in order to see the whole work context and identify opportunities and potential problems.

What is the work we expect to support?

How does this work fit into the customers' whole work life?

What are the key work tasks?

Who is involved in making the work happen?

Who are the informal helpers?

Who provides the information needed to do the job, and who uses the results?

Where does the work happen physically?

What is the cultural and social context in which the work happens?

We can use the above questions to constrain the interview situation that we can set up and guide us in thinking about how our searching system fits into uses' working environment. We can also use them to identify what kind of people we want to interview, what tasks we want to see performed, and what we want to watch for while are there. However, the above questions are just a focus but not a checklist. For example, if we are going to study online search and retrieval with mobile device, we can study how people search for information with such devices while they are away from the office or traveling on the road. This will help us understand the basic structure of finding, independent of technology and content.

In general, we have to interview two to three people in each role that is identified as important to the focus. We need 10 to 20 people in all our focus while six to ten interviews is sufficient if there is only a single role or we are only studying parts of the system (e.g. user interface) instead of overall work process. For example, financial institutions, high tech, and retail may be different market segments, but office work is done very similarly in any modern corporation. These different types of companies will not give us substantially different perspectives. In fact, office work is so similar that it is actually hard to get different perspectives. For example, many different types of company are using the same package of Microsoft Office.

Therefore, we need for choosing our subjects is focusing on those from the key areas our system are most likely to be used. On the other hand, we should not get too far ahead in lining up the visits. Therefore, we have to make sure that we talk to the people we will interview individually in advance and that they understand what will happen [68].

E. Samples of Searching Tasks

Searching Tasks of Section I
In Reuters, to find some news or information about sports and tell us the result
In HK Standard, to find some news or information about China and tell us the result
In Yahoo, to find some news or information about movie and tell us the result
In Reuters, to find some Top News and tell us the result
In HK Standard, to find some news or information about Hong Kong and tell us the result
In Yahoo, to find some news or information about university and tell us the result
In Reuters, to find some news or information about entertainment and tell us the result
In Yahoo, to find some information about travel and tell us the result
In Yahoo, to find some information about games and tell us the result
In Yahoo, to find some information about music and tell us the result
In Yahoo, to find some information about Tokyo's weather and tell us the result
In Yahoo, to find some item that you want to buy and tell us the result
Searching Tasks of Section II
To search some information about PDA and tell us the result
To search some information about on-line game and tell us the result
To search some news or information about well-known brand clothes and tell us the result
To search some information about perfume and tell us the result
To search some information about digital video or camera and tell us the result
To search some news or information about USA government department and tell us the result
Searching Tasks of Section III
Two teenage girls were shot dead and two others wounded during a party at a hairdressing salon in central England. *Please find out the full story of the news in ABC news.*
Existing technology used to detect diseases could also allow parents to select their babies' most important physical traits from eye color and hair color to brain power and even the shape of the babies' nose. *Please find out the full story of the news in ABC news.*

The state met the goal set by Gov. Ronnie Musgrove to connect Mississippi's 32,354 public classrooms to the Internet. *Please find out the full story of the news in CNN news.*
After the inaugural ceremony for the 33-kilometer Pudong track, which was presided over by Chinese Premier Zhu Rongji and German Chancellor Gerhard Schroeder on Tuesday, Shanghai officials indicated optimism that maglev trains would be the rage in the Greater Shanghai and Yangtze River Delta region. *Please find out the full story of the news in CNN news.*
The United States has announced it is to host a high-level meeting next week to discuss the crisis over North Korea's nuclear programme. *Please find out the full story of the news in BBC new.*
Oil prices have greeted the new year with a jump of almost 3%, as the markets reacted to data showing that the month-long strike in Venezuela has pushed USA reserves close to 26-year lows. *Please find out the full story of the news in BBC new.*
Searching Tasks of Section IV
Start from "Yahoo News", find some Top news, Technical news, Entertainment news and Sport news that are related to Asia.
Start from "Internet Wire", find some Top news, Technical news, Entertainment news and Sport news that are related to Asia.
Start from "ABC News – **Use IE**", find some Top news, Technical news, Entertainment news and Sport news that are related to Asia.
Start from "ABC News – **Use Klondlike**", find some Top news, Technical news, Entertainment news and Sport news that are related to Asia.
Start from "BBC", find some Top news, Technical news, Entertainment news and Sport news that are related to Asia.
Start from "Moreover", find some Top news, Technical news, Entertainment news and Sport news that are related to Asia.
Start from "7 AM", find some Top news, Technical news, Entertainment news and Sport news that are related to Asia.
Start from "ePrairie", find some Top news, Technical news, Entertainment news and Sport news that are related to Asia.
Start from BBC, find out an Israeli delegation is due to arrive in Washington to try to secure an emergency aid package worth $12bn.

Start from ABC, find out California high-tech crime detective nabs 65-year-old suspected Internet predator.
Start from Internet Wire, find out AFM Hospitality Corporation, one of North America's leading hospitality companies, announced today that it has discontinued negotiation regarding the acquisition of Marshall Management, Inc. of Salisbury, Maryland.
Start from Moreover, find out Survivor Thailand – An Interview with Helen.
Start from 7 AM, find one of the world's biggest drug marks says a federal judge has upheld one of its patents for Paxil, and anti-depressant.

F. Affinity Diagrams

1 User issues

1.1.1 User friendly

1.1.1.1 Focus: Users will focus on keywords of catalog names, content, search result, abstract and headings. Bolding or underlining keywords and highlight visited links can help to locate information.

1.1.1.1.1 Focus : "Searching keywords" at catalog names, content, search result, abstract and headings

1.1.1.1.2 Bold or underline keywords and highlight visited links

1.1.1.1.2.1 Prefer bolding wanted / important information for capturing user attention

1.1.1.1.2.2 Prefer highlighting visited links

1.1.1.1.2.3 Prefer highlighting on searching keyword

1.1.1.1.2.4 Prefer underlining wanted / important information

1.1.1.2 Performance

1.1.1.2.1 Acceptable waiting time : Able to wait for a longer time when sure loading content is useful and perform a searching task with no more than 15 minutes

1.1.1.2.1.1 Time to perform searching task

1.1.1.2.1.1.1 Able to perform a searching task with no more than 15 minutes

1.1.1.2.1.2 Wait when sure loading content is useful

1.1.1.2.1.2.1 Able to wait for 300 seconds when sure loading content is useful

1.1.1.2.1.2.2 Able to wait for 60 seconds when sure loading content is useful

1.1.1.2.1.2.3 Able to wait forever when sure loading content is useful

1.1.1.2.1.3 Wait when not sure loading content is useful

1.1.1.2.1.3.1 Able to wait for 5 seconds when not sure loading content is useful

1.1.1.2.1.3.2 Able to wait for 60 seconds when not sure loading content is useful

1.1.1.2.1.3.3 Able to wait for 10 seconds when not sure loading content is useful

1.1.1.2.1.3.4 Able to wait for 20 seconds when not sure loading content is useful

1.1.1.2.1.3.5 Able to wait for 30 seconds when not sure loading content is useful

1.1.1.2.1.3.6 Able to wait forever when not sure loading content is useful

1.1.1.2.2 Speed: Users cannot accept slow loading and accessing speed. They prefer faster speed. It was found that information fitted on single screen supporting faster accessing speed and WAP comes with faster loading speed with less pictures and additional contents.

1.1.1.2.2.1 Information fitted on single screen supporting faster searching

1.1.1.2.2.2 Prefer quickly accessing information (loading speed / selection speed)

1.1.1.2.2.3 Too slow on loading speed when viewing desktop content with PDA

1.1.1.2.2.4 WAP has less scrolling and faster accessing speed but more clicks (i.e. next page]

1.1.1.3 Priorities: Focus area, function keys, selection tools, input methods, summary and searching tools, layout components, performance and navigation tools

1.1.1.3.1 Focus : Area from top left > left > right > bottom right

1.1.1.3.2 Physical buttons would much more comfortable and frequently be used than scroll bars

1.1.1.3.3 Prefer and frequently use browser's function key > site's function key

1.1.1.3.4 Frequent of use [backward and forward button at the bottom of page] > [forward button at the bottom of page] > [backward and forward button at the top of page]

1.1.1.3.5 Pull-down menu prior to catalog, title list and scroll bar

1.1.1.3.5.1 Prefer pull-down menu > catalog > title

1.1.1.3.5.2 Prefer pull-down menu > scroll bar

1.1.1.3.6 Prefer selection based input > keyword based as difficult for inputting at PDA

1.1.1.3.7 Prefer having a local catalog/menu to summarise a Web site for local searching > local keyword search

1.1.1.3.8 Prefer abstract for summarising > picture for summarising

1.1.1.3.9 Communication cost, display area, speed, layout prior to picture

1.1.1.3.9.1 Preference on communication cost > picture

1.1.1.3.9.2 Preference on communication cost > detail > pictures

1.1.1.3.9.3 Preference on display area > picture

1.1.1.3.9.4 Preference on speed > picture

1.1.1.3.9.5 Preference on tidy layout > picture

1.1.1.3.10 Clicks and scrolling

1.1.1.3.10.1 Vertical scrolling > horizontalscrolling

1.1.1.3.10.2 Page break with more clicks > scrolling

1.1.2 User interface design

1.1.2.1 Layout: Preferred layout is a simple, clear, tidy and sharply design with common style and static standard, in which, components are clearly indicated and separated to fit on single screen by using up all the space

1.1.2.1.1 Components should be clearly indicated and separated

1.1.2.1.1.1 Catalog names/items/titles/links/function keys should be separated clearly

1.1.2.1.1.2 Should clearly indicate links/text

1.1.2.1.2 Simple, clearly, tidy and sharply design with common style and static standard

1.1.2.1.2.1 Prefer simple, clearly, tidy and sharply design rather than complicated design

1.1.2.1.2.2 Prefer common style and static standard in layout and catalog design

1.1.2.1.3 Better fitted on single screen and use up all the space

1.1.2.1.3.1 Use up all the space

1.1.2.1.3.1.1 Better use up all the space of the screen

1.1.2.1.3.1.2 Too much space results in give-up browsing, many scrolling, missing content and difficult to locate useful content

1.1.2.1.3.2 Fitted on single screen

1.1.2.1.3.2.1 Information fitted on single screen supports faster searching

1.1.2.1.3.2.2 Prefer information fitted on single screen

1.1.2.2 Text and background: Text, background and links [black and white] should be clearly separated and contrasted with stable size of text and font. .

1.1.2.2.1 Prefer stable font and text size

1.1.2.2.2 Request black and white for background and text

1.1.2.2.3 Text, background and links should be clearly separated and contrasted

1.1.2.2.3.1 Background and text colour should be contrasted

1.1.2.2.3.2 Required contrast in text and background

1.1.2.2.3.3 Should clearly indicate links/text

2 Functionality issues

2.1.1 Functional components

2.1.1.1 Function keys: Unique and users familiarized function keys should come with description and clearly shown. Moreover, function keys should be grouped together and put with other contents at top and bottom of long page and at the bottom of single screen page.

2.1.1.1.1 Function keys should being shown clearly and separately for avoiding wrong clicks

2.1.1.1.1.1 Catalog names / items / titles / links / function keys should separate clearly

2.1.1.1.1.2 Prefer function keys to be shown clearly and significantly for avoiding wrong clicks

2.1.1.1.2 Have familiarized function keys

2.1.1.1.2.1 Have similar function keys as Internet browser at desktop computer

2.1.1.1.3 Description

2.1.1.1.3.1 Prefer description of function keys

2.1.1.1.4 Unique and without duplication

2.1.1.1.4.1 Prefer one to one mapping of function keys and types of function for avoiding confusion

2.1.1.1.4.2 Do not prefer duplicated functions

2.1.1.1.5 Function keys should be grouped together and put with other contents at top and bottom of long page and at the bottom of single screen page

2.1.1.1.5.1 Function keys should not be on another page as this increases the number of clicks e.g. Google's [menu] of WAP version

2.1.1.1.5.2 Prefer function keys at the bottom of the page

2.1.1.1.5.3 Prefer function keys at the top and bottom of page and at the bottom for short page/single screen page

2.1.1.1.5.4 Prefer functions keys being put in the same position

2.1.1.2 Pull-down menu: User prefer pull-down menu as it is vertically and clearly showing information on a single screen and can quickly be accessed

2.1.1.2.1 Prefer pull-down menu for showing items in a single screen clearly

2.1.1.2.2 Prefer vertical layout, design and components e.g. pull-down menu and vertical catalog

2.1.1.2.3 Pull-down menu can help access information much quicker

2.1.1.3 Issues on scrolling: Users need to scroll for many lines because information is not grouped, concentrated and fit to a single screen. However, users do not prefer manual / automatic scrolling because scrolling causes inconvenience for locating and capturing as well as accessing to information and function. They would prefer to use page break, physical button comes that with mobile device, catalog and pull-down menu for replacing scroll bars if possible. In case scrolling is needed, single direction of scrolling depends on screen (vertical /horizontal) and no more than 2 pages are most suitable for most of users.

2.1.1.3.1 Users need to scroll for many lines because information is not grouped, concentrated and fit to a single screen and results in many scrolling

2.1.1.3.1.1 Without grouping of information

2.1.1.3.1.1.1 Do not prefer mixing of catalog names and titles that results in many scrolling and long page

2.1.1.3.1.2 Without concentrated information and too much space

2.1.1.3.1.2.1 Concentrated information with faster accessing speed and less scrolling

2.1.1.3.1.2.2 Too much space results in give-up browsing, many scrolling, missing of content and difficulty in locating useful content

2.1.1.3.1.3 Content does no fit onto a single screen of mobile device, e.g. content designing for desktop computer

2.1.1.3.1.3.1 Content is usually larger than a screen, many scrolls are required before capturing full information of a page / difficult to capture full information

2.1.1.3.1.3.2 Too many scrolling when viewing desktop content with PDA , do not prefer to have too many scrolling

2.1.1.3.2 User do not prefer manual / automatic scrolling

2.1.1.3.2.1 Do not prefer scrolling

2.1.1.3.2.2 Do not prefer horizontal scrolling

2.1.1.3.2.3 Do not prefer motion /auto scrolling on text like WAP

2.1.1.3.2.4 Do not prefer vertical scrolling together with horizontal scrolling

2.1.1.3.3 Inconvenience to locate and capture as well as access

2.1.1.3.3.1 Difficult to locate functions, useful content and capture full information and even give-up searching

2.1.1.3.3.1.1 Usually have to scroll down to locate function keys

2.1.1.3.3.1.2 Too much space results in give-up browsing, many scrolling, missing of content and difficulty in locating useful content

2.1.1.3.3.1.3 Content is usually larger than a screen, many scrolls are required before capturing full information of a page / difficult to capture full information

2.1.1.3.3.2 Increasing number of hand actions and slower accessing speed

2.1.1.3.3.2.1 As less number of hand actions as possible (e.g. scroll/click)

2.1.1.3.3.2.2 Concentrated information with faster accessing speed and less scrolling

2.1.1.3.4 Use page break, physical button that comes with mobile device, catalogs and pull-down menu for replacing scroll bars

2.1.1.3.4.1 Use page break instead

2.1.1.3.4.1.1 Page break with more clicks > scrolling

2.1.1.3.4.1.2 WAP has less scrolling and faster accessing speed but more clicks (i.e. next page]

2.1.1.3.4.2 Physical button comes with mobile device

2.1.1.3.4.2.1 Physical buttons would be much more comfortable and are frequently used than scroll bars

2.1.1.3.4.3 Catalogs and / or pull down menu

2.1.1.3.4.3.1 Prefer catalog grouping > list of titles because there will be less scrolling in this way

2.1.1.3.4.3.2 Prefer pull-down menu > scroll bar

2.1.1.3.5 Single direction of scrolling depends on screen (vertical /horizontal) and no more than 2 pages

2.1.1.3.5.1 To scroll no more than 5 pages and perfectly no more than 2 pages

2.1.1.3.5.1.1 Prefer scrolling no more than 2 pages

2.1.1.3.5.1.2 Prefer scrolling no more than 3 pages

2.1.1.3.5.1.3 Prefer scrolling no more than 5 pages

2.1.1.3.5.2 Prefer single direction of scrolling depending on screen (vertical /horizontal)

2.1.1.3.5.2.1 Vertical scrolling / horizontal scrolling depends on screen (vertical / horizontal)

2.1.1.3.5.2.2 Vertical scrolling > horizontal scrolling

2.1.1.4 Links: Text-based links are most preferred than image based links while text and links should be separated clearly

2.1.1.4.1 Prefer text-based links rather than image based links

2.1.1.4.2 Should clearly indicate links/text

2.1.1.5 Status / history tracing functions : required status / history tracing functions for sequential movement / direct jumping in between different status / location at searching

2.1.1.5.1 Required status functions e.g. backward, forward, browsing path. If no status functions, users have every time to go back to top level catalog before visiting other catalogs. This waste a lot of time when browsing

2.1.1.5.2 [Sequential movement] backward and forward can be useful and suitable at the bottom of page

2.1.1.5.2.1 Backward [input/modify query, to previous status/catalog, back to search result list], forward used for forwarding to the next page

2.1.1.5.2.1.1 Backward function used frequently for inputting/modifying query, to previous status/catalog, search result list

2.1.1.5.2.1.2 Forward functions used for forwarding to the next page

2.1.1.5.2.2 Backward and forward buttons at the bottom of page

2.1.1.5.2.2.1 Frequent of use [backward and forward buttons at the bottom of page] > [forward button at the bottom of page] > [backward and forward buttons at the top of page]

2.1.1.5.3 [Direct jumping]

2.1.1.5.3.1 Can type and go to a URL directly

2.1.1.5.3.1.1 Prefer address bar for typing in URL direcly / to visit URL directly

2.1.1.5.3.2 Directly jumping in between status

2.1.1.5.3.2.1 Jumping in between different status

2.1.1.5.3.2.1.1 Concept path for showing 3 - 4 status for back trace

2.1.1.5.3.2.1.1.1 Prefer concept path showing nearest 3 - 4 status

2.1.1.5.3.2.1.1.2 Prefer concept path, showing types of items and location

2.1.1.5.3.2.1.2 Page number list for jumping

2.1.1.5.3.2.1.2.1 Required page numbers e.g. << 1,2,3,4,5 >> for jumping to different location, showing existing location

2.1.1.5.3.2.2 History for re-query and back trace of search result

2.1.1.5.3.2.2.1 Prefer history tracing, e.g. history bar at searching result

2.1.1.5.3.2.2.2 Prefer providing input history for saving time to re-query

2.1.1.6 Catalog: Users seeking a feasible mechanism to group information and catalog is one of the possible methods helping users to identify information quickly by providing tidy display and a summary of information. They use catalog searching when cannot specify query or at no objective on searching or sure which catalog the searching items belong to. Users accept changes on catalog component based on dynamic data if necessary and they should based on common and static style. Also they prefer detailed, hierarchical catalog with multiple paths that matched common sense, user's past experience and key terms at content but no more than 30 items in each of the catalogs. As for displaying, catalog and headings should not be put together and catalog names should be separated clearly on a single screen. Moreover, they should provide searching on catalog and content, show number of items at each catalog and have sequence indicators on catalog and headings.

2.1.1.6.1 Accept slight changes on catalog component based on dynamic data if necessary and they should be based on common and static style

2.1.1.6.1.1 Able to accept addition of new catalogs but they should be few changes on the existing catalogs

2.1.1.6.1.2 Prefer common style and static standard in layout and catalog design

2.1.1.6.1.3 Prefer generating of hierarchy based on data only

2.1.1.6.2 Prefer detailed hierarchical catalogs with multiple paths that matched common sense, users' past experience and key terms at content but no more than 30 items in each of the catalogs

2.1.1.6.2.1 Prefer detailed hierarchical catalogs with multiple paths and no more than 30 items in each of the catalogs

2.1.1.6.2.1.1 Prefer detailed hierarchical catalogs with multiple paths

2.1.1.6.2.1.1.1 . Prefer hierarchical catalogs than simple catalogs

2.1.1.6.2.1.1.2 Prefer having multiple paths for the same news / content if the content /news are belong to different catalog (e.g. different paths can access the same news / content)

2.1.1.6.2.1.1.3 Prefer detailed categorization for directorial searching

2.1.1.6.2.1.2 Accept 0 - 30 items in a catalog

2.1.1.6.2.1.2.1 Able to accept no more then 10 items in a catalog

2.1.1.6.2.1.2.2 Able to accept no more then 20 items in a catalog

2.1.1.6.2.1.2.3 Able to accept no more then 30 items in a catalog

2.1.1.6.2.1.2.4 Accept empty catalog

2.1.1.6.2.2 Catalog should match the common sense and users' past experience but also match the key terms at content

2.1.1.6.2.2.1 Catalog should match to users' common sense and past experience

2.1.1.6.2.2.1.1 Better start with catalog that match users' common sense

2.1.1.6.2.2.1.2 Catalog name is useful if users know the type of searching information belongs to (by common sense or past experience)

2.1.1.6.2.2.1.3 For catalog searching, the users will find the answers quickly if they know which catalog the search item belongs to.

2.1.1.6.2.2.1.4 Will be based on past experience and common sense to perform searching with catalogs

2.1.1.6.2.2.1.5 Will not try unknown catalogs if users cannot find searching item s in known catalogs

2.1.1.6.2.2.2 Catalog name should match the keywords of the underlying content

2.1.1.6.2.2.3 Many existing catalogs can match users' common sense

2.1.1.6.2.2.3.1 Catalogs do not match users' common sense

2.1.1.6.2.2.3.2 No catalogs that match users' searching requirements

2.1.1.6.3 Catalogs and headings should not be put together and catalog names should be separated clearly on a single screen.

2.1.1.6.3.1 Catalogs and headings should not be put together

2.1.1.6.3.1.1 Do not prefer mixing catalog names and titles that results in many scrolling and long pages

2.1.1.6.3.2 Catalog names should be separated clearly and many catalogs should be show on a single screen

2.1.1.6.3.2.1 Catalog names / items / titles / links / function keys should be separated clearly

2.1.1.6.3.2.2 Show as many catalog on a single screen as possible

2.1.1.6.4 Prefer having search on catalogs and content, show number of items at each catalog and have sequence indicators

2.1.1.6.4.1 Prefer searching on content and catalog name

2.1.1.6.4.2 Prefer showing number of items in a catalog followed by catalog names

2.1.1.6.4.3 Request numbers / sequence indicators to indicate sequence of search results, headings , titles and catalogs

2.1.1.6.5 Catalogs help users to identify information quickly by providing tidy display and a summary of information

2.1.1.6.5.1 Catalog names help users to identify types and relevant information quickly

2.1.1.6.5.1.1 Title and catalog names can help users to understand and identify useful content quickly

2.1.1.6.5.1.2 Catalog names help users to know the types of news/information in the catalogs

2.1.1.6.5.2 Catalogs provide tidy display

2.1.1.6.5.3 Catalogs can summarise information for searching

2.1.1.6.5.3.1 Prefer to have a local catalog/menu to summarise a Web site for local searching

2.1.1.6.6 A feasible mechanism to group information by catalog

2.1.1.6.6.1 Prefer "favorites" of Internet Explorer (catalog) than bookmark of Klondlike (list) as good design for group information

2.1.1.6.6.2 Request grouping depending on searching purpose

2.1.1.6.6.3 Prefer catalog grouping > list of titles because there will be less scrolling in this way

2.1.1.6.6.4 Prefer having catalogs and then abstracts while loading

2.1.1.6.7 Use catalog searching when failing to specify query, doing non-objective searching or know which catalogs the searching items belong to

2.1.1.6.7.1 Use catalog searching when failing to specify the query

2.1.1.6.7.2 Use catalog searching for non-objective searching

2.1.1.6.7.3 Use catalog searching when users can specific the catalog the searching items belong to (by common sense/catalog names)

2.1.1.7 Summary: for wireless searching, only important content is required. Therefore, summary and concentrated information, such as abstract (no more than 3 complete sentences), heading, indexing and local catalog of site are preferred as they can help users to understand, search and identify useful content quickly.

2.1.1.7.1 Summary [abstract, heading, indexing, local catalog of site] is preferred as it can help users to understand, search and identify useful content quickly

2.1.1.7.1.1 Abstracts can help users to understand, search titles and identify useful content/ search results quickly

2.1.1.7.1.2 Headings can be helpful in searching

2.1.1.7.1.3 Prefer having a local catalog/menu to summarise a Web site for local searching

2.1.1.7.1.4 Prefer short abstract, description on headings and links

2.1.1.7.1.5 Prefer page indexing

2.1.1.7.2 Concentrated information is preferred as it comes with faster accessing speed and less scrolling

2.1.1.7.2.1 Concentrated information with faster accessing speed and less scrolling

2.1.1.7.2.2 Prefer concentrated approach to show information

2.1.1.7.3 Only important content is required so as to display as much information on a single screen as possible

2.1.1.7.3.1 Required important content/text only when searching with PDA

2.1.1.7.3.2 Prefer showing as many results on a single screen as possible

2.1.1.7.4 Prefer abstract in no more than 3 complete sentences

2.1.1.7.4.1 Preferred size of abstract : 1 - 2 sentences with complete sentences

2.1.1.7.4.2 Preferred size of abstract : 2 - 3 sentences with complete sentences

2.1.1.8 Presentation of search result: only necessary content (abstract and heading) is required in the search result list. Do not prefer further categorization and searching on search result. In additional, highlighting on searching keyword, numbering and total number of search results are required. Moreover, users are able to browse more pages, no more than 20 items on each page, when knowing the total number of pages and usually modify query before browsing all the results.

2.1.1.8.1 Able to browse more pages, no more than 20 items on each page, when knowing the total number of pages.

2.1.1.8.1.1 Accept no more than 20 search results at a page

2.1.1.8.1.1.1 Able to accept no more than 10 search results on page

2.1.1.8.1.1.2 Able to accept no more than 20 search results on page

2.1.1.8.1.2 Able to browse no more than 20 pages if the number of total pages is known. Otherwise, no more than 5 pages

2.1.1.8.1.2.1 Able to browse no more than 10 pages when the number of pages is known

2.1.1.8.1.2.2 Able to browse no more than 20 pages when the number of pages is known

2.1.1.8.1.2.3 Able to browse no more than 4 pages without knowing the number of pages

2.1.1.8.1.2.4 Able to browse no more than 5 pages without knowing the number of pages

2.1.1.8.1.2.5 Will only browse 1- 4 pages of searching results

2.1.1.8.2 Only necessary content (abstract and heading) is required in the search result list. Do not prefer further categorization on search results

2.1.1.8.2.1 Abstract + heading would be enough for browsing search result

2.1.1.8.2.2 Required important content/text only when searching with PDA

2.1.1.8.2.3 Do not prefer further categorization of keyword of search results, only a list of headings + abstract is required

2.1.1.8.3 Highlighting searching keyword, numbering and the total number of search results.

2.1.1.8.3.1 Prefer highlighting searching keyword.

2.1.1.8.3.2 Prefer showing number of search results, for deciding whether to modify the query or not

2.1.1.8.3.3 Request number / sequence indicator to indicate sequence of search results, headings , titles and catalogs

2.1.1.8.4 Users will modify query before browsing all the results and search on search result is not useful for them.

2.1.1.8.4.1 Search on existing search result is not useful, users will choose extending the query

2.1.1.8.4.2 Modify query after reading 50 - 60 results

2.1.2 Functions

2.1.2.1 Input: users prefer as little of input as possible when searching. They prefer and usually use familiarized keyboard (physical if possible) based input as it is more accurate than other input methods. Moreover, word completion and input history tracing can also help on enhancing input accuracy and speed.

2.1.2.1.1 As little input as possible when searching

2.1.2.1.2 Users prefer familiarized keyboard based (physical if possible) input as it is more accurate than other input methods

2.1.2.1.2.1 Do not prefer hand writing input, because not accurate enough, but if accurate, users will use

2.1.2.1.2.2 Keyboard input more accurate, similar to those used at desktop and no need to learn

2.1.2.1.2.3 Only used familiarized keyboard like input even we provided 4 types of input

2.1.2.1.2.4 Prefer using of physical keyboard for inputting

2.1.2.1.3 Word completion and input history tracing are useful

2.1.2.1.3.1 Prefer providing input history as it saves time for re-query

2.1.2.1.3.2 Word completion allows faster input and less wrong spelling.

2.1.2.1.3.2.1 Prefer word completion which causes less wrong spelling when inputting query

2.1.2.1.3.2.2 Prefer word completion which makes input much faster

2.1.2.2 Keyword search: used when failing to find or locate search items in catalogs/ no idea about searching items/ required objective search/ there are too many un-organized information. Users will need a simple text field for keyword searching based on keyword/phrase at semantic meaning or directly match. However, users will accept query length no more than 15 characters.

2.1.2.2.1 Always provide a simple text field for searching based on keyword/phrase

2.1.2.2.1.1 For pull-based searching, a simple text field e.g. "[ ]" for input keyword/phrase is enough

2.1.2.2.1.2 Prefer providing a text field for inputting query, so that users can always expand and modify their query

2.1.2.2.1.3 Prefer providing a search function

2.1.2.2.1.4 Prefer keyword + phrase searching for pull-based searching

2.1.2.2.2 Request match query and keywords at content by semantic meaning

2.1.2.2.3 Query no more than 15 characters

2.1.2.2.3.1 Prefer query of no more than 10 characters, otherwise, use catalog or voice based input

2.1.2.2.3.2 Prefer query of no more than 15 characters, otherwise, use catalog or voice based input

2.1.2.2.4 Use keyword search when failing to find or locate search item in catalog, no idea about searching item, required objective search, have too many un-organized information.

2.1.2.2.4.1 Use keyword search when failing to find or locate search in catalog or no idea about searching item (by common sense/catalog names).

2.1.2.2.4.1.1 Use keyword searching when failing to find searching item in catalogs (by common sense/catalog names)

2.1.2.2.4.1.2 Use keyword searching when failing to specific the catalog of the searching item (by common sense/catalog names)

2.1.2.2.4.1.3 Use keyword searching when users definitely do not know what the searching item is

2.1.2.2.4.2 Use keyword searching when users know the searching item exactly or doing objective searching

2.1.2.2.4.3 Prefer to having local keyword search if there is too much un-organized information on a page

2.1.2.3 Processing of information

2.1.2.3.1 Concentrated information

2.1.2.3.1.1 Catalog: users seeking a feasible mechanism to group information and catalog is one of the possible methods for helping users to identify information quickly by providing tidy display and a summary of information. They use catalog searching when can't specify query or doing non-objective searching or know which catalog the searching items belong to Users accept slight changes on catalog component based on dynamic data if necessary and they should be based on common and static style. Also, they prefer detailed hierarchical catalogs with multiple paths that match users' common sense, user's past experience and key terms at content but no more than 30 items in each of the catalogs. As for displaying, catalogs and headings should not be put together and catalog names should be separated clearly on a single screen. Moreover, there should provide searching on catalog and content, show number of items at each catalog and have sequence indicators on catalogs and headings.

2.1.2.3.1.1.1 Users accept slight changes on catalog component based on dynamic data if necessary and they should be based on common and static style.

2.1.2.3.1.1.1.1 Able to accept addition of new catalogs with slight changes to the existing catalogs

2.1.2.3.1.1.1.2 Prefer common style and static standard in layout and catalog design

2.1.2.3.1.1.1.3 Prefer generating of hierarchy based on data only

2.1.2.3.1.1.2 Prefer detailed hierarchical catalogs with multiple paths that matched common sense, users' past experience and key terms at content but no more than 30 items in each of the catalogs

2.1.2.3.1.1.2.1 Prefer detailed hierarchical catalog with multiple paths and no more than 30 items in each of the catalogs

2.1.2.3.1.1.2.1.1 Prefer detailed hierarchical catalog with multiple paths

2.1.2.3.1.1.2.1.2 Prefer hierarchical catalogs than simple catalog

2.1.2.3.1.1.2.1.3 Prefer having multiple paths for the same news / content if the content

/news belongs to different catalogs (e.g. different paths can access the same news / content)

2.1.2.3.1.1.2.1.4 Prefer detailed categorization for directorial searching

2.1.2.3.1.1.2.1.5 Accept 0 - 30 items in a catalog

2.1.2.3.1.1.2.1.6 Able to accept no more then 10 items in a catalog

2.1.2.3.1.1.2.1.7 Able to accept no more then 20 items in a catalog

2.1.2.3.1.1.2.1.8 Able to accept no more then 30 items in a catalog

2.1.2.3.1.1.2.1.9 Accept empty catalog

2.1.2.3.1.1.2.2 Catalogs should match the common sense and users' past experience but also match key terms at content

2.1.2.3.1.1.2.2.1 Catalogs better match user's common sense and past experience

2.1.2.3.1.1.2.2.2 Better start with catalogs that match users' common sense

2.1.2.3.1.1.2.2.3 Catalog name is useful if users know the catalog the searching items belong to (by common sense or past experience)

2.1.2.3.1.1.2.2.4 For catalog searching, the users will find the answers quickly if they know the catalog the searching items belong to

2.1.2.3.1.1.2.2.5 Will be based on past experience and common sense to perform searching with catalogs

2.1.2.3.1.1.2.2.6 Will not try unknown catalogs if users cannot find searching items in known catalogs

2.1.2.3.1.1.2.2.7 Catalog name should match the keywords of the underlying content

2.1.2.3.1.1.2.2.8 Many existing catalogs can match user's common sense

2.1.2.3.1.1.2.2.9 Catalog not match users' common sense

2.1.2.3.1.1.2.2.10 No catalogs that match users' searching requirements

2.1.2.3.1.1.3 Catalogs and headings should not be put together and catalog names should be separated clearly on a single screen.

2.1.2.3.1.1.3.1 Catalogs and headings should not be put together

2.1.2.3.1.1.3.1.1 Do not prefer mixing catalog names and titles that results in many scrolling and long page

2.1.2.3.1.1.3.2 Catalog names should be separated clearly and many catalogs should be shown on a single screen

2.1.2.3.1.1.3.2.1 Catalog names /items /titles /links /function keys should be separated clearly

2.1.2.3.1.1.3.2.2 Show as many catalogs on a single screen as possible

2.1.2.3.1.1.4 Prefer having search on catalogs and content, show number of items at each catalog and have sequence indicators

2.1.2.3.1.1.4.1 Prefer searching on content and catalog name

2.1.2.3.1.1.4.2 Prefer showing number of items in a catalog followed by catalog names

2.1.2.3.1.1.4.3 Request numbers / sequence indicators to indicate sequence of search results, headings , titles and catalogs

2.1.2.3.1.1.5 Catalogs help users to identify information quickly by providing tidy display and summary of information

2.1.2.3.1.1.5.1 Catalog names help uses to identify types and relevant information quickly

2.1.2.3.1.1.5.1.1 Title and catalog names can help users to understand and identify useful content quickly

2.1.2.3.1.1.5.1.2 Catalog names help users to know the types of news/information in the catalogs

2.1.2.3.1.1.5.2 Catalogs provide tidy display

2.1.2.3.1.1.5.3 Catalogs can summarise information for searching

2.1.2.3.1.1.5.3.1 Prefer to have a local catalog/menu to summarise a Web site for local searching

2.1.2.3.1.1.6 A feasible mechanism to group information by catalog

2.1.2.3.1.1.6.1 Prefer "favorites" of Internet Explorer (catalog) than bookmark of Klondlike (list) as good design for group information

2.1.2.3.1.1.6.2 Request grouping depending on searching purpose

2.1.2.3.1.1.6.3 Prefer catalog grouping > list of titles because there will be less scrolling in this way

2.1.2.3.1.1.6.4 Prefer having catalogs and then abstracts while loading

2.1.2.3.1.1.7 Use catalog searching when failing to specify query, doing non-objective searching or know which catalog the searching items belong to

2.1.2.3.1.1.7.1 Use catalog searching when failing to specify the query

2.1.2.3.1.1.7.2 Use catalog searching for non-objective searching

2.1.2.3.1.1.7.3 Use catalog searching when users can specify the catalog the searching items belong to (by common sense/catalog names)

2.1.2.3.1.2 Summary: for wireless searching, only important content is required. Therefore, summary and concentrated information, such as abstract (no more than 3 complete sentences), heading, indexing and local catalog of site are preferred as they can help users to understand, search and identify useful content quickly.

2.1.2.3.1.2.1 Summary [abstract, heading, indexing, local catalog of site] is preferred as it can help users to understand, search and identify useful content quickly

2.1.2.3.1.2.1.1 Abstracts can help users to understand, search and identify useful content/ search results quickly

2.1.2.3.1.2.1.2 Headings can be helpful in searching

2.1.2.3.1.2.1.3 Prefer having a local catalog/menu to summarise a Web site for local searching

2.1.2.3.1.2.1.4 Prefer short abstracts, description on headings and links

2.1.2.3.1.2.1.5 Prefer page indexing

2.1.2.3.1.2.2 Concentrated information is preferred as it comes with faster accessing speed and less scrolling

2.1.2.3.1.2.2.1 Concentrated information with faster accessing speed and less scrolling

2.1.2.3.1.2.2.2 Prefer concentrated approach to show information

2.1.2.3.1.2.3 Only important content is required so as to display as much information on a single screen as possible

2.1.2.3.1.2.3.1 Required important content/text only when searching with PDA

2.1.2.3.1.2.3.2 Prefer showing as many results on a single screen as possible

2.1.2.3.1.2.4 Prefer abstract in no more than 3 complete sentences

2.1.2.3.1.2.4.1 Preferred size of abstract : 1 - 2 sentences with complete sentences

2.1.2.3.1.2.4.2 Preferred size of abstract : 2 - 3 sentences with complete sentences

2.1.2.3.2 Sorting: Prefer sorting items by relevance ranking, alphabetical order or user interests if possible

2.1.2.3.2.1 Prefer sorting with search result by relevance

2.1.2.3.2.2 Prefer sorting with search result by relevance / alphabetical order

2.1.2.3.2.3 Suggest to sort according to user interest / alphabetical order

2.1.2.3.3 Presentation of search result: only necessary content (abstract and heading) is required in search result list. Do not prefer further categorization and searching on search result. In addition, highlighting searching keywords, numbering and the total number of search result are required. Moreover, users are able to browse more pages, but no more than 20 items on each page when knowing the total number of pages and usually modify query before browsing all the results.

2.1.2.3.3.1 Able to browse more pages but no more than 20 items on each page, when know total number of pages.

2.1.2.3.3.1.1 Accept no more than 20 search results on a page

2.1.2.3.3.1.1.1 Able to accept no more than 10 search results on a page

2.1.2.3.3.1.1.2 Able to accept no more than 20 search results on a page

2.1.2.3.3.1.2 Able to browse no more than 20 pages if the number of total pages is known. Otherwise, no more than 5 pages

2.1.2.3.3.1.2.1 Able to browse no more than 10 pages if the number of pages is known.

2.1.2.3.3.1.2.2 Able to browse no more than 20 pages if the number of pages is known

2.1.2.3.3.1.2.3 Able to browse no more than 4 pages without knowing the number of pages

2.1.2.3.3.1.2.4 Able to browse no more than 5 pages without knowing the number of pages

2.1.2.3.3.1.2.5 Will only browse 1- 4 pages of searching results

2.1.2.3.3.2 Only necessary content (abstract and heading) is required in search result list. Do not prefer further categorization on search results

2.1.2.3.3.2.1 Abstract + heading would be enough for browsing search result

2.1.2.3.3.2.2 Required important content/text only when searching with PDA

2.1.2.3.3.2.3 Do not prefer further categorization of keyword of search results, only a list of headings + abstract is required

2.1.2.3.3.3 Highlighting searching keyword, numbering and total number of search results.

2.1.2.3.3.3.1 Prefer highlighting on searching keyword

2.1.2.3.3.3.2 Prefer showing number of search results, for deciding whether to the modify query or not

2.1.2.3.3.3.3 Request number / sequence indicator to indicate sequence of search results, headings , titles and catalog

2.1.2.3.3.4 Users would modify query before browsing all the results. Searching existing result is not useful for them.

2.1.2.3.3.4.1 Searching existing search result is not useful, users will choose extending the query

2.1.2.3.3.4.2 Modify query after reading 50 - 60 results

3 Nature of information

3.1.1 Desktop content: although desktop content is rich in information and picture. PDA's screen is too small, while too many un-suitable pictures and much space at desktop based content results in many scrolling, slow on loading speed, difficulty in understanding and locating information. Users feel tired and difficult to capture full picture of information.

3.1.1.1 Desktop content comes with more information and pictures than content for PDA

3.1.1.2 Screen too small that content usually larger than a screen and can't be fitted onto a single screen

3.1.1.2.1 Screen too small for content when viewing desktop content with PDA

3.1.1.2.2 Content usually larger than a screen, have many scrolls before capturing full information of a page / difficult to capture full information

3.1.1.2.3 Information cannot be fitted onto a single screen when browsing desktop based content

3.1.1.3 Results in many scrolling, slow on loading speed and difficulty in understanding and locating information

3.1.1.3.1 Too many scrolling when viewing desktop content with PDA , don't prefer having too many scrolling

3.1.1.3.2 Too slow on loading speed when viewing desktop content with PDA

3.1.1.3.3 Difficult to find those wanted catalogs / functions when viewing desktop content with PDA

3.1.1.4 Too much space when browsing desktop content with PDA

3.1.1.5 Too many pictures and size of picture not suitable

3.1.1.5.1 Pictures are becoming smaller and cannot be seen clearly when viewing desktop content with PDA

3.1.1.5.2 Pictures of desktop content are too large for PDA if they are viewed directly with PDA

3.1.1.6 Users feel tired, difficult to capture full picture of information

3.1.1.6.1 Users feel much tired than browse those content at desktop when viewing desktop content with PDA

3.1.1.6.2 Cannot see the whole page as at desktop when viewing desktop content with PDA

3.1.2 Supportive

3.1.2.1 Additional information : related items at the end of page / related pictures can help information retrieval and make users comfortable

3.1.2.1.1 Prefer having related item on the same page

3.1.2.1.2 Related pictures can help identifying related content, draw users' attention, facilitating understanding of content and make users comfortable

3.1.2.2 Status / history tracing functions : required status / history tracing functions for sequential movement / direct jumping in between different status / location at searching

3.1.2.2.1 Required status functions e.g. backward, forward, browsing path. If no status functions, users have to go back to the top level of catalog before visiting other catalogs. This wastes a lot of time when browsing

3.1.2.2.2 [Sequential movement] backward and forward can be useful and suitable at the bottom of page

3.1.2.2.2.1 Backward [input/modify query, to previous status/catalog, back to search result list], forward used for forwarding to the next page

3.1.2.2.2.1.1 Backward function used frequently for inputting/modifying query, to previous status/catalog, and search result list

3.1.2.2.2.1.2 Forward function used for forwarding to the next page

3.1.2.2.2.2 Better at the bottom of page

3.1.2.2.2.2.1 Frequent of use [backward and forward button at the bottom of page] > [forward button at the bottom of page] > [backward and forward button at the top of page]

3.1.2.2.3 [Direct jumping]

3.1.2.2.3.1 Can type and go to a URL directly

3.1.2.2.3.1.1 Prefer address bar for typing in URL directly / to visit URL directly

3.1.2.2.3.2 Directly jumping in between status

3.1.2.2.3.2.1 Jumping in between different status

3.1.2.2.3.2.1.1 Concept path for showing 3 - 4 status for back trace

3.1.2.2.3.2.1.1.1 Prefer concept path which can show nearest 3 - 4 status

3.1.2.2.3.2.1.1.2 Prefer concept path, for showing types of items and location

3.1.2.2.3.2.1.2 Page number list for jumping

3.1.2.2.3.2.1.2.1 Required page numbers e.g. << 1,2,3,4,5 >> for jumping to different location, showing existing location

3.1.2.2.3.2.2 History for re-query and back trace of search result

3.1.2.2.3.2.2.1 Prefer history tracing, e.g. history bar at searching result

3.1.2.2.3.2.2.2 Prefer providing input history as it saves time for re-query

3.1.3 Concentrated

3.1.3.1 Catalog: Users seeking a feasible mechanism to group information and catalog is one of the possible methods for helping users to identify information quickly by providing tidy display and a summary of information. They use catalog searching when can't specify query or doing non-objective searching or know which catalog the searching items belong to. Users accept slight changes on catalog component based on dynamic data if necessary and they should be based on common and static style.

Also, they prefer detailed hierarchical catalogs with multiple paths that match users' common sense, user's past experience and key terms at content but no more than 30 items in each of the catalogs. As for displaying, catalogs and headings should not be put together and catalog names should be separated clearly on a single screen. Moreover, there should provide searching on catalog and content, show number of items at each catalog and have sequence indicators on catalogs and headings.

3.1.3.1.1 Accept slight changes on catalog component based on dynamic data if necessary and they should be based on common and static style

3.1.3.1.1.1 Able to accept addition of new catalogs with only slight changes to the existing catalogs

3.1.3.1.1.2 Prefer common style and static standard in layout and catalog design

3.1.3.1.1.3 Prefer generating of hierarchy based on data only

3.1.3.1.2 Prefer detailed hierarchical catalogs with multiple paths that matched common sense, users' past experience and key terms at content but no more than 30 items in each of the catalogs

3.1.3.1.2.1 Prefer detailed hierarchical catalog with multiple paths and no more than 30 items in each of the catalogs

3.1.3.1.2.1.1 Prefer detailed hierarchical catalog with multiple paths

3.1.3.1.2.1.1.1 Prefer hierarchical catalogs than simple catalog

3.1.3.1.2.1.1.2 Prefer having multiple paths for the same news / content if the content /news belongs to different catalogs (e.g. different paths can access the same news / content)

3.1.3.1.2.1.1.3 Prefer detailed categorization for directorial searching

3.1.3.1.2.1.2 Accept 0 - 30 items in a catalog

3.1.3.1.2.1.2.1 Able to accept no more then 10 items in a catalog

3.1.3.1.2.1.2.2 Able to accept no more then 20 items in a catalog

3.1.3.1.2.1.2.3 Able to accept no more then 30 items in a catalog

3.1.3.1.2.1.2.4 Accept empty catalog

3.1.3.1.2.2 Catalogs should match the common sense and users' past experience but also match key terms at content

3.1.3.1.2.2.1 The catalogs better match users' common sense and past experience

3.1.3.1.2.2.1.1 Better start with catalogs that match users' common sense

3.1.3.1.2.2.1.2 Catalog name is useful if users know the catalog the searching information belongs to (by common sense or past experience)

3.1.3.1.2.2.1.3 For catalog searching, the users will find the answers quickly if he knows the catalog the searching item belongs to

3.1.3.1.2.2.1.4 Will be based on past experience and common sense to perform searching with catalogs

3.1.3.1.2.2.1.5 Will not try unknown catalogs if users cannot find searching items in known catalogs

3.1.3.1.2.2.2 Catalog names should match the keywords of the underlying content

3.1.3.1.2.2.3 Many existing catalogs can match users' common sense

3.1.3.1.2.2.3.1 Catalog not match users' common sense

3.1.3.1.2.2.3.2 No catalogs that match users' searching requirements

3.1.3.1.3 Catalogs and headings should not be put together and catalog names should be separated clearly on a single screen.

3.1.3.1.3.1 Catalogs and headings should not be put together

3.1.3.1.3.1.1 Do not prefer mixing catalog names and titles that results in many scrolling and long page

3.1.3.1.3.2 Catalog names should be separated clearly and many catalogs should be shown on a single screen

3.1.3.1.3.2.1 Catalog names / items / titles / links / function keys should be separated clearly

3.1.3.1.3.2.2 Show as many catalogs on a single screen as possible

3.1.3.1.4 Prefer having search on catalogs and content, show number of items at each catalog and have sequence indicators

3.1.3.1.4.1 Prefer searching on content and catalog name

3.1.3.1.4.2 Prefer showing number of items in a catalog followed by catalog names

3.1.3.1.4.3 Request numbers / sequence indicators to indicate sequence of search results, headings , titles and catalogs

3.1.3.1.5 Catalogs help users to identify information quickly by providing tidy display and a summary of information

3.1.3.1.5.1 Catalog names help users to identify types and relevant information quickly

3.1.3.1.5.1.1 Title and catalog names can help users to understand and identify useful content quickly

3.1.3.1.5.1.2 Catalog names help users to know the types of news/information in the catalogs

3.1.3.1.5.2 Catalogs provide tidy display

3.1.3.1.5.3 Catalogs can summarise information for searching

3.1.3.1.5.3.1 Prefer to have a local catalog/menu to summarise a Web site for local searching

3.1.3.1.6 A feasible mechanism to group information by catalog

3.1.3.1.6.1 Prefer "favorites" of Internet Explorer (catalog) than bookmark of Klondlike (list) as good design for group information

3.1.3.1.6.2 Request grouping depending on searching purpose

3.1.3.1.6.3 Prefer catalog grouping > list of titles because there will be less scrolling in this way

3.1.3.1.6.4 Prefer having catalogs and then abstracts while loading

3.1.3.1.7 Use catalog searching when failing to specify query, doing non-objective searching or know which catalog the searching items belong to

3.1.3.1.7.1 Use catalog searching when failing to specify the query

3.1.3.1.7.2 Use catalog searching for non-objective searching

3.1.3.1.7.3 Use catalog searching when user can specific which catalog the searching items belong to (by common sense/catalog names)

3.1.3.2 Summary: For wireless searching, only important content is required. Therefore, summary and concentrated information, such as abstract (no more than 3 complete sentences), heading, indexing and local catalog of site are preferred as they can help users to understand, search and identify useful content quickly.

3.1.3.2.1 Summary [abstract, heading, indexing, local catalog of site] is preferred as it can help users to understand, search and identify useful content quickly

3.1.3.2.1.1 Abstracts can help users to understand, search titles and identify useful content/ search results quickly

3.1.3.2.1.2 Headings can be helpful in searching

3.1.3.2.1.3 Prefer having a local catalog/menu to summarise a Web site for local searching

3.1.3.2.1.4 Prefer short abstracts, description on headings and links

3.1.3.2.1.5 Prefer page indexing

3.1.3.2.2 Concentrated information is preferred as it comes with faster accessing speed and less scrolling

3.1.3.2.2.1 Concentrated information with faster accessing speed and less scrolling

3.1.3.2.2.2 Prefer concentrated approach to show information

3.1.3.2.3 Only important content is required so as to display as much information on a single screen as possible

3.1.3.2.3.1 Required important content/text only when searching with PDA

3.1.3.2.3.2 Prefer showing as many results on a single screen as possible

3.1.3.2.4 : Prefer abstract in no more than 3 complete sentences

3.1.3.2.4.1 Preferred size of abstract : 1 - 2 sentences with complete sentences

3.1.3.2.4.2 Preferred size of abstract : 2 - 3 sentences with complete sentences

3.1.4 Presentation of search result: only necessary content (abstract and heading) is required in the search result list. Do not prefer further categorization and searching on search result. In addition, highlighting on searching keyword, numbering and total number of search results are required. Moreover, users are able to browse more pages, no more than 20 items on each page, when knowing the total number of pages and usually modify query before browsing all the results.

3.1.4.1 Able to browse more pages, no more than 20 items at each page, when knowing the total number of pages.

3.1.4.1.1 Accept no more than 20 search results at a page

3.1.4.1.1.1 Able to accept no more than 10 search results on a page

3.1.4.1.1.2 Able to accept no more than 20 search results on a page

3.1.4.1.2 Able to browse no more than 20 pages if the number of total pages is known. Otherwise, no more than 5 pages

3.1.4.1.2.1 Able to browse no more than 10 pages when the number of pages is known

3.1.4.1.2.2 Able to browse no more than 20 pages when the number of pages is known

3.1.4.1.2.3 Able to browse no more than 4 pages without knowing the number of pages

3.1.4.1.2.4 Able to browse no more than 5 pages without knowing the number of pages

3.1.4.1.2.5 Will only browse 1 - 4 pages of searching results

3.1.4.2 Only necessary content (abstract and heading) is required in the search result list. Do not prefer further categorization on search results

3.1.4.2.1 Abstract + heading would be enough for browsing and search result

3.1.4.2.2 Required important content/text only when searching with PDA

3.1.4.2.3 Do not prefer further categorization of keyword search results. Only a list of headings + abstract is required

3.1.4.2.4 Highlighting searching keyword, numbering and the total number of search results.

3.1.4.2.4.1 Prefer highlighting searching keyword

3.1.4.2.4.2 Prefer showing number of search results, for deciding whether to modify the query or not

3.1.4.2.4.3 Request number / sequence indicator to indicate sequence of search results, headings , titles and catalogs

3.1.4.3 Users will modify query before browsing all the results and search on search result is not useful for them.

3.1.4.3.1 Search on existing search result is not useful, users will choose extending the query

3.1.4.3.2 Modify query after reading 50 - 60 results