

University of Strathclyde

Department of Computer and Information Science

# Personal Predictive Internet Content Pre-caching for Mobile Devices

Andreas Komninos

A thesis presented for the fulfilment of the requirements  
for the degree of Doctor of Philosophy

2005

#### Declaration of Author's Rights

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by University of Strathclyde Regulation 3.49. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

## Acknowledgements

“I am indebted to my father for living, but to my teacher for living well”

*Alexander the Great*

There is One, who I believe in, who is the provider of everything in my life, and who has blessed me with wonderful parents and wonderful teachers, and with all the persons and means that made this work possible. To Him I extend the greatest gratitude, not only for this work, but for everything.

My beloved parents, without your support I would be nothing. I can not repay you in any way for all you have endured for me, other than with a promise that I will always aspire to offer my own children the same you have offered me,

My teachers, from the first to the last, but now especially you, Mark, who has guided me with patience and genuine interest to continue and to finish this work, who has greeted me always with a smile and believed in me, as a colleague and as a friend,

My closest friends, Andreas, Sakis, Andreas, Manos, my cousins, Costas and Nikos and my dear sister, Margarita, through good times and through bad, who were always there when I needed them and for who I will always be there if they need me,

Those who have helped, in any way, small or large, those who have supported and believed in me from the beginning of this work, those that we've drank from the same wine and ate from the same food, all my friends from the Greek community in Glasgow and my friends from countries close or far away, those who waited for me, the bright sun and the infinite blue of the land I was born, and finally you, sweet Alexia,

I thank you.

## Abstract

Motivated by the disparity of desktop and mobile Internet access, both in terms of available bandwidth and in terms of cost, this thesis presents research into an alternative method of making Internet content available for mobile users. This method is based on the extraction of information regarding the user's activities and interests from their electronic calendar, and pre-loading their mobile device with Internet content, using a land-based connection.

The main aim of the thesis is to investigate whether calendars can indeed provide information that can be used to pre-fetch useful Internet content for mobile users. While it is expected that such an approach cannot fulfil the entirety of Internet content needs for a user, the thesis aims to investigate the extent to which a mobile cache can be populated with relevant documents that the user could find of interest.

Further to this, the thesis is concerned with the potential of calendar entries to be used as sources for web query generation, independently of the entry brevity and without the direct involvement of the user. This is an essential step for the investigation of the original aim of the thesis, given that an appropriately formulated web query would have a better chance of retrieving relevant documents and thus populate the mobile cache with more appropriate results. Finally, an attempt is made to show that such a predictive pre-caching system is able to adjust itself to the preferences and circumstances of the user as an individual, in order to obtain optimal retrieval performance.

In the following chapters, this thesis presents evidence that supports these main hypotheses, while presenting further research outcomes which concern the usability and interaction patterns within electronic calendars, the document reading behaviour on mobile devices and the suitability of implicit interest indicators for information retrieval on mobile devices.

## Table of Contents

1	Introduction .....	12
1.1	Current options in Internet Access .....	12
1.1.1	Desktop-based Internet Access .....	13
1.1.1.1	Broadband vs. Modem Technology .....	13
1.1.1.2	Costs .....	15
1.1.2	Mobile Internet Access .....	16
1.1.2.1	WAP & GSM .....	17
1.1.2.2	GPRS and other 2.5G solutions .....	18
1.1.2.3	3G Networks .....	20
1.1.2.4	Wi-Fi .....	22
1.1.2.5	Mobile Access Costs .....	24
1.2	Mobile Device Support for Internet Access .....	25
1.2.1	Device Characteristics .....	25
1.2.2	Device Limitations .....	25
1.2.2.1	Device Screens .....	25
1.2.2.2	Input Methods .....	27
1.2.2.3	Internet Connection .....	28
1.3	Internet content Pre-Caching as an alternative approach .....	30
1.3.1	Summary of the current state in Mobile Internet Access .....	30
1.3.2	Internet Content Pre-Caching: An alternative approach .....	31
1.3.3	Hypotheses .....	32
1.4	Structure of the thesis .....	33
2	A review of existing research .....	35
2.1	Electronic and paper-based Calendar usage .....	35
2.2	Pre-fetching on large scale for internet content providers .....	39
2.2.1	What is caching? .....	39
2.2.2	Caching strategies .....	40
2.2.3	What is pre-fetching .....	42
2.2.4	Research in Large scale (server-side) Pre-fetching .....	44
2.2.5	Implemented systems .....	49
2.3	Pre-fetching on a more personal level (client and proxy side) .....	50

2.3.1	The application of pre-fetching techniques on or near the client.....	50
2.3.2	Research in personalised pre-fetching .....	51
2.4	Anticipating users needs .....	59
2.4.1	Acquiring user context and learning user preferences .....	59
2.4.1.1	The new user problem.....	59
2.4.1.2	Solving the new user problem with filtering techniques.....	60
2.4.2	Long term acquisition of individual user preferences.....	64
2.4.2.1	The concept of Relevance Feedback.....	64
2.4.2.2	Using Implicit Relevance Feedback for the acquisition of user preferences .....	65
2.5	Query formulation.....	70
2.5.1	Manual and Automatic Query formulation.....	70
2.5.2	Enhancing queries (query expansion). .....	71
2.5.2.1	Automatic and Manual Thesauri.....	73
2.6	Summary .....	81
3	Personal predictive Internet content precaching for mobile devices .....	83
3.1	Introduction .....	83
3.2	Choosing a programming language and system environment .....	85
3.3	Data exchange mechanism.....	86
3.4	Obtaining Calendar data (Calendar Exporter).....	86
3.4.1	Theoretical Design .....	86
3.4.2	Implementation .....	87
3.5	Identifying candidate keywords (Keyword Generator).....	88
3.5.1	Theoretical Design .....	88
3.5.2	Implementation .....	89
3.6	Formulating Queries (Keyword analyser).....	93
3.6.1	Theoretical Design .....	93
3.6.2	Implementation .....	95
3.7	Pre-fetching web documents (Web Searcher).....	98
3.7.1	Theoretical Design .....	98
3.7.2	Implementation .....	100
3.8	The handheld component .....	103

3.8.1	Theoretical Design .....	103
3.8.1.1	Presenting Results .....	103
3.8.1.2	Processing relevance feedback.....	103
3.8.2	Implementation .....	104
3.9	Updating additional keyword weights (Weight Updater) .....	108
3.9.1	Theoretical Design .....	108
3.9.2	Implementation .....	109
4	Experimentation with users.....	112
4.1	Introduction .....	112
4.2	Analysing Calendar contents and Usage.....	112
4.2.1	Calendar content analysis.....	112
4.2.2	Calendar Usage Questionnaire.....	115
4.2.2.1	Question 1 .....	116
4.2.2.2	Question 2 .....	117
4.2.2.3	Question 3 .....	118
4.2.2.4	Question 4 .....	120
4.2.2.5	Question 5 .....	120
4.2.2.6	Question 6 .....	122
4.2.2.7	Question 7 .....	124
4.2.3	Summary of Findings.....	124
4.3	Web search and query formulation behaviours.....	125
4.4	Automatically identifying entry categories.....	127
4.4.1	Experiment set-up .....	127
4.4.2	Results and analysis .....	131
4.4.2.1	Analysis targets .....	131
4.4.2.2	Original results .....	131
4.4.2.3	Analysis of original results .....	133
4.4.2.4	Revised experiment design and results .....	134
4.4.2.5	Summary of Findings.....	136
4.5	Pre-caching Internet Content for mobile devices.....	136
4.5.1	Experiment design.....	136
4.5.2	Initial experiment setup.....	138

4.5.3	Actual experiment .....	140
4.5.4	Further discussion .....	143
4.5.4.1	Experimental environment .....	143
4.5.4.2	Statistical Confidence .....	143
4.5.5	Summary of Findings .....	144
5	Conclusions and future work .....	146
5.1	Review of original hypotheses .....	146
5.1.1	Hypothesis 1 .....	146
5.1.2	Hypothesis 2 .....	147
5.1.3	Hypothesis 3 .....	147
5.1.4	Hypothesis 4 .....	147
5.2	Further findings .....	148
5.3	Future Work .....	150
5.3.1	Long term testing for user-system adaptability .....	150
5.3.2	Supplementary methods for the manipulation of the additional keyword database .....	150
5.3.3	Expanding the search to include user desktop computer contents ...	151
5.4	Summary and Conclusions .....	153
	References .....	155
	Appendices .....	168
	Appendix 1: Calendar Usage Questionnaire .....	169
	Appendix 2: Calendar Entry submission form .....	170
	Appendix 3: Web query test and results .....	171
	Appendix 4: Automatic categorisation test instructions .....	181
	Appendix 5: Sample from an automatic categorisation test log .....	182
	Appendix 6: Final test list of appointments .....	183
	Appendix 7: Sample log from final test .....	198
	Appendix 8: T-Test details .....	200
	Appendix 9: Publications .....	201

## Table of Figures

Figure 1: A typical WAP site on an WAP-enabled 2G phone.....	17
Figure 2: The five radio interfaces contained in the IMT-2000 standard .....	21
Figure 3: The Thunderhawk browser.....	26
Figure 4: The Opera mobile browser .....	27
Figure 5: The T-Mobile MDA IV .....	28
Figure 6: Typical PDA devices.....	29
Figure 7: Internet-enabled mobile phones.....	29
Figure 8: System operation principles overview diagram.....	32
Figure 9: Caches in the WWW .....	41
Figure 10: Operational principles diagram (Swaminathan et al.) .....	56
Figure 11: Fidel's decision tree for term selection.....	73
Figure 12: Revised system overview .....	84
Figure 13: Base XML structure created by Outlook Export module.....	87
Figure 14: Extracts from the "travel" category identifier database.....	90
Figure 15: XML expansion caused by the Keyword Generator module .....	91
Figure 16: Sample additional keyword entries.....	95
Figure 17: XML expansion caused by the Keyword Analyser module.....	97
Figure 18: The multi-tier retrieval tree.....	98
Figure 19: The XML version of the Google pages .....	101
Figure 20: XML expansion caused by the Keyword Analyser module.....	102
Figure 21: The mobile User Interface. ....	105
Figure 22: The auto-summary function and explicit relevance feedback screens. ..	106
Figure 23: XML structure for logging user interactions.....	106
Figure 24: Claypool's findings on relevance and reading time correlation .....	110
Figure 25: Question 1 responses .....	117
Figure 26: Question 2 responses .....	118
Figure 27: Question 3 responses .....	119
Figure 28: Question 4 responses .....	120
Figure 29: Question 5 responses .....	121
Figure 30: Question 6 responses (aggregate).....	123

Figure 31: Question 7 responses .....	124
Figure 32: Amount of searches formed manually for each category .....	126
Figure 33: Entry categorisation screen.....	128
Figure 34: Explanation screen.....	129
Figure 35: Comparison of categories with original users' allocations.....	132
Figure 36: Initial group average reading times vs. feedback ratings .....	139
Figure 37: Experiment group average reading times vs. feedback ratings .....	140

## Table of Tables

Table 1 : Typical Internet connections and their bandwidth.....	15
Table 2: Typical broadband and dial-up charges in the UK (Feb 2005).....	16
Table 3: Wireless connection types and their speed .....	17
Table 4: Wireless Internet Access costs in the UK (feb 2005) .....	24
Table 5: Konstan's proposed observable behaviour. ....	68
Table 6: Categorisation of observable behaviours by Kim and Oard .....	68
Table 7: Keyword score adjustment weights .....	109
Table 8: Calendar entry categories and their frequencies .....	113
Table 9: Calendar entry category descriptions.....	114
Table 10: The entry categories and their numerical representation .....	128
Table 11: Summary of original results.....	133
Table 12: Revised summary results .....	135
Table 13: Comparison of Original and Revised Results.....	135
Table 14: Studies on average document reading times (msec) vs. perceived document usefulness .....	139
Table 15: Total documents vs. Opened documents (percentage) .....	141
Table 16: Total documents vs. Opened documents (absolute values) .....	141
Table 17: Summary viewing as a deciding factor for opening a document.....	142
Table 18: Average document scores (0-lowest, 5-maximum) .....	142
Table 19: T-test results.....	144
Table 20: Google Desktop's pre-configured search modules .....	152

## Chapter 1

### Introduction

# 1 Introduction

## 1.1 *Current options in Internet Access*

The Internet has traditionally been accessed from desktop computers, from organisations in its early origins, and more recently from businesses, offices and homes throughout the world. It has been recognised as such an important tool, not only for business but also for personal usage (infotainment), to such an extent that telecommunications providers have invested heavily in providing media for accessing the Internet even when one is on the move. Ericsson, one of the world's leading 2G and 3G mobile systems, identifies significant business opportunities for mobile Internet access in the following fields<sup>i</sup>:

- Messaging: e-mail and multimedia messages delivered in an instant
- Gaming: improved graphics and instant response for interactive game playing
- Location-based services: by-the-minute charging is slashed with GPRS always-on connectivity, which makes real-time services such as location-based services dramatically cheaper
- Personal information management: synchronization of calendar and address book between mobile phone, PDA and PC
- Personalized portals: fast, easy access to services with the user's own tag
- Entertainment: faster downloading of multimedia content
- Banking and finance: check access to account details, payment facilities, stock news and prices

With the latest technological advances, many options are offered on Internet access. These are no longer constrained to choices of speed, but have expanded to the location of the access point (fixed or roaming) and to the type of device used to access the Internet.

---

<sup>i</sup> Ericsson Mobile Systems, [http://www.ericsson.com/network\\_operators/mobilesystems/](http://www.ericsson.com/network_operators/mobilesystems/)

In the following sections, a review of the current options for Internet access will be made. The scope of this review is to highlight the advantages and disadvantages, both in terms of speed and access cost, of desktop and, more importantly, mobile Internet access. The description of the current situation in mobile Internet access provides the motivation behind the research described in later parts of this thesis.

A proposal will be made at the end of this chapter for an alternative, hybrid solution to the problems that mobile Internet access is currently associated with. This solution is based on the use of information contained in a user's calendar as a source that can be exploited in order to enable the formulation of predictions on the type of Internet content a user may need to support their future activities. Using land-based Internet access, a "docked" mobile device could then automatically pre-cache the predicted Internet content for the user, making it available on mobile situations where the cost and available bandwidth would otherwise present an obstacle to efficient Internet access. This idea is described in more detail in section 1.3.2, however, before this is presented, it is important to discuss the current situation in Internet access, both through land-based and mobile connections.

### **1.1.1 Desktop-based Internet Access**

Traditionally the Internet has required a desktop PC in order to be accessed. It is therefore natural that most Internet access options have been developed with the PC in mind. In general the amount of Internet users worldwide is estimated at 817.4 million users for February 2005<sup>ii</sup>, which is approximately 13% of the global population. So far Europe and North America have the largest penetration figures at 31.6% and 66.5% respectively.

#### **1.1.1.1 Broadband vs. Modem Technology**

Modem technology initially provided very slow access to the Internet, which was acceptable initially, as the exchange of information on the Internet consisted of mainly textual elements. While the Internet started to grow, the inclusion of

---

<sup>ii</sup> Internet World Stats, Usage and Population Statistics, [www.internetworldstats.com](http://www.internetworldstats.com)

multimedia elements in web sites soon posed a problem since the bandwidth capacity afforded by the latest technology modems (56Kbps) was unable to provide satisfactorily quick access to the content users wanted (or were forced) to see.

With the frustration caused by the slow speeds offered by modem access, much demand has been placed on the provision of broadband (high-speed) Internet access for personal use. While formerly considered a privilege of organisations, broadband access is now widely available, averaging to approximately 128m subscribers worldwide for December 2004<sup>iii</sup>. An interesting fact is that the same source quotes more than 78m of these are DSL subscribers, a figure estimated closer to 85m for Q3 2004 by [www.internetworldstats.com](http://www.internetworldstats.com). Many governments around the world, especially in developed countries, actively promote the spread of broadband access for citizens, as it is believed to be a means for promoting the growth of e-business and e-government. Special legislative adjustments or even subsidies are often provided in order to fuel the e-economy growth.

The following table shows the comparison in connection speeds for commonly encountered desktop land-line connections today. The discrepancy between the bandwidth capacities supported between even the most basic ADSL broadband connection and the speediest modem connection, shows the difference in the quality of service that can be experienced by broadband subscribers.

Line Speed	Connection name
44.736 Mbps	T3, DS3 North America
10Mbps	Thin Ethernet, Category 3 Cable, cable modem
6.312 Mbps	T2, DS2 North America
6.144 Mbps	Highest speed ADSL downstream (2 pair)
3.152 Mbps	T1c, DS1c
1.544 Mbps	ADSL, T1, DS1 North America
896 Kbps	High speed ADSL downstream
512 Kbps	Fast ADSL downstream
384 Kbps	Standard ADSL downstream
256 Kbps	Fast ADSL upstream

---

<sup>iii</sup> 128m Broadband Subscribers Worldwide, <http://www.dmeurope.com/default.asp?ArticleID=4908>

256 Kbps	Average ADSL downstream
230.4 Kbps	AppleTalk networks
128 Kbps	Standard ADSL upstream
128 Kbps	ISDN
64 Kbps	ISDN (std)
56 Kbps	K56FLEX, U.S. Robotics X2 modems, V.90
33.6 Kbps	K56FLEX, X2 modem communications rate
28.8 Kbps	V.34, Rockwell V.Fast Class modems
20 Kbps	Level 1 cable, minimum cable data speed
19.2 Kbps	V.32ter modem,
14.4 Kbps	V.32bis modem, V.17 Fax
9600 bps	modem speed of the early 1990s

**Table 1 : Typical Internet connections and their bandwidth**

An added advantage of broadband connections is that they can be left “always-on”, operating around the clock, if a user finds it necessary. The dial-up time associated with normal modem connections is almost reduced to milliseconds for those broadband connections that require it (ADSL) and the connection can be left on indefinitely, as charges are made on a flat monthly rate basis, compared to by-the-minute costs of dial-up access. Further to this, broadband connections do not tie-up the user’s telephone line; this alleviates the need for the installation of a second phone line.

#### **1.1.1.2 Costs**

Current access costs for Internet access vary from provider to provider, however, the following table is indicative of typical connection charges in the UK, which is currently in the fourth place in Internet penetration in the European Union (Internet users amount to 58.7% of the population, top country is Sweden where 73.6% of the population are Internet users)<sup>iv</sup>.

---

<sup>iv</sup> Internet World Stats, Europe Internet Usage Statistics  
<http://www.internetworldstats.com/stats4.htm#eu>

Connection type	Connection options	Price (GBP)
Modem dial-up (56K)	Pay-per-minute	From 0.003-0.01/min, no monthly subscription fee
Modem dial-up (56K)	Flat charge	12.50/month, unlimited call duration
Cable (300Kbps)	Flat charge	17.99/month
Cable (750Kbps)	Flat charge	24.99/month
Cable (1.5Mbps)	Flat charge	37.99/month
ADSL (512Kbps)	Flat charge	17.99/month
ADSL (1Mbps)	Flat charge	24.99/month
ADSL (2Mbps)	Flat charge	29.99/month

**Table 2: Typical broadband and dial-up charges in the UK (Feb 2005) (sources: NTL, AOL UK)**

From this table it becomes obvious that the costs of subscribing to broadband are not much higher than modem dial-up, so with up to 10 times quicker connections than dial-up for even the most basic access packages, it is easy to understand why more and more people are seriously considering broadband as an option.

### **1.1.2 Mobile Internet Access**

While mobile computing devices have been available for more than a decade, mostly in the form of portable (laptop) computers, connection of these devices has traditionally required a hook-up to land lines, such as a LAN or more commonly a telephone line for dial-up.

With the advent of mobile telephony, it was not long before data services became available over wireless connections in much the same way as with land-based telephony. The ability to send and receive faxes was incorporated in many mobile phones and other data transmission services, such as the Short Message System became increasingly popular. One of the major problems, however, was that the bandwidth afforded to data services by 2<sup>nd</sup> generation mobile networks was severely restrictive. The GSM standard (Global System for Mobile communications), implemented commercially since 1991, quickly became the most popular mobile telephony standard worldwide and was adopted by more than 110 countries worldwide. It provided a digital system for communications and data services at a bandwidth of a mere 9600bps. The table below shows a comparison between the speeds of GSM and later technologies, which are currently pioneered by 3<sup>rd</sup>

generation mobile telephony networks and alternative connection methods such as Wi-Fi, which are discussed in the following sections.

Line Speed	Connection name
11 Mbps	802.11, 802.11b WI-FI
2 Mbps	<a href="#">802.11</a> , Original 802.11 <a href="#">WI-FI</a>
up to 2 Mbps	<a href="#">UMTS</a> wireless
384 Kbps	Enhanced Data GSM Environment ( <a href="#">EDGE</a> ) wireless
171.2 Kbps	<a href="#">GPRS</a> wireless
56 Kbps	<a href="#">HSCSD</a> wireless
9,600 Kbps	<a href="#">GSM</a> wireless telephone

**Table 3: Wireless connection types and their speed**

### 1.1.2.1 WAP & GSM

A push to allow users to exploit the offered bandwidth was not made until late in the 1990s with the implementation of WAP (Wireless Access Protocol) browsers in mobile phones. The service depended on marked-up text sites using WML (Wireless Markup Language) and was implemented in a dial-up fashion. Users quickly became disillusioned with what was marketed as the Mobile Internet, since the experience was very poor (and very expensive) compared to what the Internet looked like on desktop computers.



**Figure 1: A typical WAP site on an WAP-enabled 2G phone.**

Although HTML and XML were both supported by WAP, a better experience for the user required the construction of dedicated sites using WML, a mark-up language which was specifically devised for small screens and one-hand navigation without keyboards. Graphics and text were both supported by WML and scripting was also an option using WMLScript.

WAP was widely considered a failure, despite heavy marketing both by phone manufacturers and service providers. Televised promotions and printed material depicted a high-tech digital world, using advanced computer graphics and colour, to attract customers, while the reality of WAP was very far away from the advertising depictions. In 2001, WAP revenue accounted for a mere 0.5% of operator revenues<sup>v</sup>. Perhaps the single most important reason behind this failure was the obvious inadequacy of mobile phones to support Internet browsing. Because of physical constraints such as small screen sizes, monochromatic displays, small physical memory, limited keypad functionality and slow speed, the browsing experience was fundamentally flawed and the need for better devices and better infrastructure services was soon driving the mobile industry.

#### **1.1.2.2 GPRS and other 2.5G solutions**

The shortcomings of WAP and early 2<sup>nd</sup> generation mobile devices were addressed gradually as time allowed the manufacture of better screens and better usability features for web browsing on mobile phones. Apart from these issues, attempts were made to address the speed problems inherent in the GSM network by deploying high-speed data services on existing GSM networks (also termed as 2.5G).

GPRS (General Packet Radio Service) was perhaps the most popular enhancement implemented on top of the circuit-based GSM system. By exploiting unused channels in the GSM network and implementing packet-switching rather than circuit-switching (traditional GSM), it allows a theoretical maximum speed of approximately 171.2Kbps, almost three times the speed of a land-based modem. One other important feature of GPRS is that it does not require a user to dial-up; instead, the user is always connected. Charges for GPRS are therefore normally made

---

<sup>v</sup> The Joy of Text, [www.economist.com/business/displayStory.cfm?Story\\_ID=780694](http://www.economist.com/business/displayStory.cfm?Story_ID=780694)

depending on the traffic to and from the user's device, rather than the actual time the user is browsing. This reflects the sharing of the available bandwidth between users, compared to GSM charging where circuit-switching meant that a user had exclusive use of a bandwidth slice for as long as they wanted it.

While the theoretical speed of GPRS is very promising, in reality the service levels provided and experienced by the average user are a little below those of a 56Kbps land-based modem. This is due to the limitations that GPRS poses on the capacity of each cell in a network and because of its packet-switching nature, which dictates that users have to share the available bandwidth. A further disadvantage of GPRS is its high latency, which is typically in excess of 1 second. This means that the transmission of a service request or a service by the provider needs at least one second before it is fulfilled, therefore prohibiting the implementation of any truly interactive application.

Another improvement on the original circuit-switched network that was GSM is HSCSD (High-Speed Circuit Switched Data). This operates on the same principles as traditional GSM, although the enhancement in speed is achieved through different coding methods and the assignment of multiple time slots to a single user (up to 4). Further to this, different levels of error checking are allowed by HSCSD, depending on the quality of the radio link. This increases the maximum transfer rate to a theoretic 14.4Kbps under ideal conditions, where traditional GSM could only afford 9.6Kbps (which was, however, guaranteed). Using up to four time slots for a typical system, HSCSD can achieve its maximum theoretic speed of 57.6Kbps.

The access costs for HSCSD are considerably larger than for GPRS to the end user, therefore the adoption of this standard was not wide, although it is still offered as an option for EDGE and UMTS 3<sup>rd</sup> generation networks, which are described in the next section.

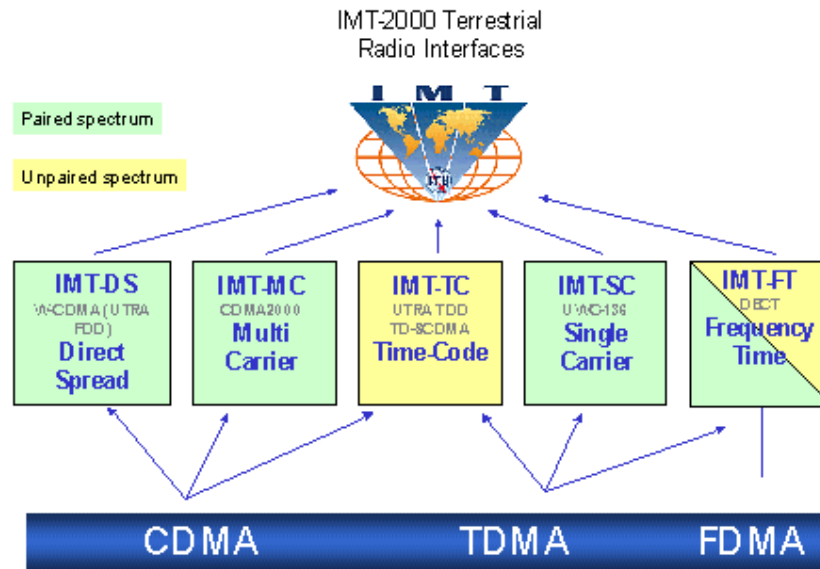
The 2.5G data transfer protocols mentioned above were deployed at a time where most mobile phones still had very limited display capabilities. However, for the first time, they enabled acceptable performance levels for mobile Internet access, especially because most GPRS-enabled phones could be paired with laptop computers (typically over infrared or Bluetooth connections) and used as wireless modems.

### **1.1.2.3 3G Networks**

The advances in mobile Internet access speeds achieved by the implementation of 2.5G solutions were important, however they did not solve the problem of mobile connectivity speeds. It became obvious that as the amount of broadband users grew, more people would come to expect the same speed and service quality on any Internet-capable platform, whether mobile or land-based. Further to this, mobile service providers began to realise that data services offered on top of existing voice services could generate even more revenue, thus came the birth of 3<sup>rd</sup> generation networks.

The International Telecommunications Union (ITU) published in 1999 a global standard for third generation wireless communications, called IMT-2000 (International Mobile Telecommunications-2000). This standard was the result of approximately ten years work by groups within the ITU and outside it, such as the 3<sup>rd</sup> generation Partnership Program (3GPP). The importance of setting this global standard lay in the fact that for the first time, full interoperability and internetworking of mobile systems could be achieved and the fragmentation in technologies that characterised the 2G networks could be forgotten.

One of the key aspects of the standard is the desire to provide seamless global roaming by allowing the users to move across countries and still use the same handset and number. The delivery of services such as voice, data, Internet and multimedia is to be made transparently via a number of media, such as satellite or fixed emitters. Finally, in order to achieve its promised services, the standard is expected to offer true broadband speeds of around 2Mbps for stationary or walking users and 384Kbps for fast moving ones (e.g. when in a vehicle). The IMT-2000 standard incorporates five possible radio interfaces, based on three different access technologies (FDMA, TDMA, CDMA)



**Figure 2: The five radio interfaces contained in the IMT-2000 standard**

FDMA (Frequency Division Multiple Access) is the term used to describe the assignment of a distinct frequency channel to each transmitter, so that receivers can discriminate amongst them by tuning to the desired channel. TDMA (Time Division Multiple Access) refers to a technology for shared radio networks that allow the sharing of the same frequency by several users. This is accomplished by the division of the frequency into *time slots* that the users employ to transmit in rapid succession. Finally CDMA (Code Division Multiple Access) is used to describe the usage of any form of *spread spectrum* by multiple transmitters to send to the same receiver, on the same frequency channel and at the same time, without interference. The spread spectrum technique in telecommunications transmits a signal in a bandwidth considerably greater than the frequency content of the original information.

The five radio interfaces approved by the ITU for IMT-2000 are:

- CDMA Direct Spread (Also known as W-CDMA or ULTRA-FDD, used in UMTS)
- CDMA Multi Carrier (Also known as CDMA2000)

- Time Code (Summarises ULTRA-TDD and TD-SCDMA)
- TDMA Single Carrier (Also known as EDGE)
- FDMA/TDMA (Also known as DECT)

Of these five standards, UMTS (Universal Mobile Telephone System), which is based on the W-CDMA technology, is currently the preferred solution in countries that operated GSM networks, especially in Europe. W-CDMA is also the main technology behind FOMA, a system launched in Japan by DoCoMo in 2001 and regarded as the world's first 3G commercial network.

CDMA2000 is the second major standard and is used by countries outside the GSM zone, more specifically in the Americas, Japan and Korea. Another standard which is not as well known is TD-SCDMA, and this is currently being developed for use in China.

Worthy of special mention is the EDGE standard, which is largely an addition to existing 2G networks rather than a clear-cut 3G approach. EDGE formally complies with the IMT-2000 definitions for a 3G network since it can achieve speeds of up to 384Kbps and has therefore been accepted formally as a 3G standard by the ITU. Initially, while UMTS was considered to be the only way forward for 2G GSM operators, given the cost of its implementation, EDGE has been reconsidered by many as an alternative mid-term upgrade to their networks.

#### **1.1.2.4 Wi-Fi**

The term Wi-Fi (Wireless Fidelity) is used to describe a new technology for wireless Local Area Networks. In 1997, the IEEE (Institute of Electronic and Electrical Engineers) established a standard for wireless LANs called 802.11, which specified two data rates of 1Mbps and 2Mbps to be transmitted either over IR (infrared) or over radio in the unlicensed band of 2.4GHz (also known as the Industrial Scientific Medical frequency). The original standard was quickly superseded by 802.11b in 1999, which defined a maximum throughput of 11Mbps. Despite this claim, due to overhead caused by the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocols used, the true bandwidth available for any application is around 5.9Mbps over TCP (7.1Mbps over UDP). The 802.11b standard was widely

adopted as prices for equipment were not high and in 2003, amendment 802.11g was ratified, which offered a theoretic bandwidth of up to 54Mbps (practically restricted to around 24).

A Wi-Fi network typically operates by enabling devices to connect to the Inter/Intranet when they are in proximity of an access point, also referred to as *hot-spot*. Because of the 2.4GHz band, the coverage range of access points is limited as metal, water and thick walls easily absorb the signals. Further to this, devices that use the 2.4Ghz band are also susceptible to (and indeed have to accept) interference from other devices that operate in the same frequencies, such as microwave ovens, Bluetooth devices, cordless phones or video senders. Coverage is therefore much worse than that of cellular networks.

Typically, access points are encountered inside offices, airports, cafeterias, hotels or other public spaces. It is common for homes to be wireless-enabled these days, in order to share a broadband connection between multiple computers. The typical ranges for a Wi-Fi hotspot are approximately 46 meters indoors and 92 meters outdoors. However, with the use of high-gain directional antennas, links up to 8Kms in range can be achieved and this is a common and viable option for the replacement of leased lines.

The greatest concern with Wi-Fi remains security, which is commonly implemented using a Wired Equivalent Privacy (WEP) key. The levels of security afforded by this are not high, as it is an easy protocol to break. Another protocol, Wi-Fi Protected Access (WPA) is also gaining in popularity and a new extension ratified in June 2004, 802.11i, provides WPA2 security which is a significant improvement.

It has been argued that with Wi-Fi, an alternative to the forthcoming 3G cellular networks' data service can be implemented. A new standard, 802.16 or WiMax, whose latest version was ratified in late 2004, offers vastly enhanced ranges of up to 50Kms (estimated) without line-of-sight to base stations. Further to this, the bandwidths afforded have risen to about 70Mbps, which, indicatively, is enough to accommodate the needs of about 1000 homes with 1Mbps connections. The WiMax standard is regarded as an ideal solution for the establishment of Metropolitan Area Networks for the provision of Internet connectivity to citizens and businesses by municipalities. Thus far, plans for WiMax implementations have been set across

several cities, although none is known to have a fully operational network until today.

For mobile applications, Wi-Fi has another drawback, the fact that it drains batteries very quickly. This restrictive factor has prevented Wi-Fi so far from being implemented in mobile phones and remains a consideration which hopefully can be addressed in the future, when better energy sources such as fuel cells will be used.

#### 1.1.2.5 Mobile Access Costs

Access costs vary significantly across countries and networks, but the table below shows some typical costs for 2.5G, 3G and Wi-Fi access (in commercial environments) in the UK. (source: T-Mobile UK)

Connection Type	Rates	Data provided
GPRS	£40/month	Unlimited (subject to 100Mb/month fair usage scheme)
3G-UMTS	£70/month	Unlimited (subject to 1Gb/month fair usage scheme)
Wi-Fi	£5/hour £7.50/3hours £13/24hours £30/7days £45/month £20/month (businesses only)	Unlimited

**Table 4: Wireless Internet Access costs in the UK (feb 2005)**

The charging structure is greatly varied through the world. In some countries, like the U.S.A., free GPRS access is to subscribers for normal HTTP traffic, although a charge is made for receiving email through POP3 servers. However, the cost guide above is typical of the charging strategies employed across Europe.

Through the comparison of the costs of mobile access, especially for broadband ADSL vs. UMTS, it becomes apparent that the disparity between the two is great. While Wi-Fi access costs are on par with the broadband costs of approximately three years ago, the dependence on hot-spots which are usually found in commercial environments, does not make Wi-Fi currently a truly mobile solution.

## **1.2 Mobile Device Support for Internet Access**

### **1.2.1 Device Characteristics**

For a mobile device to support Internet access, it needs to be able to connect using any of the wireless methods discussed above. Some devices, like laptops, are able to make use of wired connections, while some others can be paired to a desktop computer and share its wired connection. However, such connections are out of the scope of this discussion, as they do not offer Internet Access in a truly mobile scenario.

To state perhaps the obvious, apart from connectivity hardware, mobile devices also need a screen which is capable to display the downloaded information in a manner that is comprehensible by the user. Further to this, the user needs to be able to interact with the device, in order to request information for download and to be able to navigate through the downloaded information.

While the above hardware requirements place no significant issues for most desktop and laptop computers, devices such as mobile phones and PDAs have naturally very limited resources and special usability considerations need to be taken into account.

### **1.2.2 Device Limitations**

#### **1.2.2.1 Device Screens**

Given the physical dimensions of mobile phones and PDAs, their screens are naturally limited in size. Most PDAs have a screen size of approx. 3.5 inches (diagonal), which limits the resolution to approximately  $\frac{1}{4}$  VGA, i.e. 240x320 pixels. Both colour and monochrome screens exist, although the majority of PDAs are now sold with a colour screen. The colour depth for these is generally ranged from 256 to 65,536 colours, which is satisfactory for the display of images and text.

On mobile phones, the situation is often much worse as there is an apparent trend to make these devices as small as physically possible. The sizes of screens vary greatly but a typical example of a modern mid-range phone is approximately 32x39mm in size, offering 128x160 pixels at 65,536 colours (Nokia 7270). Other phones, such as the SonyEricsson P900 are closer to the PDA specifications with

208x320 pixels. Monochrome display phones are nowadays obsolete and very few, if any, are manufactured.

From these facts, it becomes quickly apparent that navigating the web using these devices poses several usability problems in terms of the inability to present websites effectively using such small screens. Given normal design of websites for resolutions of at least 800x600 pixels, the problems of scrolling in both vertical and horizontal directions becomes a grave consideration. To overcome such problems, mobile telephony providers often provide a “cut-down” version of the web to their clients, where sites that are specifically designed for mobile screens are provided. Some independent web content providers also provide low-res or text-only versions of their sites for mobiles. Lastly, special browsers for mobile phones, such as Opera, which attempt to re-structure the HTML in websites, either on the client or at a dedicated proxy, in order to make them more presentable on a mobile screen. For example, the frame structure in a page might be altered so frames appear stacked on top of each other. This does not eliminate the problem of scrolling but it reduces it to one dimension (vertical), which is offers better readability. Other browsers, such as ThunderHawk, attempt a graphical representation of a website, which the user can click in order to magnify and read.



Figure 3: The Thunderhawk browser, zoomed out website (left) is unreadable. Article of interest in yellow box is magnified and made readable (right).



Figure 4: The Opera mobile browser. The clear sections present a typical mobile screen, the dimmed sections are regions outside the screen. A standard web site (left) is rearranged to fit the width of a mobile screen, thus requiring only vertical scrolling to read.

### 1.2.2.2 Input Methods

The most common input method for PDA devices is a stylus, which is used for tapping a touch-sensitive screen. Both input in the form of “clicking” on webpage elements (input boxes, buttons, scroll bars) and textual input can be entered through the stylus. For textual input, handwriting recognition is often provided, along with alternative methods, such as a virtual clickable keyboard. While the text input speeds are not as speedy as on a normal keyboard, the functionality provided by the stylus is more than adequate for web browsing, an activity that is mainly associated with reading and clicking on webpage and browser controls.

In contrast, mobile phones lack the functionality of the stylus and restrict the user to the keypad. With this, it is difficult to access various elements of a webpage quickly. For example, in a form where multiple input boxes exist, a user would have to navigate to each box in a serial manner to get to the desired element, while with a stylus, the desired element is immediately accessible with one touch. Furthermore, text content has to be continuously scrolled up or down and there is usually no facility to allow the user to jump from one section of the page to another. This is

achievable with the stylus very easily by holding and dragging the position indicator on the scroll bar.

Some phones like the SonyEricsson P-series and the Nokia Communicator series, incorporate a PDA-like interface with touch sensitive screens and styli, while maintaining a keypad. This offers the users the ability to choose between the desired methods of input and the combination works relatively well.

### **1.2.2.3 Internet Connection**

Almost all PDAs offer at least one method of connectivity, usually to other PDAs and devices through Infrared or Bluetooth and to desktop PCs via cable. Some modern devices also offer integrated Wi-Fi capabilities. A few PDAs, such as the O2 XDA, are also able to accept mobile telephony SIM cards, allowing these to be connected to GSM 2.5G networks currently, although there are currently no PDAs that support 3G connectivity on the market. T-Mobile, a worldwide mobile telephony provider, plan to offer the first such device (MDA IV) in the Summer of 2005.



**Figure 5: The T-Mobile MDA IV offering Wi-Fi, GPRS and UMTS connectivity. (source: [www.worldofppc.com](http://www.worldofppc.com))**



**Figure 6: Typical PDA devices- O2 XDA (left), PalmOne Treo (centre), Palm Tungsten (right)**

Mobile telephones offer either 2.5G or 3G connectivity to the Internet, as currently Wi-Fi places such a big drain on the battery that manufacturers have considered its implementation impractical.



**Figure 7: Internet-enabled mobile phones- SonyEricsson P910 (left), Nokia 6170 (centre), Motorola MPX-200 (right)**

### ***1.3 Internet content Pre-Caching as an alternative approach***

#### **1.3.1 Summary of the current state in Mobile Internet Access**

With the constant progress in microelectronics technology, it has become possible to make small personal computing devices available to almost everyone today. These devices are presently distinguished between those that are used primarily for communication, such as mobile phones, and those that are used for personal information management (PIM) and are known as Personal Digital Assistants (PDAs). However, as the technology constantly allows the manufacture of more potent devices, the distinctive line between these two categories slowly starts to disappear.

A characteristic, which signifies this blending of single-purpose devices into multi-function “digital assistants”, is the incorporation of PIM functions into many of today’s mobile phones. Most importantly, the typical calendar application, which is central to PDAs, is now available in almost most modern mobile phones.

Another application which is currently available on PDAs and is slowly appearing on mobile phones is the Web Browser. Until recently, surfing the web on a mobile phone has been restricted to the use of WAP purpose-built sites, mainly due to the slow speeds available through the GSM network (~9.2Kbps) and the limited display capabilities of mobile phones. However, with the advent of wider bandwidth wireless communication standards, such as GPRS, at least part of the problem is being addressed. While mobile phone screens may still pose significant limitations to effective web surfing, at least PDA users are now able to experience better performances on their devices.

The distinction line between PDA and mobile phone is expected to disappear with the coming of 3G networks and 3G mobile devices, which are going to make full use of the available bandwidth to provide several communication services currently limited to desktop computers. Of these services, perhaps the most important one is broadband connection to the Internet and its vast amount of resources.

There is however some concern regarding the costs of this new service. The two-and-a-half generation GPRS transmission standard has been around for a while, but it is still very expensive to use. The same is expected for 3G networks, once they

become widely available, with several analysts estimating that users will be reluctant to accept the high costs of 3G network usage<sup>vi</sup>.

### **1.3.2 Internet Content Pre-Caching: An alternative approach**

Since it is widely recognised that the availability of internet content on mobile devices is a very desirable service, perhaps it is possible to devise another way of making this content available. It might be possible for a user to pre-fetch all of the data that would be required through their low-cost, land-based (and often broadband) internet connection, feed it into their mobile device and be ready for the next day. The practical problem with this scenario would be that users would have to spend a considerable amount of time browsing the web, saving websites on to their hard disks and then transferring them over to the mobile device. Indeed, such a process would require a considerable amount of time and effort on a repetitive process, which a user would promptly be bored of and would not continue to perform on a daily basis.

Given the existence of calendar applications in modern mobile devices, it can be assumed that a forecast into the future activities of the user can be obtained through the entries that have been made therein. Therefore, an intelligent software agent could examine the user's calendar and try to estimate the kind of activities they will perform and, possibly, what kind of internet content they may need, in order to support those. The web content downloaded through this predictive system, utilising the user's land-based internet connection, would be stored in the user's desktop, processed and then transferred over to the mobile device.

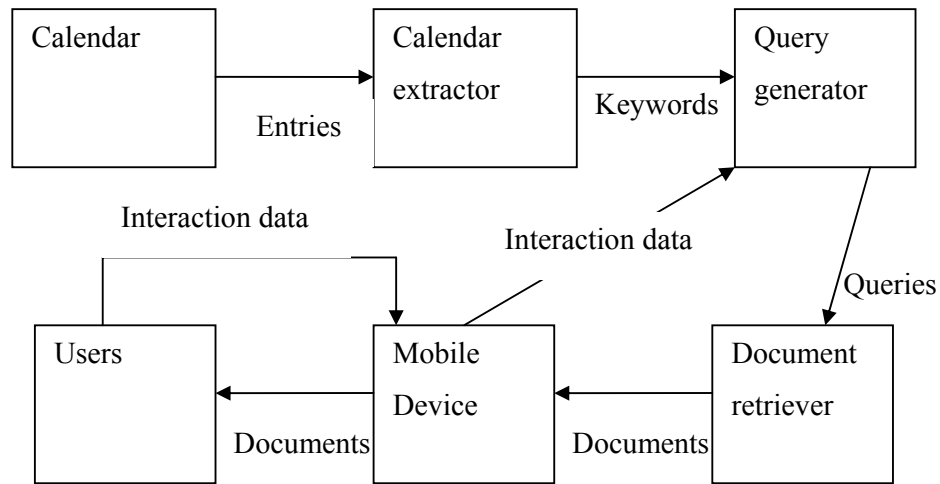
Let us consider an example: If a calendar entry mentions city (e.g. Edinburgh), it can be assumed that the user will be present at that location for some purpose which is possibly described elsewhere in the entry. Typical information about cities could then be downloaded through web searches such as "Edinburgh map", "Edinburgh accommodation", "Edinburgh Museums" etc. Also, further and more specialised searches, formed through combination with other information

---

<sup>vi</sup> Users will not pay for high 3g costs, <http://www.vnunet.com/News/1112409>

retrieved from the calendar entry, can be conducted, e.g. “Edinburgh Holiday Inn” or “Edinburgh HCII2004 conference”.

The predictive system can then be enhanced over time through use of implicit or explicit feedback to learn and remember a users’ preference for different categories of information, e.g. more interested in transport links than hotels. Such a predictive system should be able to obtain information from the user directly and indirectly. The majority of information should be obtained through indirect means, in order to minimise interference with the user’s other activities. However, the system should maintain its ability to directly interact with the user, in order to resolve any possible uncertainties.



**Figure 8: System operation principles overview diagram**

### 1.3.3 Hypotheses

This thesis will attempt to prove the following hypotheses:

- Calendars can provide information that can be used to pre-fetch useful Internet content to users.

- Queries can be formed from calendar entries, without direct user involvement and independently of the entry brevity, which will be meaningful and retrieve documents useful to the users.
- Such a pre-fetching system can learn from the users and adapt its performance in order to offer content that is increasingly relevant to the users

## **1.4 Structure of the thesis**

In order to describe the investigation the hypotheses stated above, the thesis is structured in the following manner:

- Chapter One was an introduction to the current state of mobile Internet Access and its problems. It provides some motivational background for the thesis. Also the main hypotheses of the investigation of this thesis are stated herein.
- Chapter Two takes a closer look at the research previously done on areas which are relevant to the scope of the thesis. More specifically, previous research on Calendars, the acquisition of User Context and Preferences and the Query Expansion techniques available, is examined.
- Chapter Three discusses the design for the implementation of a system developed to examine the hypotheses of the thesis.
- Chapter Four contains a description of the various experiments that were conducted, in order to obtain test results for the implementation of the system. Also, the performance tests on the final system are also discussed.
- Chapter Five presents an overview of the conclusions of this research and discusses further work that could be carried out, based on these conclusions.

## Chapter 2

### A review of Existing Research

## **2 A review of existing research**

Bearing in mind the conceptual ideas of the system, as presented in the previous chapter and the hypotheses of this research, it becomes apparent that an investigation of previous research needs to be carried out for three main fields. Firstly, an attempt should be made to examine any previous work on calendars and their content. This would allow an insight to what kind of information one can expect to find therein. Secondly, since the potential for the system to be adaptive to its user needs to be shown, a review of research on acquiring user needs and preferences is also required. Lastly, because the system will depend on its ability to form appropriate web queries from the information found in calendars, a look at the field of query generation and expansion will also provide useful insight.

This chapter presents a review of important work in the aforementioned fields. Although it would not have been possible to mention every single piece of research published, mainly due to the extent of previous work in some areas, the following review should give a sufficiently comprehensive background that describes the main ideas behind these fields of research.

### ***2.1 Electronic and paper-based Calendar usage***

Given the assumption that it might be possible to extract sufficient information from a calendar in order to create a user model and retrieve documents for a user based on this model, it would be of benefit to begin the examination of the hypothesis from the state of the current research in calendars and their use. Most of the research in this area seems to focus on the usability of the calendar applications and there seems to be very little work on the actual nature of calendar entries, which is the main interest area for this thesis' topic.

Early work by Kincaid and Dupont [Kinc85] investigated the use of electronic and paper calendars in an office environment. Though their paper focuses on the usability of existing applications and makes interesting recommendations on the essential features a calendar application should have, there is a section where the

researchers manage to identify the main uses of a calendar. It is their finding that calendars are mainly used

- As a record of meetings, appointments, events, or travel with detail remarks on each
- For reminders and notes
- For Bring Forward “ticklers”
- As “to-do” lists
- They also found that the following uses have been reported, although at a smaller frequency:
- As a planning tool
- As a record of services performed or as a diary
- As a record of telephone conversations or events of interest
- For travel and business expenses

It was also discovered that despite the office environment, only two respondents kept a separate calendar to record personal items, which means that one should expect to find adequate information in a calendar with regard to a user’s personal and business interests.

Blandford and Green [Blan01] conducted a survey of the group and time management tools used by employees at a Computing Science department of a university. Again their work focuses on the needs that have to be supported by software tools, however some other interesting by-products of their research are mentioned; some extracts are shown here, which contain pieces that are related to the nature of calendar entries. More specifically, the researchers found that calendar entries are typically used for:

- Prospective remembering of “appointments” and “to-do”. (The term “appointment” refers to a meeting or an event that has a designated start time and expected participants, and sometimes a location and various other properties)

- A note of which week of the semester it is, coursework deadlines, examination dates and similar information
- A note of when particular colleagues are away
- Pointers from the diary entry to supporting materials (e.g. papers for a meeting, an electronic copy of the agenda or contact details for the person being met)
- Reminders of deadlines and other actions to be performed on the day
- A note of events, such as people visiting the department that the individual might want to meet briefly but does not have an appointment with
- To-dos occasionally get noted in diaries
- Birthdays of close colleagues
- Record of past events
- Keeping track of the way past time has been spent (for their own use or for reporting to senior management)
- Calculating mileage claims (travel expenses)
- Supporting the planning of future events
- Recording the times and outcomes of meetings in case of queries
- Recording activities related to personal hobbies (although in this research there was only one example out of the ten who had a mixed business-personal calendar).

Most of these findings are in complete accordance with the aforementioned studies. Expanding further on the nature of the entries themselves, the researchers work highlights two important points: a) That the temporal data contained in an entry is not always reliable and b) The amount and quality of information content of the entries depends on the target readers of the calendar.

To support the first point, an example is shown that in electronic calendars, entries have a default time slot allocation of one hour, which most users tend to

accept regardless of the true duration of the commitment described by the entry. This could be intentional in order to give the individual some “slack” time after a short meeting, but could also have negative effects when a meeting is known to last longer but the default allocation is kept through lack of attention. A second supporting statement is made, which emphasises the fact that while most appointments have a known starting time, their precise end time might be unknown. Therefore the temporal placement of an entry immediately after another might equally indicate a fixed start time or that the second entry’s activity must be performed once the first one’s finished. These points are also mentioned in work by Cooper [Coop99], who argues that there are two types of time-based information: deadlines and ongoing processes.

With regard to the amount and quality of information content in an entry, the researchers highlight the fact that when users know their entries are going to be read by others, they will tend to adapt their writing to the target audience. Therefore appropriate codes might be used, cryptic notes might be avoided in entries or even explicit instructions to the reader might be included. The knowledge that the diary is shared might also prevent people from writing down all, or partial details of their activities (such as going to the doctor’s) as they might deem these to be personal and of no interest to others. On the other hand, Payne [Payn93] mentions the fact that people are helped to remember by the act of transcribing and therefore might not need to write down large volumes of information as they would be able to remember details on their own. The same is found again by Blandford and Green [Blan01], who report that many users make little use of external *aide memoires* to assist in remembering activities. This can lead to the assumption that when diaries are not shared, the quality and quantity of information therein might be very poor. The use of single or very few keywords as a reminder for further details is actually a common phenomenon in calendars, as highlighted by my own research and described later in chapter 4.

## ***2.2 Pre-fetching on large scale for internet content providers***

### **2.2.1 What is caching?**

There are several examples, both in the computing and in our everyday world, where “caching” is used as a method to improve speed and efficiency in systems. A cache is a temporary storage facility, which is intended to keep objects that are likely to be utilised in the near future, close to the resource that is going to utilise them.

Considering an example of the real world, a personal telephone and address book might be considered as a caching system, with distinct advantages over a larger resource, such as the White Pages book. The address book holds information that a person is likely to use in the near future, close to this person. While it contains only a fraction of the information that is found in the White Pages, its use has distinct benefits. Such benefits include the provision of fast access to often-used numbers, better usability and better portability. Also, it is able to hold information that comes from multiple White Page tomes. It is obvious that carrying a personal address book is more practical than having to look up names and numbers in the large numbers of White Pages tomes that exist.

Caching is often used in memory management architectures, in computer systems, as outlined by the exemplary work of Mano [Mano82] and Hill [Hill87]. Central Processing Units (CPU) depend on information provided by memory modules, in order to perform their operations. However, CPUs typically are able to operate at much higher speeds than the memory modules they depend upon. This fact is responsible for possible bottlenecks in the flow of data, which mean that the performance capabilities of the CPU are in fact limited by the performance of the memory modules.

To overcome this problem, designers have placed small memory modules close to the CPU, which are able to operate at, or very near to, the CPU’s speed. These cache modules hold information from the main memory, which the CPU is likely to need, so it may be readily available when requested. If the requested data is not found in the cache, it is fetched directly from the main memory, which of course incurs all the typical associated costs. There are several algorithms for selecting suitable candidate data from the main memory, as well as for the elimination of data

from the cache, in order to make space for more important items. I will not expand however into these, as it is beyond the scope of this work.

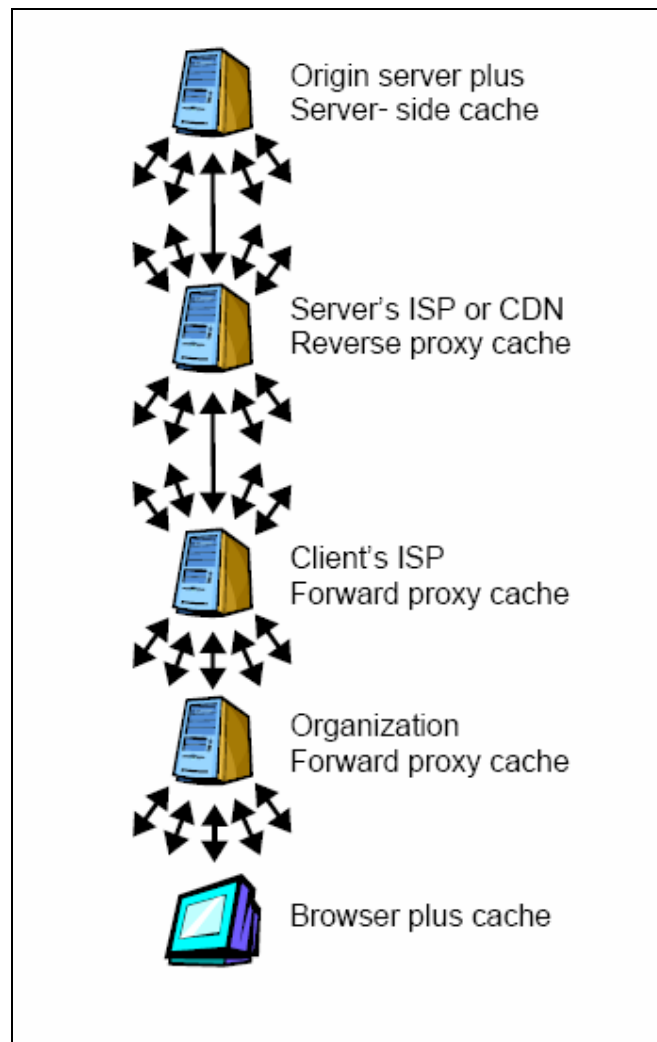
### **2.2.2 Caching strategies**

Caching on the World Wide Web (WWW) is a successful technique which has been employed, in order to help with some of the problems caused by the rapid expansion of the WWW. Internet caching works on the same principles as the memory caching described earlier, but has to accommodate for some differences, like variable resource types and sizes, and the potential or undesirable cacheability of a resource. The objective of the WWW caching methods is to improve response times for the users (clients) and performance gains and cost savings for content providers (servers), while remaining transparent.

When a client issues a request for a web resource, this request might be responded by caches that reside on many nodes, which are intermediary to the client and the server whence the resource originates. They can even reside on the client and server themselves.

Perhaps the most important question that has to be answered by anyone employing a caching scheme is what to cache. This is central to the operation of the cache facility, since a cache which does not contain items that are likely to be requested, is going to add to the problem of improving the Web experience, rather than detract from it.

A typical approach to answering this question is to store resources which have been asked for in the past. An analysis of web logs (logs of client requests) can provide information regarding the popularity of web resources and can help make informed decisions on which of them should be stored. In a study by Tauscher and Greenberg [Taus97], it was shown that 60% of the content requested by a user, was requested again by the same user within the time period of 1 month. A further study by Duska [Dusk97] showed that up to 85% of all cache hits originated from multiple users requesting the same resources. These studies prove that there is indeed value in storing resources that have been requested in the past. However, such a caching strategy can introduce problems alongside the benefits it provides.



**Figure 9: Caches in the WWW. Starting from the web browser, a request might be fulfilled by any cache before getting to the origin server.**

The most important issue that needs to be considered in caching, is the validity of the resource which has been cached. It is a well known fact that some content is un-cacheable, due to its rapidly changing nature. Information such as stock quotes and real-time images are good examples of such resources. Given also the fact that websites are updated frequently, a cache must ensure that it provides a relatively recent image of the information that is requested.

The amount of time that a resource can be cached is manageable by the content provider, who can explicitly manipulate the cache storage time of various resources, since it is included in the HTTP headers a server sends. This can be

achieved with the use of specialised web server software. It is typical therefore that static content, such as images and graphics, is given a distant expiration date, while other content, which is more susceptible to change, is given shorter ones. With careful design, a server can minimise direct requests from their own system, resulting in reduced operating costs, while allowing recent and valid information to be provided to clients through web caches.

### 2.2.3 What is pre-fetching

Having described the main caching strategies and issues that are related with them, it is time to expand on the potential improvement of these strategies. The main weakness that they entail is the fact that they rely on past user requests and web content popularity to operate. It would be a significant advantage therefore, if a cache could actively and successfully predict the kind of content a client might request, and have it available before a client requests it.

Pre-fetching is a technique that is commonly employed again in the memory architectures of modern computers, in order to preload the memory with items, which will be possibly requested in the future. The interested reader could examine examples of such work in articles by Joseph [Jose99], Kraiss [Krai98] and Smith [Smi82]. However, the proposal of the essence of this technique for usage in WWW caching solutions is also not new. It can be dated as far back as 1994, with many papers having been written on this subject, following pioneering work by Padmanabhan [Pad96] and Kroeger [Kroe97]. In his work, Davison [Davi02] defines pre-fetching as follows:

*“Pre-fetching is the cache initiated speculative retrieval of a resource into a cache in anticipation that it can be served from the cache in the future.”*

While pre-fetching is a very promising idea, there are several drawbacks associated with it, which have so far prohibited the wide use of such a technique in caching services. Perhaps the most important issue with pre-fetching is making a correct decision on what should be pre-cached. There are solutions, mostly in

browser add-on form, which will pre-fetch the links on the page a user is currently navigating, or even, periodically, the bookmarks a user has kept. Jiang and Kleinrock [Jia97] showed that such a predictive scheme can be successful and has potential for improvement. However, pre-fetching on some sites might produce undesirable results, such as items being added to a shopping cart or automatic log-out due to multiple simultaneous requests on some secure sites. It is interesting to also note that with particular reference to web resources, in his work, Davison defines the pre-fetchability of a resource:

*“A web resource is pre-fetchable if and only if it is cacheable and its retrieval is safe”*

Further from the unknown cacheability or un-prefetchability of a resource, pre-fetching is also able to increase loads on content servers to an unnecessary extent, due to the number of resources that are provisionally downloaded. There is little guarantee that these resources will actually be used and it is clear that decision-making strategies are an important factor for the conservation of bandwidth and memory resources both on the client and on the server sides. Indiscriminate pre-fetching is also able to alter usage statistics for various servers, making the investigation of their usage more complicated. In their work, Crovella and Barford [Crov98] argue and prove that “...source burstiness caused by pre-fetching on the WWW result in increases in the variability of aggregate traffic on a wide range of scales”. However, they manage to prove that controlled-rate pre-fetching offers opportunities for traffic shaping in a network; therefore an appropriately designed pre-fetching scheme is required in order to obtain the advantages of pre-fetching without imposing unnecessary load upon the network.

In the following text of this thesis, the terms pre-caching and pre-fetching are used almost interchangeably to specify the action of pre-loading a computer’s storage media with Internet content.

## 2.2.4 Research in Large scale (server-side) Pre-fetching

There have been considerable research efforts into the matter of establishing algorithms for the large-scale pre-fetching of web content. While initially one may consider this work unrelated to this thesis' aims, from the careful reading of the research performed in this area, one can observe that many of the techniques described therein can also be used for pre-fetching on a personal level. Combined with other research specifically targeted to this domain (section 2.2.5), the potential for the combination of this thesis' work and other pre-caching methods can be revealed.

One of the earlier and simple ideas proposed by Markatos and Chironaki [Mark98] in 1998 was a top-ten approach. Their proposal was to pre-cache the top ten documents for each server, in terms of popularity. This way, clients or proxies could attain these documents without any significant cost in network traffic. Their experiments show that this approach expects more than 40% of the client requests, while achieving a hit ratio of approximately 60%. This comes at a cost of around 10% increase in network traffic. The implementation of this algorithm can be expanded to preserve the top  $n$  documents of each server, with appropriate increases in the hit ratio, although at a higher network usage cost.

The previous approach is regarded as a naïve approach to pre-fetching but it is useful in the demonstration of the fact that even basic pre-fetching can offer reasonable advantages. Another naïve technique which can be utilised, as stated in Jiang et al.'s [Jian02] work, is the pre-fetching of objects according to their life span. Obviously, the cost in bandwidth for pre-fetching items is reversely analogous to the lifespan that they have, hence pre-fetching items with large life spans can, expectedly give us low bandwidth costs.

Another approach, described by Venkataramani [Venk01] is entitled the Good-Fetch algorithm. It attempts to address the problem of accessing a cached web resource that has become stale (i.e. the cache does not contain the latest modification of the resource). An object that ends up being referenced before becoming stale, is considered a Good-Fetch. The algorithm calculates the probability that an object will be referenced, before it becomes stale, and pre-fetches it accordingly, if that probability is above a given value. For an object  $i$  that has a life-time of  $l_i$ , a

probability that it will be accessed of  $p_i$ , and where the arrival rate of user requests per second is  $a$ , then the probability of the object being accessed before it expires is

$$P_{goodFetch} = 1 - (1 - p_i)^{a \times l_i}$$

The algorithm, as proposed here, is highly tunable since it allows objects to be pre-fetched at various threshold values. In turn, this allows the optimisation of a server's performance and can help specify its physical characteristics (such as memory and bandwidth), through the analysis of the cost of pre-fetch requests.

Jiang, Wu and Shu [Jian02] propose a different algorithm, called APL, while comparing some of the previously described approaches in their work on web pre-fetching. Considering an object on the web ( $i$ ), again they define its life-time as  $l_i$ , the probability that it will be accessed as  $p_i$ , and the arrival rate of user requests per second as  $a$ . In this case,  $a \times p_i$  is the request rate for the object and  $a \times p_i \times l_i$  should represent the amount of requests for the object, before it expires. Therefore, the algorithm suggests the inclusion of the objects with the highest APL value in the pre-fetching set. On a more interesting variation of this algorithm, they propose the formula of  $a \times (p_i)^n \times l_i$ , in order to emphasise object popularity. This way, when  $n > 1$ , the algorithm works closer to working similarly to pre-fetching by popularity, thus employing higher bandwidth to improve response time. In contrast, when  $n < 1$ , the algorithm works closer to pre-fetching by lifetime, placing less strain upon the network bandwidth. This approach comes with the potential advantage of allowing a dynamic shift from one strategy to another, depending on the given network conditions. Their results show an effectiveness which is very close to the one obtained by Venkataramani et al., although it maintains the flexibility of being able to adapt to network conditions.

Markov Chain models have also been extensively examined in various studies for the improvement of pre-fetching strategies [Cade00], [Saru00], [Palp98],[Fan99], either explicitly or as part of the Prediction by Partial Match (PPM) predictor. Markov models, as described by Papoulis [Papo91], are well suited for the prediction of actions that a user might perform next, and as such, were good candidates for application into the field of pre-fetching. These models are represented by three parameters, A,S, and T. Parameter A represents a set of all the

possible actions that can be performed by the user.  $S$  is a set of all the states for which the Markov model shall be built. Finally,  $T$  represents a  $|S| \times |A|$  *transition probability matrix*, where each entry  $t_{ij}$  corresponds of the probability of action  $j$  being performed while the process is in state  $i$ . Markov models generally fall under two categories, 1<sup>st</sup> order and 2<sup>nd</sup> order (a generalisation of which is  $K^{\text{th}}$  order models). First order models refer to those that predict the next action of the user by looking only at the last action, hence in these models, each action represents a single state of the system. Second, or  $K^{\text{th}}$  order models, attempt to predict the next  $K$  actions of the user, looking at the previous  $K$  actions that have been performed.

From this description, it is easy to understand that the higher order models, while able to offer significantly better prediction results than 1<sup>st</sup> order models, operate with a set of disadvantages. Complexity becomes an issue as the order rises, since the number of possible states also rises exponentially. This has a significant effect on the physical requirements (memory, processing power) for the real time processing of such a model. Furthermore, when considering the matter of web pages, it is not obligatory that all examples have as many states as assumed by the order of the Markov model. This reduces the coverage of the model and its accuracy, as it is forced to make generalised and lower accuracy predictions for such cases. In their work, Pitkow and Piroli [Pitk99] recommend an approach called the All-  $K^{\text{th}}$ -order, to overcome the shortcoming of  $K^{\text{th}}$  order model coverage. In this approach, all models from 1<sup>st</sup> to  $K^{\text{th}}$  order are built and the highest model that covers an instance is used for prediction. For example, if a 3<sup>rd</sup> order model does not contain the given state, the 2<sup>nd</sup> order model is used and so forth. However, it is obvious that again such a system places even larger demands on the already significant problem of complexity, since several models have to be maintained simultaneously.

An interesting variation of the All-  $K^{\text{th}}$ -order algorithm (otherwise known as the PPM algorithm) is discussed by Nanopoulos et al.[Nano03], who propose the  $WM_o$  algorithm. In essence, this addresses the fact that PPM might overlook a candidate sequence for pre-fetching, since it produces candidate sets that are completely unordered. A noteworthy element of this algorithm is the fact that it is a generalisation over existing algorithms, since the researchers adequately prove that with the application of appropriate constraints, it becomes equivalent to existing

algorithms. The researchers proceed to test the  $WM_o$  algorithm with synthetically generated data sets and discover significant gains over Kth order PPM and Decision Graphs (described below), which are verified by further tests on real data sets.

As a potential solution to the All-  $K^{\text{th}}$ -order shortcomings, Dashpande and Kapyris [Dash01] describe three alternative variations on this theme, which they title *Selective Markov Models (SMM)*. These models attempt to eliminate certain states across the different order Markov models in an All-  $K^{\text{th}}$ -order scheme, and utilise the remaining states for the final prediction model. This is done in an effort to reduce the state complexity and also improve the prediction accuracy that can be achieved.

The first SMM is called the Support Pruned Markov Model. It eliminates the states that have very low levels of support in the training set, based on the observation that these tend to attain low prediction accuracy. The second, more complex model is the Confidence Pruned Markov Model, which uses statistical techniques to determine whether the probability of the most frequent action is significantly different from the probabilities of other actions that can be performed from a given state. The action is pruned if its difference in probability is not great, since this way it is bound to offer low prediction accuracies. Finally, Dashpande and Kapyris propose the Error Pruned Markov Model, which is based on the idea of performing a validation step on the trained models, using a part of the training set that was not used during the model building. Given the fact that the sequence of actions is known in advance (because of the training set), the error rates can be identified and used for pruning. They proceed to propose two different error discovery strategies, Overall Error Pruning and Individual Error Pruning, which are similar in the procedure of discovering subset states and pruning the ones that have high error rates. However, their difference lies in the way error rates are calculated.

Having compared 1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup>, All-Kth and SMM models, the researchers conclude that the Error Pruned SMM (in particular the Overall Error Pruning model) offers relatively low state-space complexity, while providing results that are significantly more accurate than those of the other algorithms.

In 1996, Padmanabhan [Padm96] et al. proposed an algorithm based on the work by Griffioen and Appleton [Grif94]. Their algorithm was based on the generation of a dependency graph, which depicts the pattern of access for all files

that are stored at a server. There is a node in the graph for each file that has even been accessed, and an arc from node A to node B, if and only if node B was at some point accessed after  $w$  accesses after node A. The symbol  $w$  represents the lookahead window size. The arcs in this algorithm are weighted by the measure of the ratio of the number of accesses to B after A (within the window  $w$ ), to the number of accesses to A itself.

$$W_{arc} = \frac{A_{B(w)}}{A_A}$$

However, even though it is a good indicator, this weight does not reflect the probability that B will be requested after A. The node will become a candidate for pre-fetching if the arc weight is greater than a pre-determined weight  $p$ , which is the decision threshold. Padmanabhan et al. proceeded to show that this method works well for unshared low-bandwidth client connections to a proxy and for high-bandwidth, high-latency links, such as satellite.

There have been other approaches to the problem of pre-caching, which focus on the exploitation of web logs and data mining techniques, in order to generate suitable algorithms. The main objective of these approaches is to determine access patterns for web servers through the statistical analysis of their web logs. A recent and interesting approach by Yiang et al. [Yian01], shows an extension of Cherkasova's GDSF (Greedy Dual-Size Frequency) algorithm [Cher98], which is utilized successfully, as the core of a replacement strategy for documents in caching servers. The GDSF algorithm ranks an object  $p$  by assigning a key value ( $K_{(p)}$ ), which is computed through this formula:

$$K_{(p)} = L + F_{(p)} \times \frac{C_{(p)}}{S_{(p)}}$$

Here,  $L$  is an inflation factor to avoid cache pollution,  $F_{(p)}$  is the frequency that  $p$  occurs in the past,  $C_{(p)}$  is the cost of fetching  $p$  and  $S_{(p)}$  is the size of  $p$ . The researchers calculate the probability  $P_{i,j}$  that an object residing on a server  $O_i$  will be requested in session  $S_j$  through the use of association rules, which are automatically created through the analysis of web logs. They progress then further to establish the

future frequency  $W_i$  of requests for the object  $O_i$  (assuming that all sessions on a server are independent) as:

$$W_i = \sum_j P_{i,j}$$

Therefore, having established the probability of future demand on a page  $p$ , the GDSF algorithm can be extended to incorporate it, in a manner that emphasises that the key value of a page is not depended only on its past, but also its future occurrence.

$$K_{(p)} = L + (F_{(p)} + W_{(p)}) \times \frac{C_{(p)}}{S_{(p)}}$$

In their conclusions, while underlining the encouraging results of their method, the researchers are keen to point out that a balance has to be struck between the performance, as measured, in object hits and network load.

Further work by A. Pandey et al. [Pand02], seems to verify the success of approaches that employ association rules, as the one described previously. A comparison was performed of three approaches to the problem of pre-fetching, namely, using a Top-N, a Pairwise Interaction (2<sup>nd</sup> order Markov Chain) and an Association Rule model. Six data sets were used to compare the performance of the three approaches. Of these sets, five were synthetic server logs and one was real. It was found that through the data sets, the association rule model was the one that clearly and overwhelmingly provided the best performance when considering the first three logs. For the remaining logs, it offered competitive performance which was directly comparable to that of the other models, indicating that overall the association rule model provided the best overall performance.

### 2.2.5 Implemented systems

A number of commercial system use pre-fetching techniques as part of their caching strategy. For example, CacheFlow [Cach02] uses a mechanism for the retrieval of in-line content, such as images, while proactively searching for expired objects. Another system, Avantis ContentCache [Cont05], is a learning material server that

can be configured to pre-cache web content ahead of lessons, or when internet usage is low. Broadband enabled schools can also make use of another system, RM SmartCache2 [Smar05], which allows the scheduling and pre-caching of large files during idle connection times.

## **2.3 Pre-fetching on a more personal level (client and proxy side)**

### **2.3.1 The application of pre-fetching techniques on or near the client**

Pre-fetching (or pre-caching) is an active field of research which has a focus on client-side storage of the cached content. In light of the drawbacks associated with this method, which were described in the previous section, pre-caching at a client or proxy level has been considered as an alternative implementation which allows the circumvention of many obstacles encountered in large scale pre-caching systems. The hypothesis which forms the main argument in favour of this alternative is that there exists a greater probability of making accurate predictions on the need for internet content of a *single user* or a *group of users from a similar background*, rather than of a large group of potentially un-related individuals.

Currently, there are various commercial products that employ a pre-caching mechanism, such as Webcelerator, NetSonic and PeakJet2000. An alternative approach for mobile devices, AvantGo<sup>vii</sup>, utilises “channels”, such as entertainment or news, to which users subscribe to. The system then pre-caches web documents that might be of interest to the user, although there is little in the way of predictive pre-fetching, therefore it cannot be compared directly with other pre-caching methods discussed in this review. There are also many experimental implementations of such systems and also of browser extensions, such as the ones proposed in Klemm’s work [Klemm94]. In the following section, some of these approaches for personal pre-fetching will be described.

---

<sup>vii</sup> AvantGo, <http://www.avantgo.com>

### 2.3.2 Research in personalised pre-fetching

One of the earliest attempts at the automatic prediction and retrieval of internet content was described by Balabanovic et al. in 1995 [Bala95]. The system described there forms a model of each user's preferences and continuously adapts itself to reflect the user's opinions of the content that is prefetched. The user is presented with a collection of hyperlinks to documents that the system has identified as potentially interesting. There is an option for the user to explicitly rate each link (from +5 to -5), therefore providing the system with simple relevance feedback. Even though the scheme employed by the researchers is a relatively straightforward approach, they succeeded in proving that there are significant gains that can be made through the personal profiling of users.

Wang and Crowcroft [Wang96] discuss the some tradeoffs between pre-fetching and the improvement of latency in the WWW. Also, they present an implementation of a deterministic pre-fetching approach, called Coolist. Their system is layered between the client and the proxy server and organises websites in folders. These folders can then be assigned three methods of pre-fetching. Batch pre-fetching is the first method, where a site is scheduled for downloading at a given date or time. Another method of pre-fetching is described by the term "start-up" and means that a site will be pre-fetched when Coolist is invoked. Finally, their third proposed method is pipeline pre-fetching, where sites are grouped for pre-fetching. When the first page in a group is requested, the next one will be automatically pre-fetched, regardless of the fact that a user may have not requested it.

Another discussion of the advantages of pre-fetching was carried out by Cunha and Jaccud [Cunh97], who proposed two algorithms for the prediction of the user's next action while browsing the web. Their first algorithm, using Random Walk approximation, projects the long-term interaction trend, while a second algorithm focuses on the short term trends. Using a model described by Thiebaut in 1989 [Thie89], which relates the accumulated number of cache misses to a program's random walk range, the researchers show that it can be successfully applied to characterise users' strategies, under the hypothesis that these relate to an infinite browser's cache. This model is mathematically described as follows:

$$N(r) = Ar^{1/\theta}, r \gg 1, \theta \geq 1$$

In this equation,  $r$  is the number of references,  $N(r)$  is the accumulated number of misses,  $\theta$  sets the curve growth pace, and  $A$  is a constant. A second method is described within the same report, which uses an algorithm of two phases: Firstly, a *preparation* phase computes the first order difference of the envelope of the user's profile curve, displaced by a factor of 0.5 (for ease of detecting behaviour changes). Secondly, the *prediction* phase determines how conservative the user was in the last  $t$  accesses. Also, a determination of how much history is made, based on that count, in order to compute the desired set of coefficients that minimise the short-time prediction error, around a vicinity of size  $n$ , for a sample at virtual time  $r$ . A routine, based on Durbin's method to calculate the linear prediction coefficients is then called, and lastly, the predicted value is computed as a linear combination of the past  $NCOEF$  terms. The authors show that both user models manage to achieve a degree of accuracy around 85%, which can be applied in conjunction with pre-fetching techniques.

In his technical report, Palpanas [Palp98], investigated the feasibility of using a model based on the partial-match prediction algorithm, for pre-fetching documents from the web. In his model, a pre-caching agent acts as an intermediary between the client and the server(s) that a user is connected to in a session. Having taken into consideration the special characteristics of the Web and after tailoring the algorithm to accommodate those, the author concludes that his proposed scheme's implementation is feasible and that it would be assistive to users who "consistently follow regular access patterns, when searching for information". This conclusion is reached through simulations, run on the access log files of the web server of the department of computer science, at the University of Toronto.

Jiang and Kleinrock [Jian98a] presented in 1998 a system in which pre-fetching is decided by the client, based on usage statistics about embedded HREF tag attributes. In their work, the client monitors its available bandwidth continuously and pre-fetches web content, choosing however not to pre-fetch images, in order to save bandwidth. An algorithm to decide which pages should be pre-fetched is used, based on the client's access history combined with the server's access histories for each file they hold. Further filtering on the decision process is placed by placing an upper

bound on the pre-fetch threshold, which is a function of the system load, capacity and cost of a time unit and a system resource unit. This two-tier decision process allows the system to maximise the performance gaining that can be achieved through pre-fetching.

Further application of Jiang and Kleinrock's work is found in another paper that investigates pre-fetching for mobile users [Jian98b]. Interestingly, in this paper, the authors extend their prediction algorithm to achieve a higher number of hits, by assigning users to a category (such as those interested in database research), amongst other things. The second component of their scheme is a server threshold model, which judges whether a page should be pre- fetched based on:

- The amount of time that may be saved by pre-fetching a file that may be needed
- The amount of bandwidth that will be wasted if the file is not used;
- The impact of the pre-fetch request on other users whose normal requests may be delayed by the pre-fetch request.

The latter is a necessary consideration in order to ensure that the overall system performance can be improved by pre-fetching the file. The authors proceed to conclude that their approach is well suited to mobile users, who may need to switch between different network connection methods (modem, broadband, satellite, wireless). This is because the separate server threshold module allows the adaptability of the prediction algorithm, ensuring the best possible performance under each network condition.

A simpler implementation than the one by Ziang and Kleinrock was proposed with the WCOL system, by Chinen and Yamaguchi in 1997 [Chin97]. Their system is a research prototype that pre-fetches embedded hyperlinks top-to-bottom without regard to likelihood of use. Embedded images of pre-fetched pages are also pre-fetched. Bandwidth waste can be capped by configuring WCOL to pre-fetch no more than a certain number of hyperlinks, and no more than a certain number of images embedded within pre-fetched hyperlinks.

In 1999, Dan Duchamp [Duch99] presented his own work on pre-fetching hyperlinks, based on a predictive algorithm at a client level, which is however able to

communicate usage statistics from the server. Because the client is unable to form an objective view of the usage for a given web page, unless that page is visited often by the user, it passes on to the server the current usage statistics it has obtained, but also demands the aggregated statistics for that page, as held by the server. The performance results obtained by an implementation of the aforementioned idea were strongly encouraging. For example, of all the pre-fetched pages, a figure of 62.5% was eventually used. An improvement in latency of the order of 52.3% was observed, while notes for the consideration of network overhead due to the usage reports are being addressed. Further concerns regarding the size of the modifications necessary to the browser (Mozilla) and the execution time overhead due to these, are eased, since these do not appear to be significant.

Continuing on the theme of document pre-fetching that is done outside a server and on a more personalised level, more related work was carried out by Fan et al. [Fan99] in 1999. They propose pre-fetching at a proxy level, under the argument that because proxies can collect access histories for limited numbers of users, this would present significant advantages over server-level pre-caching, since servers would be able to only collect access histories for the entire WWW population. An investigation of prediction by partial match algorithms follows in their work, followed by a simulation which shows an improvement in latency between low-bandwidth clients and proxies, of the order of ~23%

Pitkow and Pirolli [Pitk99] explore predictive modelling techniques that attempt to reduce model complexity while retaining predictive accuracy. The techniques merge two methods: a web-mining method that extracts significant surfing patterns by the identification of longest repeating subsequences (LRS) and a pattern-matching method that embodies the principle of weighted specificity. Their work is largely motivated by previous studies by Schechter, Krishnan, and Smith [Sche98], who utilized path and point profiles generated from the analysis of Web server logs to predict HTTP requests. Also, much reference is made to the work by Padmanabhan and Mogul, who describe the efficiency of Markov Models for pre-fetching. The authors use the definition of Longest Repeating Subsequence by Crow and Smith [Crow92], which contains the following terms:

- Subsequence means a set of consecutive items

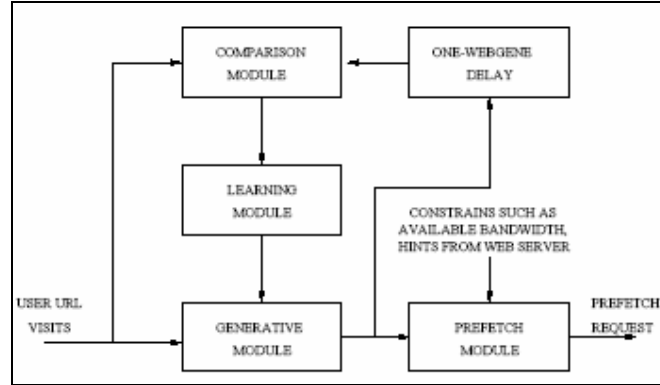
- Repeated means the item occurs more than some threshold  $T$ , where  $T$  typically equals 1
- Longest means that although a subsequence may be part of another repeated subsequence, there is at least one occurrence of this subsequence where this is the longest repeating.

Two models are proposed, firstly a hybrid LRS model which extracts LRS patterns from a training set and uses them to estimate a 1st order Markov Model. This model is compared against a 1st order Markov model estimated from all the paths in the training data set. The second hybrid model proposed is one that decomposes the extracted LRS subsequences into all possible  $n$ -grams of various lengths. This is called the All Kth Order LRS model, as all orders of  $k$  are able to make predictions. This model is compared against an All-Kth-Order Markov model derived from all the possible subsequences, decomposed into varying length  $n$ -grams. Further simplification is added to their web surfing model, by assuming that surfing paths have an average branching factor  $b$ . Surfers may start in  $b$  places and from each page, they move on to one of  $b$  pages on average. By assuming that surfing paths of length  $S$  can be divided into  $S/k$  subpath partitions ( $0 < k \leq S$ ), the complexity cost  $C(k)$ , in terms of the number of patterns as a function of  $k$  can be described as

$$C(S) = \sum_{i=1}^S (S/i)b^i$$

Through analysis of the applied methodology, the authors showed that in the case of modeling paths under the first hybrid model, the reduced LRS model was able to match the performance accuracy of the 1st order Markov model while reducing the complexity by nearly a third. They also then showed that overall hit rates could be raised by including the principle of specificity, with the All-Kth-Order LRS model almost equaling the performance of the All-Kth-order Markov model while reducing the complexity by over an order of magnitude. Finally, within their findings, it was further shown that increasing the prediction set has a dramatic impact on predictive power, with the predictive power of each method nearly doubling by increasing the set size to four elements.

Swaminathan et al. presented in 2000 [Swam00] a study of web pre-fetching which is based on the characterisation of the web client alone, without depending on server or proxy side algorithms.



**Figure 10: Operational principles diagram (Swaminathan et al.)**

The above figure shows the principle of the system as described by the authors. The input stream consists of symbols representing the URLs. The learning module learns the trend in visit counts associated with the past  $n$  URLs and the order of URLs visited and it is implemented using genetic algorithms. The comparison module compares the predicted and the actual streams and provides feed back of the error in prediction to the learning module. The generative module attempts to predict the next  $k$  URLs that might be visited for possible pre-fetching. Finally, the pre-fetching module decides which URLs to actually pre-fetch based on constraints such as available bandwidth, recommendation from the server and the state of the web client.

In more detail, the authors use a genetic algorithm to learn the trend in visit counts of URLs and the order of visit of URLs by a web client, in order to predict the next few URLs that might be visited. The URLs visited by the user are considered in blocks of  $n$  URLs, which are called loadgenes. The genetic algorithm takes  $k$  previous loadgenes as input and predicts the next loadgene, by learning from the relationship among previous loadgenes. The relationship that is considered among the previous loadgenes is the visit counts and the order of URL visits. Each loadgene is associated with an incidence vector and a transition matrix. When comparing the two loadgenes, the difference in incidence vectors is determined. This is yet another vector whose

elements represent the difference in visit counts of the URLs of the two loadgenes under consideration. The difference in incidence vectors for the past  $k$  loadgenes provides a range within which the visit count should lie for the URLs in the predicted loadgene and the unique URLs to be present in the same. Thus the difference in visit counts and the URLs that are present in a loadgene provides a measure of the fitness of a candidate loadgene. Candidates with high fitness values are preferentially chosen to propagate and reproduce further.

In their research, the authors highlight the problem of dynamically generated Internet content, which renders pre-fetching approaches useless. Indeed, given the trend to deliver highly customisable content to users and the generic structure that is “populated” by dynamic articles, which now forms the basis of many major sites, the validity of the pre-fetched content becomes an important issue. An interesting point in this research was that the authors manage to prove, through simulation on actual client traces, that their proposed pre-fetching technique allows the maintenance of a client cache hit ratio of around 13% on average, even when all the visited URLs are dynamic.

Interesting research on proxy cache was also presented by Foygel and Strelow in 2000 [Foyg00]. The authors propose a system of hierarchical proxy caches, where their algorithm observes requests to a cache and its ancestors, before initiating pre-fetching for the predicted future requests. This would only happen if the pre-fetching action is deemed likely to reduce the overall latency experienced by the cache’s clients. Their algorithm is based on the continuous evaluation of the usefulness of each document in the cache, but also of documents that are not in the cache, but are likely to be needed in future requests. The set of documents (fetched or un-fetched) which has the greatest esteemed value is kept in the caches. In their conclusions, the authors argue that a hierarchical cache network structure is the ideal foundation on which pre-fetching can yield significant performance gains. Again, however, they highlight the concern over increased network utilisation, although they argue that it is often the case that because traffic is added to under-utilised networks, the performance gains can be obtained without significant cost.

Brian Davidson [Davi02] presented an article in 2002, which relates to predicting web actions from HTML content. In his work, he compared the simplistic

approaches so far taken for pre-fetching based on HTML content, with an information retrieval-based one that ranks the list of links using a measure of textual similarity to the set of pages recently accessed by the user. These simple approaches vary, but examples are namely pre-fetching all hyperlinks in a page, or pre-fetching all hyperlinks in a serial manner, as time allows.

The algorithm used for measuring the similarity between two text documents ( $D_1$ ,  $D_2$ ) is

$$TextSim(D_1, D_2) = \sum_{all\_w} TF(w, D_1) * TF(w, D_2)$$

( $TF(w, D_1)$  = the number of times term  $w$  appears in  $D_1$ )

Having compared text-similarity-based ranking methods to simple original link ordering and a baseline random ordering, the author found that similarity-based rankings performed 29% better than random link selection for prediction, and 40% better than no pre-fetching in a system with an infinite cache.

The ideas of Davison are implemented in Mozilla, an open-source web browser in a technical report published in 2003 by Zhang et al. [Zhan03], who incorporated the aforementioned content-based prediction algorithm with the history-based prediction described in another work by Davison [Davi04].

Further research presented by Cohen and Kaplan [Cohe00] in 2000 could be considered related, although it does not explicitly discuss a document pre-caching technique. Their proposal is that in order to overcome potential problems in the validity of cached documents, and other problems that relate to the increased network utilisation, which in turn might cause clients to experience even more latency, one could pre-fetch (rather, pre-execute, one might add) the *means* of getting a document, rather than the document itself. This is because of the observation that the actual steps required for the setup of a connection, are relatively costly in terms of time. A suggestion is made that this *pre-transfer prefetching* could be accomplished by

- Pre-resolving, which means that the browser or a proxy could perform a DNS lookup before a request to a server is issued, therefore eliminating the DNS query time from user-perceived latency

- Pre-connecting, where the browser or a proxy establishes a TCP connection to a server, prior to a user's request. This should address the problem of connection establishment time, which is significant compared to HTTP request response times.
- Pre-warming, or, in other terms, sending a dummy HTTP HEAD request prior to the actual request, in order to address the problem of start-of-session latency at the server, which tends to be larger for first-time requests than follow-up requests (the server is referred to as being “cold” or “hot”, once a request has already been issued).

The tests conducted in this research show a significant decrease in average latency times, proving that pre-fetching the means for getting a document is a useful technique that can be applied to the problem of reducing overall latency.

## ***2.4 Anticipating users needs***

### **2.4.1 Acquiring user context and learning user preferences**

#### **2.4.1.1 The new user problem**

Searching the web for information is a task that is difficult for the users, given the size of the information space whose elements need to be combined, in order to produce firstly a meaningful query, and secondly, a set of useful results. The complexity of the information that needs to be combined in order to produce a query that will yield the desired results makes the task difficult for humans, and to an even greater degree, for automated systems that attempt to substitute or support the users in these tasks. Even when dealing with groups of users from similar backgrounds, it is impossible to apply a solution that would cover everyone's needs completely. Given the individuality and variance of the circumstances, preferences and needs, however slight, of different users, a personalised and adaptive solution is perhaps the only approach that can ensure a satisfactory user experience.

For an automated system that would substitute the user in the task of generating web queries, the single most difficult challenge is the comprehension of

the user's environment, preferences and needs, simply referred to as the user's context. However, before attempting to solve this problem, one must look first at another question: What information should be assumed for the user's context, when very little or nothing has been done in order to acquire accurate information (e.g. in the case of a new user). This problem has been encountered in research for recommender systems and it is called the *new user problem*. This problem has been referred to by Avery [Aver99], Balabanovich [Bala97] and Good [Good99].

#### **2.4.1.2 Solving the new user problem with filtering techniques**

A technique that could be potentially useful for addressing the new user problem is known under the term of "collaborative filtering". This term has been used to describe an approach for recommender systems to suggest options to a user, based on the preferences of a larger group of users. The basic principles behind collaborative filtering form two basic schools of thought, one that uses "memory based" algorithms, and one whose approach is "model based", as argued by Breese et al. [Brees98]. Memory based algorithms tend to form a database of all the users' preferences for all items. Examples of such implementations can be found in work by Resnick [Resn94], Shardanand [Shar95], Breese [Brees98] and Basu [Basu98]. In contrast, model based algorithms firstly form a descriptive model for all users, items and preferences, based on the compilation of the user preferences; Secondly, they provide recommendations that are relevant to the given model. Such algorithms are again described by Breese [Brees98] and Ungar [Unga98].

Inspired by the work on memory and model based algorithms, Pennock and Horvitz proposed in 2000 the use of a "value-of-information" calculation to discover the most valuable ratings information to next gather from a user [Penn00]. Their approach is a hybrid memory-model algorithm, where "all data is maintained through the process, new data can be added incrementally and predictions have meaningful probability semantics". A discussion is made of their proposed collaborative filtering method called personality diagnosis (PD), where each user's reported preferences are interpreted as a manifestation of their underlying personality type. For the purposes of their research, personality type is encoded as a vector of the user's "true" ratings for titles in a movie database and it is assumed that users report ratings with Gaussian error. A summarised description of this approach considers the active user to be

“generated” by choosing one of the other users uniformly at random and adding Gaussian noise to his or her ratings. Given the active user’s known ratings, one can infer the probability that he or she is actually one of the other users, and then compute the probabilities for ratings of other items. Their experiment compared the Personal Diagnosis algorithm against two memory-based and two model-based algorithms and found statistically significant evidence that proves this approach makes better predictions than the single-scope approaches, although the researchers have not conducted exhaustive studies in multiple data sets.

The aforementioned approaches to the new user problem are not the only ones that have been investigated. Kohrs and Merialdo [Kohr01], for example, make use of entropy and variance in their ratings data in order to generate more accurate predictions for new users. Another approach to solving the new user problem is to create pre-made user categories and to assign new users to one of them. Determining the category to which a user belongs can be accomplished by asking the user pre-determined questions that build a user preference structure. This helps introduce the user into the system without requiring a substantial number of ratings, as shown in work by Ha et al [Ha98]. and Nguyen et al [Nguy98]. These approaches address the question of what to present first by starting with a small set of preference models (e.g. demographic models, models based on attributes of items) and asking questions that help choose an appropriate model for a user. When these models are accurate they can be quite useful, but the premise of personalized recommender systems and collaborative filtering is that a person’s preferences are a better predictor of other preferences than other attributes.

An interesting variation of the aforementioned approach suggests the integration of agents into a collaborative filtering environment to extract user preference information transparently, as presented by Wasfi in 1999 [Wasf99]. This method has the advantage of collecting implicit information in addition to explicitly provided ratings, and should gather data for new users more rapidly.

Work by Rashid et al. [Rash02] in 2002, investigated the problem of acquiring new user preferences in more depth, by performing comparisons on different preference acquisition strategies and the relative cost they presented to the users. They conclude that the optimum approach varies with the nature of the

recommender system's field of application. In relation to my own research, they suggest that for systems that need to make recommendations without any knowledge of the user, it is better to start with the most popular items and then adopt an *item-item* recommendation approach. The latter is a technique where once a user has provided at least one rating, a recommender is used that computes similarity between items to select other items that the user is likely to have seen. Then, the system updates the list of similar items whenever the user submits more ratings, remembering items that the user has already seen so that they are not presented again.

Investigating the problem of presenting a new user with information, but also of learning user preferences, under an influence from the Information Retrieval field, Billsus [Bill00] et al. presented an article in 2000, which described a learning agent for wireless news access. In their work, a short-term model for new users employs a list of tf-idf weighted words from the full text body of the news story to obtain "similar" documents that might be of interest. This is employed at a first level of preference. However, in the long term, a list of "informative words", indicative of news story categories, is compiled. These are words that constantly receive a high tf-idf score in stories from a given category and these are meant to reflect the users' general preferences. A naïve Bayesian classifier is then used to probabilistically estimate the interest of a story but only if it contains a given number of "features". Otherwise a default score is given. This long-term interest history is applied when the short-term algorithm cannot classify a story. Interest in a news story is obtained exclusively through implicit means, i.e. the observations of the user's interaction with the news client (opening a headline, reading a story etc.). In an experiment during a period of ten days in October 1999 the researchers used the system's personalized relevance prediction for half of the users, while the other half received news stories in static order determined by an editor at the news source (Yuhoo! News). On odd days, users with odd account registration numbers received a personalized news order and even users received a static order. On even days, this policy was reversed. To quantify the difference between the two approaches, the mean rank was measured, i.e. the mean display position, of all selected stories for the personalized and static operating modes. Expressing these mean ranks as a function

of the number of previously selected stories showed a significant difference between the two approaches. Restricting the analysis to stories selected by users that previously retrieved 10 or more stories, resulted in an average rank of 6.0 in the static mode, and 3.8 in the adaptive mode (the selection criteria hold for 20 users that selected 14.5 stories out of 777 headlines). The practical implications of this difference became apparent when we looked at the distribution of selected stories over separate headline screens (every screen contains 4 stories). In the static mode, 72.8% of the selected stories were on the top two headline screens, while this was true for 93.6% of the stories in the personalized mode. The limited availability of the test platform (Palm VI) at the time of the experiment provided a small number of user-subjects, thus limiting the reliability of the results, although they are a good indicator of the performance potential of the adopted approach.

Morita and Shinoda [Mori94] presented their findings on research conducted over the problem of filtering information, based on user behaviour. In their work, they assume that reading time in news articles would indicate greater user interest in these. Also their second hypothesis is that users perform syntactical pattern-matching in order to decide upon the relevance of an article, rather than read the entire text. An argument is made that information can be filtered based on these two hypotheses and the authors describe a human-based experiment and a simulation. Based on the data acquired from these, their findings suggest that both hypotheses are true, despite concerns over the effect of the article readability on reading times and the fact that the ideographic nature of the language of the data set (Japanese) might contribute to different behaviours.

Filtering the information that is presented, or in our case, pre-fetched for the user can only be achieved through the understanding of the user's preferences, as demonstrated in the examples above. This can be achieved by making assumptions regarding the context of a user by assigning a category to them. For example, one could safely assume that if a user is a lecturer in the field of Computer Science, then they may be interested in documents that relate both to the advances of Computer Science, but also to matters related to academic administration procedures, teaching methods and research-related information. Further granularisation of the user categories might help even more in the provision of meaningful documents.

Continuing the previous example, a system could assume that a lecturer for Operating Systems might be more interested in documents from that particular field and therefore present fewer documents from fields such as Information Retrieval and Programming. When the amount of documents presented to the user is smaller, but more representative of the user's interests, the overload of information is avoided and the task of finding appropriate information is greatly simplified.

## **2.4.2 Long term acquisition of individual user preferences**

### **2.4.2.1 The concept of Relevance Feedback**

Despite the advantages of an approach such as collaborative filtering, and its suitability for solving the new user problem, the greatest drawback this technique has is perhaps one of critical importance in some applications. We refer to the inability of such a group-model based approach to capture the individuality of a user and therefore provide content that is immediately relevant and tailored to the needs of the user, rather than the needs of a social, cultural, professional or otherwise contextual group they may belong to. Furthermore, a user can simultaneously belong to more than one of these groups, making the combination of the results from such an approach difficult, since complex decisions about which group should be dominant for a given query have to be made.

Another approach leads to a more personal end result and is concerned with obtaining training data from a single user, rather than assuming conditions based on group preferences. This approach comes from the field of Information Retrieval and uses *Relevance Feedback*, an assortment of methods from obtaining user preferences either directly or indirectly. These preferences are used to train user models (either at given intervals or continuously), in order to customize the retrieval mechanism to each individual. Because relevance is a deeply subjective term, as it is solely defined by each user based on their context (which remains extremely difficult to capture in whole), group preferences seem an attractive approach for dealing with users we know very little about. In contrast, relevance feedback and trainable user models are perhaps a more appropriate approach for recommender and pre-caching systems, such as the one my research is concerned with.

The term *Explicit Relevance Feedback* is used to describe those methods that are designed to enquire the user directly, with regard to the perceived relevance of a given document. Explicit relevance feedback can be obtained through several methods, such as specifying keywords, selecting, marking and rating retrieved documents or by forcing users to answer direct questions regarding their preferences. One can immediately see that this direct approach is possibly very efficient in capturing the user's preferences and context. However, the trade-off is the cost of implementing this approach, in terms of additional actions that are imposed on the user. A constant load that is placed on the user is bound to produce a negative perception of a system which is in constant need of explicit instruction.

On the other hand, obtaining information implicitly (*implicit relevance feedback*) can be used to collect information for the user's context, without placing any additional loads to them. This is generally accomplished by the constant but unobtrusive monitoring of the user's interactions with and behavior within the system. Examples of appropriate metrics that can be monitored are the reading time, bookmarking actions, scrolling, text selection and saving. Nichols [Nich97] highlights the fact that implicit relevance feedback is regarded as less accurate than its explicit counterpart. However, given the large volume of information that can be obtained virtually without any cost to the user and the possibility to combine it with other information that can be obtained explicitly, implicit relevance feedback appears as an attractive approach that is worth further investigation.

#### **2.4.2.2 Using Implicit Relevance Feedback for the acquisition of user preferences**

Extensive work has been carried out in the field of implicit relevance feedback, of which one early example (Morita and Shinoda, 1994) was mentioned earlier. In summary, they examined reading time and its relation to document relevance, and found that a correlation exists (with a coefficient of 0.49) between the two. Some further confirmation of the work pioneered by Morita and Shinoda is given by Konstan et al. [Kons97], who conducted a larger scale experiment based on USENET articles. In their work, a clear and again almost linear correlation seems to appear between the average reading time and the interest level for a given document.

Further work by Claypool et al. [Clay01], supported the findings of Morita and Shinoda, by examining the reading times of web documents versus explicit interest ratings, which were on a scale of 1-5. Their investigation found a strong correlation between the two (an almost linear trend is shown) and also found that scrolling a page and reading time combined with scrolling were good interest indicators. Their examination of the amount of mouse clicks and time of moving the mouse over a web document, against the explicit interest ratings, did not provide conclusive evidence of any correlation.

Despite this work with regard to reading time, opinions, with regard to using reading time as a metric, so far seem to be in contrast today and there has not been a single study that can unquestionably prove the reliability of its use for all types of documents and under all circumstances.

Kim et al. [Kim00a] also examined the relationship between the mean reading time and the binary indication of relevance/irrelevance of documents. Two user groups, one from the Telecommunications and one from a Pharmaceutical background, were provided an array of scientific papers and were asked to indicate their relevance, using a scale of 0-3. Their findings suggest that it might be impossible to distinguish between the degrees of relevance, based on reading time, since

*“..In both experiments, we noted a decline in mean reading time between articles rated as moderate interest and those rated as high interest. In fact, a consistent decline in reading time in the second experiment was evident as interest increased..”*

The researchers analysed their results using a revised scale where 0 is “irrelevant” and all other scores indicate a “relevant” state. While the two groups had a very similar mean reading time for relevant documents (50.49 and 53.19 seconds), the corresponding figures for irrelevant documents were rather far apart (32.85 and 42.97 seconds respectively). Their findings suggest that reading time can be used to measure relevance, although not at great resolution. Further to this, it must be noted that these findings show that the context of the users and the nature of the presented documents, as well as the settings for each experiment, play a significant part in the determination of appropriate thresholds for using reading time as a metric.

Kelly and Belkin [Kell01] examined the reading time hypothesis proposed by Morita and Shinoda under a different context. In their work, they used data extracted from trace files generated during the TREC-8 interactive searching study, which implemented relevance feedback techniques in two experimental Information Retrieval systems. After analyzing the data, the researchers found that there was no statistical significance between the reading time and the relevance of the presented documents (Relevant: mean reading time=27.62s, standard deviation=25.99s, Irrelevant: mean reading time=25.63s, standard deviation=23.65). Similar conclusions were reached for the amount of document scrolling and other interaction, i.e. command button clicking within the IR systems. It is noted however that these results may be biased by the additional tasks that users had to perform in this experiment (construct queries, evaluate, save and label documents), in contrast with the original experiment by Morita and Shinoda, where users only had to evaluate incoming articles. Also, in the original experiment, the tasks were interesting and familiar to the user, while in Kelly and Belkin's experiments, the tasks were artificial.

Further from time, scrolling and clicking, researchers have proposed other metrics for observable behaviour that might be perceived as interest indicators. These, however, seem to be dependent on the nature of the system that is under investigation in each experiment. Konstan, for example, proposed printing, saving, replying to, forwarding and posting a follow-up message to an article as potential interest indicators. Nichols [Nich97] proposed a more extensive list of observable behaviour for implicit relevance feedback, which included purchase, access, repeated use, print, save, delete, refer, mark, examine/read, glimpse, associate and query. These are shown together with explanatory annotations in table 5.

Further work by Kim and Oard [Oard98] categorized the behaviours mentioned by Nichols into three broad categories: Examination, Retention and Reference. These are shown with further clarity in table 6.

Apart from using reading time, the researchers examined retention behaviour (namely printing) in an experiment based on the users of the Powerize.com servers. In this study [Kim00b], they concluded that both reading time and retention behaviours provide good interest indicators for implicit relevance feedback.

Action	Notes
Purchase (Price)	Buys item
Assess	Evaluates or recommends
Repeated Use (Number)	e.g. multiple checkout stamps
Save/Print	Saves a document to personal storage
Delete	Deletes an item
Refer	Cites or otherwise refers to item
Reply (Time)	Replies to an item
Mark	Adds to a “marked” or “interesting” list
Examine/Read (Time)	Looks at whole item
Consider (Time)	Looks at abstract
Glimpse	Sees title/surrogate in list
Associate	Returns in search but never glimpses
Query	Association of terms from queries

**Table 5: Konstan’s proposed observable behaviour.**

Category	Observable Behaviour
Examination	Selection Duration Edit Wear Repetition Purchase (object or subscription)
Retention	Save a reference or save an object With or without annotation With or without organisation
Reference	Object → Object (forward, reply, post follow-up) Portion → Object (hypertext link, citation) Object → Portion (cut & paste, quotation)

**Table 6: Categorisation of observable behaviours by Kim and Oard**

Chan [Chan99] underlined the unreliability of using reading time as a metric alone by identifying cases where time spent on a page did not necessarily reflect user interest in it. Such cases include occasions where the user is distracted by other activities, such as answering an incoming phone call or where the page is relatively small, thus requiring less time to read in full. Chan proposed the normalisation of

reading times by taking page length in mind, as well as imposing a reasonable upper bound on measured reading time, which in his study was 15 minutes. Methods such as eye-tracking in order to capture the actual reading time of a document are also proposed by Chan; however, such an implementation for the purposes of implicit feedback does not seem to exist in current bibliography. The aforementioned observations were also made by Huang [Huan01] in his Masters thesis, who proposed a further solution by estimating the reading time of an article based on the user's average reading speed and using this as a better informed threshold.

An important observation on reading times is that their distribution curves tend not to be normal. This hinders the statistical analysis process but a method to overcome this problem is proposed by Rafter and Smyth [Raft01], who recommend a two-step process to prevent unreasonable reading times from interfering with the use of time as an implicit relevance metric. They analysed the access logs from a job-finding site, where profiles were created for each user. These profiles included the calculated reading time for each job entry and the clicking on each job (it was assumed that a revisited job was of more interest). The median of medians read-time values per individual job access (as opposed to the total read time over a number of visits to a job) for both users and jobs was used to calculate a normal read-time for the system. Spurious reading times were then identified using this normal reading time and outliers were replaced by this value. Graded reading times per job were then computed by calculating, in each user's profile, the number of standard deviations each job's newly adjusted read time was above or below the user's mean reading time. The researchers concluded that using the adjusted reading time data resulted in better predictions, which in turn suggests that the normalisation of behavioural data might be necessary to produce more accurate results.

## **2.5 Query formulation**

### **2.5.1 Manual and Automatic Query formulation**

One of the most important and difficult challenges in my proposed model is the generation of suitable queries from the data found within calendars, so that relevant and useful documents can be retrieved from the web. As it is immediately obvious, the quality of the generated queries will have a significant impact on the quantity of the relevant documents that will be returned. It is therefore critical for the system to achieve a level of query formulation skills, which would match that of seasoned web users.

Ouellet et al. [Ouel00] presented some work on a system that automatically enriches the personal information space for a user browsing or querying the Internet. In this work, they identify some basic steps in the process of automatic query generation and web document retrieval, namely:

- Choose terms that are representative of the concept
- Organise these terms into a query
- Choose a search engine and formulate the query for this particular engine
- Launch the query and retrieve the results
- Filter out links that are strictly publicity or that are judged to be irrelevant
- Present the results to the user

Of all these steps, the researchers identify that the most important and perhaps most difficult to accomplish with some success, is choosing the terms which are representative of the concept (the theme that the user is interested in). The method they chose to employ is extracting terms from the name of the concept, e.g “web site design and management”, although mention is made that more sophisticated methods can also be employed. Such methods are the use of a thesaurus to fetch descriptive terms, using the most frequent terms in documents already attached to a concept,

using the terms that are involved in the title or headers, or those that have some typographic emphasis or are declared as keywords. The researchers achieved a precision indicator of 23% for documents retrieved, which were relevant to intended concept. Considering the fact that Internet search engines have a precision of 23 to 38% [Hawk99], these results seem to confirm the effectiveness of Ouellet's approach in generating queries that retrieve relevant documents. Moreover, some documents, although irrelevant for the target concept, may still be useful for some other concepts in the structure.

The work by Ouellet et al is influenced by previous publications from Pazzani et al [Pazz96], who proposed a system (Syskill & Webert) that could generate user models and then employ these to automatically generate and submit queries to the LYCOS search engine. In this implementation, the researchers use the words found most frequently in highly rated pages, along with words that discriminate the topic from other topics, and submit a list of these words to LYCOS. Since LYCOS cannot accept very long queries, they use the 7 most discriminating words that are found in a higher proportion of hot pages than all pages and the 7 most commonly occurring words as a query. The choice for the length of the query is not otherwise justified so it can only be assumed that the inclusion of this many terms in a query is made in order to achieve as high accuracy in topic discrimination as possible. This is quite an interesting approach, when it has been observed that the average length of a query to a search engine, as reported by Jansen et al. [Jans98], is 2.35 terms. This is supported also by the findings of Croft et al. [Croft95], who identified that the average length of a world-wide-web search query is approximately two words. Another interesting observation by Jansen was that less than 10% of the queries actually contained any Boolean operators (e.g. queries like "Edinburgh AND map OR hotel").

### **2.5.2 Enhancing queries (query expansion).**

There are several problems with regard to querying for documents, which lie mostly in the natural diversity of the humans that generate the documents and those that search for them and the inherent attributes of natural language that contribute to ambiguity. Synonymous words that can be used to describe the same concept are an

adequate example of this problem. In fact, studies by Borgman [Borg98] indicate that even experienced system users encounter problems concerning search strategies and output performance, while Fenichel [Feni81] shows that even experienced searchers lose sight of the search logic, miss obvious synonyms and search far too simply. This vocabulary problem was also discussed by Furnas et al. [Furn87] in 1987, who indicate that the problem is aggravated when queries are especially short, or when the corpus under search is very large (e.g. the Web).

To overcome such difficulties in query formulation and to assist users obtain results, which they might have not otherwise been able to access, researchers have in fact worked on the field of query expansion for more than two decades. Early work by Bates [Bate79] in 1979 identified a number of search tactics and organised a catalogue of those, however, without proceeding further into proposing a recommendation on the circumstances, under which each tactic might be used. She identified four main areas of search tactics, namely monitoring, file structure, term manipulation and search formulation. Of these four, perhaps the most interesting and applicable areas for this research are the manipulation of terms and the formulation of searches. The tactics proposed for search formulation and term manipulation describe the available techniques to broaden and narrow queries. The search formulation tactics include “*the selection of appropriate initial search terms and the manipulation of query structure*”; the term manipulation tactics describe “*the use of context, thesaural terms, and stemming to modify queries*”.

Further work by Fidel [Fide91] proceeded to the generation of a formal decision tree based on the intuitive term selection methods employed by online searchers. Fidel describes the options available to the searchers, as well as the conditions under which each option is selected. Furthermore, rules are defined for deciding when to use text words or descriptors or both to search indexed databases, and guidelines for including thesaural relationships. These are shown in figure 11.

There have been several approaches to solving the problems described above, however, these can be mostly grouped into three main areas, i.e. the manual or automatic creation of thesauri, the exploitation of content relationships of documents within a collection and, lastly, the automatic extraction of terms from top-ranking retrieved documents. The latter two approaches are also referred to as Global and

Local document analysis respectively. In the following section, reference will be made to some of the most significant work in the field. Given the early interest and therefore substantial volume of work in this field, the interested reader can find a more comprehensive bibliography by Efthimiadis [Efth96], which contains work from 1971 to 1996.

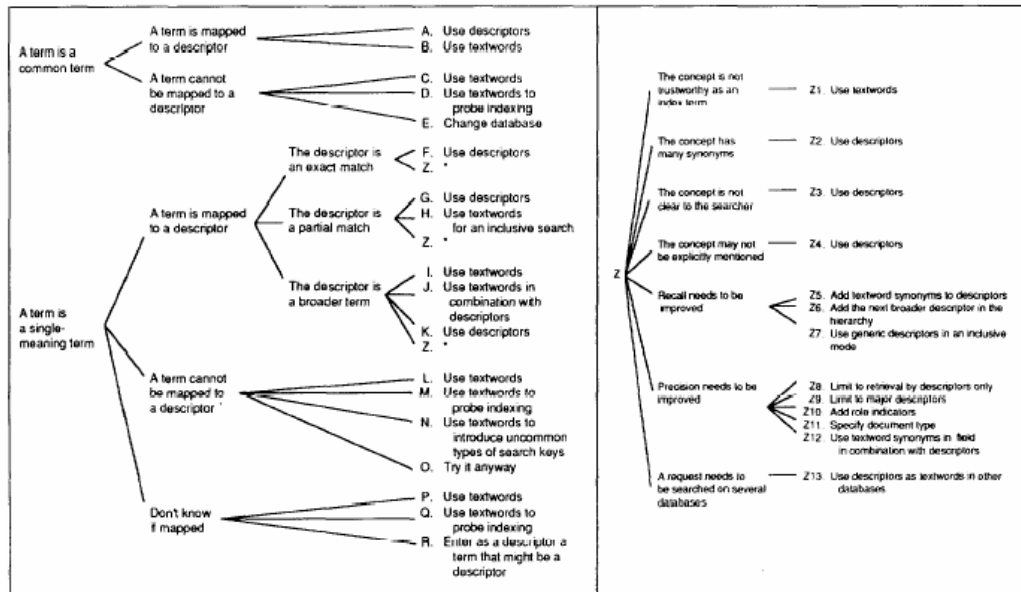


Figure 11: Fidel's decision tree for term selection

### 2.5.2.1 Automatic and Manual Thesauri

Perhaps the most interesting technique employed for the automatic generation and reformulation of queries is the use of a *thesaurus*, in other words a large list of synonyms and related words. If a query was expanded with the use of words found inside a thesaurus, the chances of matching words in relevant documents would potentially be increased, as different authors would use different words to describe the same context. The problem therefore shifts to the creation of a suitable thesaurus, either by hand or in an automatic fashion, from which further appropriate keywords can be identified and used to broaden (or narrow) a query.

### 2.5.2.1.1 *Manually created general thesauri*

Manual thesauri have been used for a multitude of years in experimental and commercial systems and are a fairly widely adopted practice. Thesauri of this kind are either general word thesauri (such as WordNet, which is discussed below, and Roget's), but they are seldom used for IR practices. Other thesaurical collections are exemplified by phrase-based implementations such as MeSH (also discussed below), LCSH and INSPEC. The main disadvantages related to manual thesauri are the cost of building these and the relative inflexibility that they offer, in terms of ease and speed of updating.

Using a general, manually created thesaurus might seem a good solution to traditional IR problems, however, in a study by Voorhees [Voor94], little evidence is found that they offer any improvement in the effectiveness of the search, even when words are hand-picked by the searchers. Fox [Fox80] furthermore had suggested much earlier that query expansion based on manually created thesauri could only be successful *“if a domain-specific thesauri is used which corresponds closely to the domain-specific document collection”*.

Using WORDNET<sup>viii</sup> as a tool for query expansion, Voorhees conducted experiments using the TREC collection. Through the use of synonyms, hypernyms and hyponyms, she attempted to expand all the queries with additional terms, which were appropriately weighted using a weight set  $S=[0.1, 0.3, 0.5, 1.0, 2.0]$ , while giving the original query terms a weight of 1. The results showed that retrieval performance was only improved with short queries, while longer ones did not show any significant improvement.

Another approach using WORDNET was that of Smeaton et al. in 1996 [Smea96]. They attempted to expand queries to the TREC-4 set by adding terms from WORDNET. At first, a set of hierarchical concept graphs was derived directly from WORDNET. Comparisons were drawn on the similarity of concept classes

---

<sup>viii</sup> WordNet is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organised into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets (Source: <http://wordnet.princeton.edu/w3wn.html>, valid 02/05)

using an algorithm. The approach suffered from the lack of inclusion of relationships other than IS-A (*something IS-A type of thing*) and from the polysemous nature of some words (i.e. bank – commercial bank vs. river bank). Previous work by the researchers had shown that a distinction between WORDNET word senses from free text cannot be made with a degree of accuracy greater than 65%. As a second element in their experimentation, a disambiguation of word senses was attempted and it was found that performance was not degraded in neither the training sets nor in the actual TREC-4 set runs. There were in fact approximately 30% more query terms after disambiguation. In their final experiment runs, and in order to emphasise on the effectiveness of the WORDNET-based query expansion alone, the researchers omitted all of the original query terms from the queries. The final results of the experiment were discouraging; however, they did manage to find 347 relevant documents out of 6501 by using just the expanded query terms, which means that potentially these documents contained none of the original query terms. In turn, this indicates that while the technique is inefficient as a stand-alone approach, it might prove valuable combined with other methods.

#### ***2.5.2.1.2 Automatically created Thesauri and Query Term Generation***

With the discouraging results from the use of manually created general thesauri, alternative methods for automatic thesaurus constructions have been looked at by several researchers. Such a possible alternative approach is to analyse the text of the corpus being searched in order to obtain a more effective thesaurus. Again, two slightly different approaches have been investigated in a number of studies for the automatic generation of thesauri. One approach looks at the word occurrences and relationships in the corpus as a whole, while the second approach only looks at the top-ranked documents retrieved by the original query. These approaches are referred to as Global and Local Analysis respectively by Xu [Xu96], whose work is discussed later in this section.

##### **Global Document Analysis**

Global Analysis was pioneered by Sparck Jones [Spar71] in 1971, who clustered words based on the co-occurrence in documents. These clusters were then used to

expand the queries. Since then, it was not until the early Nineties until consistent effectiveness improvements were obtained by using the analysis of entire corpora as a basis.

An approach that uses clustering to determine the context for document analysis is reported by Crouch and Yang in 1992 [Crou92], who performed four experiments with automatically generated thesauri that provided very encouraging results. Their thesaurus generation algorithm is summarised by the three following steps:

- The document collection is clustered via the complete link clustering algorithm
- The resultant hierarchy is traversed and thesaurus classes are generated, based on three user-supplied parameters
- The documents and queries are indexed (augmented) by the thesaurus classes

The link clustering algorithm mentioned above is described by the authors to work in the following manner: “At each stage, the similarity between all clusters is computed, the two most similar are merged and the process continues until only one cluster is made. Since the similarity between clusters is defined as the minimum of the similarities between all pairs of documents (where one document of the pair is in one cluster and the other document is in the other cluster), the resultant clusters are difficult to join and hence tend to be small and tight.”

In 1993, Qiu and Frei. [Qiu93] presented further work which was based on global analysis techniques. Their presentation is that of a probabilistic query expansion model, which is based on a similarity thesaurus that is constructed automatically. The similarity thesaurus is essentially a matrix, which consists of term-term similarities. However, while the study by Sparck-Jones uses the probabilities of the terms representing the documents to build a co-occurrence matrix, this approach uses the probabilities of the documents representing the meanings of the terms. This addresses the problem of selecting additional search terms. Another aspect, the weighting (according to importance) of these additional terms, is dealt in a probabilistic manner. This allows the selection of terms that are

closer to the *concept* of the query, rather than just those that are just similar to the query terms. The results from Qiu's approach were very encouraging. After experimentation with three distinct document collections, results showed an average of 20-30% increase in the retrieval effectiveness performance. Despite this value being smaller than the results obtainable by relevance feedback, one must bear in mind that the entire process is fully automatic and requires no input from the user.

Similar work was carried out by Schutze and Pederson [Schu97], who constructed a thesaurus based on word co-occurrence. The co-occurrence of words is defined as the appearance of words within 41 words of each other. Under this context, the co-occurrences were represented by a matrix where each entry represented the number of co-occurrences of the word for that column, with the word for that row. The similarity of words was determined by the similarity of their corresponding columns. Having observed that the construction of the matrix and the subsequent calculations would be computationally expensive, if a large number of words were used, the researchers used Singular Value Decomposition to reduce the dimensionality of the matrix. In the end, their thesaurus was a matrix where similarity was measured by second order of co-occurrence, that is to say by the sharing of neighbours. In order to avoid the coverage problems of thesauri like WORDNET, they disambiguated word senses by clustering the context vectors retrieved from a corpus and creating a sense vector which represented each cluster. The sense of each word was then chosen by finding the closest vector generated by the algorithm. In experiments with the TREC-1 corpus, the researchers observed a small query performance improvement of around 14%.

Work by Jing and Croft in 1994 [Jing94] showed results for PhraseFinder, an automatically built association thesaurus which was based on grammatical analysis. The co-occurrence between phrases and terms is considered by PhraseFinder as associations. These associations are extracted and represented as *text feature dependences*. Such features include terms (i.e. any word apart from stop words) and parts of speech (they used the Church tagger [Chur98] to give each word a part of speech). Further to this, the researchers added a few rules to define items such as paragraphs, sentences and phrases. Using this as a basis, several experiments were performed on different collections with encouraging results, however the researchers

highlighted the fact that a larger corpus appears to lend itself better to the automatic formation of thesauri. Indeed the retrieval performance for query expansion improved in larger collections. Further to this, an important observation was the superiority of noun-phrase based thesauri (especially noun-only phrases) over word-based thesauri. Finally, another noteworthy find was the fact that sampling overly large collections in order to generate a thesaurus seemed to have no significant impact on the quality of the thesaurus. The researchers however do not conclude on the optimal sample size.

Such results as those of Qiu and Schutze appear encouraging. As stated by Peat and Willet [Peat91] in 1991 however, “The weight of the experimental evidence to date hence suggests that query expansion based on term co-occurrence data is unlikely to bring about substantial improvements in the performance of document retrieval systems”. The reasons for this are possibly not due to the use of co-occurrence data, but could be found in the way these data are used, as maintained by Ferber [Ferb96] in an unpublished, although interesting study:

- Some studies used similarity measures that favoured frequent terms
- The expansion was often done for each single query term in isolation, and not for the query as a whole
- The size of the text collections from which the co-occurrence data was extracted was rather limited resulting in weak estimations of probabilities of co-occurrence

Ferber’s study provides some evidence that the use of co-occurrence data can provide significant improvements to the IR process. However, he highlights the fact that a suitable model for the use of this data has to be carefully designed and that the corpus from which co-occurrences are taken has to be large enough.

Interesting critique on the use of thesauri was performed by Mandala [Mand99] in 1999, who identified three main categories of thesauri: hand-crafted, co-occurrence based and head/modifier based. The shortcomings of each type of thesaurus are identified and a proposal is made for an approach that uses the combination of all three types, under the theory that each type of thesaurus should

help overcome the constraints posed by the other types. Furthermore, the expansion terms are weighted, in order to eliminate misleading expansion terms. Their experiments on the TREC7 collection shows that there is a considerable gain when using a combination of all types of thesauri over any single thesaurus approach.

### **Local Document Analysis**

Local document analysis, as mentioned earlier, does not look at the whole corpus in order to generate collections of additional query terms. Instead, the analysis involves only the top-ranked documents retrieved from each original query. The idea was pioneered by Attar and Fraenkel [Atta77] in 1977, who used the top ranked documents for a query as a source of information for building an automatic thesaurus. After examining the terms found within those documents, the researchers clustered them and treated them as *quasi-synonyms*. Having tested their ideas on a Hebrew legal case text database and also on a small English text database (U.S. patents in electronics), the researchers found encouragingly similar performance improvement, despite the linguistic complexities of the Hebrew language.

In a study on probabilistic models for document retrieval, Croft and Harper [Croft79] also used a form of local feedback (pseudo or blind relevance feedback) by using information from the top ranked documents in order to estimate the probability of a term occurring in the relevant set for a query. Each term of the query was re-weighted accordingly but no extra terms were added to the query, therefore the technique was outside the strict notion of query expansion. Nevertheless, an improvement in the retrieval effectiveness was noted, although the test collection was not extensively large.

Work on local document analysis has been carried out at Cornell University by Buckley et al [Buck95], [Back96], using the TREC3 and TREC4 sets. In the latter experiments, the most frequent 50 terms and 10 phrases (defined as pairs of adjacent words, excluding stop words) from the top ranked documents, were added to (or in some cases, removed from) the query. Both experiments used the same approach of weighting terms, using the relevance feedback approach of Rocchio. According to this approach, expressed in vector terms, the final query vector

becomes the initial vector moved toward the centroid of the relevant documents and away from the centroid of non-relevant ones, through this formula:

$$Q_{new} = \alpha * Q_{old} + \beta * W_{rel} - \gamma * W_{nrel}$$

(where  $\alpha$ ,  $\beta$ ,  $\gamma$  are parameter weights,  $W_{rel}$  is the average weight in relevant documents and  $W_{nrel}$  is the average weight in non-relevant documents)

The difference between the tests on the TREC3 and TREC4 collections was that for the latter, there was a separate stage for tweaking the query term weights even further, on a per-query basis (Dynamic Feedback Optimisation). The weights are altered by testing whether a mildly changed term weight performs better when run on the learning set of documents, i.e. those documents already seen and judged.

One of the obvious shortcomings of Local techniques is the dependence on search engines to yield the top ranking documents. While the performance of these is potentially satisfactory in retrieving relevant documents, questions arise on how many documents should the “top sample window” contain. To illustrate this problem, if the top 20 documents were assumed to be of some relevance to the original query but only 5 were actually highly relevant, it becomes immediately apparent that the quality of information that will be used in order to obtain additional query terms is such that will yield very unsatisfactory results. This problem was addressed by Mitra [Mitr98] in 1998, who added a document re-ranking step in the process of local feedback. His method was as follows:

- To use  $K$  documents in the feedback process, retrieve a larger number  $T (>K)$  using the original user query
- For each of the retrieved document, compute a new similarity score  $Sim_{new}$  based on the occurrence of additional relevance indicators in the document
- Re-rank the retrieved (top  $T$ ) documents based on  $Sim_{new}$ , breaking ties by the original score
- Select the top  $K$  documents in the new ranking and use them in the Rocchio relevance feedback process to expand the query

- Use the expanded query to retrieve the final list of documents returned to the user

Using automatic and manually generated Boolean filters to compute the new similarity scores, the researchers arrive to the conclusion that their approach can give significant performance improvements. Using the TREC3-6 collections, they found a 7-22% improvement over blind local feedback using manual filters and 6-13% improvement using automatic ones.

## **2.6 Summary**

In this chapter, previous research was examined on the field of calendar research, as electronic calendars are the information source for the experimental system that was developed for the purposes of this thesis. Based on the model overview described in Chapter 1, a review of previous work on user preference acquisition was also attempted. Finally, In order to complete a theoretical background that would enable the full implementation of the model in Chapter 1, a review of the field of query formulation was made.

Based on the background information gathered from these reviews, the following chapter discusses the design and implementation choices that were made for the final pre-caching system, which relies on information that is found within user calendars.

## Chapter 3

Personal predictive Internet content Pre-caching  
for mobile devices

### **3 Personal predictive Internet content precaching for mobile devices**

#### **3.1 Introduction**

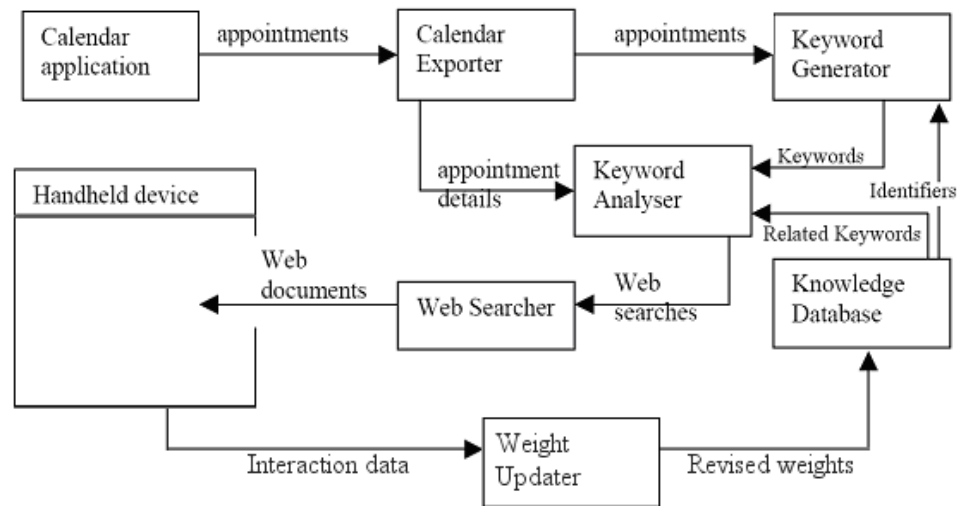
In Chapter One, a system was described which would be used to prove the hypotheses of this thesis. Such a system would comprise of two main software components: The first one, which would reside on the user's desktop PC, should be responsible for extracting keywords from the calendar entries of a user, forming appropriate web queries, retrieving documents with the user's desktop broadband connection and passing them on the user's handheld device for off-line browsing. All of these functions should take place on the user's desktop PC.

The second software component, which should reside on the user's handheld device, should have the role of presenting the documents to the user. Furthermore, it should be able to monitor the user's interactions with the retrieved documents, which, when passed on to the desktop component, will allow the latter to adapt its performance to suit the user's preferences.

Words such as “know”, “guess”, “remember” have been previously used in our theory regarding the way the predictive system should work. These prompt an insight of the human memory as a starting point to model the pre-fetching system. By looking at theories from cognitive psychology, a working model for the system could be produced that will make efficient use of memory by designing a flexible, adaptable and interconnected “knowledge-base” structure. By combining data in this memory, the agent can subsequently perform “guesses” at the user's needs. With the knowledge of general facts about the world (such as “people need maps when travelling to unfamiliar locations”) and knowledge about the users themselves, the agent can provide a meaningful and personalised experience to each individual user.

In the human brain, various tasks, such as storing information into working memory, are performed with the simultaneous work of various centres. A paradigm of this is the central executive, which uses other agent-type centres, like the rehearsal loop, in order to process and store information in the working memory area. This means that the central executive, which is the “main processor” for the system can

offload results from previous tasks to other, smaller capacity agents, so it is free to engage in other difficult processing tasks<sup>ix</sup>. From this paradigm, we adopt a modular and more refined form for the agent's structure, such as depicted below in figure 12.



**Figure 12: Revised system overview**

This is a helpful paradigm for our model, as the advantages of such a structure are obvious. Instead of relying on one main executive to perform tasks in a serial manner, it is possible to have several “sub-agents” working collaboratively in order to perform tasks with enhanced efficiency, in order to minimise the total running time of the retrieval process. The boxes in the previous diagram denote key “**activity areas**” of the agent. These activity areas are responsible for the performance of subtasks, the sum of which will form the existence purpose of agent (i.e. to pre-fetch internet content).

Following in the current chapter, a more analytical discussion on how this system operates will take place. Based on the schematic shown above (Figure 12), the main elements of the system will be discussed by presenting their design and implementation details jointly, since some implementation considerations have an

---

<sup>ix</sup> DANIEL REISBERG, Cognition – exploring the science of the mind, 2<sup>nd</sup> edition, W.W. Norton & co, New York. p. 15-17

impact on design decisions. The intertwining of these development phases would make addressing one impossible without referring to the other.

### ***3.2 Choosing a programming language and system environment***

Before discussing the model described in the previous section, mention needs to be made of the choices of the programming language and the environments that the system development would require. This necessity rises from the fact that several important design and implementation decisions, as well as system characteristics, would be dictated by these choices.

It was agreed that a desktop PC running Microsoft Windows 2000 would be the platform where the desktop software component would run. The combination of a x86 processor architecture running a Microsoft Windows environment is the most common configuration for desktop PCs and reflects a typical user scenario. Further to this, to represent the handheld device component of the system, an IPAQ PDA running the PocketPC2003 operating system was chosen. These types of handheld devices are very common, with 40.2% of the shipped units in Q1-2004 being such devices (Palm control 40.7% segment)<sup>x</sup>. Furthermore, a derivative of this operating system can now be found in many mobile phone devices (smartphones). The combination of desktop and handheld device is adequately suitable to represent a typical user scenario.

The programming language chosen for the implementation was C++. The first reason behind this choice was the availability of development environments for the devices the system would be implemented on (Desktop PC running Windows 2000, IPAQ PDA running PocketPC2003). The second reason behind the choice of language was due to the fact that all mobile operating systems currently support C++ programming, while only a few support other languages such as Visual Basic. Given an initial uncertainty on the mobile platform that would be used, it was deemed

---

<sup>x</sup> Windows CE ties PalmOS for Q1'04 market share,

<http://www.windowsfordevices.com/news/NS8063885791.html>

reasonable to restrict the choice of programming language to one that would be universal and could provide easily transferable code.

The code for the desktop components was written using the Microsoft Visual Studio 6 development environment. For the handheld component, Microsoft Embedded Visual Tools 3.0 environment was used, along with the PocketPC 2003 Software Development Kit.

### ***3.3 Data exchange mechanism***

Because the two components of the system would need to cooperate and exchange information, the need for a communication mechanism was apparent from the early phases of the design. It was decided that information would be exchanged through XML-styled data structures.

The goal of the desktop component would be to pass an appropriate XML file to the handheld component, along with the pre-cached documents. This would enable the handheld component to present the calendar entries, the keywords found therein, the associated web queries formed from the keywords and links to the relevant retrieved documents retrieved by the queries, to the user. Another XML structure would be used to pass information about user interactions related to web queries back to the desktop component. The final structures of these structures are presented in the next sections of this chapter, along with the relevant modules that were used to create them.

### ***3.4 Obtaining Calendar data (Calendar Exporter)***

#### **3.4.1 Theoretical Design**

As explained in the previous sections, the source of information for the pre-caching system will be the user's electronic calendar. From the entries contained therein, an attempt will be made to judge the nature of the user's activities, and once that has been done, keywords from the calendar entry will be matched with relevant keywords from the knowledge base in order to form retrieval queries.

For the implementation of the system, Outlook 2000, Microsoft's electronic calendar application was chosen as a calendar source, although the implementation works with other versions as well. In fact the software that was written under this

research is compatible with all Outlook versions beyond 2000, but there have been no tests with earlier versions. This platform was chosen mainly for three reasons, of which the third is arguably the most significant:

- Wide user base and multiple revisions mean the calendar application contains most (if not all) of the crucial characteristics highlighted and pinpointed in research by Kincaid and Dupont (see Chapter 2).
- Has the ability to synchronise with the calendar applications on most handheld devices (Nokia, SonyEricsson and PocketPC).
- Has an openly available API which can be exploited by any purpose-built application to control the export of items.

### 3.4.2 Implementation

In order to extract calendar entry entries, the Outlook automation feature was used in order to invoke an instance of the application and automatically extract all entries in a given date. The entry title, location and notes were inserted into small XML structures, such as the one depicted in the figure below. The structures are later expanded by other modules of the system, in order to hold information relevant to the calendar entry.

```
<appointment id=0>
  <ap_title>(data)</ap_title>
  <ap_loc>(data)</ap_loc>
  <ap_notes>(data)</ap_notes>
</appointment>
```

**Figure 13: Base XML structure created by Outlook Export module**

The relevant code for this module can be found in:

Filename	Line	Function Name
DeSnagDlg.cpp	278	OutlookXport(CString &theDate)

### **3.5 Identifying candidate keywords (Keyword Generator)**

#### **3.5.1 Theoretical Design**

One of the most important problems that the system needs to solve is the identification of keywords within the calendar entries, which can be used to initiate the process of web query formulation. To achieve this, a database of known keywords (identifiers) could be kept, against which the system would compare the content of the calendar entries.

In order to train the system with the ability to recognise potential keywords, it was decided that an analysis of the contents of a sample of real world calendars would be necessary. The scope of this analysis would be to determine firstly which suitable words and of what types, would be frequently encountered in a calendar. Subsequently, in order to confirm previous early research on calendars, an attempt would be made to identify calendar entry categories, so the list of keywords for these could be supplemented with other words that belonged to similar contexts. For example, if the analysis was to show that placenames (“Glasgow”, “Edinburgh”) were common occurrences, then an appropriately extensive list of placenames would need to be compiled. These keywords in essence are *identifiers* for the categories they represent.

Although early research by Kincaid and Dupont [Kinc85] had highlighted some of the categories of entries commonly encountered in calendars, it was felt necessary to re-investigate this matter, firstly due to the age of the preceding research (which was not based entirely on electronic calendars), and secondly to tune the system performance for use by individuals in an academic context. Because the subjects of our final experiments were expected to be from an academic environment, it appeared reasonable to attempt to gear the system towards the particularities of the academic community. The system could have similarly been geared towards other professional areas or could be tailored towards a general population. This would require the selection of appropriate test subjects and since the comparison of performance on different target groups is not part of the scope of this research, the academic sample and subjects were judged to be appropriate and

adequate. Further discussion on the collection of calendar samples can be found in Section 4.2.

### 3.5.2 Implementation

The database of identifiers which would be compiled, would not exclusively contain words, but also short keyphrases, such as “travel to”. The reason for this choice was that it would be almost impossible to compile a list of every possible associated identifier for every category. Therefore by including keyphrases, we could infer that an unknown word which would follow these, might actually be a good identifier candidate. For example, “trip to Garnethill<sup>xi</sup>” is a sentence where a human can immediately identify Garnethill as the name of a place, even though they might never have heard of it. It is also very unlikely that the name of Garnethill would come up in any general placename list. It was desired that same functionality for inferred knowledge should be implemented for the system as well.

The database was split into separate files, one for each category of calendar entry. Further to this, some further data such as names and surnames were added separately on distinct files. Although these did not belong to a category explicitly, they are useful as identifiers for multiple types of categories. Again this will be better described in the following section about the formulation of queries.

While for most of the categories the identifier list has not been overly extensive, one which did have a very extensive list was the “travel” category. The compilation of this category is based on the assumption that the user habitual location is known. In the case of this research, we assumed all our test subjects to be resident, or employed in the city of Glasgow (UK). Because through observation it was noticed that “travel” type entries often contain the name of the destination the calendar user intends to travel to, most of the identifier entries for this category are placenames. Due to the impracticality of storing every single placename in the entire world, a model which would depend on a pyramidal “level of resolution” was implemented. Centered on Glasgow, the following were included:

---

<sup>xi</sup> Garnethill, a region in the Glasgow City Centre area (UK).

- A list of all local placenames (city suburbs, local surrounding towns)<sup>xii</sup>. These were manually picked out from local maps.
- A list of all major cities and towns in the country (UK)<sup>xiii</sup>. The list for these came from two different sources.
- A list of all major world cities<sup>xiv</sup>. These refer to the 815 world's largest places, according to population.
- A list of all world countries<sup>xv</sup>.

The following figure illustrates a few extracts from the “travel” database itself.

[rules]	[countries]	[world major cities]	[country place names]	[local city surroundings]
travel	afghanistan	adan	epping	airdrie
travel to	albania	amman	aberdeen	alexandria
away to	algeria	aba	abingdon	balloch
departure	andorra	abeokuta	accrington	barrhead
airport	angola	abidjan	aldershot	bearsden
	antigua and	abu zabi	alloa	bellshill
	barbuda	abuja	alton	blantyre
	argentina	acapulco	altrincham	clydebank
			andover	coatbridge
			Angus	cumbernauld
				dumbarton

**Figure 14: Extracts from the “travel” category identifier database**

A further rule that was implemented was to dictate that irrespectively of the existence of a keyword from the “Location” field of the entry in the databases, the keyword would have to be treated as though it belonged to the “travel” category. Special provision was also made for the identification of first and second names. It was decided that if a word started with a capital letter then it might be considered to be a name. Further confirmation to this would come if the following word also started

---

<sup>xii</sup> Compiled data from <http://www.multimap.com>

<sup>xiii</sup> Compiled data from [http://www.citymayors.com/gratis/uk\\_topcities.html](http://www.citymayors.com/gratis/uk_topcities.html) and [www.pcgraphics.uk.com/list2.htm](http://www.pcgraphics.uk.com/list2.htm)

<sup>xiv</sup> Compiled data from <http://www.gazetteer.de>

<sup>xv</sup> Compiled data from [http://www.un.org/pubs/cyberschoolbus/infonation/e\\_information.htm](http://www.un.org/pubs/cyberschoolbus/infonation/e_information.htm)

with a capital, or was included in the names/surnames lists. The list of names and surnames was compiled from the US census statistics and consisted of the 2400 most common first names (male and female) and the most common 2000 surnames in the country. The choice of US census data was made because of the wide multicultural communities existent in that country, a fact which allows the list to include common names that originate in many other countries.

This process of recognising keywords and inferring their meaning based on rules and lists is well-established practice in the field of Information Extraction. This practice is one of the two approaches generally available for solving the problem of machine text analysis and understanding, the other one being automatically trained systems that do not rely on hand-crafted rules and databases. Given the recommendations of Appelt and Israel [App99] on the choice of approach and based on the good availability of resource lists (e.g. name and place lists) and the lack of extensive training data (calendar entries), the manual approach was elected. Appelt and Israel provide in their work further explanation on the concepts of Information extraction for the interested reader.

The principle behind the operation of this module is to take the XML generated by the calendar exporter and split the title, location and notes fields in words. These are then examined against the identifier database in order to determine their nature, in order to help formulate appropriate queries for these keywords. The original XML is then expanded to encompass the keywords found (if any). The following image depicts the XML expansion caused by the keyword generator module:

```
<appointment id=0>
  <ap_title>(data)</ap_title>
  <ap_loc>(data)</ap_loc>
  <ap_notes>(data)</ap_notes>
  <keyword id=xx, category=yyy>(data)
  </keyword>
  ....
  <keyword id=xx, category=yyy>(data)
  </keyword>
</appointment>
```

**Figure 15: XML expansion caused by the Keyword Generator module (new statements in blue)**

The relevant code for this module can be found in:

Filename	Line	Function Name
DeSnagDlg.cpp	701	Thread2()

The relevant keyword databases can be found in:

Subdirectory	File Name
Rules	Birthdayid.txt
Rules	Classid.txt
Rules	Generalid.txt
Rules	Meetingid.txt
Rules	Miscid.txt
Rules	Names.txt
Rules	Reminderid.txt
Rules	Socialid.txt
Rules	Surnames.txt
Rules	Titles.txt
Rules	Travelid.txt
Rules	Worktaskid.txt

## **3.6 Formulating Queries (Keyword analyser)**

### **3.6.1 Theoretical Design**

With the appropriate keywords identified by the Keyword Generator module, the next step the system should take is to generate appropriate queries for these keywords. Sometimes a keyword on its own might be a good candidate for a query. However, in order to obtain information that is closer to the needs or preferences of the user, the keyword will have to be combined with other keywords to form longer queries. Therefore a separate database of additional keywords for the formulation of extended queries will be necessary.

To be able to adapt the system to the user's preferences and needs, it becomes apparent that all additional keywords will need to be weighted in order of importance to the user. Because some of the additional keywords that will initially be provided might be irrelevant to the user's context, the system should be able to either negatively weigh these, or increase the weight of additional keywords for which the user was presented queries that retrieved "good" documents, while leaving other weights intact. A combination of positive and negative weighting can also be used.

One approach to the problem of constructing an additional keyword database would be to create a list of potential associations for each keyword contained in the identifier database. In this manner, the level of relevance between identifiers and additional keywords would be high, although this would come at a considerable cost. The costs of this approach would firstly be the large degree of duplication of data, as a keyword might be relevant to a multitude of identifiers. Secondly the construction of appropriate additional keyword lists for every single keyword would have to be done manually and there is no guarantee that every single identifier would have an additional keyword to reflect all context scenarios. Finally, because of the specificity of the additional keyword associations, a sudden or temporary change of context for the user might be completely ignored or take a long time to adapt to, which is of course undesirable.

An alternative approach would be to cluster the additional keywords into smaller databases, which will reflect the association of each keyword category, rather

than each keyword individually, with additional keywords. The term of *clustering* is borrowed from the Information Retrieval field, where it is defined as follows:

*'...We define the organisation as the grouping together of items (e.g. documents, representations of documents) which are then handled as a unit and lose, to that extent, their individual identities. In other words, classification of a document into a classification slot, to all intents and purposes identifies the document with that slot. Thereafter, it and other documents in the slot are treated as identical until they are examined individually. It would appear, therefore, that documents are grouped because they are in some sense related to each other; but more basically, they are grouped because they are likely to be wanted together, and logical relationship is the means of measuring this likelihood...'*<sup>xvi</sup>

This is a more natural step to take, considering that calendar entries are already clustered into categories by nature. Therefore web queries will be made by combining the original keyword (category identifier) with either all or the top  $n$  additional keywords that are associated with the category. In fact, as the system becomes increasingly accustomed to the preferences the user, the user-defined  $n$ -sized window of additional keywords ranks should reflect the true preferences of the user by returning a smaller amount of keywords within the same window.

The additional keywords can initially be set to have the same score, which would indicate an equal opportunity for each additional keyword to rise up in the ranks and therefore reflect a user's preference, independently of that preference's commonality. Alternatively, additional keywords can all be given different initial scores which would reflect the general preferences and assumptions that could be made for an average user. This would impede the training process for users that have very particular requirements, although for the majority of the users this method should provide adequately promising performance even from the early phases of use.

---

<sup>xvi</sup> HAYES, R.M., 'Mathematical models in information retrieval', in *Natural Language and the Computer* (Edited by P.L. Garvin), McGraw-Hill, New York, 287 (1963).

### 3.6.2 Implementation

The creation of separate database files for each calendar entry category formed the basis of the implementation. Furthermore, because some additional keywords could form sub-clusters of their own (e.g. hotel, bed&breakfast and hostel are all types of accommodation), these were separated from the master lists. This separation was also made in order to allow master categories to have access to common elements rather than have to duplicate entries in both.

Each additional keyword file was populated with a list of additional keywords and their associated scores, using a simple XML structure as depicted below in figure 16. Distinctions are made between terms (additional keywords) and categories, i.e. further lists of additional keywords that the system has to look into. The additional keywords and the additional categories are both weighted.

```
<search>map
  <s_type>term</s_type>
  <s_rank>0.94</s_rank>
</search>
<search>accommodation
  <s_type>category</s_type>
  <s_rank>0.94</s_rank>
</search>
```

**Figure 16: Sample additional keyword entries.**

In order to obtain additional keywords for each category, three different methods were used in conjunction:

- Interviews where people were asked to give details of typical searches they might have performed for calendar entries of each category (during the calendar entry sample collection)
- Calendar entry samples were given to independent subjects who were asked to produce as many web queries as they could for each entry sample (see section 4.3)
- Google's keyword suggestion tool<sup>xvii</sup>, where category identifiers were entered and a list of potential associated keywords was returned.

---

<sup>xvii</sup> Google Adwords keyword suggestion tool:

The latter tool by Google is worth special mention. It is mainly designed to help businesses who want to advertise with Google so they can get their advertisement displayed along with the results of a web query. By associating each advertisement with keywords, Google ensures that the advertisement will show when a related query has been submitted. The tool proposes further keywords that a business might like to consider by providing a list of *“popular synonyms and related phrases, based on billions of searches”*.

Although obtaining further meaningful additional keywords that were not covered by the interviews was not possible for the majority of category identifiers, once category where we found the tool extremely useful was the travelling category. The names of the 21 largest cities of the world were inserted the city names in the google keyword suggestion tool, to obtain a list of keywords (searches) for each city, under the assumption that there would be more searches for these cities than any others. The searches that were common to all these cities were grouped in categories, which formed the basis for the additional keywords list for travelling to locations.

The keyword analyser module looks into the expanded XML that has been formulated by the keyword generator module, in order to find the identified keywords and associate them with appropriate relevant keywords. It proceeds into forming the following two types of web queries:

- Original keyword (or keywords, e.g. name+surname) only
- Original keyword + additional keyword

The system can currently be set-up to fetch the top N rated queries for each user, according to their “s\_rank” value (fig. 16), although for the main experiment (see section 4.5), the amount of generated queries was limited to the top 5 rated items. When two or more queries happen to be ranked equally, they are both included in the generated query list. Once the process of query formulation is complete, the keyword

---

<https://adwords.google.com/select/main?cmd=KeywordSandbox>

analyser further expands the XML structure created by the previous two modules. This is accomplished by adding the search. The extended XML is shown below:

```
<appointment id=0>
  <ap_title>(data)</ap_title>
  <ap_loc>(data)</ap_loc>
  <ap_notes>(data)</ap_notes>
  <keyword id=xx, category=yyy>(data)
    <search>(data)
    </search>
    ...
    <search>(data)
    </search>
  </keyword>
  ....
</appointment>
```

**Figure 17: XML expansion caused by the Keyword Analyser module (new statements in blue)**

The relevant code for this module can be found in:

Filename	Line	Function Name
DeSnagDlg.cpp	1146	Thread3()

The relevant keyword databases can be found in:

Subdirectory	File Name
Keywords	Birthdaysearches.txt
Keywords	Classsearches.txt
Keywords	Generalsearches.txt
Keywords	Meetingsearches.txt
Keywords	Miscsearches.txt
Keywords	Namesearches.txt
Keywords	Remindersearches.txt
Keywords	Socialsearches.txt
Keywords	Travelsearches.txt
Keywords	Worktasksearches.txt
Keywords\subcategories	Accommodation.txt
Keywords\subcategories	Embassy.txt
Keywords\subcategories	Entertainment.txt
Keywords\subcategories	Localtravel.txt
Keywords\subcategories	News.txt

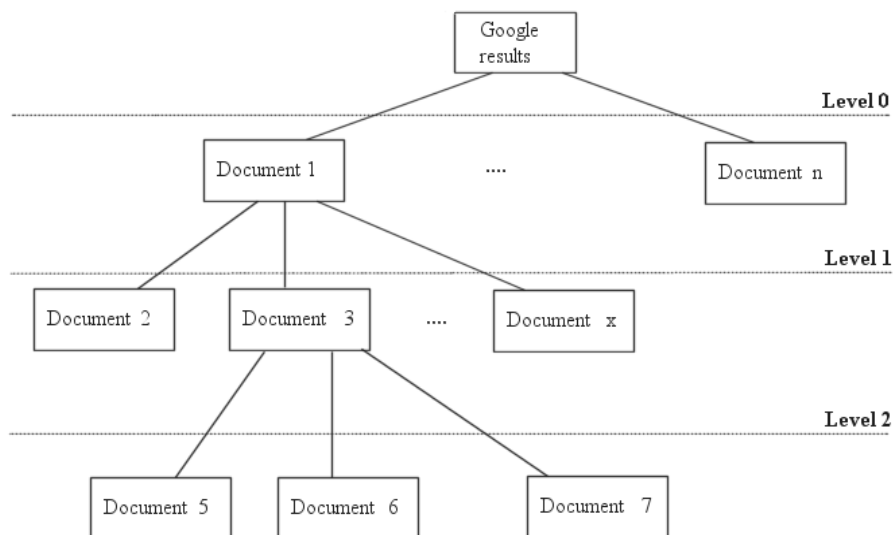
Keywords\subcategories	Travel.txt
------------------------	------------

### 3.7 Pre-fetching web documents (Web Searcher)

#### 3.7.1 Theoretical Design

The previous module is responsible for formulating the web queries that will be used in order to pre-fetch documents that are relevant to the calendar entry from the web. The next logical step in the sequence of operation events for the system is to submit these queries into one or more search engines on the Internet, which in turn should be able to retrieve links to relevant documents.

Those links should be followed in order to retrieve the documents. From within the documents, further links can be followed to documents within the same web site, or to documents that are external to the site. If those external documents were fetched and analysed, the system would encounter yet more links, which could also be retrieved. The retrieval process might be likened to a multi-tier tree, where a document from the original query can be considered a root, further linked from it are second-level nodes and links to further documents from these form edges that lead to even lower-level layers of nodes.



**Figure 18: The multi-tier retrieval tree**

It becomes apparent that the system could be locked into a fetching mode, which will exponentially increase the number of documents retrieved, without guarantees that all of the lower-level nodes will be relevant to the original query. In fact, it is logical to expect that the degree of relevance will reduce with the increase of the depth of the retrieval tree. Considering Jansen's observation that only 50% of the original web documents retrieved from web queries are classed as "interesting" by users, the lack of any guarantee of relativity of all the supplementary retrieved documents is apparent. For this reason a control mechanism should exist, which will either limit the amount of levels the retrieval tree can have, or be able to intelligently predict whether a linked document might be worth pre-fetching. Based on the pre-fetching methods researched earlier for desktop computers and proxy caches, it becomes obvious that such decisions have to be based either on data from server request statistics or an analysis of the target document, in which case it needs to be fetched so it can be analysed. Such an implementation technique as the latter might be considered an interesting exploration for the adjustment of performance levels, however it is beyond the scope of this investigation as its development would warrant altogether a separate PhD. Therefore the logical solution for this problem under these circumstances would be to implement a mechanism that would limit the depth of the retrieval tree levels for each document.

Further considerations would have to include the type of Internet content that the system should be able to store. Further from HTML documents which contain formatted text, other elements such as images, .pdf or .doc files, audio and video, might be desirable to have. All of these however place a considerable stretch of the memory limitations of current handheld devices. Therefore an interview of users should be able to indicate exactly what kind of content should be prioritised for pre-fetching. The results of this interview showed an overwhelming preference to text and images, while other media such as .pdf, audio and video were considered largely irrelevant, unless a user was specifically looking for such types. Further details about this interview can be found in chapter 4.

### 3.7.2 Implementation

Of the search engines available today on the Internet, the most successful is the Google engine<sup>xviii</sup>. This engine was chosen for the submission of the formulated web queries. An observation by Jansen and Spink [Jans03] indicated that the majority of users (80%) only view between ten and twenty results for each query, i.e. no more than one or two pages of results. Because the Google search engine can retrieve thousands of relevant results, a choice was made to limit the returned results to the top ten, as ranked by Google. It was felt that the choice to restrict the results to the top ten was justified, bearing in mind the memory constraints that are present on handheld devices. Furthermore, because Google is widely considered to be the best Internet search engine currently available, the choice was made not to submit the queries to other search engines. The comparison between the performance of Google and other search engines was not of interest for the purposes of this thesis and beyond this, such an attempt would greatly increase the amount of retrieved documents without any expected significant increase in the amount of retrieved relevant documents. This would place a significant and unnecessary burden on the subjects of the main experiment. Although collective filtering could be applied to perhaps include only the top K results from a number of search engines, it should be kept in mind that the optimisation of the cache, while desirable, was beyond the purposes of this thesis. Thus no services other than Google were employed in this instance.

The Google result page is trimmed from all unnecessary HTML elements in order to remove graphics, advertisement and unnecessary links. Subdirectories are created to store all the documents of the desired tree depth levels. The retrieval procedure then works for each document of each tree level, including the Google result page, in the following manner: Firstly, the document is parsed and searched for links. Once a link is encountered, the linked document is retrieved and stored in the appropriate subdirectory. The link in the document under analysis is changed to reflect the local relative URL of the newly-fetched document. The process continues

---

<sup>xviii</sup> <http://www.google.com>

until no further links can be found in the document, in which case, the system moves on to the next document of the same depth level. In this manner, one can envisage a horizontal breadth-first traversal of the document traversal tree, in order to generate the lower levels of pre-fetched document. The entire process ends when the user-specified retrieval depth has been reached.

Once the system is finished retrieving the top ten documents for each Google page, further transformation of the Google page is made. A second-level parsing is performed in order to separate the document titles, summaries and URLs contained therein, and store them in an XML structure. This XML structure will later be passed to the handheld device, along with the relevant documents, and will be used to display the retrieval results to the user. This XML structure is depicted below:

```
<document>
  <url>(data)</url>
  <title>(data)</title>
  <description>(data)
</description>
</document>
...
```

**Figure 19: The XML version of the Google pages**

The module also expands the XML created by the previous modules, by adding the location of the relevant XML-based Google results page for each search. This is the final step before passing the retrieved documents and the accompanying XML structures on to the handheld device. Therefore the final version of the desktop component's XML, used to communicate its generated output to the handheld component, has the following structure:

```

<appointment id=0>
  <ap_title>(data)</ap_title>
  <ap_loc>(data)</ap_loc>
  <ap_notes>(data)</ap_notes>
  <keyword id=xx, category=yyy>(data)
    <search>(data)
      <gpage>(data)</gpage>
    </search>
    ...
    <search>(data)
      <gpage>(data)</gpage>
    </search>
  </keyword>
  ....
</appointment>

```

**Figure 20: XML expansion caused by the Keyword Analyser module (new statements in blue)**

The relevant code for this module can be found in:

Filename	Line	Function Name
DeSnagDlg.cpp	1686	Thread4() This module retrieves the Google pages
DeSnagDlg.cpp	2067	Thread5() This module retrieves the documents

## **3.8 *The handheld component***

### **3.8.1 Theoretical Design**

#### **3.8.1.1 Presenting Results**

Once the documents have been retrieved and sent to the handheld device, a separate software component therein should be responsible firstly for displaying the documents to the user, and secondly for observing the interactions of the user with these documents.

It is clear that for the first task, the design needs to consider the physical characteristics of the handheld device and especially the constraints placed by the dimensions of the device screen. Taking in mind Nielsen's observations that users tend to dislike long pages which require lots of scrolling [Niel97], an implementation of the results browser should be considerate of this natural tendency and contain facilities that will allow the users to minimise the scrolling needed. A collapsible tree-structured list of calendar entries, identified keywords, searches and retrieved document titles/summaries is potentially a good way of addressing the scrolling problem. Unfortunately, due to the nature of web pages, scrolling to view their content is inevitable. However the quality of the browsing is beyond the system's control and will be fully dependent on the device's integrated web browser. As this issue is beyond the scope of this research, no attempt to write a dedicated web browser was made.

#### **3.8.1.2 Processing relevance feedback**

Observing the interactions of the users poses several questions that need to be answered. Firstly, which actions of a user do reflect interest and therefore should be monitored? Secondly, once such actions are identified, do they all indicate the same amounts of interest or should their importance be weighted with different measures? Based on previous related research, an instinctive negative answer to the last question is probably the right one. Indeed, as mentioned previously in chapter 2, it has been shown that not all kinds of interaction can provide dependable implicit information on relevance.

On the handheld device, a log is kept of the user interactions with the content. Based on this information, an attempt should be made to judge the relevance of a given keyword from the knowledge base to the user's context and determine which of these are likely to be wanted as part of a query in the future. Given the collapsible presentation structure, a log can be kept for:

- The viewing of the document index for a given search
- The viewing of the summary of a given document
- The viewing of the document
- The amount of time spent on a document that has been opened
- Any explicit feedback rating that a user might provide for the document.

It would be sensible to assume initially that these measures are in order of importance from least to most.

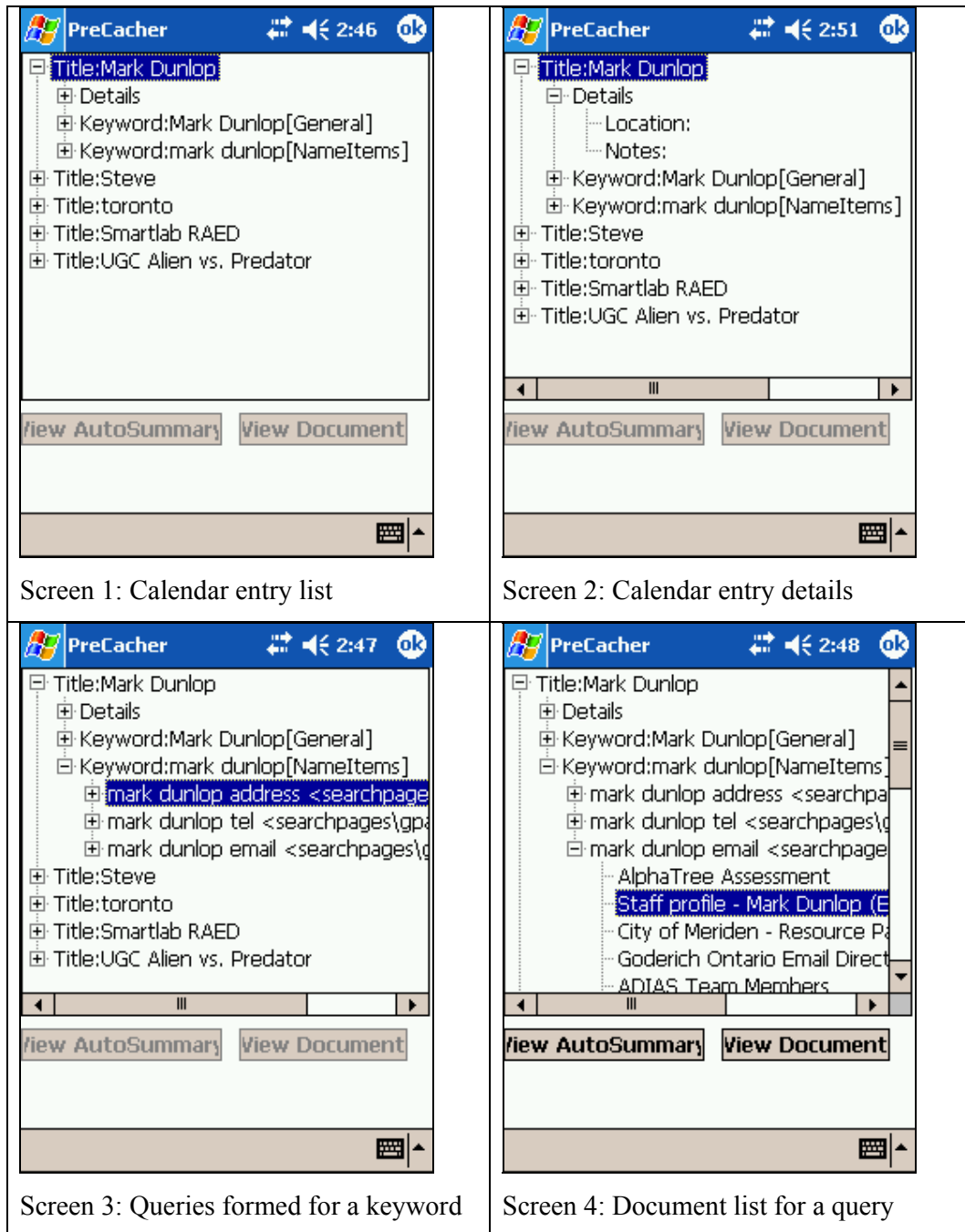
The incorporation of further heuristics, such as pointer movement, highlighting of text, bookmarking and scrolling in the document, would also be desirable. However, given the implementation was on a PocketPC platform, programming Pocket Internet Explorer to trap such behaviours was not feasible. In addition, some of these heuristics (e.g. scrolling) would not be reliably applicable, given the lack of previous research on small screen devices for such measures.

### **3.8.2 Implementation**

Upon loading, the handheld component examines the XML structure passed to it and the XML-formatted Google pages, in order to load the necessary details for presentation to the user. The pre-fetching activity details are presented using a collapsible tree structure list, which gives details of the appointments, keywords identified, web queries formed and document titles of each search (see figure 21).

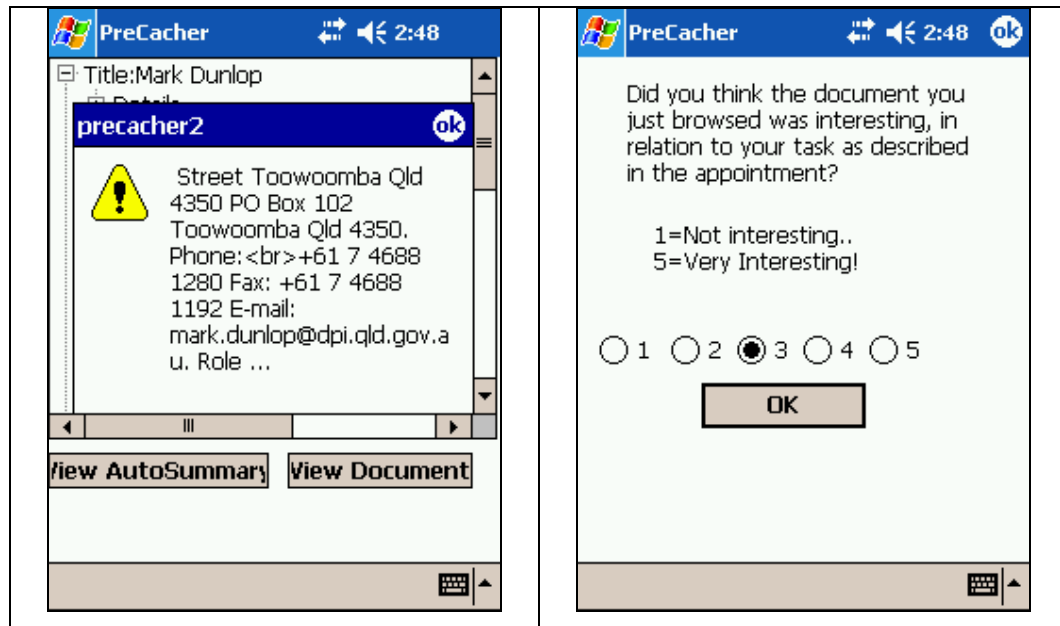
When a document title is highlighted, the user is given the option via two buttons to either launch a descriptive summary of the document, or open the document with Pocket Internet Explorer. The system monitors the collapse of the document list for a web query, the viewing of a summary, the launch of Pocket Internet Explorer and the duration for which it is active, i.e. in the foreground. It is assumed that the user will be reading the document for that time. Also, when Pocket

Internet Explorer exits and the user returns to the handheld software that is running in the background, an option is given to explicitly rate the quality of the document just read on a scale of 1-5 (figure 22).



**Figure 21: The mobile User Interface, showing the collapsible tree list. Each keyword under a given calendar entry (title), can be expanded to show the queries formed for it. For each query, a list of retrieved documents is provided.**

To implement the document summary function, the short document narrative that Google provides directly under the document title in its results page is used. This is displayed as a pop-up dialog box to the user, upon request, therefore imposing an interaction cost on its viewing (see figure 22 and section 4.5.3).



**Figure 22: The auto-summary function and the explicit relevance feedback screens.**

The observed user interactions are logged using an XML structure, which is passed later on back to the desktop component for updating the additional keyword details. The XML structure is depicted through a sample below:

```
<search>
  <name> accommodation</name>
  <expanded>0</expanded>
  <doc>
    <url>"level1\\113615.htm"</url>
    <opened>0</opened>
    <time>0</time>
    <summary>0</summary>
  </doc>
  ....
</search>
```

**Figure 23: XML structure for logging user interactions. Binary data (0,1) are used for most of the fields except time, which is measured to the nearest second.**

The relevant code for this module can be found in:

Filename	Line	Function Name
Precacher2.cpp	55	OnInitDialog() Displays and loads data into the user interface, creates XML to pass back to desktop
Precacher2.cpp	287	OnButton1() Launches Pocket IE, monitors time elapsed and writes relevant information in XML file
Precacher2.cpp	550	OnButton2() Launches summary view, writes relevant information in XML file
Precacher2.cpp	741	OnItemExpandedTree1() Monitors the expansion of a query's document list, writes relevant information in XML file

### 3.9 Updating additional keyword weights (Weight Updater)

#### 3.9.1 Theoretical Design

Having monitored and kept a log of user interactions with the system, the handheld device should somehow communicate this information back to the desktop component, which is responsible for making the pre-fetching decisions. In order for this task to be accomplished, the XML data structure as described in the previous section is analysed by the desktop component, in order to calculate the appropriate weight modifications for each retrieved document and subsequently each keyword.

For the interest indicators mentioned in section 3.8.1, appropriate weights and a method to combine their values to form a score for each keyword should be devised. A definition of the importance  $I(k)$  of a keyword  $k$  can be given as follows:

$$I_{(k)} = w_{E(k)} + \sum D_{(i)}$$

where  $w$  is the weight associated with a viewed document index for the keyword  $E(k) \in \{0,1\}$ , and  $D_{(i)}$  is the importance of each document  $i$  that has been retrieved for keyword  $k$ . Furthermore, a definition of  $D_{(i)}$  is as follows:

$$D_{(i)} = \alpha O_{(i)} + \beta S_{(i)} + \gamma F_{(i)} + \varepsilon T_{(i)}$$

where  $O(i)$  indicates the viewing of document  $i$ ,  $S(i)$  indicates the viewing of its summary,  $F(i)$  is the explicit feedback given by a user to the document,  $T(i)$  is the time spent reading the document and  $\alpha, \beta, \gamma, \varepsilon$  are the respective weights for each of these measures.

The weights  $w, \alpha, \beta, \gamma$  and  $\varepsilon$  take positive or negative values. A mechanism that would promote the appearance of preferred keywords in the queries was desired. However, the negative marking for undesirable keywords would allow firstly the improved promotion of desirable keywords, but also, should one of the latter become undesirable, due to perhaps the change of context of the user, it would not take too long for it to start disappearing from the queries.

### 3.9.2 Implementation

Experimentation was made with several weights for the relevance feedback system and the conclusion was reached that it is not only important to consider the relationship between the weights, but also the bias towards positive or negative marking. Current research by Jansen (see Chapter 2) shows that users will view only two or three (on average) documents per web query and the vast majority will visit at most 2 pages or results (approximately 20 results in all). The same research shows that an estimate of 50% of documents viewed from these results are expected to be relevant to the query. Therefore it was decided that a bias of approximately 1:7 in favor of positive marking was reasonable. Because the queries would not generate more than 10 documents each, this means that two documents with a positive overall rating will indicate a successful and relevant query was made.

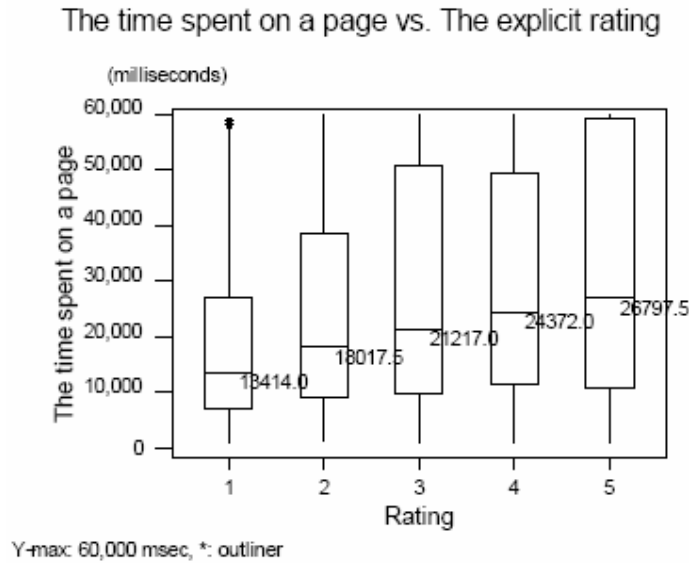
The weights that were used were as follows:

Weight	Value
$\alpha$	-0.03 (document not opened) +0.03*7 (document opened)
$\beta$	-0.021 (summary not viewed) +0.021 (summary viewed)
$\gamma$	-0.15 (feedback=1) -0.075 (feedback=2) 0.00 (feedback=3) +0.15*6 (feedback=4) +0.15*7 (feedback=5)
$\epsilon$	Inconclusive

**Table 7: Keyword score adjustment weights as used in the system implementation**

There was some confusion as to the weights  $\epsilon$  which should be used for association with the reading time interest indicator. As mentioned in chapter 2, previous research highlights a possible correlation between average reading time and relevance of a document. However, all previous research had been conducted on desktop computers, where screen readability issues were not as much of a problem. The initial inclination was to apply a tiered weighting system according to the reading time averages reported by Claypool [Clay01]. The weights for each metric were chosen on

an ad-hoc basis, mainly through consultation with the main experiment's control group subjects and their observed interactions with the system (see section 4.5.2).



**Figure 24: Claypool's findings on relevance and reading time correlation**

From the above figure, it becomes immediately obvious that there is quite some overlap between the distributions of reading times for each explicit rating, something which makes it difficult to accurately infer document relevance from reading time alone, in most cases. After experimentation, described later in chapter four, it was decided also that these average reading times were completely irrelevant to the handheld device environment. In fact, it was discovered that no distinct correlation could be made between reading time and document relevance on a handheld device, so the decision was made to abandon this metric completely.

The relevant code for this module can be found in:

Filename	Line	Function Name
DeSnagDlg.cpp	2352	OnButton11() Loads interaction XML, calculates scores for each additional keyword and updates additional keyword database

## Chapter 4

### Experimentation with users

## **4 Experimentation with users**

### ***4.1 Introduction***

In this chapter, a description will be made of the experiments and observations that were undertaken, in order to arrive to the full implementation of the system described in chapter 3. The first three sections describe the initial experiments conducted, while sections 4.5 discusses the final experiments on the full system and 4.6 shows another test that was carried out, inspired by previous findings of this research.

### ***4.2 Analysing Calendar contents and Usage***

#### **4.2.1 Calendar content analysis**

The starting hypothesis behind this thesis is that adequate information can be found in user calendars, which will allow the pre-caching of useful Internet content for these users. In order to establish whether such a hypothesis is indeed true, it is necessary to begin the investigation by examining real world calendars and their contents.

A group of twenty volunteers from the University of Strathclyde agreed to provide original, unedited entries from their calendars, for the purposes of this research. The users all worked within an academic environment, although not all of them were academics. The group included lecturers, students, researchers and secretaries. Under the agreement for full confidentiality, unrestricted access was allowed to their calendars, in order to extract entries. Having taken a sample of 161 original entries from both electronic and paper based calendars, several important observations were made on the nature of calendar entry contents, which are discussed below.

Firstly, it immediately became clear that the calendar entries tend to fall within specific categories. The categorisation of the entries was done manually, with guidance from the original entry authors, where ambiguity made it necessary. This categorisation comes as a confirmation of the findings of previous research, as described in Chapter 2, and especially those of Kincaid and Dupont [Kinc85], who discovered that users tend to use their calendars mainly to keep a record of meetings,

appointments, events, travel, reminders, notes and as “to do” lists. This, in turn, suggests that there is a consensus to the items that form appropriate calendar entries and that there is a common mental representation model for the organization of these entries amongst humans. There is a slight variation in the categorisation of entries that was made with these new findings, although this can be considered natural, due to the particularities of the academic environment. However, the categories with the highest frequency are the same as those in previous research. The table below shows the categories and entry frequencies as derived from this research’s sample.

Category	Frequency
Meeting (Group)	53
Meeting (another person)	25
Reminder	18
Travel	13
Social event	13
Work task	10
Class (to attend)	10
General Task (to-do)	7
Miscellaneous	7
Birthday	5

**Table 8: Calendar entry categories and their frequencies**

Further to the distribution of calendar entries, table 9 presents short descriptions of each category, in order to show exactly the type of entry each one contains. It is important to stress here that these categories reflect the opinions of the calendar users, who are highly familiar with the context of their entries. One can observe that there is some overlap between the calendar entries, for example, Birthday could be considered a subset of Social. However, where such overlap is maintained, it is because there was a strong indication from the users that such a low-level category is significant and should exist separately from its high-level parent. A revised alternative category grouping might be considered, by grouping the categories Class, Work Task and General Task under a more general category called TASK, and also the categories Social and Birthday, under SOCIAL.

Category	Description
Birthday	Indicates someone's birthday
Class (to attend)	User has to attend a class (either as a student or lecturer)
General Task (to-do)	General tasks to complete (non-work related), such as buy an item or email someone
Meeting (group & personal)	A meeting that has to be attended or an appointment with someone
Miscellaneous	Unclassifiable items
Reminder	A reminder that an event is happening, such as "Mary is off sick" or "Exams start today"
Social	A social event, such as dinner or going to the movies
Travel	User has to travel to some destination out of their habitual location
Work – related task	A task to do that is related to the user's work, such as "write a report" or "mark exam scripts"

**Table 9: Calendar entry category descriptions**

As mentioned earlier in Chapter 3, the actual word contents of the calendar entries were also analysed, in order to obtain the category identifiers for each of the categories. Most of the entries were easily categorised as they obviously fell under certain categories (e.g. "Meeting with Mark"). For the few entries that were ambiguous, the authors were consulted for the disambiguation, either during collection or by subsequent communication. The relevant derived category descriptors can be found in the "Rules" subdirectory on the accompanying CD-ROM (see end of section 3.5). Further rules derived from the lexical/syntactical analysis of the calendar contents are described in more detail in Chapter 3.

From the analysis of calendar entry wording and structure, it was determined that users typically fall into three categories, according to their style of input. There are users who write very little in their entries: they use these un-meaningful (to the foreign eye) notes as reminders for all the details they hold in their head. They form

approximately 60% of the number of calendar users interviewed. The second category is those who tend to be very organised and are keen to include many details in their entries. These are, however, a very small percentage of around 10%. The remaining 30% tend to include only very important information in their entries, but normally this information is sufficient for the foreign observer to interpret successfully the nature of the entry.

This important observation highlights the necessity to formulate rules for inferring additional information from hastily written entries (e.g. it has been found that when an entry's subject is the name of a person, it is 90% certain that the user will be meeting that person) and for determining which words and in which context can become keyword candidates.

The hypothesis of the thesis should perhaps then be re-worded. From the analysis of calendar entries, it is apparent that there is indeed a plethora of information that can be found about a user and their activities. The real question is, can we infer additional details relevant to this information to match the users, in order to support both the “organised” and the “disorganised” uses, which form the majority in our sample?

#### **4.2.2 Calendar Usage Questionnaire**

Simultaneously with the collection of the calendar entries, users were asked to voluntarily complete a questionnaire. For the purposes of establishing general patterns in the use of calendars, the questionnaire was designed to contain the following questions:

- Do you use your calendar to remind you of things to do, instead of just noting appointments and important dates?
- Do you synchronise your calendar with other PIM devices or do you tend to keep more than one calendar?
- Would you be willing to change your style of input (wording) if you knew it would help a program pre-fetch related internet sites for your PIM device (handheld, mobile)?

- Are there any internet sites that you visit on a regular basis (e.g. daily, weekly)? How many approximately?
- How big a “grace” period would you give to a program if it needed to adapt to your personal needs? Would you expect the program to work perfectly straight from the box?
- Please rate your internet content needs (5-most important, 0-least important):

Documents (html)	
Documents (.pdf, Word etc.)	
Images	
Video	
Sound	

- Does your calendar contain information about your personal/social life or is it used for business only?

All twenty users took the additional time to complete the questionnaire, and the findings are presented below.

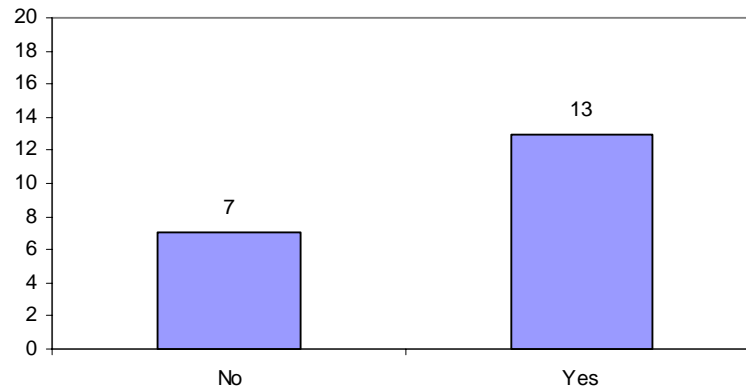
#### **4.2.2.1 Question 1**

“Do you use your calendar to remind you of things to do, instead of just noting appointments and important dates?”

In a hypothetical scenario, a user might explicitly desire to visit websites or retrieve other internet content as part of their everyday activities, for example, one might need to make a note to visit a given site and read an article more carefully. This is something a person would add to a to-do list, rather than to a calendar and most PIM software has facilities that accommodate, but also distinguish, between these two.

This question has the intention of establishing whether users are familiar with associating calendar entries with to-do items and to what extent do they blend these distinct categories of input items.

**Question 1:**  
**Do you use you calendar to remind you of things to do**  
**instead of just noting appointments and important dates?**



**Figure 25: Question 1 responses**

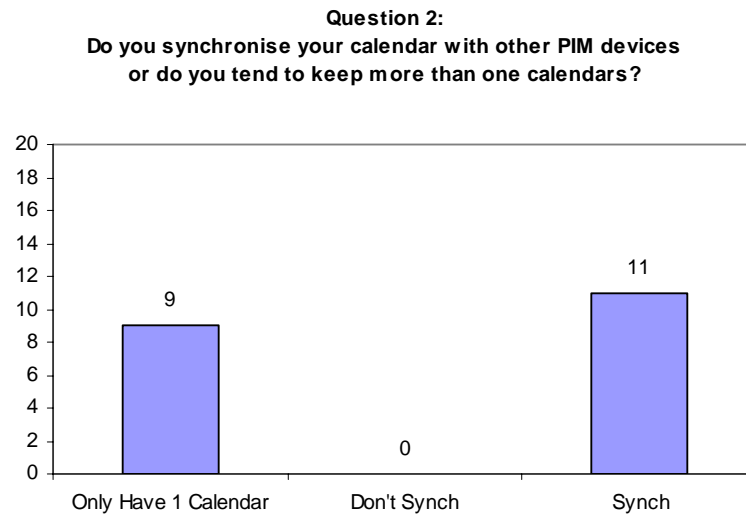
From the above table, it becomes apparent that the majority of users (64%) tend to insert both types of items into their calendars and therefore are likely to have reminders about things they need to do in their calendars, rather than on a separate list.

This should indicate that users are likely to accept the concept of being able to use the calendar as a means of communication with the software agent. It also indicates that the software agent should find information, on which it can act with a relatively high level of certainty. Certainty, in this case, refers to the ability to guess whether the user would like some internet content related to the entry or not. Interestingly enough, the broader TASK category still only has a frequency of approximately 17%. This further supports the need for inferring details about other categories, since the instruction/task based TASK category where activity details are more specific does not hold the largest proportion in category distribution.

#### **4.2.2.2 Question 2**

“Do you synchronise your calendar with other PIM devices or do you tend to keep more than one calendars?”

This question intends to determine whether users are habituated to the concept of synchronising their calendars, in the case they have more than one.



**Figure 26: Question 2 responses**

Figure 24 shows that all the people who do have more than one calendar (or more than one PIM device), synchronise them. This is a good indication that the software agent should be able to find current and valid information when it opens the user's calendar for information extraction.

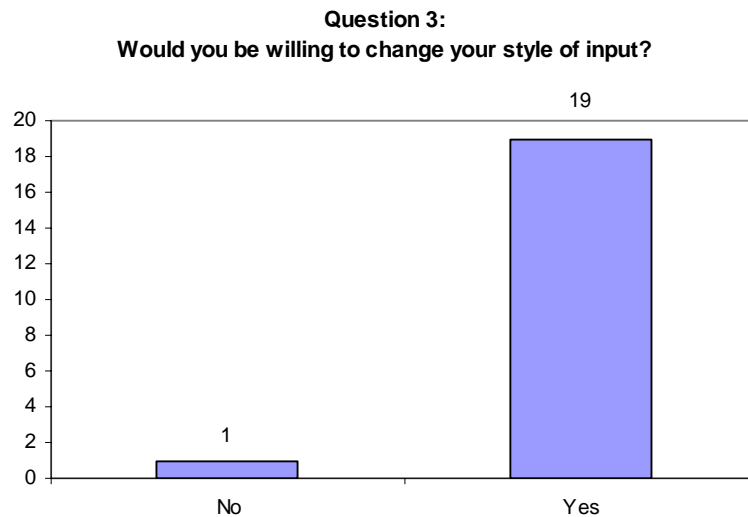
#### **4.2.2.3 Question 3**

“Would you be willing to change your style of input (wording) if you knew it would help a program pre-fetch related internet sites for your PIM device (handheld, mobile)?”

Human behaviour does not change easily and users are not likely to change their habits, unless of course they perceive that something useful will be gained from this change. In this survey, most of the appointments written down contain information and are syntaxed in such a way that their meaning is apparent almost always only to the user. This is understandable, as a calendar is obviously something personal, which users do not actually tend to share with others. However, even in the case of

one user who does share his calendar with others on-line, there are appointments that are worded in a way that make sense only to him.

This question attempts to determine whether potential users of the software agent are willing to alter their input habits if they were to gain specific advantages from doing so. In the light of the previous findings, where it was established that “disorganised” users were the majority in our sample, this question becomes important in helping establish the tolerance of the users to a system that attempts to infer knowledge of their preferences.



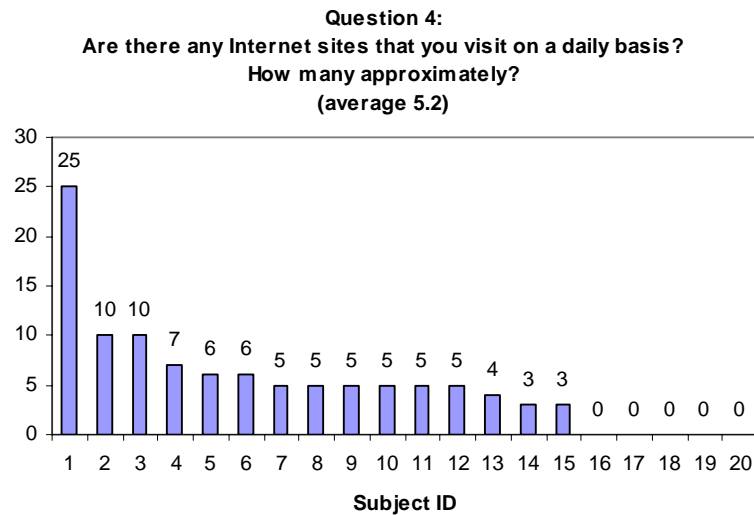
**Figure 27: Question 3 responses**

It is apparent that almost all users interviewed indicate a willingness to change writing behaviour if they were to gain all the advantages the software agent could offer. However, it would be interesting to observe at a later stage, exactly how many of these users would actually change and how many users would revert back to their old way of input, as persistent behaviour is only occasionally rewarded [Fers82]. Also it would be interesting to establish whether such a thing would happen because of improper behaviour by the software agent or by other factors.

#### 4.2.2.4 Question 4

“Are there any internet sites that you visit on a regular basis (e.g. daily, weekly)?  
How many approximately?”

The purpose of this question is to determine the approximate amount of content that users require that is not related to their calendar entries. If, for instance, a person likes to read a newspaper on-line every day, they are not bound to have any entries mentioning their desire to have this content in their calendar.



**Figure 28: Question 4 responses**

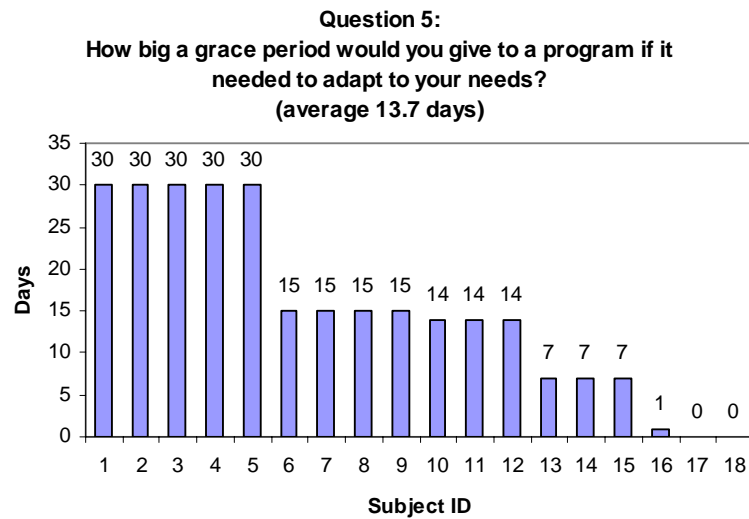
Out of the 20 users, a total of 5 (20%) stated they do not have any internet sites they visit regularly and would not therefore like to have their content available immediately. While this percentage sounds initially high, it must be stated that all of them are people from non-computing backgrounds who normally use the computer for purposes other than surfing the internet. It is expected that this percentage will fall drastically as more people are interviewed.

#### 4.2.2.5 Question 5

“How big a “grace” period would you give to a program if it needed to adapt to your personal needs? Would you expect the program to work perfectly straight from the box?”

It is expectable that a program that does not perform as advertised is bound to be eventually rejected and neglected by its users. The software agent proposed in this research is a piece of software which, by nature, is unable to perform to its maximum ability “straight out of the box”. It will need time to adapt to the users’ needs and preferences and it can be assumed that it will be susceptible to negative feedback.

For this reason it is highly desirable to establish a period of time, within which the agent should begin to perform at high efficiency levels. This period of time will also aid in the decision of how much “knowledge” should the agent have at the time it is given to a new user and an insight as to what levels of performance the positive and negative keyword weighting should be able to achieve.



**Figure 29: Question 5 responses**

It is interesting to observe that two users refused to give an estimate answer to this question, but stated that the time they would give would depend on the learning ability of the program. More specifically, it was stated that if there was a positive and clear indication that the program was indeed beginning to learn, then they would be prepared to invest more time in the program. A further two people gave an estimate but also stated that it could change according to the agent’s demonstrated ability to learn. These findings could be combined with earlier research by Ruvini

and Gabriel [Ruvi02], who found that as long as an electronic categorisation assistant achieves reasonable performance, users are prepared to tolerate errors from it. Their system for automatically classifying emails into user-defined folders, indicated a performance level of approximately 80%.

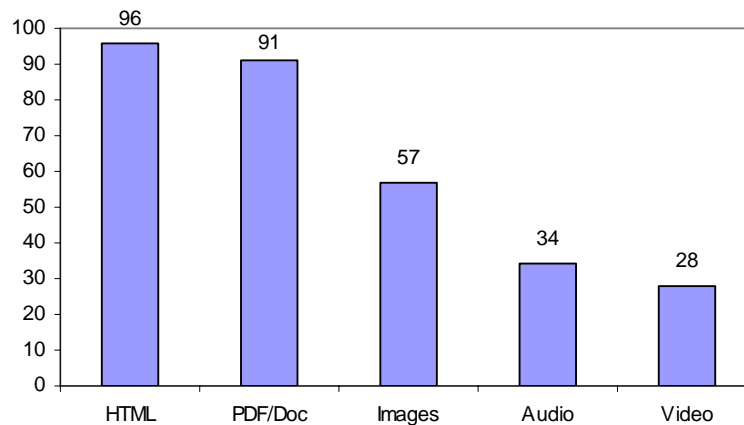
It could therefore be assumed that as long as the system described in this thesis achieves a performance level of 80% for the automatic categorisation of calendar entries, within the average timeframe of 14 days of usage, the users' acceptance of the system as a success would be highly likely. The overall performance of the system depends also on the quality of the queries which are formed, however, a pre-requisite for the formation of good queries, is the correct automatic categorisation. In subsequent experiments (section 4.4), it is shown that these conditions are immediately met with the implementation of the categorisation algorithms, as described in Chapter 3.

#### **4.2.2.6 Question 6**

“Please rate your internet content needs (5-most important, 0-least important):”

In this question, users were asked to indicate a preference on each of the five types of internet content most frequently available on the web today. This was in order to establish what kind of internet content and links should be followed in the program's default behaviour. Obviously the users should be able to specify that they want to store one or more specific types of internet content, however, for resource economy, the program should by default be able to store items that are of most importance.

**Question 6:**  
**Please rate your Internet content needs (0-5)**



**Figure 30: Question 6 responses (aggregate)**

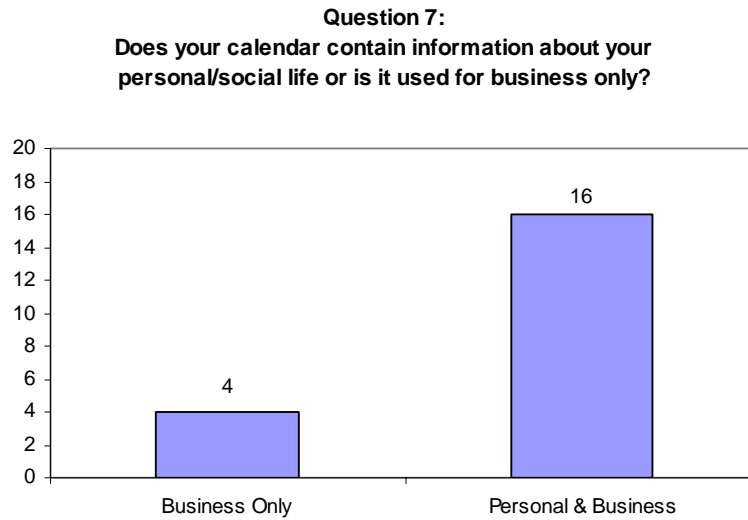
From this graph, it is obvious that, bearing in mind a maximum of 100 (20 x 5) points for each answer, HTML text takes precedence with an overall score of 96%. It also appears that other types of electronic document (such as Adobe's PDF format and Word documents) rank highly, with an overall of 91%. Further from text-based information, the total scores of the multimedia content is rather low. Visual information scores the highest (57%), followed by audio, with a score of 34%. The combination of visual and audio information (video) information scores the lowest mark with 28%.

It is interesting to see that the importance score of the textual information (html and other e-documents) is by 66% larger than the score of all the multimedia content combined. The obvious conclusion from these statistics, is that the emphasis should rely on pre-caching textual information and that images should only have a 60% chance of being desirable. This of course is a hypothetical maximum, as one should remember that a good proportion of images on web sites are cosmetic graphics or advertisement banners, something that the users would be highly unlikely to have great demand for.

#### 4.2.2.7 Question 7

“Does your calendar contain information about your personal/social life or is it used for business only?”

This question tries to establish whether the software agent will need to learn things about the users’ personal lives and whether such information can be retrieved through entries in their calendars.



**Figure 31: Question 7 responses**

From the results it would seem that users tend to keep information about their personal lives in their calendars, and it should therefore be feasible to obtain, at least some of it, implicitly. It also means that users would be interested in knowledge which is not only business-focused. It would be interesting to see whether users might begin to entrust their personal life affairs to the software agent by providing sufficient information for it to act on.

#### 4.2.3 Summary of Findings

The first piece of research regarded the nature of calendar entries, the wording and information that could be found therein and also the attitudes and behaviour of the

calendar users. The most important findings of this research can be summarised as follows:

- Calendar entries seem to fall under specific categories. While some of these categories might be perceived as subsets of other, broader categories, the actual separation is desirable by the calendar entry authors and might be helpful in retrieving more specialised and relevant documents for each.
- A plethora of information can be found in user's calendars, not only about their business activities, but also their personal lives. A calendar based pre-caching system should therefore be accommodative of all types of categories.
- The calendar entries tend to be short-worded and meaningful to perhaps only the author. The majority of authors prefer monolectic entries which act as reminders for the rest of the details to them. Therefore the need for inferring those additional details is apparent, in order to achieve effective pre-caching of documents.
- The information found in calendars can be expected to be current and valid, as users either synchronise multiple calendars or tend to keep only one.
- The pre-caching system should be able to adapt to a good extent to a user's preferences in an average of approximately 12 days from the beginning of use. However this average is not absolute and more time might be given to it by users, if an improvement in performance is perceived.
- The types of Internet content that should primarily be pre-cached is text and images. Other multimedia sources do not seem to be of much interest to the users, unless there is a specific requirement for these.

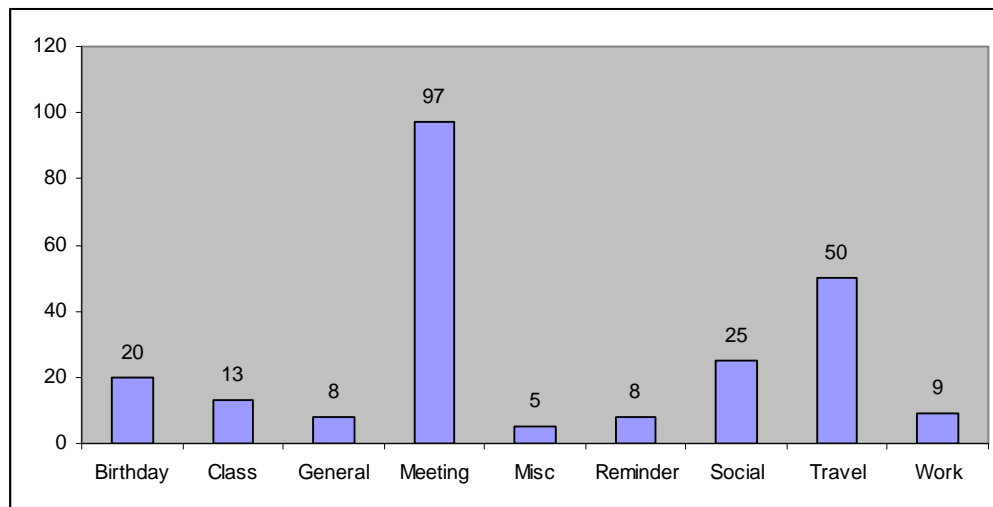
### ***4.3 Web search and query formulation behaviours***

In an attempt to capture the search behaviour patterns of users, when given calendar information as a basis to act upon, a survey was made with another ten persons, again from an academic background, although unrelated to the authors of the calendar entry sample. The unfamiliarity of the test group with the calendar entries would simulate the unfamiliarity of a software agent which would essentially be attempting

to perform the same task. This was a desirable characteristic of this experiment, as it would allow the setting of a performance benchmark, which the software agent should compete against. Another useful outcome of this experiment would be the resulting ability to add more keywords to the additional keyword database, as specified by real users, through the analysis of the queries they would provide. Finally, it was expected that the categories for which the users were more likely to formulate a search would be pinpointed. This way, special attention could be paid to the software agent's performance for these categories, as they should be regarded as highly important to the users.

The persons that partook in the experiment were given a subset of the original calendar entry collection, which contained 50 random entries. The entry categories were represented with the same frequency percentages as in the original collection and a few duplicate entries were also purposely included. This was done in order to help capture alternative ideas that users might have missed when encountering an entry for the first time and also to observe whether the comprehension of an entry might change or remain the same regardless. The test group were allowed an indefinite period of time to complete the task of formulating web queries based on the information contained in the calendar entries (see appendix 3 for the instructions given to subjects).

The following diagram presents the frequency of searches that were encountered for each category.



**Figure 32: Amount of searches formed manually for each category**

From this diagram, it is immediately obvious that the most important categories in terms of usefulness for web searching are Meetings, Travelling, Social and Birthdays (which again might be considered as a subset of Social). It would also appear that Meeting entry types are by far the most important ones to users, although an analysis of the searches proposed by the test group revealed an important fact: 42 out of the 195 total searches excluding travel, were actually related to travelling details mentioned in the entries. Out of these 42 searches, 33 were encountered in the meeting category searches. In relative terms, this means that one in three searches (34%) formulated out of a meeting entry, actually relate to the travel category instead. Under the light of this observation, it becomes apparent that the focus of the agent's performance should primarily be with the entries which contain travelling details.

## ***4.4 Automatically identifying entry categories***

### **4.4.1 Experiment set-up**

To assess the quality of the automatic categorisation and keyword identification rules that were established in previous stages of this research, a small application was written, that would read in a pre-compiled static list of entries and present them, one at a time, to the members of a test group. For each entry, each subject would be asked to assign a category to it, and to provide the level of confidence that accompanied their decision. To limit study time and user frustration the collection was sub-sampled by approximately 50%, giving a total number of entries of 100 per subject. The entries provided in the test collection were randomly selected from the original collection while preserving the category distribution. In the original collection some entries had two or more instances caused by recurring appointments – these were preserved in the test collection. The choice was made not to eliminate any duplicate entries, in order to observe the perseverance of the test subjects to their original perception of the category recurring entries belong to.

Category	%	Category	%
Birthdays	4	Reminder	11
Class (to attend)	4	Social	9
General task (to do)	6	Travel	8
Meeting (group & personal)	48	Work-related task	6
Miscellaneous	4		

**Table 10: The entry categories and their numerical representation in the test collection**

Sample screenshots of the actual test executable are depicted below. Figure 31 shows the entry categorisation dialog, with the entry details displayed on the top part. The user is obliged to select one choice from the Category radio button group and one from the Confidence group.

Figure 34 shows the dialog displayed when further classification information is asked from the user. The entry details are displayed again, along with the user's choices from the previous dialog. The user can enter further information in the box at the bottom of the dialog.

**Figure 33: Entry categorisation screen**

Please provide more information

Please could you provide the reason behind the categorisation of this appointment?  
Just enter a short explanation in the box and click OK!

Title: ms. Janet Deale

Location: outside of the class

Notes:

Selected Category: workt

Confidence: 4

OK

**Figure 34: Explanation screen**

When the user clicks on Next in Figure 33, the application compares the category allocation with the keyword-based prediction of the entry’s category. If the user’s choice agrees with the prediction, then the next entry is displayed. Otherwise, the user is asked to provide some more information on the rationale behind their choice (Figure 34). To reduce users from feeling that they were somehow “wrong” and to encourage them to feel confident and honest in the explanation provided, they were told these explanations would be requested randomly. The information from each assessment of is logged using an XML data structure

The test program was distributed to 10 individuals to run unsupervised on their own computers. The test subject group did not include any of the original providers of appointment entries in order to eliminate any possibility that some of the ambiguous entries were familiar to the subjects. This was done in order to simulate the algorithm’s natural uncertainty and unfamiliarity with the user’s environment. The test subject group consisted of 2 postgraduate students, 2 non-academic professionals and 6 undergraduate students, all of which were made aware of the different category types and the typical kinds of entries they might include. Finally, the algorithm was allowed to run a simulation of the user interaction on the test collection, independently, and produce its own log of results, which would be compared against the correct choices and the choices of the subject group. It should be noted here that there really isn’t an objectively “correct” choice for most of the entries, since the meaning of an entry strongly depends on the user’s context.

However, the word “correct” is used to describe the categorisation of entries, in the same way as described by the entry’s author.

Another important note regarding the selection of the test subjects is that the aim of this experiment was not to assess whether the system performs adequately for the users that provided the original entries. Such an attempt would be extremely biased due to the fact that the original system is based on the information gathered from these people. It could be therefore expected that the system would perform in a satisfactory manner, providing however results that are meaningless due to the bias. A more logical approach would be to test the system against human subjects, who are unfamiliar with the original entries, therefore simulating the systems initial lack of any knowledge about the user it is meant to be helping.

The aim of the pre-caching system is to begin with some initial knowledge, which will, in time, adapt to a user’s particularities through the process of implicit relevance feedback. The scope of this experiment is to test the sufficiency of the basic knowledge for initial system operation in unfamiliar user environments and contexts. An appropriate analogy, to illustrate this rationale, is that of a secretary on her first day at work for a new person. She would have very little knowledge of this person’s context and preferences, although she would be aware of basic facts, such as “Meetings should not normally be scheduled outside office hours”. In time, the secretary would adapt to the preferences of the person she is assisting and might learn that this person might in fact have no problems with working for up to two hours outside normal office hours, for example.

Unfortunately, due to an error in the execution of the test program on one subject’s computer, one of his entries was lost. Therefore, to maintain a consistent result, that entry was excluded from analysis, resulting in 99 entries tested per subject – as the entry was of type “Meeting”, there was little impact on the frequency distribution of categories from that given in table 10.

## 4.4.2 Results and analysis

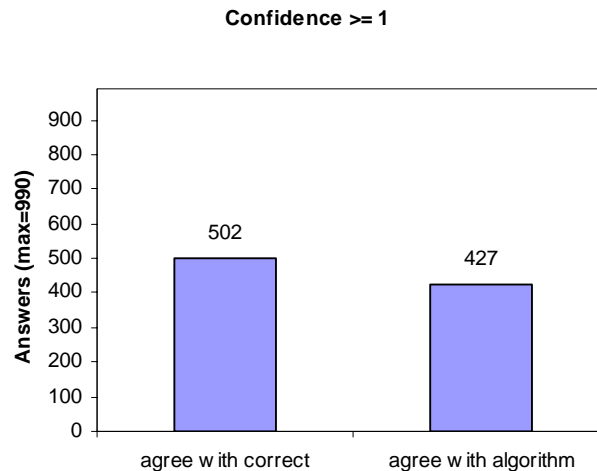
### 4.4.2.1 Analysis targets

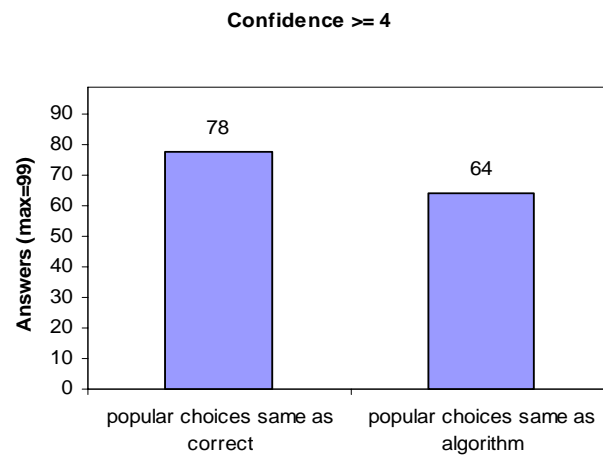
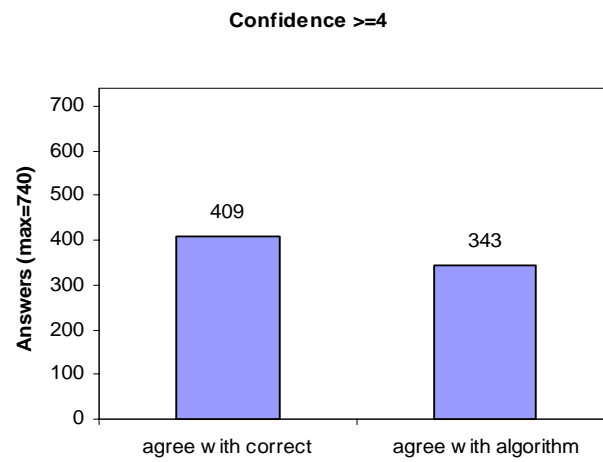
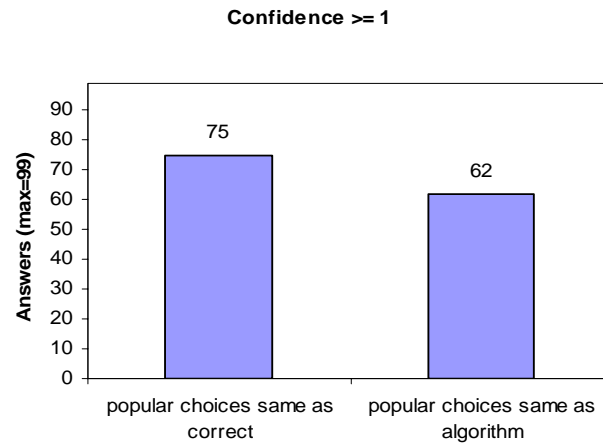
The post-processed data collected for each calendar entry is displayed in the following graphs (figure 35):

- The number of answers that agreed with the correct choice
- The number of answers that agreed with the algorithm's choice
- Whether the most popular choice agrees with the correct choice
- Whether the most popular choice agrees with the algorithm's choice

This data is reported for all levels of user confidence and separately for all the answers that were made with a user confidence level greater or equal to 4. This was done in order to assess the impact of potential lucky guessing. Also, it would be interesting to observe the level of confidence with which the users decide on their perception of the entries.

### 4.4.2.2 Original results





**Figure 35: Comparison of categories with original users' allocations**

	Confidence ≥1	Confidence ≥4	Difference
Agree with Correct	50.7%	55.3%	4.6%
Agree with Algorithm	43.1%	46.4%	3.3%
Popular choice same as Correct	75.8%	78.8%	3.0%
Popular choice same as Algorithm	62.6%	64.6%	2.0%
Algorithm's correct guesses	73.7%	n/a	n/a

**Table 11: Summary of original results**

#### **4.4.2.3 Analysis of original results**

Table 3 summarises these results. It is interesting to observe that approximately half (50.7%) of the answers given by subjects actually coincide with the answers provided by the entry owners, a percentage which rises slightly to 55.3% when considering only the answers given with a strong degree of confidence. This could be interpreted as indicating that the success rate of our algorithm, without any training, should not have to exceed 55% to be considered equivalent to human performance.

Having measured the performance of the keyword category identification algorithm on the same collection of test data, we have found it to score an approximate 73.7%, which is well beyond what the expectations based on the human performance should be. It is true that the algorithm has been formulated using, amongst other things, rules and keywords derived by the analysis of our original entry collection. However, similar performance levels were also found under a smaller scale test, which was conducted to test the functionality of the software. That test executed on two subjects and with smaller collections of 20 entries each, which were obtained after the analysis of our initial collection.

Another noteworthy observation is that while only half of the answers given actually were in accordance with the “correct” answers, when considering the most popular category choice for each entry, the percentage amounts to 75.8%. The close proximity of this number to the success rate of the algorithm shows that the

algorithm is very close to electing the same choice as the “majority” of the subjects, therefore it is close to adopting the best/most appropriate elements of human rationale for the completion of its task.

It is noteworthy also to observe that the number of answers that were given with a high degree of confidence is rather large and makes up for approximately 75% (740) of the total amount of answers (990). From this one can conclude that users appear to be quite confident about their choices, even though less than half of them are “correct”. Since the active prediction of the user’s choice is paramount, in order to provide personalised and meaningful results that are particular to the user, it appears that the target for an acceptable, perhaps tolerable, success rate for the prediction and suggestion of categories, and thus related information, should lie in these levels of 75%.

Finally, for the 11 recurring appointments, the most popular choices at each occurrence were measured. It was discovered that with a confidence level greater than zero, for four of the appointments (36%) the category was changed but only one change was actually to a “correct” choice. With a confidence level greater or equal to four, three entries (27%) were given a different popular choice, with none coinciding with a “correct” choice. Naturally this sample is fairly small and one cannot be conclusive, however, this seems an unlikely large percentage.

#### **4.4.2.4 Revised experiment design and results**

In section 4.2.1, the fact was mentioned that there is some overlap between the categories as assigned by the entry owners, which could result in some ambiguity. The results, as described above, compare the low-level categorisation of the entries by their owners with that of people who are completely unfamiliar with the context under which the entries were made. It can therefore be argued that there might exist a bias towards error, against the test subjects. To remove such a suspicion, the same results were analysed again, having grouped the categories Class, Work Task and General Task under a more general category called TASK, and also the categories Social and Birthday, under SOCIAL. The revised results can now be summarised as follows:

Revised results	Confidence ≥1	Confidence ≥4	Difference
Agree with Correct	51.7%	56.5%	4.8%
Agree with Algorithm	59.2%	67.0%	7.8%
Popular choice same as Correct	74.7%	78.8%	4.1%
Popular choice same as Algorithm	62.6%	63.6%	1.0%
Algorithm's correct guesses	73.7%	n/a	n/a

**Table 12: Revised summary results**

Table 13 shows the comparison of the original versus the revised results. From this table, it is apparent that any differences, where they occur, are not significant as their magnitude is of approximately 1%. It can be argued therefore that the analysis of the original results is valid for the revised experiment, despite the change in the experiment design. An average rise of about 18.5% is observable for the percentage of choices which agrees with the algorithm's choice. This is expected as the merged categories are similar in concept and thus, the grouping of narrow concepts under broader ones is expected to yield better performances. It should be noted here that while this statistic is of interest to observe, it has little significance in the measuring of the algorithm's performance.

	Confidence. ≥1		Confidence ≥4	
	Original	Revised	Original	Revised
Agree with Correct	50.7%	51.7%	55.3%	56.5%
Agree with Algorithm	43.1%	59.2%	46.4%	67.0%
Popular choice same as Correct	75.8%	74.7%	78.8%	78.8%
Popular choice same as Algorithm	62.6%	62.6%	64.6%	63.6%
Algorithm's correct guesses	73.7%	73.7%	n/a	n/a

**Table 13: Comparison of Original and Revised Results**

Furthermore, it can therefore be concluded that the breakdown of high-level categories into subcategories and that any overlap that may appear in the original categorisation scheme does not have any significant impact in the performance of the average test subject (our algorithm seems to remain thoroughly unaffected). However, the assignment of sub-categories might actually be desirable, as it would allow the production of more appropriate web searches. Indeed, our system is already designed to support inheritance from “master” or other categories.

#### **4.4.2.5 Summary of Findings**

This experiment reports work on using a keyword-based algorithm for predicting the category of diary entries in order to support mobile services (e.g. pre-caching of probably relevant internet content). In particular the experiment shows the results from the comparison of the effectiveness of the keyword-based algorithm with that of people other than the entry authors. The results show that, on average, people can individually correctly classify a diary entry only 49% of the time, while the majority decision from a group of users achieves 75% accuracy when compared with the original entry’s author’s categorisation. Furthermore, the automatic keyword-based algorithm tested here achieved 73% accuracy – nearly matching that of the majority of users and well exceeding that for individual users. While the test could have been conducted on a larger scale, the performance of the system is such as to show very promising potential and arguably enough to warrant very few changes in its design.

### ***4.5 Pre-caching Internet Content for mobile devices***

#### **4.5.1 Experiment design**

Having implemented a full system, as described in Chapter 3, it was time to test its performance and determine whether the hypotheses proposed by this thesis could actually be met. Ideally the system should be given to several users and they should be allowed to run it for a period of time which should be as long as possible. However, given the lack of volunteers that would be willing to run the experiment as part of their everyday routines and also given disproportionately large timescale the

experiment would take, a decision was made to test the final system under supervised conditions which would simulate real world scenarios as closely as possible.

Two groups of users were given the same scenario with some details of their imaginary living location, job and a list of some names of people and how they would be related to them. Furthermore, over the duration of three weeks, the users would be given five tasks per week that form their hypothetical schedule of activities for that week. These activities were given in the form of a calendar entry that contained a title, location and notes for each one. Some activities did not contain items in the location or notes fields, as they were based on real-world entries that we had collected in previous studies. The users were also given clear instructions on the exact meaning of each entry, through the provision of accurate descriptions of the entries (see appendix 6).

The users were then allowed to freely navigate through the pre-cached content that was fetched for these hypothetical schedules, and try to locate content that they thought might be helpful to them. We would also ask the users to give an indication of whether they found the provided content for each activity useful.

It was decided that the users should not be told that their behaviours would be logged. Also, one of the groups would have their logs analyzed and we would attempt to provide them with content that was personalized on the basis of these logs. Again, the groups would not be made aware of this discrepancy until after the experiment had ended. The analysis of the logs was done automatically by the system, as described in section 3.9, for each of the monitored subjects individually. Their respective profiles were maintained and updated at the end of each session, therefore influencing the retrieval process to personally match each of the monitored group's subjects.

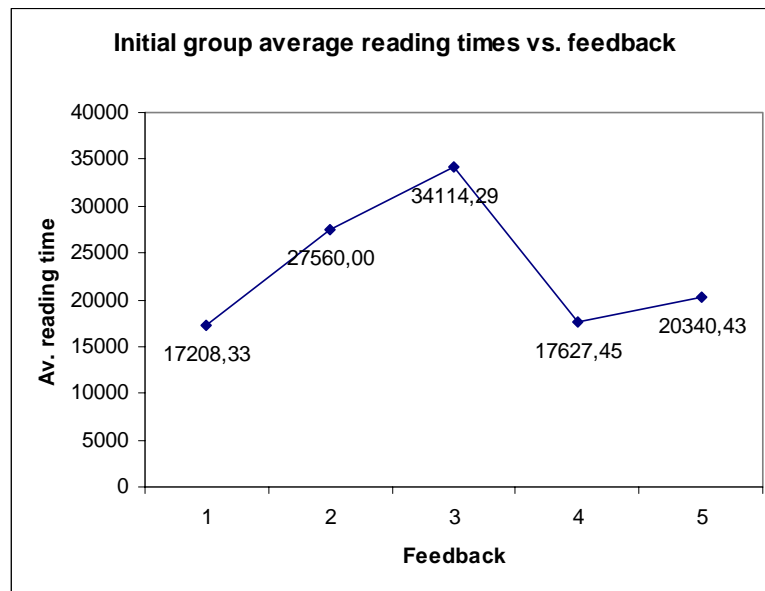
Finally, another factor which was considered in the experiment, was that the physical storage limitations for the devices used led to a choice to pre-cache only HTML documents, and furthermore, these were restricted to the documents proposed by Google for each search. Effectively this meant that the retrieval tree was limited to just one level. For a first-level document that contains three hyperlinks, a two-order retrieval means a total of three documents retrieved. The amount of generated documents from just the one-level tree (table 16) was very large and this would only

grow further with the implementation of additional levels. Therefore, in order to avoid overloading the user with documents and to overcome storage limitation problems, the choice was made to restrict the retrieval tree to just one level. Further to this, the focus was placed on the four most popular categories, according to the findings of the query test. Therefore the calendar sets given contained only entries of type Meeting, Travel and Social (including Birthday)

#### **4.5.2 Initial experiment setup**

An initial group of ten subjects volunteered to test the system before we proceeded with the actual experiment. All of the subjects were from a similar background and considered themselves computer literate, although most did not have previous experience with a PDA. The initial group was given different, but similar in context, data than those that would participate in the actual experiment; however, the rules of the experiment were the same, apart from the duration of the experiment, which would only encompass the virtual timeframe of one week. The goal of this initial experiment was to ensure the system ran smoothly with users that were unfamiliar with it. A further, and more important goal, was to observe the average reading times for the web documents and their relation to explicit feedback, as we planned to use this metric for implicit relevance feedback.

Having analysed the results of these initial groups, it was observed that the average reading times were not what we expected, and were certainly in contrast with previous research such as that mentioned in Chapter 2.



**Figure 36: Initial group average reading times vs. feedback ratings**

It is clear from this graph that the users take, on average, the same amount of time to distinguish between either relevant or largely irrelevant documents. Therefore it is apparent that the use of time as a metric is not a reliable source of information, since there is not much significant discrepancy between the average reading times for each feedback score. This observation brought about the decision to eliminate this metric from the weight recalculation formula, as it is in contrast with other findings, such as those by Morita [Mori94] and Claypool [Clay01], but seem to confirm Kelly's [Kell04] conclusion that reading time is an unreliable source for implicit relevance feedback. While my own research and that of Claypool's use scales of 1-5, Kelly uses a scale of 1-7. However what is more interesting than the direct comparison of observations is the fluctuation between these.

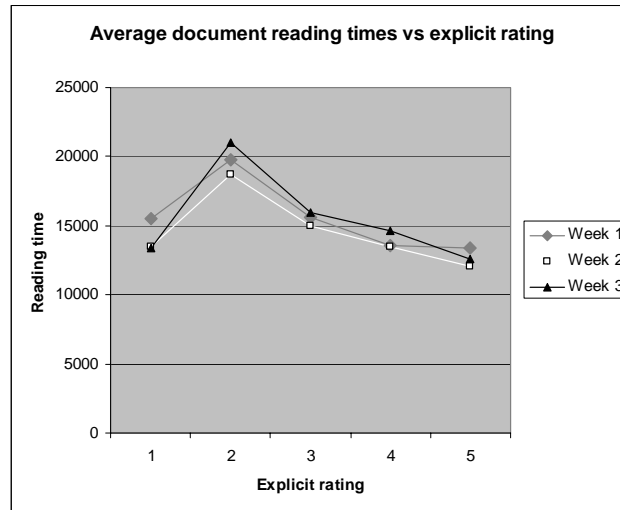
Author	Relevance category (1-least, 5/7-most)						
	1	2	3	4	5	6	7
Komminos	17208.33	27560.00	34114.29	17627.45	20340.43	n/a	n/a
Claypool	13414.00	18017.50	21217.00	24372.00	26797.50	n/a	n/a
Kelly	69000.00	77000.00	82000.00	46000.00	57000.00	71000.00	124000.00

**Table 14: Studies on average document reading times (msec) vs. perceived document usefulness**

### 4.5.3 Actual experiment

For our actual experiment, two groups of ten people each were used. The rules were applied in full this time and we were able to obtain some interesting results at the end of the experiment. Unfortunately, due to data corruption on the logs of two members of one group, we were forced to exclude them from the analysis, removing also two random members from the other group, to make the figures directly comparable.

In the following graphs, a representation of the average reading times for both groups, over the three experiment weeks is depicted.



**Figure 37: Experiment group average reading times vs. feedback ratings**

The trend shown here is slightly different from the results of the initial group. This is expected as the data for the two groups were not the same. However, again from this graph one can clearly see that determining a relationship between feedback and reading time is not feasible. The average reading times for the top two and the worst mark are very close, making any secure distinction between the two almost impossible.

Further analysis of the results should show whether there is a trend in the improvement of cache hits for the group whose data was tuned according to their previous logs. The following two tables show the numerical and percentile quantities of the opened documents (cache hits) vs. the total documents offered.

	Group 1	Group 2	Joint
Week 1	8,6%	24,4%	16.5%
Week 2	16.5%	18%	17.7%
Week 3	24.8%	30.5%	27.8%

**Table 15: Total documents vs. Opened documents (percentage)**

	Group1		Group2		Joint	
	Total	Opened	Total	Opened	Total	Opened
Week 1	1464	127	1464	357	2928	484
Week 2	1879	31	1880	357	3759	667
Week 3	1065	264	1248	381	2313	645

**Table 16: Total documents vs. Opened documents (absolute values)**

From table 15 above, one can clearly see an almost linear trend developing for group 1, who were the group that had their data adjusted according to their logs. This is a strong indication that for these users, the system provides an improvement in potential efficiency, if one considers that opening a document indicates the user's potential interest in it. For group 2, a solid conclusion cannot be made, as the percentage rates seem to fluctuate almost randomly, affecting of course the joint outcome as well. It would appear that for this group, the nature of the entries in their given schedules is the only determinant in the percentage of documents opened. For group 1 however, it appears that the application of interest indicators found in previous logs, has a restraining effect to the fluctuation of the variation in percentages and an overall effect that shows increased performance of the system.

As mentioned in Chapter 3, a further indicator of interest in a document is its summary. A look at how many documents were judged by the summary reveals the following results:

	Group1		Group2		Joint	
	Immediate Open	After summary	Immediate Open	After summary	Immediate Open	After summary
Week 1	90 (65.6%)	37 (34.4%)	255 (71.4%)	102 (28.6%)	345 (71.3%)	139 (28.7%)
Week 2	282 (90.9%)	28 (9.1%)	252 (70.6%)	105 (29.4%)	534 (80%)	133 (20%)
Week 3	209 (79.1%)	55 (20.9%)	205 (53.8%)	86 (46.2%)	504 (78.1%)	141 (21.9%)

**Table 17: Summary viewing as a deciding factor for opening a document.**

The percentages compare each figure with the total number of opened documents. From these trends we see that approximately only 1 in 4 times did the users consult the summary before making a decision. This suggests that a user will be inclined to navigate to a website based on the information contained in its title solely. A higher percentage was expected in this situation, especially since visiting a document is costly (in terms of loading and reading time) and also because in the implementation, explicit feedback was requested after each user had finished reading. According to Nielsen [Niel97] also, users tend to like summaries and will read them before resuming with the rest of the text. However, it must be noted here that the summaries were only displayed on demand, while the document titles were immediately available. This was a necessary tradeoff in order to reduce the scrolling required for the retrieval results overview, although the action of opening a summary incurs an additional cost to the subjects.

Finally, the following table takes a look at the average document scores for each session:

	Group 1	Group 2	Joint
Week 1	2.26	2.37	2.44
Week 2	2.60	2.38	2.48
Week 3	2.49	2.36	2.42

**Table 18: Average document scores (0-lowest, 5-maximum)**

It would appear from this table that the scores remain at a constant level and in fact, at around the middle of the scoring table. This in turn is consistent with the results by Jansen [Jans03], where it is mentioned that users should expect 1 of 2 web documents they view to be relevant.

The choice was made not to measure the individual scores attained by each group in order to establish a trend. These would be actually just measuring the ability of Google to return relevant results, where as this research is only concerned with measuring the relevance of the web query in the context of the calendar entry and the user's needs.

#### **4.5.4 Further discussion**

##### **4.5.4.1 Experimental environment**

The experiment was performed in a quiet room. This setting might not appear to be realistic in the sense that there were no external distractions for the subjects, although they were given food and drink and were allowed to communicate and interact with each other. Mobile devices are used in both mobile and stationary environments and since the experiment shows that reading times are not long (around 25 seconds), it is expected that a mobile user could easily dedicate such small times to interact undisturbed. I perceive the notion of “mobile” to mean “out of office” rather than “walking” or “driving”, therefore the setting seems adequate for the purposes of the experiment. In any case, a maximum time limit of 2 minutes, based on the observations from an initial test group (see 3.1.3), is imposed on the measuring to eliminate gross inaccuracies due to user distraction. Therefore it can be concluded that the environment settings for the experiment were appropriate for its purposes and did not deduct from its credibility.

##### **4.5.4.2 Statistical Confidence**

Further to the results of the test, a two-paired T-Test was conducted in order to investigate the statistical significance of the findings that were observed, in relation to the improvement of cache hit-rate improvement for the two groups. The t-test was the recommended approach as the experiment dealt with two groups of different subjects, who came however from a homogenous background, for one of which an

external factor was applied and its effect was observed. This external factor was the monitoring and consideration of interaction and feedback behaviour, and its implication in the retrieval process.

The cache hit rates between week 1 and week 3 were measured for each individual subjects and their difference was analysed. The findings of the t-test are summarised in table 19 below. With a statistical probability of error of approximately 1.2% when considering whether the external factor was indeed responsible for the cache hit-rate improvement, the credibility of the results is further enhanced. The full details of the t-test can be found in appendix 8.

Meana— Meanb		t	df
0.1003		2.4985	14
P			
one-tailed	0.01277		
two-tailed	0.02554		

**Table 19: T-test results**

#### **4.5.5 Summary of Findings**

Several important conclusions were reached by this experiment. Firstly, the system shows that useful Internet content can indeed be pre-cached based on calendar information alone. This is shown by the cache hit rates, which rose close to 30%. Another important finding was that the reading behaviour of the subjects when faced with documents on a small screen, showed that the time spent on a document does not accurately reflect the quality of the document. A correlation between these two cannot be established, therefore the use of reading time for implicit relevance feedback on small screen devices is not recommended. Finally, and perhaps most importantly, the results of this experiment show a gradual, almost linear improvement of the retrieval performance for the group whose behaviours were taken into account. Although the duration of the experiment could have been longer, one could say that the results are solid enough to provide adequate confirmation of a promising learning curve performance.

## Chapter 5

### Conclusions and future work

## **5 Conclusions and future work**

In Chapter three, and with the experiment discussed in Chapter four, a pre-caching system was described and implemented, which is based on the information found in electronic calendars. The main function of this system is to provide useful content for a user with a small mobile computing device. This chapter will discuss the conclusions that were reached from the implementation of this system, with particular reference to their relevance to the original hypotheses of the thesis. Further findings that were discovered will also be laid out in this chapter. Finally, some critical considerations regarding the experimentation will be discussed and directions for future research on this topic will be given.

### **5.1 Review of original hypotheses**

#### **5.1.1 Hypothesis 1**

*“Calendars can provide information that can be used to pre-fetch useful Internet content to users.”*

While a system, such as the one discussed in this thesis, would not be able to completely satisfy all of a user’s internet content needs or desires in its own right, it was shown that it can indeed provide useful content for the appropriate entry categories. This conclusion rises firstly from the experiment presented in section 4.3, and secondly from the observation of the cache hit rates in the final experiment (4.5).

More specifically, it was discovered that while almost each entry category could be used to formulate at least one query which the user might find relevant, the categories where queries and documents seem to be most important to are Travel, Social and Meeting.

### 5.1.2 Hypothesis 2

*“The majority of calendar entries can be organised into a small number of categories for all users whose professional circumstances are similar.”*

This hypothesis is shown to be true through the analysis of the calendar entries obtained by staff and students that work within a University environment. It appears that most of these categories are indeed common to most calendar users, irrespectively of their professional circumstances, as the comparison with previous research reveals.

### 5.1.3 Hypothesis 3

*“Queries can be formed from calendar entries, without direct user involvement and independently of their brevity, which will be meaningful and retrieve documents useful to the users.”*

Based on the proof of the previous hypothesis, the clustering of category identifiers and the association with relevant additional keywords, the system is able to form a multitude of web queries, especially for the top three categories identified in the previous section. More importantly, even in the case of entries where the information contained therein comprises of a single word, the automatic generation of web queries based on common knowledge and the users’s preferences proves to be able to provide meaningful and useful content. This is again evident from the cache hit rates in 4.3.

### 5.1.4 Hypothesis 4

*“Such a pre-fetching system can learn from the users and adapt its performance in order to offer content that is increasingly relevant to the users.”*

Given the opened document trends as described previously in section 4.5, there is reason to believe that the system is able to adapt accordingly to the individual

preferences of a user. In these results, a linear increase in the pre-cached document hit rate was observed for the group whose interactions with the system was taken into account. The fluctuating cache hit rates of the other group show that despite the nature of the searches (caused by keywords that are better candidates than others, e.g. better internet content is probably available for a large town rather than a village), the monitored group is helped by the system to obtain increasingly more relevant results.

The adaptation of the system to the users' preferences is achieved through the constant implicit monitoring of user interaction with the system but also on the explicit ratings provided by the users for documents of interest. Because of the unreliability of the implicit relevance feedback indicators and the uncertainty regarding the factors that might affect their measure, the conclusion has to be drawn that implicit relevance feedback for this situation cannot be used exclusively to create reliable user profiles. However, these indicators are helpful as a complimentary method to explicit relevance feedback and are useful in order to resolve situations where explicit ratings would result in ambiguity. To illustrate the latter point, one can imagine a situation where two search-forming keywords (e.g. "hotel" and "bed & breakfast") may have been explicitly weighted equally by a user. Implicit interest indicators could then indicate a slight preference towards one type of accommodation than another, although the user might be willing to settle with either type of accommodation.

The study described in section 4.5 was limited to three weeks, which seem to provide evidence for the aforementioned conclusions that are adequate for the purposes of this thesis. However, a further trial over an extended period of time would be able to show the fluctuation between improvement rates and whether a peak is reached, which would indicate the system's optimum performance level.

## **5.2 Further findings**

Apart from the findings that were part of this thesis' main target, several other interesting facts were encountered. The similarity between decision times for judging positively or extremely negatively against a document prohibits the use of such a metric from any further studies.

Further to this, one can't help but be impressed by the low reading times, which are in stark contrast with other studies that are concerned with the average reading time of a web document, such as that by Jansen. Other studies report average reading times closer to the ones we experiences, but again higher (Kelly, Claypool), although these were not based on web documents. However, all of these previous studies relate to documents viewed on a desktop, where a large monitor facilitates the viewing of documents.

It is possible for one to arrive to the conclusion that the smaller reading times on the handheld indicate a tendency for users to “skim” through the document in order to decide on its usefulness. This should be considered normal, given that the need for immediate and full comprehension of the information in the text was not there (due to the virtual environment). Therefore the users would try to acquire a general “feel” for the quality of the searches, and refer to these later on when they have more time or immediately need the information. Nielsen argues that scanning the text in a web document is common practice [Niel97b]. Furthermore, in his work, long pages that cause lots of scrolling are considered to be largely disliked by users. Since the limited size display on a handheld causes websites to appear unproportionately large and causes lots of scrolling, the findings of reduced reading times seem to be further supported by these statements.

In addition to the tests which referred to the main hypotheses of the thesis, a further test was conducted to investigate whether the usability of small screen calendars could be improved through the categorisation of entries, and more specifically by the use of colour to discriminate between categories [Kohn05]. Contrary to standard practice from most mobile device manufacturers, it was discovered that colour can indeed enhance the usability levels of a calendar by conveying information faster than, and at least just as accurately as, a monochromatic representation.

## **5.3 Future Work**

### **5.3.1 Long term testing for user-system adaptability**

It would be of interest to observe the behaviour of the system over extended periods of time, and under real world situations, where users would actually engage in cooperation with the software to assist in everyday tasks. Interesting observations could then be made with regard to the adaptation of the human component to the software, rather than the opposite which is currently measured. In section 4.2, the replies to the questionnaire that was issued showed willingness for users to adapt, if they were aware of the advantages that this change could have. However, whether such claims would hold true is an unanswered question.

The greatest challenge to conducting long-term tests, especially under real world environments, would be to find subjects that would be willing to participate. Long-term commitment is not easy to ask of people who are unrelated to one's research and furthermore, the dedication of their owned devices and adherence to a schedule that would be required for the testing is not guaranteed. Although the distribution of the software over the Internet to a multitude of users was considered as an option at one stage, the difficulty in tracking these subjects for the retrieval of logs and the difficulty in ensuring that they followed the practices required for testing, made this idea unfeasible.

### **5.3.2 Supplementary methods for the manipulation of the additional keyword database**

Further work could be carried out in the additional keyword database. One shortcoming of the current implementation is that it does not allow users to manually insert or remove additional keywords in order to fine-tune the database of results to their preference. It is expected that such a facility could have a positive effect on the adaptability of the software.

Based on the recommendations of Rashid et al. [Rash02] that indicate a recommender system without any knowledge for its user should start by presenting the most popular items, the system in its final test was preconfigured with a list of additional keywords which were assigned predetermined scores (see section 3.6.2).

Having made these choices and ratings for the users was reasonable after Rashid's recommendations and special care was taken to ensure these choices were representative of the majority's rationale, after Mackay [Mack90] and Palen's [Pale99] showed that users tend to accept and adapt to default settings. However, an added facility that might enable the users to inspect these original ratings and explicitly re-order query formulating keywords according to importance, might give the system a better chance to show higher cache hit rates from the beginning of its use. Such a facility could also allow the users to directly affect the pace of the learning process if they so desired.

Further experimentation could be made with the implementation of non-static thesauri, instead of the manually generated ones that are currently in use. It would be interesting to see whether additional keywords could automatically be inserted into the database, through the analysis of retrieved text. Variations on this theme could include analysis of all retrieved text for new keywords, analysis of explicitly marked relevant documents and analysis of implicitly marked documents. After Mandala's [Mand99] investigations, it might be sensible to hypothesise that the combination of manual and automatic thesauri will have a positive effect on the query formulation process. The proof of this hypothesis would require extensive comparative studies to be performed on "automatic-only" and "automatic+manual" thesauri versions of the system with the same subjects, on top of the existing "manual-only" study presented in this thesis.

### **5.3.3 Expanding the search to include user desktop computer contents**

At the moment, the system focuses on the pre-caching of Internet documents only. During the tests of the system, some of the subjects indicated that the pre-caching of other documents (e.g. meeting notes and email conversations) would also be a useful addition to the system, especially for calendar entries of the meeting category. A variant of this system could extend the notion of pre-caching to include the pre-caching of relevant documents from the file system of the user's desktop computer.

Incidentally, while the system was undergoing the final testing phase (section 4.5), Google released a tool which applies its searching technology on a personal

computer's file system, which could be employed as a provider of the searching technology required to retrieve items from a user's personal computer. This tool, called Desktop Search<sup>xix</sup>, is able to search not only the user's file system for particular types of files, but also is able to look into data and logs kept by other applications, as shown in table 20. The Google desktop search mechanism uses HTML to present its search results and can be queried in a fashion that is almost the same as the normal Google website, therefore the inclusion of the desktop searches in the system would not be difficult to implement.

Email Clients	Documents	HTML	Multimedia	Instant Messengers
Outlook	Word	Internet Explorer	Images	AOL Instant Messenger
Outlook Express	Excel	Netscape	Video	
Netscape Mail	Powerpoint	Firefox/Mozilla	Music	
Thunderbird	PDF			

**Table 20: Google Desktop's pre-configured search modules**

The applications listed in this table come pre-configured with the Google Desktop download, although an interested party can write add-ons which allow the search of, or for, further file types and application data. It is expected that the inclusion of the Desktop search component from Google would greatly enhance the cache hit rates for certain categories, such as Meeting, for which it is currently difficult to formulate many useful queries for.

---

<sup>xix</sup> <http://desktop.google.com>

## **5.4 Summary and Conclusions**

This thesis was written as an examination of the suitability of electronic calendar entries as a source of information that can assist the pre-caching of Internet content, for use on mobile devices. In summary, with the implementation of a system as described in Chapter 3 and with the tests presented in Chapter 4, it is shown that such a hypothesis can be proved true.

Users are naturally different in the way that they use their calendars and in the amount of information they deposit therein. Some users prefer analytical accounts of every activity or event in their life. Others prefer monolectic entries to act as reminders, while they keep all the details in their memory. Furthermore, some types of entries (e.g. travelling arrangements) denote activities for which, by nature, it is possible to pre-cache considerable amounts of Internet content, while others can contain activities or events that can not be supported by Internet content (e.g. “buy bread”).

Despite these variations in input style and content, the system described in this thesis is capable of making informed assumptions and predictions in order to generate relevant web queries, thus supporting not only those users that are analytical in their input style, but also those who are by nature monolectic. The system uses a simple mechanism of basic common-sense rules, derived from the observation and analysis of people’s calendars and their contents, together with a database for known keywords and associated queries, to achieve this purpose. It is shown that even with this relatively simple structure, the system manages to achieve good cache hit rate levels, which might only be improved with a more extensive keyword database or complex rules. This is in turn adequate proof that calendar entries are indeed a valuable source of information for the personal pre-caching of Internet content.

Further to these findings, this thesis shows that the adaptation of the system to an individual user’s preferences and context is possible through the monitoring of implicit and explicit interest indicators. The adaptation of the system to its users is shown to provide continuous improvements to the cache hit rate, which suggests that the profiling of users is an essential process for any system that might attempt to provide personal Internet content pre-caching. The thesis also highlights the fact that

implicit indicators, even though they are a valuable source of information for the profiling of users and have a very low cost to the user interaction aspect, cannot be entirely relied upon for this purpose, and need to be combined with explicit interest indicators to obtain accurate results.

Apart from results which are targeted to the main purpose of this thesis, several other observations have been made during the research process, which are worthy of mention. Firstly, useful conclusions from the observation of the interaction of humans with their calendars can be drawn and used for the improvement of electronic calendar applications. The current lack of support in many mobile calendars for entry categories is a drawback which, if addressed, can have a significant effect on the usability of calendars, as I show in (yet unpublished) research [Kohn05]. Secondly, the interaction patterns of humans with electronic documents on small devices are considerably different than those on desktop computers, as is shown by the average document reading times and the corresponding conclusions that can be drawn from these.

## References

- [Appe99] Appelt, D., Israel, D.J., Introduction to Information Extraction Technology, *International Journal of Communications in Artificial Intelligence* 12, pp. 161-172, 1999
- [Atta77] Attar, R., Fraenkel, A.S., Local Feedback in Full Text Retrieval Systems, *Journal of the Association for Computing Machinery*, 24(3), pp. 397-417, 1977
- [Aver99] Avery, C., Resnick, P., and Zeckhauser, R. The Market for Evaluations. *American Economic Review*, 89(3), pp. 564-584, 1999.
- [Bala95] Balabanovic M., Shohav Y., Yun Y., An Adaptive Agent for Automated Web Browsing, *Journal of Visual Communication and Image Representation* vol. 6 n.4, 1995
- [Bala97] Balabanovic, M., Shoham, Y., Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66-72, 1997
- [Basu98] Basu, C., Hirsh, H., Cohen, W., Recommendation as Classification: Using Social and Content-based Information in Recommendation, *Proceedings of the 15th National Conference on Artificial Intelligence*, pp. 714-720, July 1998
- [Bate79] Bates, M.J., Information Search Tactics, *Journal of the ASIS* 30(4), 205-214, 1979
- [Bill00] Billsus, D., Pazzani, M., Chen, J., A Learning Agent for Wireless News Access, *Proceedings of the 5th International Conference on Intelligent User Interfaces*, New Orleans, pp.33-36, 2000
- [Blan01] Blandford, A.E., Green, T.R.G., Group and Individual Time Management Tools: What you get is not always what you need, *Personal and Ubiquitous Computing* 5, pp. 213-230, 2001.

[Borg98] Borgman C.L., Why are Online Catalogs Hard to Use? Journal of the ASIS, 37(6), 387-400, 1986

[Brees98] Breese, J. S., Heckermann D., Kadie, C., Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the 14th annual conference on Uncertainty in Artificial Intelligence, pp. 43-52, 1998.

[Buck95] Buckley C., Salton, G., Allan, J., Singhal, A., Automatic Query Expansion Using SMART: TREC-3, Proceedings of the Text REtrieval Conference, pp. 69-80, 1995

[Buck96] Buckley C., Singhal, A., Mitra, M., Salton, G., New Retrieval Approaches Using SMART: TREC-4, In Harman D. (ed.), Proceedings of the Trec-4 Conference, National Institute of Standards and Technology Special Publication, 1996

[Cach02] CacheFlow Inc. Active caching technology.  
<http://www.cacheflow.com/technology/whitepapers/active.cfm>, link valid Oct02.

[Cade00] Cadez I., Heckerman D., Meek C., Smyth P., White S., Visualisation of navigational patterns on web sites using model based clustering, Technical Report MSR-TR-0018, Microsoft Research, Microsoft Corp. 2000

[Chan99] Chan, P. K., Constructing Web User Profiles: A Non-Invasive Learning Approach, Proceedings of Web Usage Analysis and User Profiling International Workshop, San Diego, 1999

[Chin97] Chinen, K. Yamaguchi S., An Interactive Prefetching Proxy Server for Improvement of WWW Latency. Proceedings of INET97, June 1997.

[Cher98] Cherkasova, L., Improving WWW caches performance with Greedy Dual-Size Frequency Caching Policy, Hewlett-Packard Technical Report HPL-98-69 (R1), November 1998

- [Chur88] Church, K., A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text, Proceedings of the 2nd Conference on Applied Natural Language Processing, pp.136-143, 1988
- [Clay01] Claypool M., Le, P., Waseda, M., Brown, D., Implicit Interest Indicators, Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI '01), USA, pp. 33-40, 2001
- [Cohe00] Cohen, E., Kaplan, H., Pre-fetching the means for document transfer: a new approach for reducing Web latency, Proceedings of the 2000 IEEE INFOCOM conference, pp. 854-863, Tel-Aviv, 2000
- [Cont05] Avantis ContentCache  
[http://www.avantisworld.com/02\\_cddvd\\_cache\\_overview.asp](http://www.avantisworld.com/02_cddvd_cache_overview.asp), link valid Feb05
- [Coop99] Cooper, A., The Inmates are Running the Asylum, Indianapolis: Sams Publishing 1999
- [Croft79] Croft, W.B., Harper, D.J., Using probabilistic models of document retrieval without relevance information, Journal of Documentation 35, pp. 285-295, 1979
- [Croft95] Croft, W.B., Cook, R., Wilder, D., Providing Government Information on the Internet: Experiences with THOMAS, Digital Libraries Conference, DL'95, pp. 19-24, 1995
- [Crou92] Crouch, C. J., Yang, B., Experiments in Automatic Statistical Thesaurus Construction, Proceedings of the 15th Annual ACM Conference on Research and Development in Information Retrieval, Copenhagen, pp. 77-88, 1992
- [Crov98] Crovella, M., Barford, P., The network effects of pre-caching, Proceedings of IEEE Infocom, 1998
- [Crow92] Crow, D., Smith, B., DB\_Habits, Comparing Minimal Knowledge and Knowledge-Based Approaches to Pattern Recognition in the Domain of User-

Computer Interactions, Neural Networks and Pattern Recognition in Human Computer Interaction, pp. 39-63, NY, Ellis Horwood, 1992.

[Cunh97] Cunha C., Jaccoud C., International Symposium on Computers and Communication 97, Alexandria, Egypt 1997

[Dash01] Dashpende M., Kapyris G., Selective Markov Models for Predicting Web Page Accesses, SIAM International Conference on Data Mining, April 2001

[Davi02] Davison B. D., The design and Evaluation of Web Prefetching and Caching techniques, PhD Thesis, University of the State of New Jersey, 2002

[Davi02] Davison, B., Predicting Web Actions from HTML content, Proceedings of the 13th ACM conference on Hypertext and Hypermedia, College Park, MD, pp. 159-168, 2002

[Davi04] Davison, B., Learning Web Request Patterns, in A. Poulovassilis and M. Levene, editors, Web Dynamics: Adapting to Change in Content, Size, Topology and Use, Springer 2004

[Duch99] Duchamp D., Prefetching Hyperlinks, Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems, Boulder, CO, 1999.

[Dusk97] Duska, M. B., Marwood, D., Feely, M. J., The measured access characteristics of World Wide Web client proxy caches, Proceedings of the USENIX Symposium on Internet Technologies and Systems, Monterey CA, pp. 23-36, 1997

[Efth96] Efthimiadis, E., Query Expansion, In Williams, Martha E. (ed), Annual Review of Information Systems and Technology (ARIST) 31, pp. 121-187, 1996

[Fan99] Fan L., Cao P., Lin W., Jacobson Q., Web pre-fetching between low bandwidth clients and proxies: potential and performance, ACM conference on Measurement and Modelling of Computer Systems, (ACM SIGMETRICS '99), 1999.

- [Fan99] Fan L., Jacobson Q., Cao P., Lin W., Web prefetching between low-bandwidth clients and proxies: Potential and Performance, Proceedings of the Joint International Conference on Measurement and Modelling of Computer Systems (SIGMETRICS 99), Atlanta, Georgia, 1999
- [Feni81] Fenichel, C.H., Online Searching: Measures that Discriminate Amongst Users with Different Types of Experience, *Journal of the ASIS*, 32(1), pp. 23-32, 1981
- [Ferb96] Ferber, R., Using Co-Occurrence Data for Query Expansion: Wrong Paradigm or Wrong Formulas?, unpublished, <http://information-retrieval.de/ferber/homepage/pdf-ps/using-cooc.pdf>
- [Fers82] Ferster, C. B. & Culbertson, S. A., Behavior principles. Englewood Cliffs, NJ: Prentice-Hall, 1982
- [Fide91] Fidel, R., Searchers' Selection of Search Keys, *Journal of the ASIS* 43(7), pp. 490-500, 1991
- [Four87] Fournas, G. W., Landauer, T. K., Gomez L. M., Dumais, S. T., The vocabulary problem in human-system communication, *Communications of the ACM* 30, 11 (Nov.), pp. 946-971, 1987
- [Fox80] Fox, E.A., Lexical Relations Enhancing Effectiveness of Information Retrieval Systems, *SIGIR Forum* Vol 15, No. 3, pp. 6-36, 1980
- [Foyg00] Foygel, D., Strelow, D., Reducing Web Latency With Hierarchical Cache-based Prefetching, Proceedings of the International Workshop on Scalable Web Services (in conjunction with ICPP0), Toronto, Canada, 2000
- [Good99] Good, N., Schafer, J.B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J. Combining Collaborative Filtering with Personal Agents for Better Recommendations. Proceedings of AAAI-99, pp. 439-446, 1999

- [Griff94] Griffioen J., Appleton R., Reducing file system latency using a predictive approach, Proceedings of the 1994 Summer USENIX technical conference, Cambridge MA, 1994
- [Ha98] Ha, V., Haddawy, P., Towards Case-Based Preference Elicitation: Similarity Measures on Preference Structures, Proceedings of UAI 1998, pp 193-201, 1998
- [Hill87] Hill, F. J., Peterson, G. R., Digital Systems: Hardware Organization and Design. John Wiley & Sons, New York (Third Edition) 1987.
- [Huan01] Huang, Y., An Intelligent Adaptive News Filtering System, Masters Thesis, Graduate College of University of Illinois at Urbana-Champaign, 2001
- [Jans03] Jansen, B., Spink, A., An analysis of web documents retrieved and viewed, Proceedings of the 4th International Conference on Internet Computing, Las Vegas, Nevada, pp. 65-69, 2003
- [Jans98] Jansen, B.J., Spink, A., Bateman, J., Sarasevic, T., Real Life Information Retrieval: A Study of User Queries on the Web, ACM SIGIR Forum 32(1), pp 5-17, 1998
- [Jian02] Jiang Y., Wu M., Shu W., Web Prefetching: Costs, Benefits and performance, Proceedings of the 7th International Workshop on Web Content Caching and Distribution, Boulder, CO, 2002
- [Jian97] Jiang, Z., Kleinrock, L., Prefetching Links from the WWW, IEEE International Conference on Communications, Montreal, pp. 483-489, 1997.
- [Jian98a] Jiang, Z. Kleinrock L., An Adaptive Network Prefetch Scheme, IEEE Journal on Selected Areas in Communications, Vol. 16, No. 3, pp. 358-368, 1-11, 1998.
- [Jian98b] Jiang, Z., Kleinrock, L., Web prefetching in a mobile environment, IEEE Personal Communications 5, pp. 25-34, 1998

- [Jing94] Jing, Y., Croft, W.B., An association Thesaurus for Information Retrieval, Proceedings of RIAO'94, Intelligent Multimedia Information Retrieval Systems and Management, pp. 146-10, Paris, France, 1994
- [Jose99] Joseph, D., Grunwald, D., Prefetching using Markov predictors. Transactions on Computers, 48(2), 1999.
- [Kell01] Kelly, D., Belkin, N., Reading Time, Scrolling and Interaction: Exploring Implicit Sources of User Preferences for Relevance Feedback, Proceedings of the 24th annual international ACM conference on Research and Development in Information Retrieval, New Orleans, pp. 408-409, 2001
- [Kell04] Kelly, J. D., Understanding Implicit Feedback and Document Preference: A naturalistic user study, PhD Thesis, State University of New Jersey, USA, 2004
- [Kim00a] Kim J., Oard, D., Romanlik, K., User Modelling for Information Access Based on Implicit Feedback, Technical report, University of Maryland Computer Science Department, HCIL-TR-2000-11, 2000
- [Kim00b] Kim, J., Oard, D., Romanlik, K., Using Implicit Feedback for User Modelling in Internet and Intranet Searching, Technical report, College of Library and Information Services, University of Maryland, 2000
- [Kinc85] Kincaid, C. M., Dupont, P.D., Kaye A. R., Electronic Calendars in the Office: An assessment of user needs and current technology, ACM Transactions on Office Information Systems 3(1), pp. 89-102, 1985
- [Klem94] Klemm, R., WebCompanion: A friendly client-side Web pre-fetching agent, IEEE Transactions on Knowledge and Data Engineering 11(4), pp. 577-594, 1994
- [Kohr01] Kohrs, A., and Merialdo, B. Improving Collaborative Filtering for New Users by Smart Object Selection, Proceedings of International Conference on Media Features (ICMF) 2001 (oral presentation).

- [Komn05] Komninos, A., Dunlop, M., Enhancing the usability of calendar applications in mobile devices through entry categorisation, Submitted for acceptance at MHCI05, Salzburg, Austria, 2005, available online at <http://www.cis.strath.ac.uk/~andreas/paper2.pdf>
- [Kons97] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J., GroupLens: Applying Collaborative Filtering to Usenet News, Communications of the ACM 40 (3), pp. 77-87, 1997
- [Krai98] Kraiss, A., Weikum, G., Integrated document caching and prefetching in storage hierarchies based on Markov-chain predictions. VLDB Journal, 7, pp.141-162, 1998
- [Kroe97] Kroeger, T. M., Long, D. D. E., Mogul, J.C., Exploring the bounds of Web latency reduction from caching and prefetching. In Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS '97), 1997.
- [Mack90] Mackay, W.E., Users and customisable software: A co-adaptive phenomenon, Dissertation, Sloan School of Management, Cambridge, MA, MIT, 1990
- [Mand99] Mandala, R., Tokunaga, T., Tanaka, H., Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion, SIGIR'99: Proceedings of the Association for Computing Machinery Special Interest Group on Information Retrieval, pp.191-197, New York, 1999
- [Mano82] Mano, M. M., Computer System Architecture. Prentice-Hall, Englewood Cliffs, NJ, Second Edition, 1982.
- [Mark98] Markatos, E. P., Chironaki, C. E., A top 10 approach for prefetching the web, INET98 Proceedings, Internet Global Summit, 1998
- [Mitr98] Mitra, M., Singhal, A., Buckley, C., Improving automatic query expansion, Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 206-214, 1998

- [Mori94] Morita, M., Shinoda, Y., Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval, Proceedings of the 17th annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 272-281, Ireland, 1994
- [Nano03] Nanopoulos A., Katsaros D., Manolopoulos Y., A Data Mining Algorithm for Generalised Web Pre-fetching, IEEE Transactions on Knowledge and Data Engineering, Vol.15, No. 5, Sept/Oct. 2003
- [Nguy98] Nguyen, H., Haddawy, P., The Decision-Theoretic Video Advisor, Proceedings of the AAAI Workshop on Recommender Systems, pp. 76-80, 1998.
- [Nich97] Nichols, D. M., Implicit Ratings and Filtering, Proceedings of the 5th DELOS workshop on Filtering and Collaborative Filtering, Hungary, pp. 31-36, 1997
- [Niel97b] Nielsen, J., Be Succinct! Writing for the Web, <http://www.useit.com/alertbox/9703b.htm>, Link valid Feb 2005
- [Niel97] Nielsen, J., Morkes, J., Concise Scannable and Objective: How to write for the web, <http://www.useit.com/papers/webwriting/writing.html>, 1997, Link valid Jan 2005
- [Oard98] Oard, D. W., Kim, J., Implicit Feedback for Recommender Systems, AAAI Workshop on Recommender Systems, Madison, WI, pp.81-83, 1998
- [Ouel00] Ouellet, M., Gecsei, J., Nie, J.Y., Discovering Internet Resources to Enrich a Structured Personal Information Space, Proceedings of the 2000 RIAO conference, Paris, pp.1450-1463, 2000
- [Padm96] Padmanabhan V., Mogul J. C., Using Predictive Prefetching to improve WWW Latency, ACM SIGCOMM Computer Communication Review, 26 (3), pp. 22-36, 1996

- [Pale99] Palen, L., Social, individual & technological issues for groupware calendar systems, Proceedings of the ACM CHI99 conference, Pittsburg, PA, pp.17-24, 1999
- [Palp98] Palpanas T., Web Prefetching Using Partial Match Prediction, Proceedings of the 4th International Web Caching Workshop, San Diego, CA, 1998
- [Panda02] Panday A., Vatsavai R. R., Ma X., Srivastava J., Shekhar S., Data mining for Intelligent Web Prefetching. proceedings of the Workshop on Mining Data Across Multiple Customer Touchpoints for CRM (MDCRM02), 2002
- [Papo91] Papoulis A., Probability, Random Variables and Stochastic Processes, McGraw-Hill, 1991
- [Payn93] Payne, S. J., Understanding Calendar Use, Human-Computer Interaction 8(2), pp 83-100, 1993
- [Pazz96] Pazzani, M., Muramatsu, J., Billsus, D., Syskill & Webert: Identifying Interesting Web Sites, Proceedings of the National Conference on Artificial Intelligence, Portland, OR, 1996
- [Peat91] Peat, H.J., Willet, P., The limitations of term co-occurrence data for query expansion in document retrieval systems, ACM Transactions on Information Systems 10 (2), pp. 115-141, 1991
- [Penn00] Pennock, D., and Horvitz, E. Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Modelbased Approach. Proceedings of UAI 2000, pp. 473-480, 2000
- [Pitk99] Pitkow J., Pirolli P., Mining Longest Repeated Subsequences to Predict WWW surfing, Proceedings of the Second USENIX Symposium on Internet Technologies and Systems, October 1999.
- [Pitk99] Pitkow J., Pirolli P., Mining longest repeating subsequence to predict world wide web surfing. 2nd USENIX symposium on Internet Technologies and Systems, 1999

- [Qiu93] Qiu, Y., Frei, H.P., Concept based query expansion, Proceedings of the 16th International ACM SIGIR conference on the Research and Development in Information Retrieval, Pittsburgh, pp. 160-169, 1993
- [Raft01] Rafter, R., Smyth, B., Passive Profiling from Server Logs in an Online Recruitment Environment, Proceedings of the IJCAI Workshop on Intelligent Techniques for Web Personalization, pp. 35-64, 2001
- [Rash02] Rashid, A.M., Albert, I., Cosley, D., Lam, S., McNee, S., Konstan, J., Riedl, J., Getting to know you: Learning new user preferences in recommender systems, Proceedings of the 7th international conference in Intelligent User Interfaces, San Francisco, pp. 127-134, 2002
- [Resn94] Resnick, P., Iacovou N., Suchak, M., Bergstrom, P., Riedl, J., GroupLens: An Open Architecture for Collaborative Filtering or Netnews, Proceedings of the ACM Conference on Computer Supported Collaborative Work, pp. 175-18, 1994.
- [Rocc71] Rocchio, J.J., Relevance Feedback in Information Retrieval, In Gerald Salton (ed.), The SMART Retrieval System – Experiments in Automatic Document Processing, Chapter 14, Prentice Hall, Englewood Cliffs, NJ, 1971
- [Ruvi02] Ruvini, J.D., Gabriel, J.M., Do users tolerate errors from their assistants? Experiments with an email classifier, Proceedings of the 7<sup>th</sup> International conference on Intelligent User Interfaces, San Francisco, pp. 216-217, 2002
- [Saru00] Sarukkai R. R., Link prediction and path analysis using Markov chains, 9th International World Wide Web conference, 2000
- [Sche98] Schechter S., Krishnan M., Smith M. D., Using path profiles to predict HTTP requests, Proceedings of the Seventh International WWW Conference, Brisbane, Australia, pp. 457-467, 1998
- [Schu97] Schutze, H., Pederson, J.O., A co-occurrence based thesaurus and two applications to Information Retrieval, Information Processing and Management 33 (3), pp. 307-317, 1997

- [Shar95] Shardanand, U. Maes, P., Social Information Filtering: Algorithms for automating “word of mouth”. Proceedings of Computer Human Interaction, pp. 210-217, 1995
- [Smar05] RM SmartCache 2  
<http://www.rm.com/Primary/Products/Product.asp?cref=PD141057&catref=219.1.2.43>, link valid Feb05
- [Smea96] Smeaton, A., Kelledy, F., O’ Donell, R., TREC-4 Experiments at Dublin City University: Thresholding Posting Lists, Query Expansion with WordNet and POS Tagging of Spanish, Proceedings of TREC-4 D. Harman (ed.), Gaithersburg USA, 1996
- [Smit82] Smith, A. J., Cache memories. ACM Computing Surveys, 14(3), pp.473-530, 1982.
- [Spar71] Sparck Jones, K., ed., Automatic keyword classification for information retrieval. London: Butterworths, 1971
- [Swam00] Swaminathan, N., Raghavan, S. V., Intelligent pre-fetching in WWW using client behaviour characterization. Proceedings of the Eighth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2000.
- [Taus97] Tauscher, L., Greenberg, S., How people revisit Web pages: Empirical Findings and implications for the design of history systems. International Journal of Human Computer Studies, 47(1), pp. 97-138, 1997.
- [Thie89] Thiebaut D., On the Fractal dimension of computer programs and its applications to the prediction of the cache miss ratio, IEEE transactions on Computers, 38(7), pp. 1012-1026, 1989
- [Unga98] Ungar, L., Foster, D. P., Clustering Methods for Collaborative Filtering, Workshop on Recommendation Systems, 15th National Conference on Artificial Intelligence, 1998

- [Venk01] Venkataramani A., Yalagandula P., Kokku R., Sharif S., Dahlin M., The potential costs and benefits of long term pre-fetching for content distribution, Sixth International Workshop on Web Caching and Content Distribution, 2001
- [Voor94] Voorhees E., Query Expansion using Lexical-Semantic Relations, Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval, pp.61-69, 1994
- [Wang96] Wang Z., Crowcroft J., Prefetching in the World Wide Web, Proceedings IEEE Global Internet Conference, London, 1996
- [Wasf99] Wasfi, A. M. A., Collecting User Access Patterns for Building User Profiles and Collaborative Filtering, Proceedings of IUI 1999, pp. 57-64, 1999
- [Xu96] Xu, J., Croft, W.B., Query expansion using Local and Global Document Analysis, Proceedings of the 19th annual international ACM SIGIR conference on Research and Development in Information Retrieval, Zurich, pp. 4-11, 1996
- [Yian01] Yiang Q., Zhang H., Li T., Mining web logs for prediction models in WWW caching and prefetching, 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001
- [Zhan03] Zhang, W., Lewanda, D. B., Janneck, C. D., Davison, B. D., Personalized Web Prefetching in Mozilla. Technical Report LU-CSE-03-006, Dept. of Computer Science and Engineering, Lehigh University, 2003

## Appendices

## Appendix 1: Calendar Usage Questionnaire

Name of interviewee: \_\_\_\_\_

### General Questions

1: Do you use your calendar to remind you of things to do, instead of just noting appointments and important dates?

Yes / No

2: Do you synchronise your calendar with other PIM devices or do you tend to keep more than one calendars?

I synch	I don't synch	I have only 1 calendar
---------	---------------	------------------------

3: Would you be willing to change your style of input (wording) if you knew it would help a program pre-fetch related internet sites for your PIM device (handheld, mobile)?

Yes / No

4: Are there any internet sites that you visit on a regular basis (e.g. daily, weekly)? How many approximately?

5: How big a “grace” period would you give to a program if it needed to adapt to your personal needs? Would you expect the program to work perfectly straight from the box?

6: Please rate your internet content needs (5-most important, 0-least important):

Documents (html)  
Documents (.pdf, Word etc.)  
Images  
Video  
Sound

7: Does your calendar contain information about your personal/social life or is it used for business only?

Business only / Personal & business

## Appendix 2: Calendar Entry submission form

Name of interviewee:

---

### Calendar Appointments

Subject:	
Location:	
Notes/Description:	
Items desirable for appointment	

Subject:	
Location:	
Notes/Description:	
Items desirable for appointment	

### **Appendix 3: Web query test and results**

The following pages show the handout that was given to subjects for the web query test, as per section 4.3. The collections for each subject were randomly picked from the master entry collection and are different for each subject. Therefore we present here a sample.

## Web Query Test

Test Instructions: Read carefully before proceeding!!!!

Imagine that the following lists contain items extracted from your calendar (actually they are a random sample from a collection of other peoples' entries!!). What I would like you to do, is think about things that you might search for on the Internet, based on the information contained in these calendar notes, then write down your queries on the last box of each row.

Because some of the entries might be ambiguous, I am providing you with the category they belong to, for example, an entry titled "Andreas" and categorised as "M", means it's a meeting with someone called Andreas. Also imagine that you are based in Glasgow, therefore each location that does not appear to be in Glasgow, can be considered as a place you need to travel to. Again, for example, if you saw something like "IEEE conference, London", then you might search for "London Hotels, flights to London, IEEE conference".

Try to think of as many queries as possible, but if you can't think of any, that's perfectly OK!!!! It's rare that someone would like to see internet content about someone they are meeting or for going to the supermarket. Another thing to keep in mind, is that you might come across duplicate entries. This is intentional, but if you can't be bothered thinking about it again, you can leave it blank, which again is perfectly OK. If you don't understand what an appointment is about, again don't worry too much about it, just move on to the next one.

For any questions you may have, just send me an email: [andreas@cis.strath.ac.uk](mailto:andreas@cis.strath.ac.uk).

I will collect the answers from you tomorrow around 3pm, so please try to complete it by that time. If you wish, you can also leave the answers at my pigeonhole as well.

Thanks very much for your participation!!!

Cat.	Title	Location	Notes	<i>Queries</i>
B	Birthday			
R	Simon Scheisher		Arrive centre until march 2003	
WT	babysit			
Misc	Mairead Home?			
M	Team Briefing		Just to confirm that Wednesday's team briefing will occur at 9am in room B219	
M	Oral - Ben Lower	Cork	Dep. Gla 11.30 check-in 10.45	
M	See dr. X			
M	Aarhus meeting			
M	Prog. Languages group		David Lievens on ad- hoc polymorphism	

M	Meeting	Righead industrial estate		
WT	Email		person1, person2 re. photo, person3, person4: person4@yahoo.com	
M	Joanne Mitchell			
T	LHR -YUR AC897 Departs T3, seat 9h			
GT	Queen's Park vs Sterling Albion		Details 250 words Sunday Herald 170 words	
M	Louise T. Campbell			
C	513 systems and control tutorial	M415		
M	Honours shadow board	S3.15		
T	Departure City1-City2			

M	John Lafferty - Sister Elisabeth			
B	Michael Tagg's birthday			
Misc	MIMS online review			
GT	Library book return			
T	Flight details		B7 RUBB london stansted (STN) to glasgow (GLA)	
M	EH board	col 416		
M	FRAP			
M	Meeting with spanish project student Francisco			
S	Going out with the girls from the office			
M	Writing for publication	CAP level 2 GHB		
T	Check in for flight BE830 dept 1955 arrive 2105			

S	Spiderman			
S	UGC unfaithful		8 for 8.30	
R	Unavailable			
M	Victoria			
M	Program Board	TBC		
M	Meeting with Gillian		Lit Review	
M	meeting with andy			
M	meet students			
M	AM1 Marketing meeting			
S	Judith Dinner			
R	Ann at doctor's			
M	Gillian Scott's thesis committee			
M	Joanne Mitchell			

R	Exam Diet			
M	Andreas			
R	available		Every Friday from 14.30 to 16.30	
S	Chamber choir	Assembly hall		
M	Meeting Sally	staff room	principles of marketing	
C	AM1 Maria			
WT	questionnaires	Sauchiehall str		
<i>GT</i>	<i>100 to BOS to cover incentives and travel expenses to Manchester</i>			

## Query test Results

Birthday	Class
birthday presents	513 systems and control
web cards	Java Strathclyde continuing education
birthday presents	java evening class
e-cards	SU maps
egreetings card	513 homepage
rabs pub	sustainable products course notes
free greeting card	sustainable products course handbook
egreetings card	sustainable products timetable
birthday gift men	sustainable products student photos
gift idea lady	POM strathclyde
gift ideas	flow nets
rab pub map glasgow	strath uni 17246 timetable
gifts birthday UK buy	strath uni sustainable products timetable
books amazon	
cds amazon	
birthday gift men	
rabs pub location	
birthday gift	
novel birthday gift male	
novel birthday gift male	

General	Misc	Reminder
tickets queens park	hairdresser glasgow stylist	g. burt notes
travel queens park	street map area X	balfron
Martin Atonga	248 albert drive glasgow	balfron activities
Mr X	Mairead	G.Burt
news queens park	Mairead	edmedia
news stirling albion		balfron map
hannah homepage		balfron activities
Chewing machine		rent

Social	Work
wine cheap uk	railway club jokes
cinema listings spiderman	richard brown
review spiderman	natalie
Spiderman 2 movie	leslie heart
UGC spiderman	Minutes court university of Strathclyde
Glasgow cinemas movies	railway club directions glasgow
UGC glasgow showing times	railway facts figures
UGC unfaithful	TV guide
Marketing Department strathclyde university	strath uni chaplaincy
wedding gift ideas	
Unfaithful movie times	
recorder charity concert homepage	
staff outing bbq details	
spiderman review	
happy hour pub	
bacchus	
aviemore	
spiderman review	
spiderman IMDB	
recorder charity concert location	
gift	
5pm.co.uk	
spiderman review	
strath uni chamber choir	
novel engagement presents gifts	

Travel
flights aberdeen glasgow
train aberdeen glasgow
coach aberdeen glasgow
flight time B7 RUBB
budget flights canada glasgow
Sweden weather
City2 hotels
City1 trains
City1 flights
National car rental
accommodation birmingham
entertainment birmingham
map of area X
map st lukes dukes road 22
YXY map
National car rental prices
Sweden hotels
Sweden places to visit

Sweden map
Sweden time
Sweden offers
Paisley hotels
Paisley maps
Paisley timetable train
Paisley timetable bus
Maps sweden
travel info sweden
journey planner uk
flight glasgow heathrow
National car rental YXY
YUR map
YUR hotels
Heathrow transfer info
guide to sweden
town maps
town hotels
stanstead airport
birmingham airport
birmingham transport
birmingham bus
birmingham train
bus paisley
flight confirmation
dukes road bus
flight confirmation
cheap car rentals YXY
B7 RUBB glasgow london
KL 82
YXY national car rental
Canada

## **Appendix 4: Automatic categorisation test instructions**

These instructions were emailed to potential subjects.

Dear all,

I'm looking for a few people to run a test for my research. It's really very straightforward, all you need to do is download a small executable from my site and run it. The whole thing should take about 15 minutes. The exe is at <http://www.cis.stath.ac.uk/~andreas/test1.exe>

When the program installs, leave the Launch box ticked, that will start the test. Tick the "work from set" box and click Start. You will be presented with several dialogs that contain appointment entries collected from other calendar users.

All you need to do is assign them to a category of your choice and also say how confident you are about this decision. The program may ask you for a reason behind your rationale, in which case you will be presented with another dialog.

Please give reasonable answers and try not to click the Skip button. At the end of the test, the program generates a file called "applog.txt", which I need you to mail back to me. There's even an uninstall option to remove everything from your PC.

## Appendix 5: Sample from an automatic categorisation test log.

The following is an extract from the log of a categorisation test, as described in section 4.4

```
<app>ms. Janet Deale
  <cat>meeting</cat>
  <loc>outside of the class</loc>
  <note></note>
  <reason></reason>
  <confidence>4</confidence>
</app>
<app>AM1 Maria
  <cat>misc</cat>
  <loc></loc>
  <note></note>
  <reason>i don't know exactly what it means so i put it under misc.</reason>
  <confidence>3</confidence>
</app>
<app>Honours workshop
  <cat>classt</cat>
  <loc></loc>
  <note></note>
  <reason>because i think its a class</reason>
  <confidence>5</confidence>
</app>
<app>POM
  <cat>misc</cat>
  <loc>M422B</loc>
  <note></note>
  <reason></reason>
  <confidence>1</confidence>
</app>
<app>Call mr X
  <cat>reminder</cat>
  <loc></loc>
  <note>(telephone number)</note>
  <reason>because it looks like a reminder to phone mr X</reason>
  <confidence>5</confidence>
</app>
<app>Library book return
  <cat>reminder</cat>
  <loc></loc>
  .....
```

## Appendix 6: Final test list of appointments

### Analytical list of Appointments (W1)

---

Your Name:

Appointment 1

Title: MIMS Joint Board

Location: S3.15

Notes: (blank)

Comments:

This is a board meeting that concerns the final marks of the students at the Marketing Department. It happens twice a year in room S3.15 (Stenhouse Building, 3<sup>rd</sup> floor, room 15)

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

## Appointment 2

Title: QMU Board

Location: Edinburgh

Notes: (blank)

Comments:

This is a board meeting at the Queen Margaret University College, in Edinburgh.  
You will have to travel to Edinburgh and come back the same day for this meeting.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

### Appointment 3

Title: Meeting with Francisco

Location: (blank)

Notes: (blank)

Comments:

Francisco is one of your students; he wishes to consult you with regard to a piece of homework.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

#### Appointment 4

Title: Barcelona

Location: (blank)

Notes: (blank)

Comments:

You are jetting off to the city of Barcelona, in Spain, for a short stay of 3 days over the weekend. There you will meet some Spanish friends and have an opportunity to relax for the duration of the weekend.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

Appointment 5

Title: Staff outing BBQ

Location: (blank)

Notes: (blank)

Comments:

Given the nice weather forecasted for the end of the week, a barbeque has been arranged for all staff working at your department. This event will take place at the university's campus, after work on a Friday.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

## Analytical list of Appointments (W2)

Your Name:

Appointment 1

Title: meeting Alan Poulter

Location: (blank)

Notes: confirmation numbers outbound EIPHW51 return EIPHW21

Comments:

You are meeting dr. Alan Poulter to discuss a journey you will both be making to London next month. You have booked some flights for the both of you and you will give the booking details to him, for further reference.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

## Appointment 2

Title: Nina Kowalska

Location: (blank)

Notes: (blank)

Comments:

Nina Kowalska manages St. Luke's Communications, an advertising agency in London. You will be meeting her along with dr. Poulter, in order to discuss a project that is to be undertaken jointly. Ms. Kowalska is a friend of Chris Seeley, a common acquaintance who works at the University of Bath.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

### Appointment 3

Title: Sarah Dinner

Location: (blank)

Notes: (blank)

Comments:

You are going to dinner with Sarah, one of your close friends. Perhaps after dinner, you would both like to continue the evening together and might go somewhere else.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

#### Appointment 4

Title: Check in for BE 817 flight to Birmingham

Location: (blank)

Notes: (blank)

Comments:

You are jetting off to Birmingham on university business. You have made a note of the flight number you need to check into. The length of your stay in Birmingham is not great: you will be coming back the following day.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

Appointment 5

Title: EDSRC platform meeting

Location: (blank)

Notes: (blank)

A meeting of the EDSRC committee has been scheduled

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

# Analytical list of Appointments (W3)

Your Name:

Appointment 1

Title: Mark Dunlop

Location: (blank)

Notes: (blank)

Comments:

You are meeting dr. Mark Dunlop, a colleague, for coffee and an informal chat. There is no special agenda for your meeting, which is happening mostly for social reasons.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

## Appointment 2

Title: Toronto

Location: (blank)

Notes: (blank)

Comments:

You have been contemplating on a trip to Canada later this year, to see relatives who live permanently there. It's not entirely certain whether your relatives will be able to accommodate you for the stay, so potential accommodation information would be of interest.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

### Appointment 3

Title: Steve

Location: (blank)

Notes: (blank)

Comments:

You are meeting Steve Neely, a colleague at work. He mentioned he wanted to discuss some departmental administrative matters with you; however, you are not sure what that may be.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

#### Appointment 4

Title: Smartlab RAED

Location: (blank)

Notes: (blank)

Comments:

Smartlab, the name of your research group, is holding a meeting with regard to the RAED project (Role-based Access- control for Evolution of Distributed Software). You need to attend in order to discuss the project's progress and issues that have come up.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

### Appointment 5

Title: UGC Alien vs. Predator

Location: (blank)

Notes: (blank)

You are heading off to watch one of the latest films, Alien vs. Predator, at the UGC cinema in the city centre, with some friends.

How useful did you find the results for this appointment? (1=not much, 5=very much)

1	2	3	4	5
---	---	---	---	---

Would you have added any other searches to the results?

## Appendix 7: Sample log from final test

The following is a sample extract from the logs obtained for the final test.

```
<search>
  <name> Joint Board</name>
  <expanded>1</expanded>
  <doc>
    <url>\searchpages\level1\154715.htm</url>
    <opened>1</opened>
    <time>13000</time>
    <feedback>1</feedback>
  <summary>0</summary>
</doc>
<doc>
  <url>\searchpages\level1\154717.htm</url>
  <opened>1</opened>
  <time>11000</time>
  <feedback>2</feedback>
<summary>0</summary>
</doc>
<doc>
  <url>\searchpages\level1\154722.htm</url>
  <opened>0</opened>
  <time>0</time>
  <feedback>0</feedback>
  <summary>0</summary>
</doc>
<doc>
  <url>\searchpages\level1\154725.htm</url>
  <opened>1</opened>
  <time>8000</time>
  <feedback>1</feedback>
<summary>0</summary>
</doc>
<doc>
  <url>\searchpages\level1\154726.htm</url>
  <opened>0</opened>
  <time>0</time>
```

```
<feedback>0</feedback>
<summary>0</summary>
</doc>
<doc>
  <url>\searchpages\level1\154727.htm</url>
  <opened>0</opened>
  <time>0</time>
  <feedback>0</feedback>
  <summary>0</summary>
</doc>
<doc>
  <url>\searchpages\level1\154729.htm</url>
  <opened>0</opened>
  <time>0</time>
  <feedback>0</feedback>
  <summary>0</summary>
</doc>
<doc>
  <url>\searchpages\level1\154730.htm</url>
  <opened>0</opened>
  <time>0</time>
  <feedback>0</feedback>
  <summary>0</summary>
</doc>
<doc>
  <url>\searchpages\level1\154732.htm</url>
  <opened>1</opened>
  <time>4000</time>
  <feedback>1</feedback>
  <summary>0</summary>
</doc>
</search>
```

## Appendix 8: T-Test details

	Subject	w1	w2	w3		Difference	w3-w1
Group 1	S1	0.005464	0.707627	0.165289			0.159825
	S2	0.054645	0.114894	0.251656			0.197011
	S3	0.065574	0.106383	0.288732			0.223159
	S4	0.098361	0.055085	0.197183			0.098822
	S5	0.120219	0.123404	0.28169			0.161472
	S6	0.120219	0.055556	0.28169			0.161472
	S7	0.098361	0.099138	0.178862			0.080501
	S8	0.131148	0.055085	0.343137			0.21199
						mean	0.161781

Group 2	S9	0.125683	0.217021	0.275641			0.149958
	S10	0.213115	0.212766	0.384615			0.171501
	S11	0.360656	0.119149	0.262821			-0.09784
	S12	0.136612	0.212766	0.198718			0.062106
	S13	0.333333	0.217021	0.275641			-0.05769
	S14	0.31694	0.191489	0.384615			0.067675
	S15	0.26776	0.131915	0.294872			0.027112
	S16	0.196721	0.217021	0.365385			0.168663
						mean	0.061436

Test Summary		Group 1		Group 2
		Xa		Xb
n		8		8
sum		1.294251		0.491488
mean		0.1618		0.0614
sumsq		0.2275		0.1024
SS		0.0181		0.0722
variance		0.0026		0.0103
St. dev.		0.0509		0.1016

Variances and standard deviations are calculated with denominator = n-1,

Meana— Meanb		t		df
0.1003		2.4985		14
P				
one-tailed	0.01277			
two-tailed	0.02554			

## Appendix 9: Publications

The following publications have been made based on the work presented in this thesis:

Komninos, A., Dunlop, M.D. (2003): Towards a model for an Internet content pre-caching agent for small computing devices,  
Proceedings of the 10th International conference on Human Computer Interaction (HCI2003), Crete, Greece, 2003

Komninos, A., Dunlop, M.D. (2004): Keyword based categorisation of diary entries to support personal Internet content pre-caching on mobile devices,  
Proceedings of the Mobile and Ubiquitous Information Access Workshop in association with Mobile HCI04, Glasgow 2004

Komninos, A., Dunlop, M.D. (2005): Calendar based Contextual Information as an Internet Content Pre-Caching Tool, 28th Annual International ACM SIGIR Conference on Research and and Development in Information Retrieval, Information Retrieval in Context Workshop, Salvador, Brazil