Mobile Search
University of Jyväskylä MIT Department, Finland

Pedro Tiago Evangelista

July 24, 2007

Acknowledgments

To all teachers and colleagues that have accompanied me during my academic formation and especially to my family for their support, education and constant presence in my formation as human being.

Thanks to Researcher Mikko Vapa who helped me during the project execution and gave me unconditional support.

Thanks to Heikki Kokkinen and Jukka Nurminen from Nokia Research Center Peer-to-Peer Research group for the guidance during the project.

Summary

This report describes the work developed during the internship period by the undergraduate student Pedro Tiago Evangelista in the context of Mobile search project.

During the last years progress in web search engines has been made to the point that relevant information can be reached easily most of the times. However very little empirical research has been carried to study web search in highly dynamic social network environments composed of mobile devices.

The aim of this work was therefore to investigate novel approaches that took advantage of the social network environment inherent to that mobile P2P paradigm.

The work focused mainly on the development of prototype for the Mobile search concept. The prototype was built on top of Drupal content site management system. A simulator was also developed to test different query forwarding algorithms.

This study suggests that the methods presented can be a complement to traditional web search engines.

This document describes the main activities while focusing in the more interesting problems solved.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Academic Context

This report was written in the context of *Projecto de final de curso* subject of Computer Science Engineering degree of Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia.

Researcher Mikko Vapa, of the MIT Department of University of Jyväskylä gave academic and advisory guidance to this project. The project was developed from March 2007 through July 2007 in the MIT Department of University of Jyväskylä in the Peer-to-Peer Research group.

1.2 Technological and Scientific Context

The main objective of the internship was to study and explore the concept of mobile search. A prototype was designed in order to test and validate the feasibility of concepts developed.

In order to pursue these objectives a series of technologies/opportunities were carefully explored in order to determine the best way to develop this system. A considerable effort in knowledge elicitation practice's were also required before starting the design and implementation phases.

1.3 Internship Goals

The internship described in this document focused on the development of a prototype for testing mobile search inherent concepts.

The internship could be divided into the following main tasks :

- Knowledge elicitation: The first thing to be considered is the global system comprehension on the eyes of the stakeholder. First the Preview[8] methodology was used in requirement phase. Several fruitful meetings were also held with all the parties involved in the project(Nokia Research Center Helsinki Peer-to-Peer group representatives and Mikko Vapa from MIT Department of University of Jyväskylä Peer-to-Peer Research group).

  The information gathered during this phase was required for an effective understanding of the system and for the overall project development.

- Prototype: A prototype was designed and implemented to test the feasibility of Mobile search.

- Implementation of a simulation environment: A simulator was implemented to test the system behavior in different scenarios. This simulator can be a valuable tool for future research.

- Project documentation: All aspects related with knowledge elicitation and implementation have been documented. The major supporting documents are presented in this report.

1.4 Document Structure

The organization of the document body is similar to overall structure of the application development. The report is divided to the following chapters:

- Mobile search overview introduces the scope of the project and gives a global overview. The concepts developed during the project are also presented.

- System Requirement Analysis.

- System Specification.

- Prototype.

Under each of these chapters, there will be sub-chapters each with a relevant issue. The document ends with future work and conclusion.

Chapter 2

Mobile Search Overview

2.1 Introduction

Mobile phones' computational power has been improving approaching the capabilities of general purpose computers. Nowadays is possible to host a web site in a mobile device. It is also expected that number of mobile web sites will out number the static web servers [13].

Mobile phones possess an extra set of concerns that are not present in normal web servers(e.g. Personalization; Interactivity; Location and context dependence;dynamic)[13]. Those concerns can be further expanded by taking into consideration the social network formed by the contacts in the address book. This fact introduces paradigm shifts in relation to the Peer-to-Peer web search paradigm and the traditional centralized approach.

Recently, there has been a growing interest in how to explore the mobile phone capabilities in the web search context and how to merge them with existing phone functionalities[13][11]. However the research has tended to focus on centralized approaches or Peer-to-Peer web search, rather than on the Peer-to-Peer web search in the social network context. The purpose of this report is to present different set strategies that take advantage of the described type of environment and extend current web search mechanisms giving the end user new possibilities of exploring information.

The road map of this chapter is as follows. Initially the motivation behind the need for mobile search is presented. Continuing with the core concerns and major differences between this type of search and traditional centralized web search. To accomplish this goal first a detailed description of the mobile search and it's core concerns is shown. After a comparison between the two types of search are shown. Subsequently a brief description of the prototype and related work within the topic are presented.

2.2 Principles

The concept of mobile search rose from the specifities presented in this section. In the future will be common to have a web server running in mobiles devices. This represents a shift in normal web servers webware. The biggest change is the possibility of users to freely manage it's own contents without being restricted by third entities. There is the need to categorize content in different ways in order to create new forms of navigation and searching data. The content in mobile phones can be divided in two distinct logic groups: dynamic and static. Dynamic content usually is unique and generated by the mobile phone sensors. Static content on other hand is not context dependent and is generated by the user. Both types of content can be easily replicated. Usually dynamic content can be easily characterized by tags, although common content can be categorized in a similar way. Content is distributed in overlapping data islands. Each user may belong to several data islands simultaneously because each user is connected to users who belong to different interest groups(even unknowingly)[3]. The connections are created based on the address book contacts forming presumably a power law graph[3]. It's assumed that the nearest neighbors of a node have higher probability of owning relevant content to that node. In the information searching context is important to have the ability to search through relevant data and take advantage of the overall network topology.

2.2.1 System Description

In this subsection is presented a description of a system for mobile search. The system is based on pure Peer-to-Peer architecture and it offers scalability, efficiency, resilience to failures and privacy at a higher degree than current centralized solutions.

To take advantage of the portrayed scenario a new set of concepts were introduced. One of the most important concepts is how to navigate through neighbor data. Users search one graph level at the time. Every time a user issues a search query the mobile device forwards it to all it's neighbors. The neighbors answer back by returning a result set and a list of their neighbors. If the user who issued the query is not satisfied by the results he can always ask new results from the neighbors as long there are non visited nodes in the network. A user can start issuing a query anywhere in the network. This concept was named *manual multi hopping*. *Manual multi hopping* can be extended to *automatic multi hopping* if an algorithm is used to sort which of the non-visited nodes to query further.

Other way of navigating is by searching neighbor content tags and getting the result set composed by the content links with the tags and the list of next level neighbors. Tags work as links between content categorized similarly. At each hop the user gets the list of contents tagged in a similar way by nodes in it's neighborhood.

The mobile search system can be divided in two logical parts: meta crawling and local web search engine. The part responsible for the meta crawler gets the results from the direct neighbors. The way results are presented can always be changed thus the mobile device bears the load of processing the returned references. It can be employed any specific method to sort out the references in any specific order. For example more relevance can be given to results from a certain source so they appear first in result list.

There is also the possibility of merging different types of mobile phone data with different type of content. For example a user A may search for user B's meetings after getting the results he may merge the results with his own agenda and display the meeting locations on a map.

The local web search engine gives the user power to tailor the search results to it's own needs. The search index can be updated every time the content changes. The user may allow certain information to be only searched by a specific group of users or to influence certain query results in a certain context.

This feature allows users to create *groups of trust*. They can decide which information source is more relevant to them in different contexts.

2.3 Comparison

It may be pointed out that centralized solutions have one single point of failure, load balance and trust issues and may censor certain entities[7]. Although nowadays they have grown incredibly robust.

For example Google presents in it's back end a highly scalable architecture [2] but it cannot address the premise that our friends are more likely to have interesting results to us and may not even be connected or linked to our content. In this scenario the hyperlink concept is expanded by our network of connections formed by the mobile phone address book. These types of linkage enables the blend of several groups of interest along the network. In several situations the link web structure of documents doesn't portray possible relations between people [6].

Normal web search engines don't allow tailoring results to individual needs. For example user A only wants to display a specific result list to a certain query from user B. Centralized solutions provide an efficient way of finding popular content but lack the ability to find more personal/social proximity content. This situation is evident in a corporate setting where many documents are not available to the outside world. Other type of personal/social proximity content that is not indexed by normal web search engines is mobile phone data. One clear example is searching for a phone number or meeting information that is available in one of our neighbors. This capability allows to avoid the use of third entities(e.g. number services, central servers) and enhances the information availability. In the other hand mobile search due to the topic oriented network nature is not suited to find popular content. Conversely, it's a powerful mechanism in restricted topic set environment.

One major issue of mobile search in relation with the centralized approach is the quality of the results returned. Different sites may have different criteria to classify and rank information. This poses a problem on how to merge the different results sets returned for a query.

This can highly increase the quality of the results in some scenarios. For example in work context user A can give more weight to Document X in searches made by users from the group work because that document is more relevant to them.

Other issue is the high number of neighbors and free riding. Those factors are a risk to network traffic. They can be overcome first by limiting the search query to a pre-selected group of users, second by only returning back neighbors who have a higher probability of having meaningful content.

One main advantage of mobile search is the total independence of the nodes. The system can operate without using any central server. System load is fully distributed. In this type of system the owner can index the searchable content whenever he/she desires.

As long a user sets the right permissions for the different contents, other users linked to that user won't have to know the exact location of the information and they can search for it.

2.4 Drupal Prototype

Drupal was chosen to be used as a test prototype of features taught to be important in this type of system.

Concerns	Centralized solutions	Mobile search
Load	centralized/single point of failure	highly distributed
Trust	censorship/pressure from external entities	highly distributed
Number of results	millions (single set)	thousands to billions (multiple different sets)
Index update	days to months	every second
Content type	popular	personal/social proximity

Table 2.1: Centralized web search solutions vs Mobile search

Drupal is an open-source content type management system. It allows to manage, organize and publish several types of content.

The meta crawler described in a previous section was built as a weakly coupled component on top of Drupal local web search engine. This component allows *automatic-multi hopping* and result interleaving.

The current implementation is single threaded because mobile Apache doesn't support multiple threads[12][11].

Drupal tac_lite module and Drupal module were also used as fundamental elements in the prototype setting. These modules allow to set content access rules and to process user authentication in distributed fashion without any central servers.

It was implemented an extra component that allows to do queries to local mobile phone content such as location, address book and meeting data. This feature was built as a proof of concept.

The prototype is also able to gather search results from unmodified Drupal web sites.

The more significant drawbacks during the elaboration of the prototype are related with the single-threaded nature of the meta crawler. That aspect can have very negative impact on response time because site crawling is done in a serial way. A multi-thread implementation would speed up the system considerably.

2.5 Related Work

The concept of Peer-to-Peer web search has been harnessed before in the literature. Different approaches have been tried before[16][10][9]. Although these studies tended to focus on Peer-to-Peer web search, rather less attention has been paid to how to take advantage in this scenario of mobile sites' concerns and integration in the social network context.

Zhou states that evaluation of resources by human users it's more important to resource quality than the traditional machine based approach[16]. In his paper presents a novel page ranking algorithm - Peer-Rank. In this report a simpler version to rank remote results is presented. First of all, in the problem context described in this study it's assumed that the content on the mobile phone can be divided in two sub-types: dynamic/unique(photos taken with mobile phone camera); static/common(music files). It will be rare to have different sites returning the same content. Secondly, it's also considered that the majority of the content will be dynamic/unique due to mobile phone nature. Furthermore, each mobile site can employ it's own human/machine based methods to rank results. With these details in mind it's two ways of ranking the results are proposed: Explicit(Tagging content); Implicit(Machine based methods).

Galanx puts an extra focus on query forwarding in Peer-to-Peer web search context[10]. Traditionally Peer-to-Peer web search studies try to "emulate" the behavior of centralized solutions. Those approaches are completely orthogonal to the one presented in this report. One of the main concepts derived from the social network environment is the ability to navigate through neighbor sites and explore them like in a common social network site where users are able to follow friends' links and explore them. In this case links are created based on the search results. If users are not satisfied with results they can always jump to the next set of nodes and continue searching. In the Galanx case like in a centralized web search only a set of results is provided and the users are unable to explore themselves the network. The sites are presented as fully separated entities, although they can have hyperlinks between them allowing a partial network navigation.

The query forwarding mechanism employed can be described as directed breadth first search with manual iterative deepening. The algorithm is similar to the one described by Garcia-Molina [15] with the exception of using manual iterative deepening. A search is only continued if the user is not satisfied with the results.

Other major source of inspiration was the social network tagging system. Similarly the same principle was applied to the system with minor modifications. Users are able to tag content freely. Some predefined tags related with mobile phone concerns will be always available(e.g., photo location). Generally user tags have only a local significance in the network[5]. The predefined tags try to create general tags present all over the network enhancing the navigation. Each time a user in a site can search for neighbor tags and navigate through them like in the normal web search presented in this report.

Chapter 3

Prototype Requirements

3.1 Requirement Analysis

This section contains the Requirement analysis made for the Mobile search system design. The requirements were extracted from stakeholders' specifications and as preliminary studies of information available on Internet.

The requirement engineering Preview methodologies were used[8] during this stage.

3.1.1 Ambiguities

Some requisites were semantically dubious and their meaning changed slightly during the project execution. It is important to bear in mind that during the different process phases the stakeholder were always present. When it was not possible to get feedback due to minor inconveniences, decisions were taken trying to conform the stakeholder beliefs in order to eliminate the ambiguities.

The following section presents the ambiguities found and their resolution:

- User Level Expertise: The average user only has basic user level knowledge.
- Mobile phone hardware: It's assumed that a mobile phone would have the same memory and computing power as a regular PC.
- Mobile phone software: It's assumed that mobile phones have full support for the LAMP[1] platform.

[1] Stack of software programs used together to run dynamic websites - Linux; Apache; MySQL; PHP

- Threads: The mobile Apache will be single threaded.
- Battery consumption: This item will not have a big weight on the system due to recent technological innovations(e.g., Powercast).
- Number of users: The system is expected to have millions of users.
- Search result set: Each site is responsible to rank it's contents.

3.1.2 Viewpoints

The following viewpoints are proposed :

- Mobile phone user(Admin): The owner of the mobile device.
- Anonymous user: User not authenticated in the system and without permissions to access certain content.
- Authenticated user: User with permissions to access a specific content.

3.1.3 Concerns

The following concerns are proposed :

- Response time : The amount of time the mobile device takes to return a search result.
- Privacy : The system only displays information allowed by the mobile phone owner.
- Scalability : Being able to support thousands of nodes operating independently.
- Compatibility : The system should be able to operate with different software platforms and offer backward compatibility with existing Drupal web sites.
- Accuracy: The relevance of the results returned by the neighbor sites.
- Usability: Ease of use of the mobile search.

Concerns/Viewpoints	Mobile phone user	Authenticated user	Anonymous user
Response time	x	x	x
Privacy	x	x	
Scalability	x	x	x
Compatibility	x	x	x
Accuracy	x	x	x
Usability	x	x	x

Table 3.1: Concern/viewpoint relationship Table

3.1.4 Contribution Tables

In the table 3.1 the relationships between the different concerns and viewpoints are displayed. As can be seen from the table all the major concerns intercept all viewpoints except the Anonymous user view point.

The contribution table displays how the concerns affect each other. The plus sign means that the concerns are affected positively. The minus sign means there will be some kind of a trade off between those concerns.

From table 3.2 it can be concluded that the concern response time affects all the major system concerns. The concerns are affected in the following way:

- Response time is affected by:
  - Privacy: An anonymous user in order to authenticate has to access the server at least two times.
  - Scalability: Due to the single threaded limitation on mobile Apache search queries are made on a serial fashion. Thus the response time grows linearly as the number of queried nodes grows.
  - Compatibility: Processing results from different types of systems can increase the CPU usage.
  - Accuracy: More complex ways of merging the result sets can increase the processing time.
- Privacy is affected by:
  - Scalability: Increased network traffic in order to login in remote sites to do a query.
  - Compatibility: How to authenticate a user in different types of systems.

- Scalability is affected by:
  - Compatibility: The increase in number of platforms that support this feature can enhance the use of this application.
- Accuracy is affected by:
  - Usability: The possibility of organizing the results to the user linking can highly improve accuracy. More relevant information is displayed first to the user.

Concerns	Response time	Privacy	Scalability	Compatibility	Accuracy	Usability
Response time	x	-	-	-	-	
Privacy		x	-	-		
Scalability			x	+		
Compatibility				x		
Accuracy					x	+
Usability						x

Table 3.2: Contribution Table

In the following table the contribution table with weights is displayed.

The weights give an idea how the system will be developed. The bigger the weight more important will be the concern. All the weights are between zero and one. The weights were biased by the stakeholder views during the meetings.

The compatibility has weight 0 because the system will only be used with other Drupal web sites.

Concerns/Viewpoints	Mobile phone user	Authenticated user	Anonymous user
Response time	0.4	0.4	0.5
Privacy	0.15	0.15	
Scalability	0.15	0.15	0.2
Compatibility	0.1	0.1	0.1
Accuracy	0.1	0.1	0.1
Usability	0.1	0.1	0.1

Table 3.3: Contribution Table with Weights

3.2 Requirement Elicitation

This section contains the list of requirements for the Mobile search prototype.

3.2.1 Viewpoints/Requirements

The main requirements related with each major view point are the following.

- Mobile phone user
  - Login.
  - Create new content.
  - Set content and user permissions.
  - Search all the system contents.
  - Search remote neighbors data.
  - Manage neighbor list.
- Authenticated user
  - Login.
  - Create new content if the permissions allow it.
  - Search all the content for that specific permission set.
  - Search remote neighbors' data.
- Anonymous user
  - Login.
  - Create new content if the permissions allow it.
  - Search all the content for that specific permission set.
  - Search remote neighbors' data.

3.2.2 Full List of System Concerns

In order to uniquely identify each requirement, a naming with coding aware convention has been followed. For each requirement the following identifier is present : Type-SequenceNumber where:

- Type is the requirement type and can be:

- FR: Functional requirement.
- SER: Software engineering requirement.

- SequenceNumber: is a sequence number unique to each requirement.
- Prioritization : Each requirement was prioritized using the following enumeration:
  - Must have(MH): The "must haves" define a minimum set of functionalities for the system to be functional.
  - Should have(SH): For requirements that would be classfied mandatory in a less time-constrained development.
  - Could have(CH): For requirements that can be easily left out of the increment under development.

Requirement	Description	Priority
FR-01	The Prototype shall be able to query neighbor sites	MH
FR-02	The Prototype shall be able to query mobile phone data	SH
FR-03	The Prototype shall be able to use the tagging concept	CH
FR-04	The Prototype shall not allow to search certain contents if the user doesn't have enough permissions	MH
FR-05	The Prototype shall allow the users to authenticate themselves	MH
FR-06	The Prototype shall allow the users to create new content	MH
SER-01	The methodology used for analysis and design is UML	SH
SER-02	The prototype must be developed in PHP	MH
SER-03	The prototype shall be developed using free software tools	MH
SER-04	The prototype shall be developed on top of Drupal CMS	MH

Table 3.4: System Requirements

Chapter 4

System Specification for Mobile Web Search Prototype

This chapter presents the Mobile search overview from a user's perspective.

4.1 Instalation Diagram

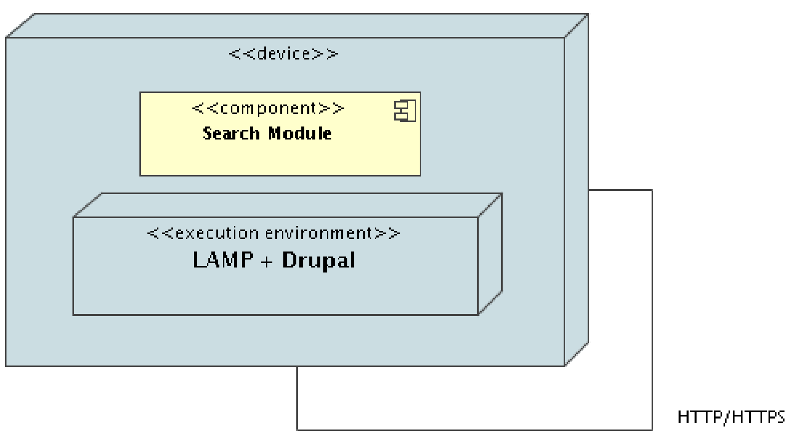

Figure 4.1: Installation Diagram

4.2 Use Cases

The following sections describe the functionalities provided by the Drupal prototype. Also a small UML diagram showing the user interaction with system accompanied by a small description is provided.

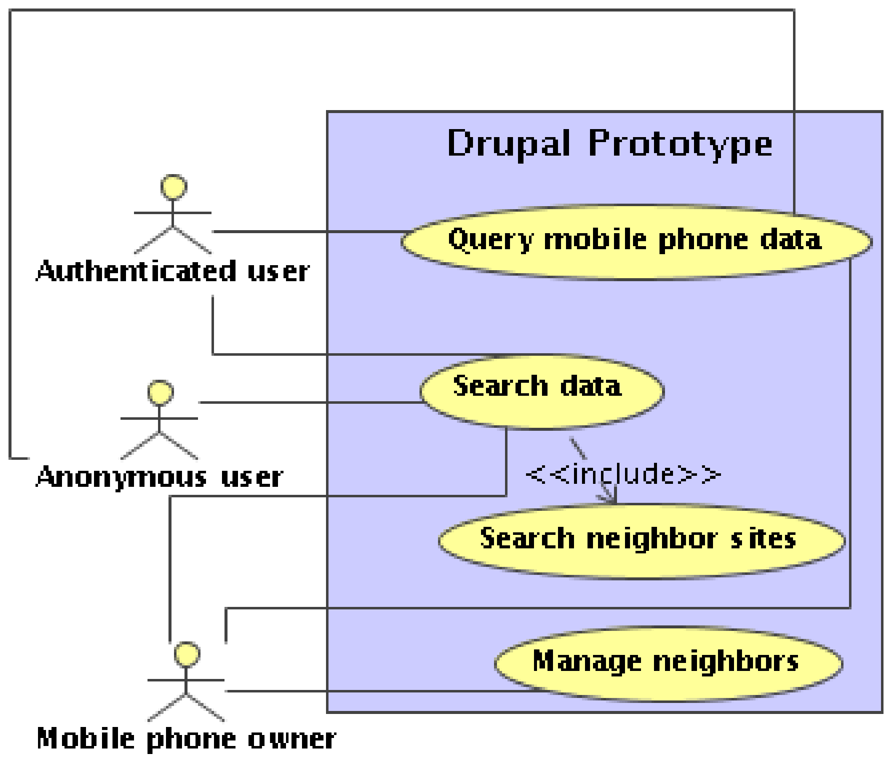

Figure 4.2: Drupal prototype use case diagram

System Actors

- Mobile phone owner: Responsible for managing the mobile website neighbor list. Has access to all contents in the system.
- Authenticated user: User authenticated in the system that can search for content based on the permissions.
- Anonymous user: Non-authenticated user that can search only for content available to anyone.

Use case structured description

Use case: Manage neighbors.

- Actors: Mobile phone owner.
- Pre-conditions: There are no repeated neighbors; A neighbor url written per line.
- Steps:
  1. The user writes a text file with the neighbor addresses.
  2. The users save the file in a specific location.
- Post-conditions:None

Use case: Search data.

- Actors: Mobile phone owner; Authenticated user; Anonymous user.
- Pre-conditions: the query must be at least 3 characters long.
- Steps:
  1. The user enters the search keyword in the search box.
  2. The users presses search.
  3. A local search is done.
  4. The search query is sent to the neighbor sites.
  5. The local result set is merged with the remote result set.
  6. The neighbors are extracted from the remote result set.
  7. Local neighbors are appended to the result set.
  8. The final result set is displayed to the user.

- Post-conditions: None

Use case: Query mobile phone data.

- Use case: Query mobile phone data.
- Actors: Mobile phone owner; Authenticated user; Anonymous user.
- Pre-conditions: The xml files with data file are available.
- Steps:
    1. The user enters the query in the browser navigation bar.
    2. The search result is displayed in a normal result page.
- Post-conditions: None

Chapter 5

Future Work

The concept of mobile search can be easily expanded and integrated as an extension to existing systems.

Query forwarding algorithms should be considered in order to minimize several problems like free riding[1] though in a different setting than previous studies. Algorithms like Ant search[14], K-Random walk, Expanded Ring and Hybrid approaches should be considered. During the elaboration of this work a simulator was also built in order to test those algorithms' behaviors in different settings.

Other way of extending the mobile search functionalities is by creating different ways of accessing the same content. Information could be accessed by a search result or by different entry point. An entry point is a link to a specific content. Tags are an example of creation of different entry points. A different way of creating an entry point is by merging different types of data.

This technology enables the creation of multi social network fusion. With the mobile search the user doesn't need to know exactly where the different entry points are. The returned results will allow exploring vicinities following the links of the different tags or by asking for new results. The same user may present in it's own site several data related to it's own interests. Certain data may only be available to a specific group of users. The data also may be presented in different ways for different groups. This features could be particularly valuable in an enterprise setting. One example would be a fully distributed enterprise portal[6] using the technology described in this report.

Other feature worth exploring is adaptive ranking. Historical behavior of users who conducted similar searches or may have a similar role in an organization may be used to boost document rating. This concept may be expanded if more data is available by creating a profile to generate suggestions for documents based on user context and role in that particular social network[6].

All those features can be tweaked at different granularity's for the different group of users that access the system. For example a user may only generate profiles of work mates in order to make suggestions.

Other topic of interest is search results usability, and new paradigms of displaying different types of information and user interaction. Current Web2.0 may not be fully suitable for mobile devices paradigm of interaction. This could also be an excellent opportunity to use a query language to the web applied to this type of systems for example an adaptation of webSQL[4]. This would likely create a bigger interoperability and homogenization in this type of systems with easier deployment of new functionalities.

Chapter 6

Conclusions

Mobile search is a complement to traditional web search engines. It gives the user means to explore the neighbors contents's by traveling to the friends network topology. It covers a multitude of environments not covered by the centralized solutions.

One of the main advantages in relation to current centralized social network sites is the possibility to manage the site without interference from an external entity. Currently in a normal social network site a user can only display or use modules made available by a third entity. Due to this characteristic it is possible to merge different network sites that cover different topics and create a social network "melting pot". Each user can have what type of content he/she wishes in the site and display different content for different users.

This type of system is better suited for mobile devices due to the always "on" characteristic[13]. Content can be always updated on spot.

Mobile search has an enormous potential to evolve and become a major tool in knowledge management technology. Adaptive Ranking, Role-based recommendations, Locating Experts and Communities [6] can be taken to extreme.

To sum up mobile search can be used to enhance the ability to search for critical information.

Appendix A

A.1 Examples

In this section several examples are presented related to mobile search intrinsic mechanisms.

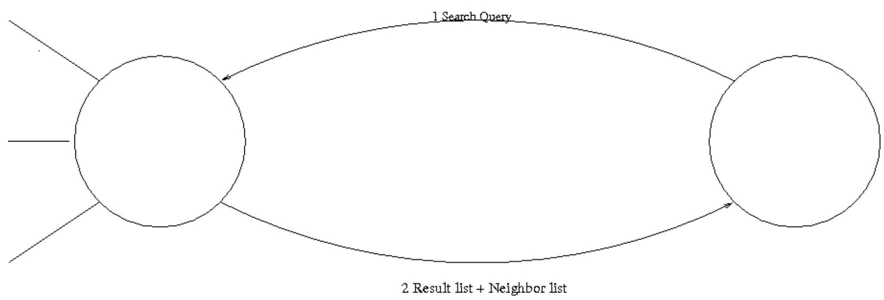

Figure A.1: Mobile search diagram between two nodes

Diagram A.2 shows the schematization of mobile search functionality. The node connections were not explicitly represented instead it's considered that the first level nodes are connected to the source node and the n level nodes are linked to the n-1 level nodes.

The numbers denote the messages sent between the nodes. Messages 1, 3 denote a query request. This message only contains the query String. Messages 2 and 4 are responses to the query request. The response is composed by the search results plus the next level neighbors as it can be seen in Figure A.1. The numerical values denote the order of the messages.

The figure A.3 shows a possible intersection of topics in a network. Each topic has a different color. For simplicity it is assumed that nodes are connected to nodes within a topic. It's possible to have topic partitions

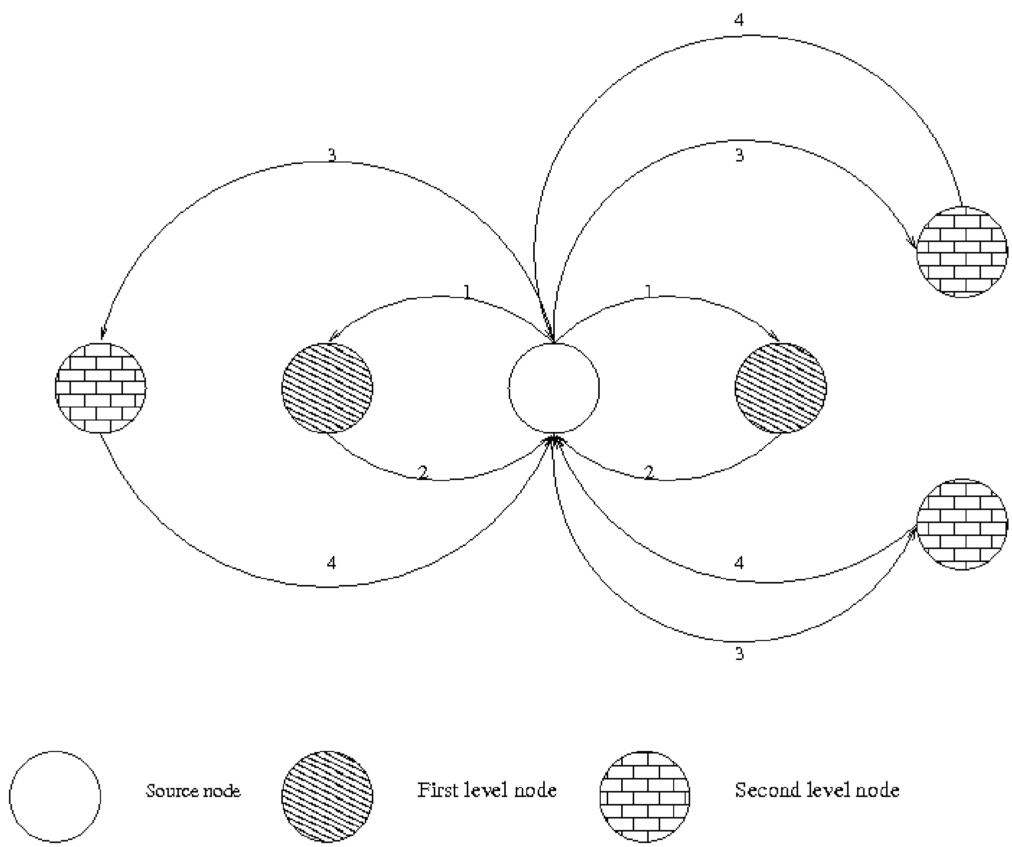

Figure A.2: Mobile search diagram

but when users have overlapping interests they create a bridge that allows to reach information in different islands.

Figure A.4 shows the representation of the tagging concept. The black dot represents the source node who issued a query searching for the tag Portugal. The figure are represents the results returned by the neighbor nodes in different network levels (each image corresponds to a neighbor who returned a result).

For example if the source node issued the query Portugal it would obtain six results. If then the user chooses to navigate by the tag Lisboa he would get one result(the Trolley image). If instead the user chooses the keyword Portugal he would get three results(The trolley, the caravel and Figo).

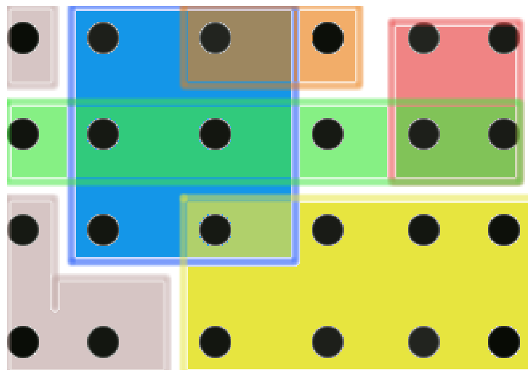

Figure A.3: Data islands

A.2 Prototype code API

Slight changes were made to the Drupal search module.

A.2.1 DomXmlDocument

Object that managed the mobile phone content queries.

A.2.2 ExternalSearchEngine

Object responsible for the multicrawler part of the prototype described earlier.

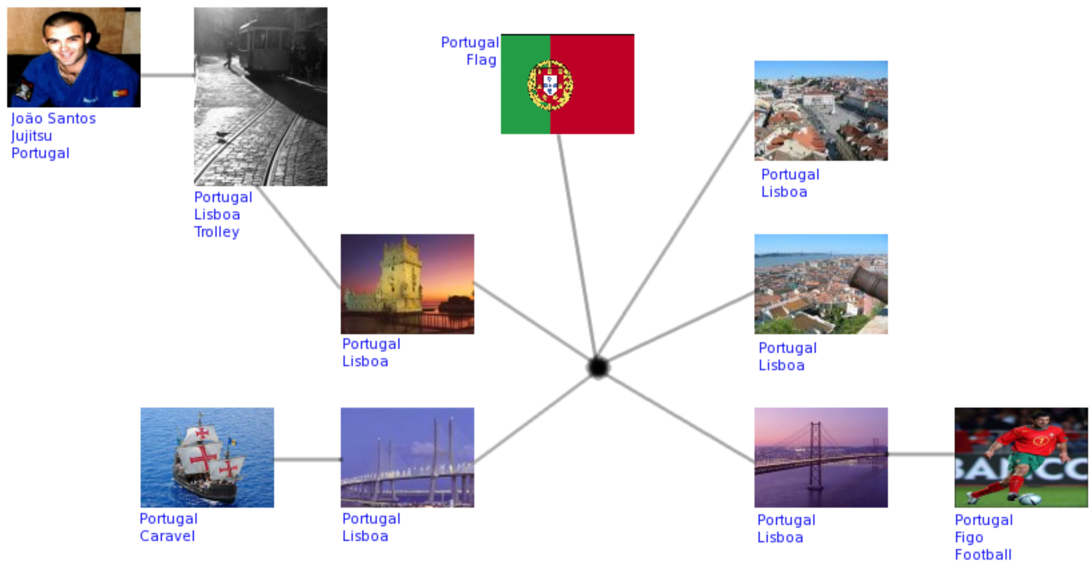

Figure A.4: Tagging concept

A.3 Drupal Prototype Examples

In this section several examples related with the Drupal prototype will be presented. The pictures were retouched in order to emphasize certain details and improve the overall readability.

In figure A.5 can be seen the network topology representation used in the following example.

Let's assume a user in mobilewebsearch node issues the query Jesus. The figure A.6 displays one possible way of sorting the results and displaying them to the user. The results on the neighbor sites of mobilewebsearch are also displayed in a colorful frame in order to pinpoint their origin.

The next examples will use the network topology shown on figure A.7. If a user issues the query "Era" as an anonymous user in mobilewebsearh3,

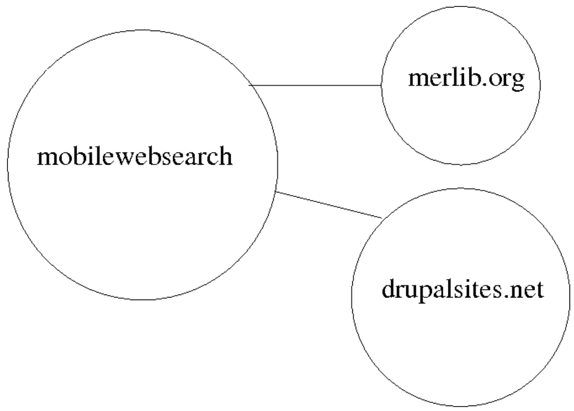

Figure A.5: Network topology.

he/she will get the result list on figure A.8. On the other hand if the user authenticates as PT and issues the same query the result list will contain one extra result. This result is only available to a specific set of trusted users that can be remote or local. If an anonymous user issues a similar query from mobilewebsearch2 the result list will be empty as it can be seen in figure A.10. If the user presses the "Get more results" link he/she will obtain a similar list to figure A.8 as shown in figure A.11.

Let's assume a user authenticates as PT in mobilewebsearch2 issues the query "Era". He/She gets one result from mobilewebsearch3 because that node trusts the user PT from mobilewebsearch2 as shown in figure A.12. Afterwards if the user presses the link "Get more results" the result list will be similar to the ones obtained as an anonymous user in mobilewebsearch3 and mobilewebsearch2 because the neighbors of those sites don't trust the user PT from mobilesearch2 as displayed in figure A.13.

The figures A.14, A.15, A.16, show the sequence of results obtained by an anonymous user after issuing the query "fear" on mobilewebsearch and pressing the "Get more results" link until there are no unvisited nodes.

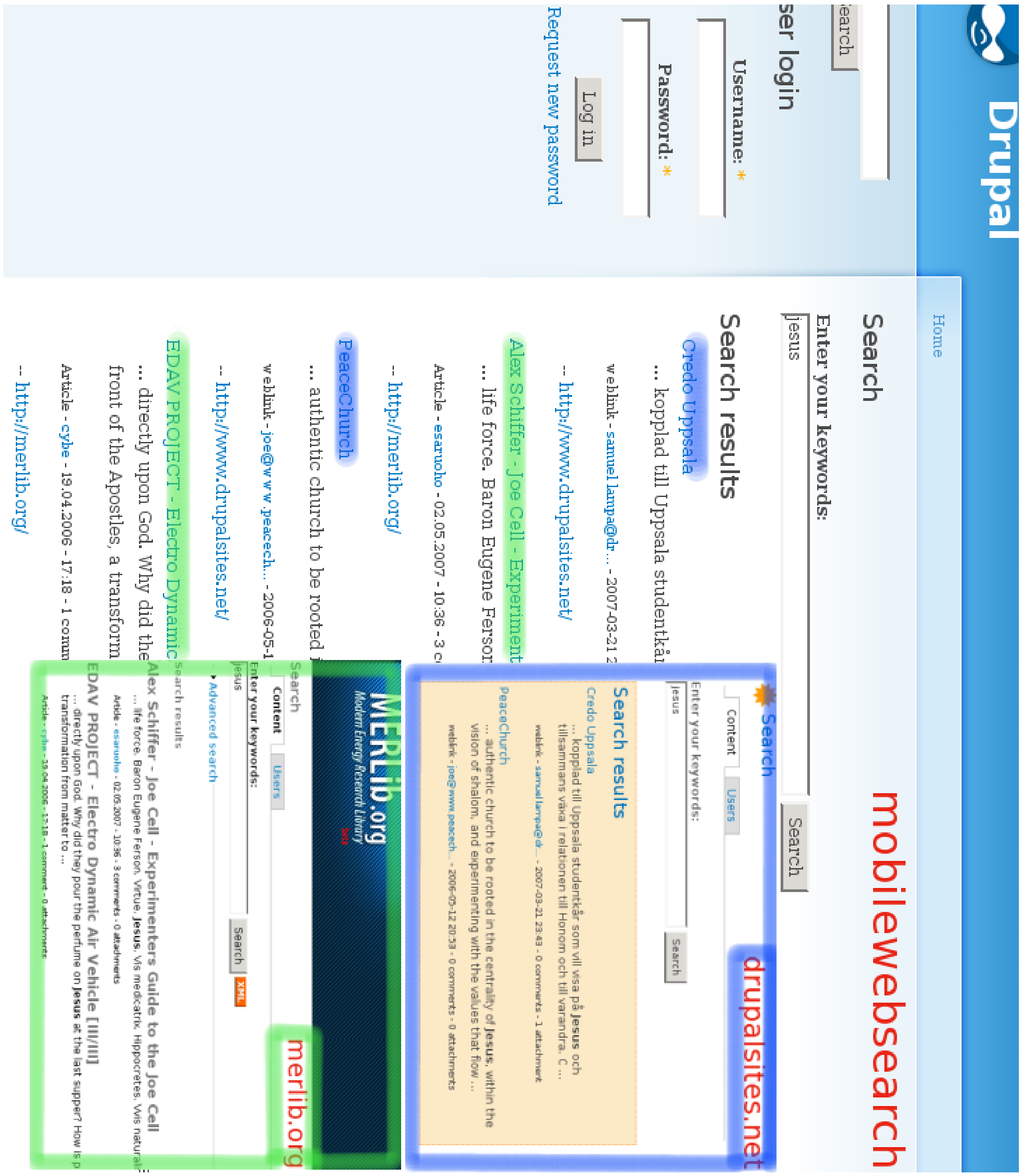

Figure A.6: Drupal web site results for the keyword Jesus.

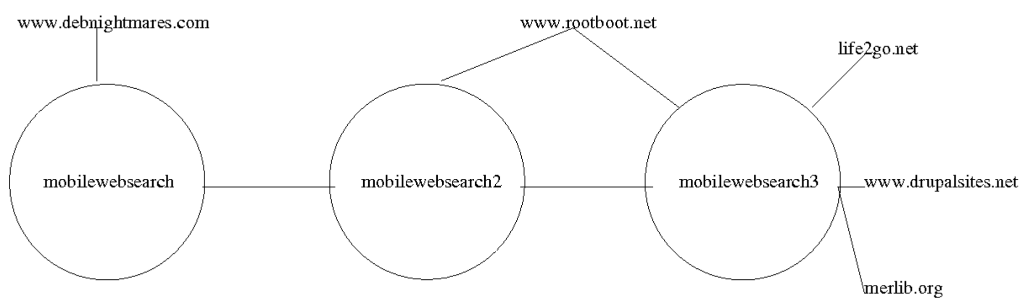

Figure A.7: Network topology.

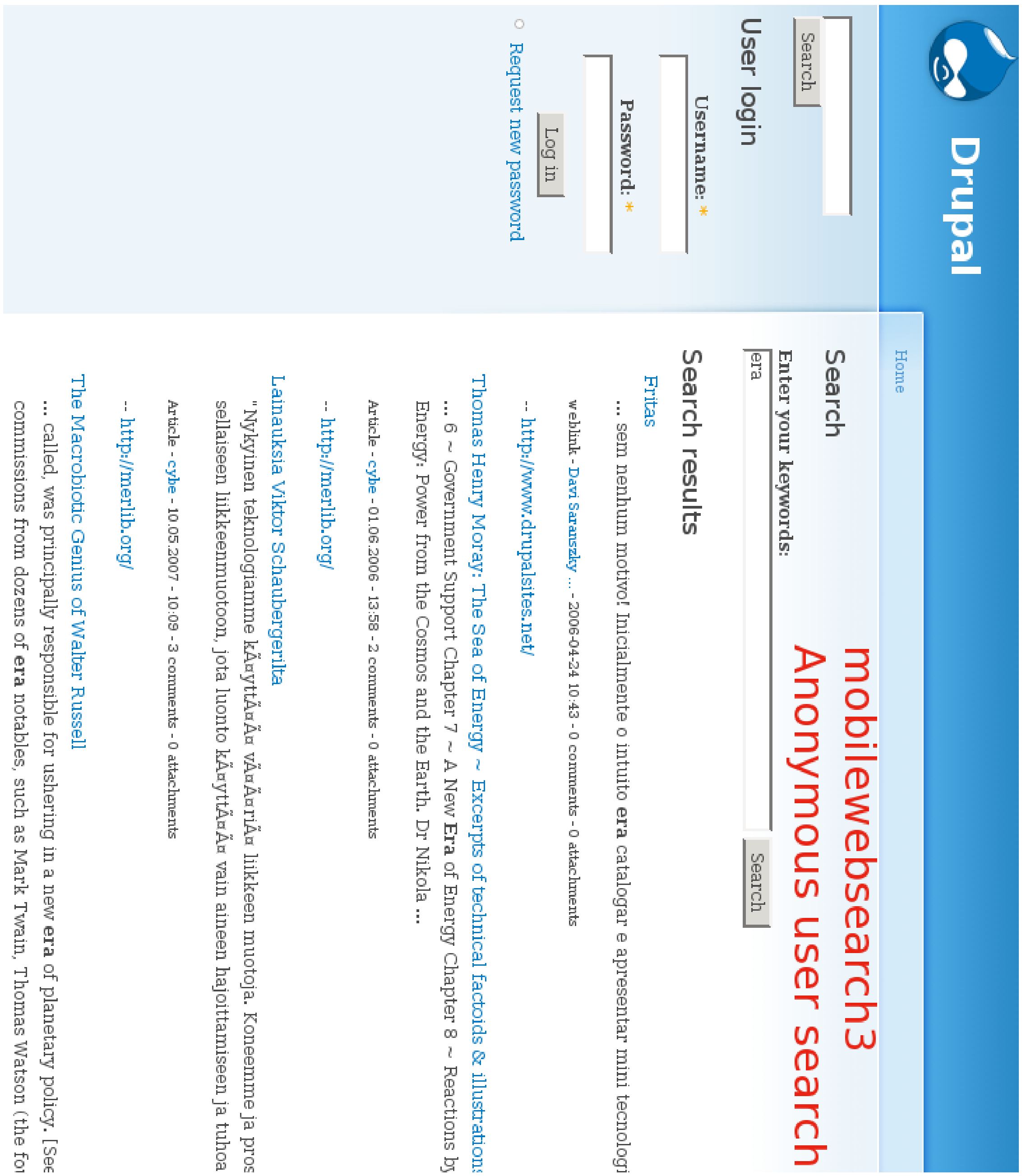

Figure A.8: Search result in mobilesearch3 for the query Era as anonymous user.

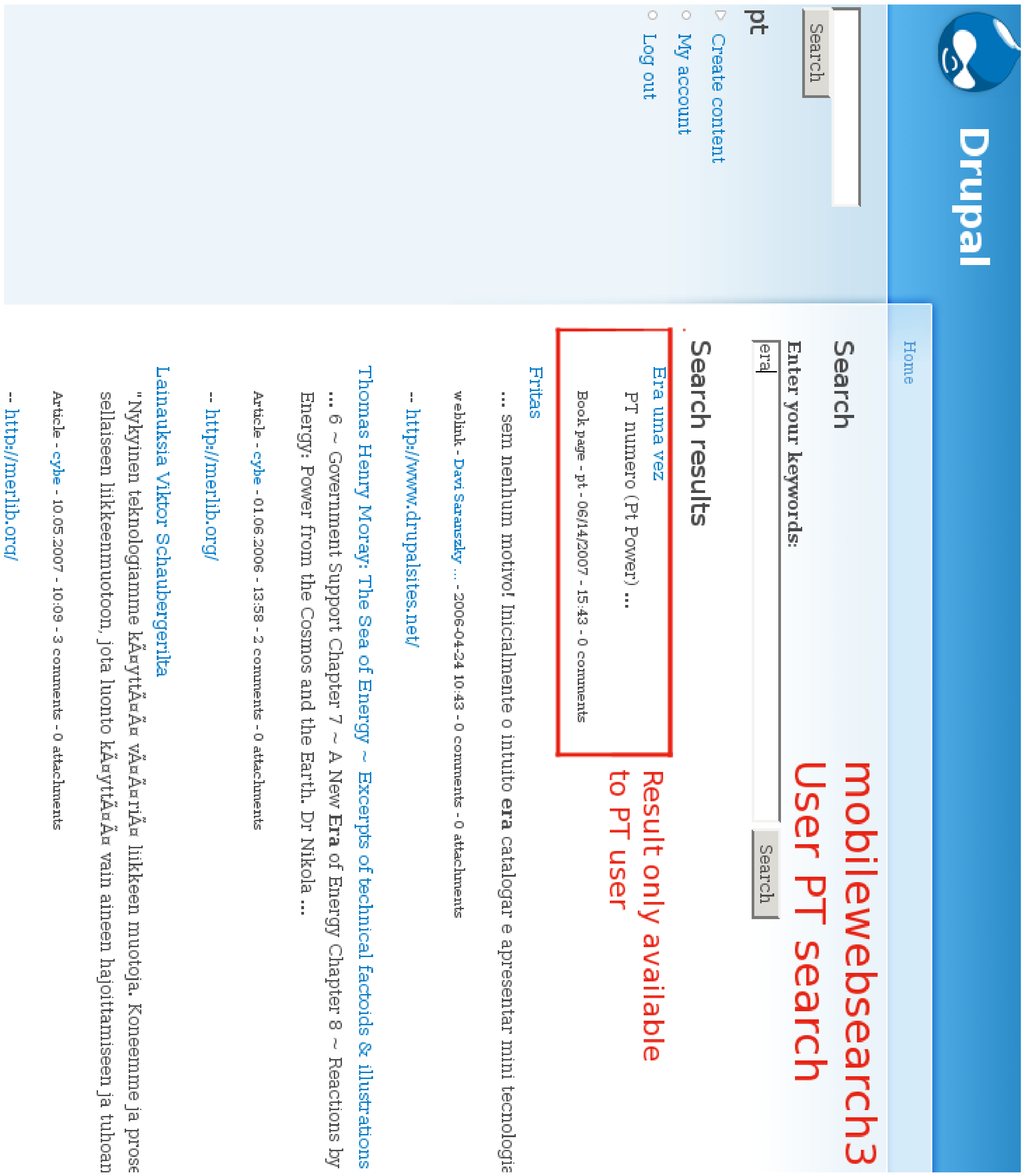

Figure A.9: Search result in mobilesearch3 for the query Era as PT user.

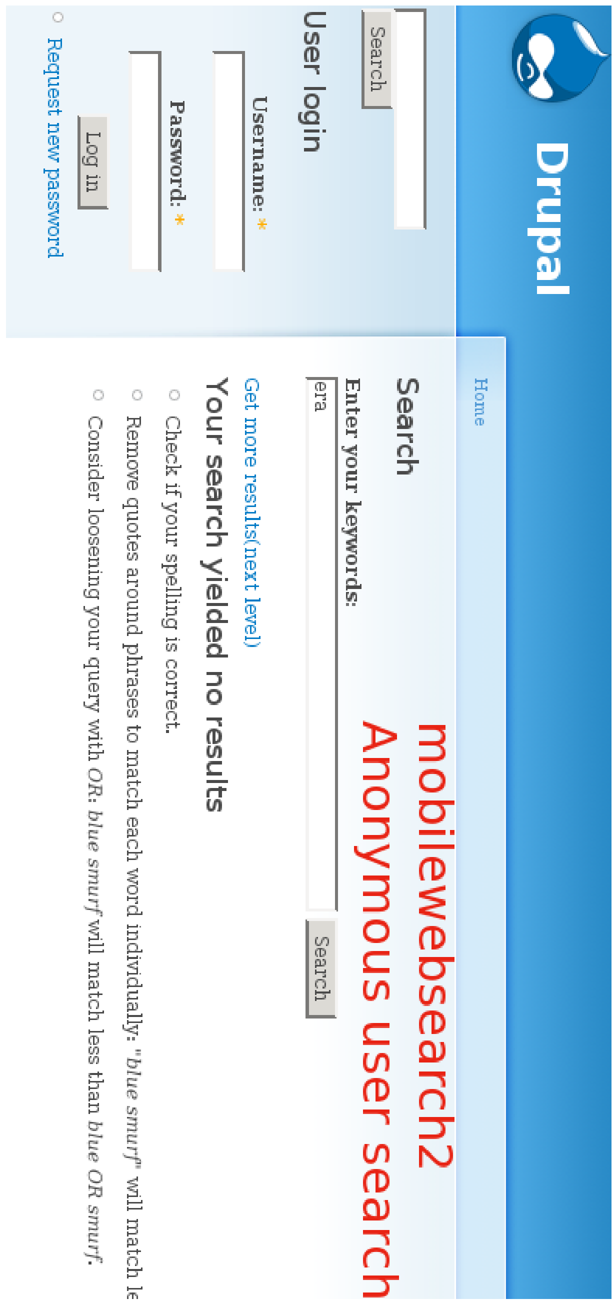

Figure A.10: Search result in mobilesearch2 for the query Era as anonymous user.

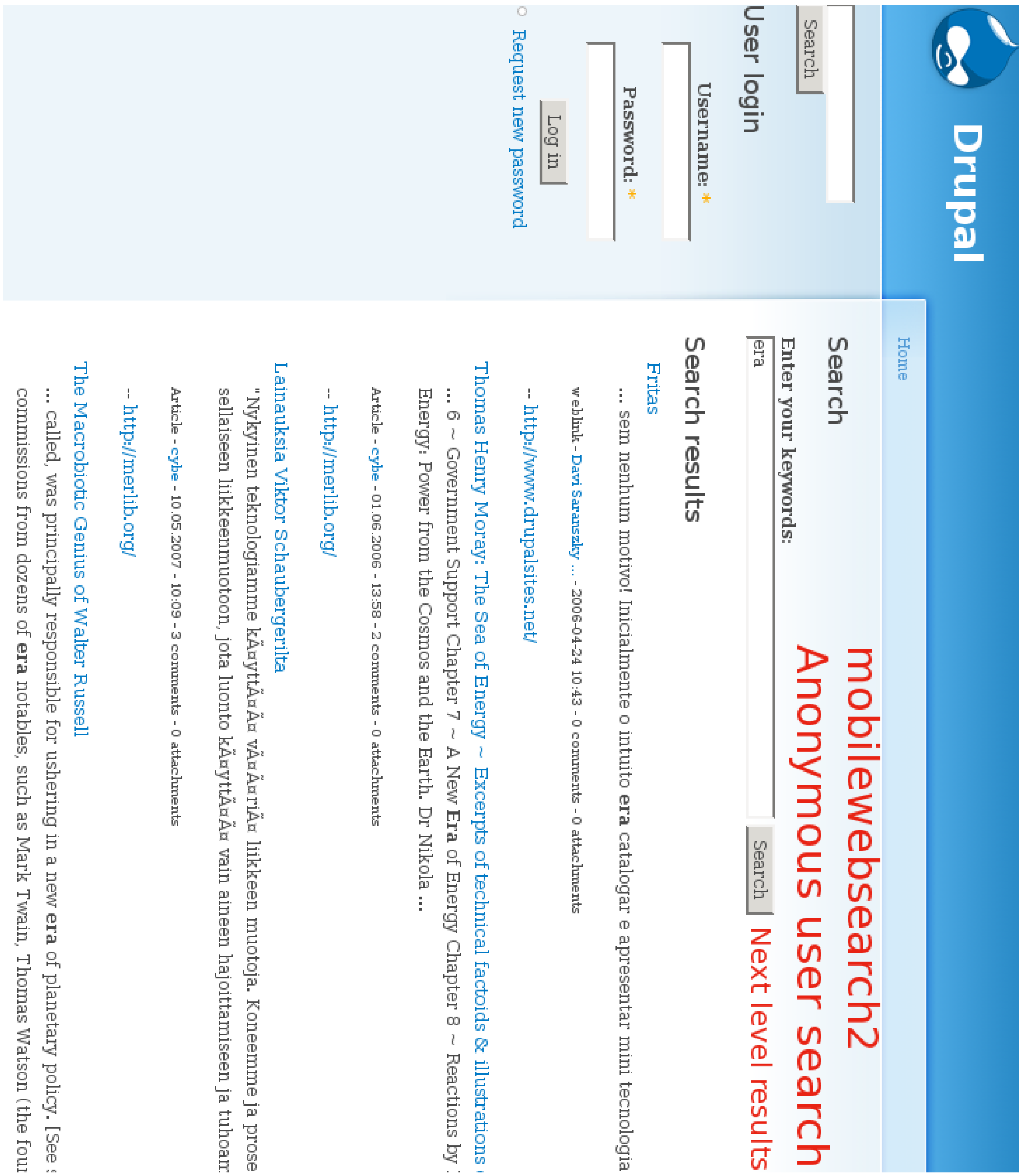

Figure A.11: Search result in mobilesearch2 for the query Era as anonymous user - First level.

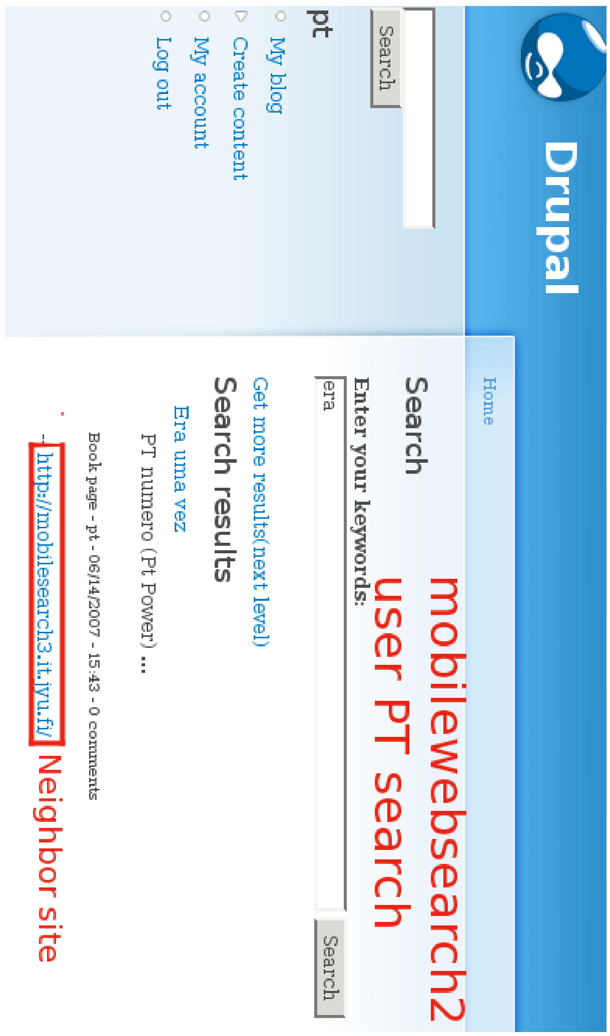

Figure A.12: Search result in mobilesearch2 for the query Era as PT user.

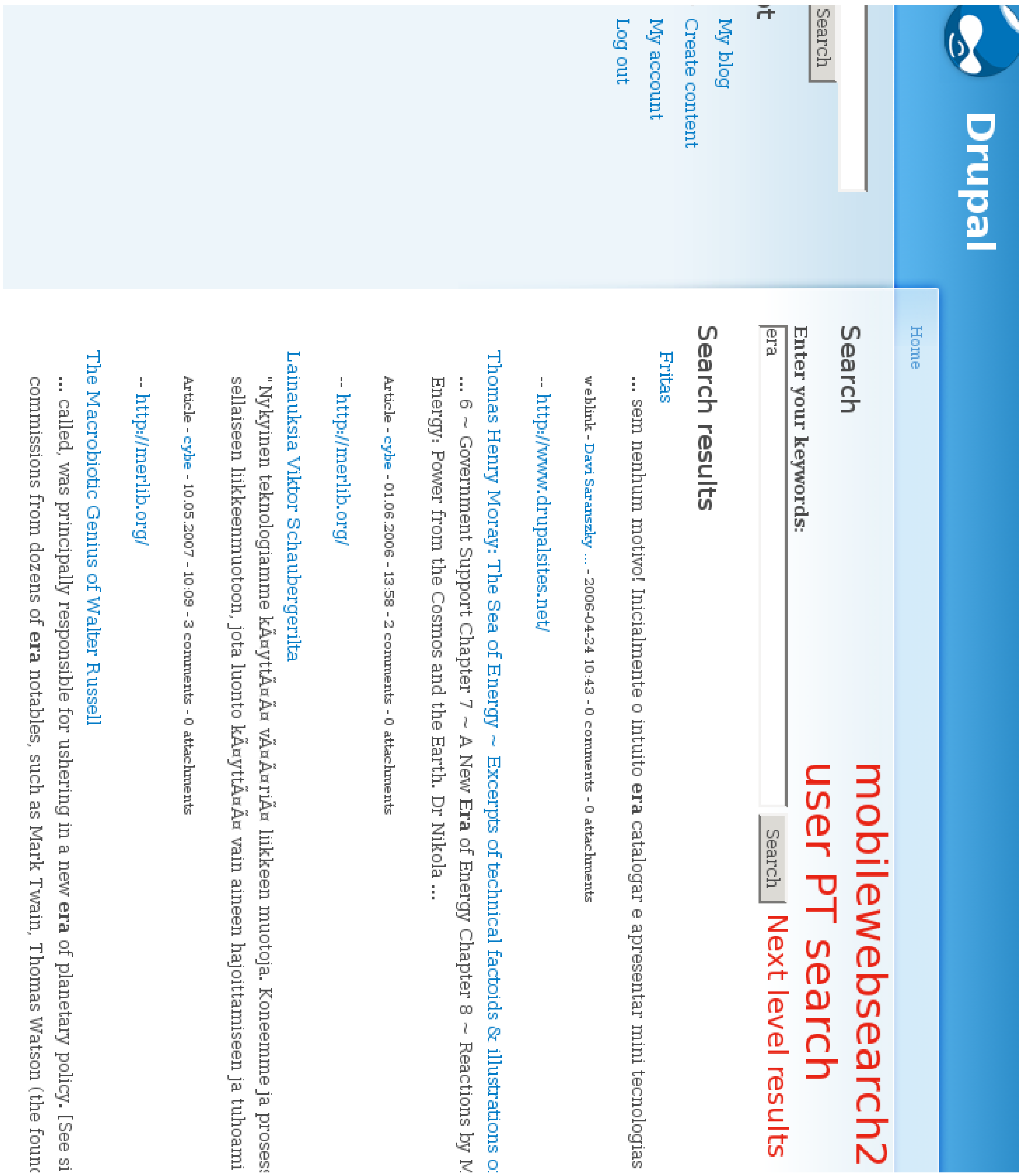

Figure A.13: Search result in mobilesearch2 for the query Era as PT user - First level.

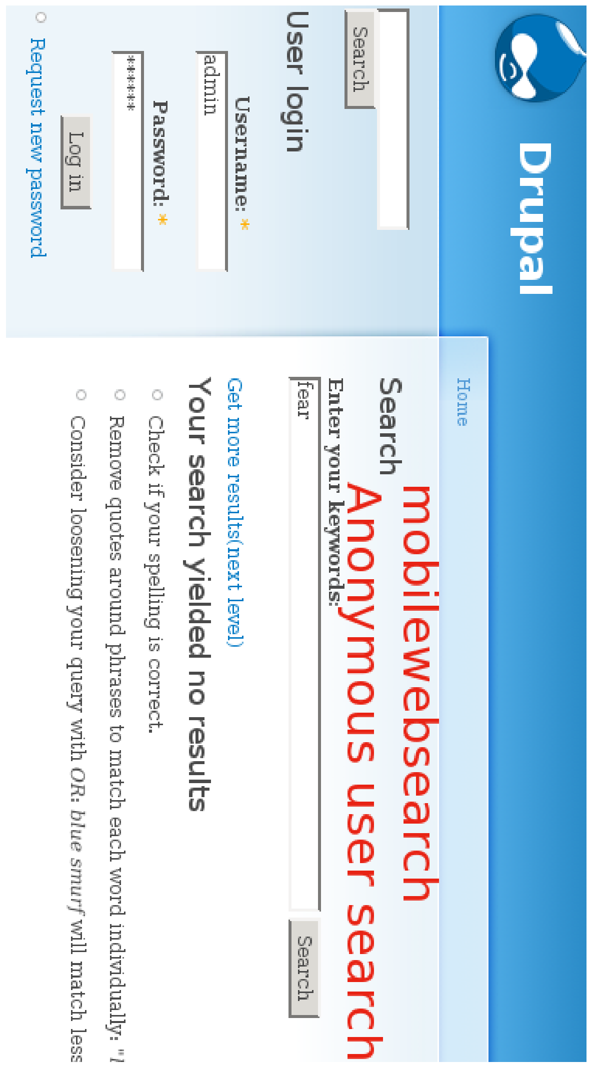

Figure A.14: Search result in mobilesearch for the query fear as an anonymous user.

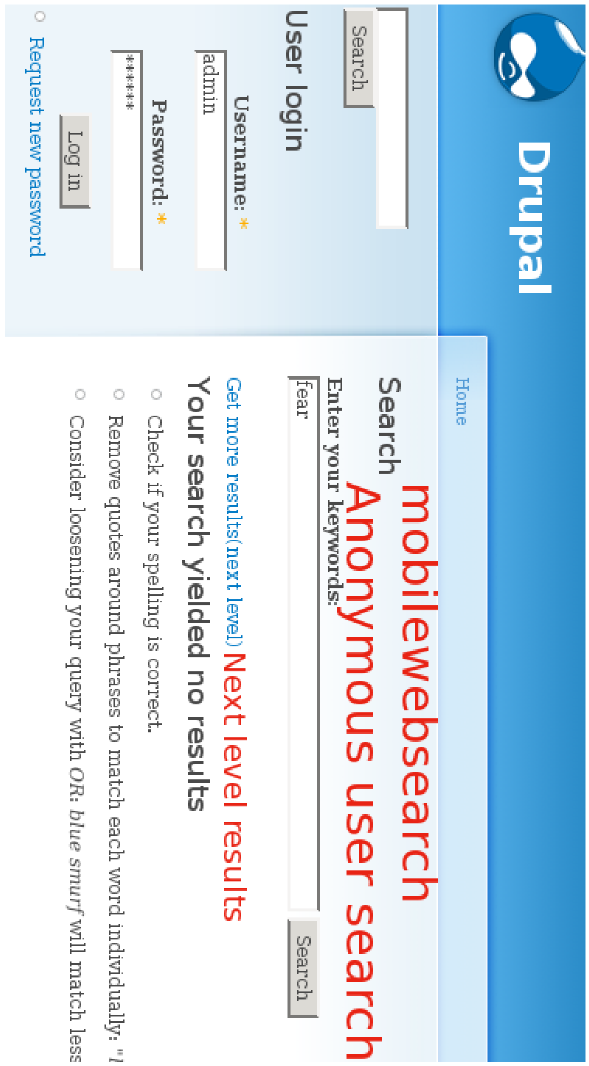

Figure A.15: Search result in mobilesearch for the query fear as an anonymous user - First level.

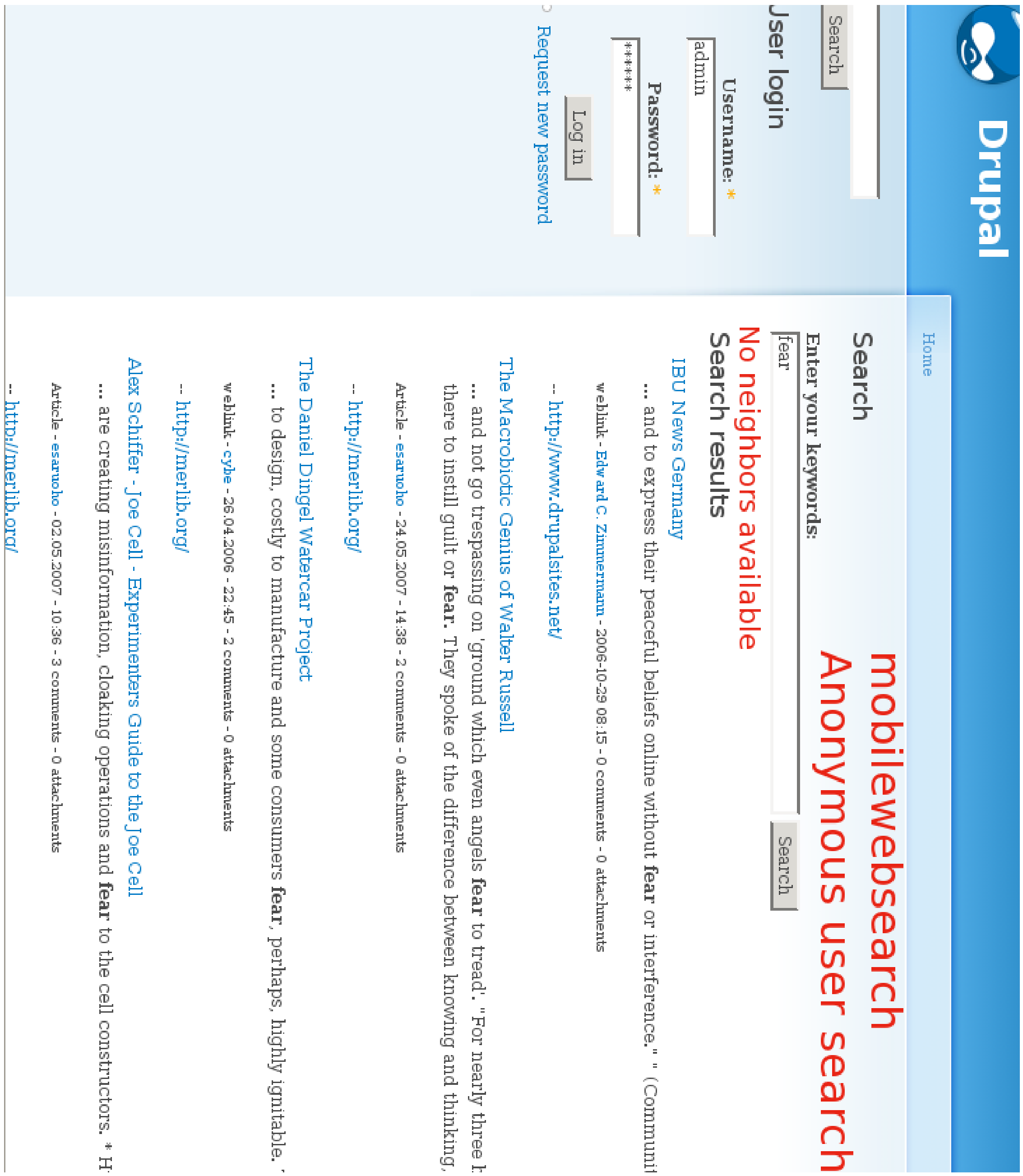

Figure A.16: Search result in mobilesearch for the query fear as an anonymous user - Second level.

Bibliography

[1] E. Adar and B. Huberman. Free riding on gnutella. *Technical report, Xerox PARC*, 2000.

[2] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[3] Elizabeth F. Churchill and Christine A. Halverson. Social networks and social networking. *IEEE Internet Computing*, 9(5):14–19, 2005.

[4] Alberto O. Mendelzon, George A. Mihaila, and Tova Milo. Querying the world wide web. *Int. J. on Digital Libraries*, 1(1):54–67, 1997.

[5] John C. Paolillo and Shashikant Penumarthy. The social structure of tagging internet video on del.icio.us. In *HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, page 85, Washington, DC, USA, 2007. IEEE Computer Society.

[6] Prabhankar Raghavan. Social networks from the web to enterprise. *IEEE Internet Computing*, 6(1), 2002.

[7] Patrick Reynolds and Amin Vahdat. Efficient peer-to-peer keyword searching. In *Proceedings of International Middleware Conference*, pages 21–40, June 2003.

[8] P. Sawyer, I. Sommerville, and S. Viller. Preview: Tackling the real concerns of requirements engineering. *Cooperative Systems Engineering Group, Technical Reports*, 1996.

[9] Hirokazu Tomiyasu, Takuya Maekawa, Takahiro Hara, and Shojiro Nishio. Social network applications using cellular phones with e-mail function. In *Proceedings of 21st International Conference on Data Engineering*. IEEE Computer Society, 2005.

[10] Yuan Wang, Leonidas Galanis, and David J. DeWitt. Galanx: An efficient peer-to-peer search engine. *Technical report, University of Wiscosin - Madison*, 2003.

[11] Johan Wikman. Mobile web server - eurooscon presentation, 2006.

[12] Johan Wikman and Ferenc Dosa. Providing http access to web servers running on mobile phones. *Technical report, Nokia Research Center*, May 2006.

[13] Johan Wikman, Ferenc Dosa, and Mikko Tarkiainen. Personal website on a mobile phone. *Technical report, Nokia Research Center*, 2006.

[14] Chi-Jen Wu, Kai-Hsiang Yang, and Jan-Ming Ho. Antsearch: An ant search algorithm in unstructured peer-to-peer networks. In *ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications*, pages 429–434, Washington, DC, USA, 2006. IEEE Computer Society.

[15] Beverly Yang and Hector Garcia-Molina. Improving search in peer-to-peer networks. In *Proceedings of 22nd International Conference on Distributed Computing Systems*, Viena, Austria, 2002.

[16] Jin Zhou, Kai Li, and Li Tang. Towards a fully distributed p2p web search engine. In *Distributed Computing Systems, 2004. FTDCS 2004. Proceedings. 10th IEEE International Workshop on Future Trends of*, pages 332–338. IEEE Computer Society, 2004.