Mobile Multi-domain Search over Structured Web Data

Atakan Aral[1], Ilker Zafer Akin[2], and Marco Brambilla[2]

[1] Department of Computer Engineering, Istanbul Technical University, Istanbul, Turkey
aralat@itu.edu.tr
[2] Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milano, Italy
ilker.akin@mail.polimi.it, mbrambil@elet.polimi.it

Abstract. Text-based web search that is primarily designed for personal computers, can be enhanced and optimized while moving to mobile devices. New methods on web search may let user conduct the search without being hampered by the limitations of the device. Moreover, appropriate solutions may also exploit the advantages of such devices. This paper summarizes new trends and technologies of searching, especially multi-domain and exploratory search, as well as demonstrating how they can be best applied to mobile environments.

Keywords: Web search, web-based mobile application, multi-domain search, search computing.

1 Introduction

Searching is the key activity of web browsing. While basic text-based search had been acceptable until recently, technological advances such as broadband internet connectivity, device mobility and trends such as Web 2.0 and semantic web have led to higher expectations. In this paper, we aim to propose design and implementation ideas as well as introduce a novel mobile search application in order to meet most of these expectations. Covered topics can be categorized roughly in four areas: mobile search, multi-domain search, exploratory search and enhanced presentation of results.

Web integrates to our daily life more than ever with the introduction of smart phones and tablet PC's. It is essential for searching to evolve to keep up with the mobile migration that offers both new opportunities and threats. Mobile search allows users to search for information anywhere and anytime, moreover search experience can be enriched with location data which is made available by most mobile devices. On the other hand, factors such as screen size and input method complicate the search process.

Multi-domain search is about handling data coming from different semantic fields of interest. When the user enters the query "good physic conference October 2012 Milan reasonable 5-star hotel", a multi-domain search application

should perform three searches for three semantic fields (conference, city and hotel) and somehow relate them to each other. One solution for this specific case is to build a list of physic conferences in Milan in October 2012 along with nearby 5-star hotels and order it according to conference rating, hotel price, proximity, etc.

Exploratory search proposes that user should be aided in formulating his/her interest, in exploring most relevant and credited information sources and in correlating the elements of those sources. These can be accomplished by asking the user to choose a topic then specialize on the topic and information sources step by step, and finally asking for input data specific to that sub-topic. It is also possible to offer results from related topics during or after the search process in order to allow user enhance the query. Exploratory search comes into prominence especially while using mobile devices as it allows constructing more complex queries with less textual input and in shorter duration.

A natural outcome of multi-domain and exploratory search is the requirement of more advanced ways of presentation. Result sets from different semantic fields should be treated differently and presented through different interface elements such as maps, lists or tables. Moreover, these elements should also allow users to filter and sort results according to various criteria as well as further specify their query.

Our focus is the exploration of applicable solutions for recent and innovative ideas on web search including mobile search, multi-domain search and exploratory search. We also intend to demonstrate how such solutions can work together in order to enhance and ease search process for complex needs on mobile devices and what kind of user interface elements can be used to support them. Moreover, a practical application of the discussed solutions is presented to clarify technical issues.

Remaining of this paper is structured as follows: Section 2 summarizes the studies in related areas to give an idea about the state of the art. Next section, explains our proposed ideas for the topics explained here as well as illustrating how these ideas were implemented in a web based mobile application. Finally, remaining two sections conclude the solutions and discuss future research directions respectively.

2 Related Work

Web search is a thoroughly researched field and there is extensive amount of studies. In this section, we mainly focus on studies in three categories: (1) Multi-domain and Exploratory Search, (2) Search Computing, (3) Mobile Design.

2.1 Multi-domain and Exploratory Search

Multi-domain queries are defined as queries that are over more than one "semantic fields of interest" [1]. There exists domain-specific search engines but they are applicable to only one domain. Multi-domain search engines, on the other hand,

intend to automatically combine the results of domain-specific searches and provide answers originating from various domains. Without multi-domain search, only expert users can access such an answer by conducting individual searches on different domain-specific search engines and manually combining findings, which is an exhausting and time-consuming work.

A Model for the search process by a multi-domain search engine is described by Bozzon et al. [2]. It begins with query submission, proceeds with query computation and ends with result visualization. In query computation phase, which is on our focus in this sub-section, search engine first needs to identify one or more domains referred in the query. In their approach, domains are predefined and each domain has a set of attributes to define it. In addition, there are domain-specific sub-engines for each domain and they are invoked when a query is identified to be related to that domain. Thus, a multi-domain search engine can be considered as an integration of many domain-specific search engines.

Marchionini [3] introduced the idea of exploratory search which "blends querying and browsing strategies from retrieval that is best served by analytical strategies". He categorized search activities into three overlapping groups: (1) Lookup search where the user simply needs "fact retrieval" or "question answering" and the returned answers are discrete and well-structured; (2) Learn search that returns objects in various media forms and used for cognitive processes such as examining, comparing and making judgments; (3) Investigate search that takes longest time and includes critical assessment of results. While current systems are quite adequate in answering lookup search queries, latter two groups require more human participation. Exploratory search aims to include more human interaction into the search process by means of interactive user interfaces.

In a recent application of exploratory search [4], user begins the search process with an initial topic and then progressively develops it by discovering his/her needs and exploring additional related information. To achieve this, user selects the initial topic from a list and inputs a query. Then, among the top ranked results displayed for that query, user chooses the one he/she is most interested in and the system offers additional related topics to explore. Development of the result set continues with the topic the user selected and this time results are ranked considering previously chosen results as well.

A new paradigm called, Liquid Query [1] is proposed for multi-domain and exploratory search over structured information sources. It aims to allow searchers to develop their query by adding another search service, requesting more results specifically from a certain search service, ordering or filtering results, changing the visualization type etc. in order to get closer to the desired information step by step. While the user makes such changes in the query, result set dynamically accommodates to the modified query.

2.2 Search Computing

Goal of the Search Computing (SeCo) Project is to construct a platform to address multi-domain queries by integrating various search services [5]. SeCo

provides an alternative to the conventional web crawling and indexing techniques of horizontal search engines that are not adequate for multi-domain search [1].

Architecture of the SeCo system, described by Brambilla and Ceri [6], contains two activity flows: (1) Registration flow is used by administrators and responsible for the addition and configuration of domains and search services, (2) Execution flow is used by final users and processes the queries. As the domains and related search services are entered by the administrator, and they are chosen by users, automatically identifying domains from the user query is not a concern in SeCo applications.

Two-tier (client and server tiers) and three-layer structure of the SeCo is explained by Bozzon et al. [4] and Campi et al. [7]. At the conceptual level, web objects or domains are represented by Service Marts which hide the underlying physical structure and provide a simple interface. A Service Mart has a name and both atomic and repeating attributes. Each attribute can be input or output depending on which Access Pattern is used at the logical level. A Service Mart may have multiple Access Patterns. Finally, there are Service Interfaces at the physical level and they mapped to specific concrete data sources. As an example, Cinema Service Mart may have name, address, city, country atomic attributes and movies repeating group. Name attribute may be an input value in one access pattern to allow user search by name. Same attribute can also be an output value in another Access Pattern where the user searches by address, city and country inputs. Under each Access Pattern there may be multiple Service Interfaces which are mapped to search services (such as IMDb.com or Yahoo! Movies) supporting that combination of inputs and outputs. Another notion to allow combination of Service Marts is Connection Patterns. A Connection Pattern is characterized by two coupled Service Marts and the logical connection between their attributes. A Cinema and a Restaurant can be related in a Connection Pattern according to their geographic proximity by having the same values for country, city and address attributes.

2.3 Search Interfaces

In her book [8], Hearst provided a comprehensive background research about mobile search interfaces and proposed dynamic term suggestion, query anticipation and spoken queries to overcome input difficulty in mobile devices. In addition, presenting alternative visualization methods for certain types of results (for example, map for results containing location data) is encouraged.

In addition to the mobile device issues, multi-domain search adds other challenges to the visualization problem [9]. One important factor is that, search results do not have to correspond to a web page and it may be a combination of objects from the web. As a consequence, result set for multi-domain query can be highly dimensional. Scheme of their proposed solution for these problems can be summarized in four steps: (1) Most relevant dimension is chosen. (2) Best visualization method for that dimension is identified. (3) All dimensions that are applicable for that method are visualized. (4) Repeat for the remaining dimensions. They exemplify their approach with map view for geo-referenced objects,

timeline view for time-located objects, and other methods when suitable interval dimensions are not available.

Church et al. [10] examined seven mobile search engines and argued that simply applying traditional query-based search and list-based result presentation in mobile cases is not optimal. They proposed an approach called search result gisting which aims to generate shorter but more informative result snippet texts by making use of terms from previous queries that have led to the selection of that result. They also evaluated their approach on a user base and validated usefulness of it.

3 Mobile Exploratory Search

In this section, we propose solutions for the topics presented in Introduction and Related Work sections and demonstrate them on a real world application. Our aim is to increase usability and functionality of mobile search by applying these solutions. Here we briefly explain requirement analysis, design and implementation phases of the application. More about the technical details can be found in the M.Sc. thesis by Akin and Aral [11].

3.1 Requirements

First requirement we want to satisfy is that the application should work on mobile computers such as smart phones and tablet computers. Compatibility for most kinds of devices and multiple browsers is also a desired feature.

Another need is the support for multi-domain and exploratory search. These two terms were introduced thoroughly in Introduction and Related Work sections. To sum up with one sentence: the application should be able to search for combinational results from multiple semantic fields in an incremental fashion and should guide the user during query development.

User should be able to visualize search results in multiple, customized perspectives in order to determine the best result for him/her. Moreover, application should also be capable of storing linked results from various semantic fields and presenting to the user clearly.

Last but not the least, user interface should be as natural as possible since the main target of the application is the end user who cannot easily conduct separate searches for different domains and combine the result. It should be straightforward enough to search in a domain, examine the results, and combine them with another domain without any necessity of tutorial.

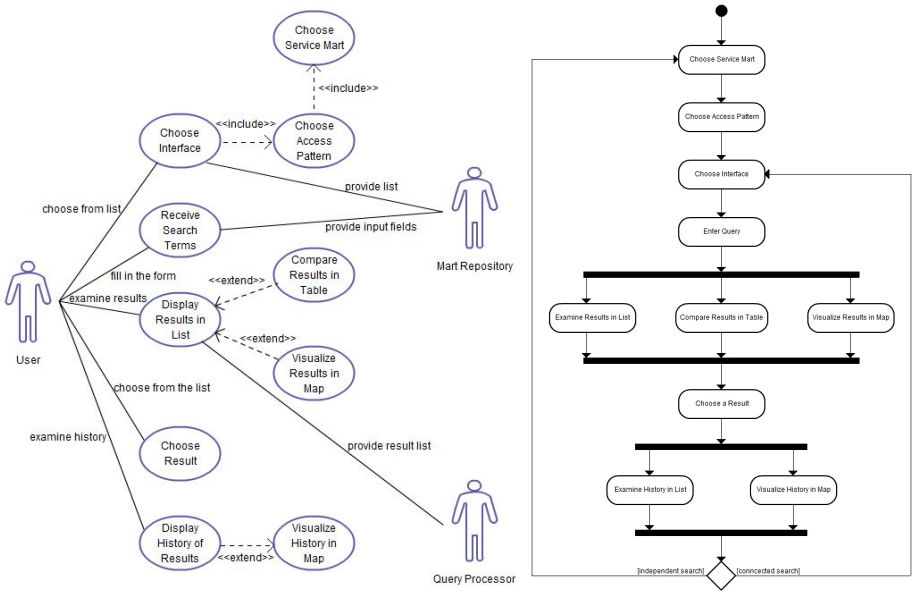
3.2 Overview

Overall picture of the application can be considered as a loop of connected domain specific searches. User searches for a result from a single domain at a step of the loop but previously chosen results from other domains may also affect the result set.

First, user decides which domain his/her initial search belongs to, by which input he/she will search and which information source will be used. Then, user enters the input data and submits the initial query. Once the user examines the result set and decides which result is most convenient, he/she can move to the second step of the loop by selecting a related domain. Same operations should be carried out at each step until the user includes all the domains to the query and is satisfied with the chosen results. As a matter of course, result set of the domain specific searches except the first one also depend on what user have selected in previous searches and in what ways the domains of these selections are connected to the current domain. In between each step, user is presented with an overview of the ongoing multi-domain search. This overview is stored for later reference or modification until user discards it to launch a new search.

3.3 Engineering

Design phase of the development is summarized in this subsection by means of UML diagrams. Actors, their actions and dependencies among the actions are described by the Use Case diagram in figure 1(a). On the other hand; Activity diagram in Figure 1(b) describes the overall workflow steps of the system and the control flow. Finally, figure 2 describes the components of the systems and their dependencies inter se.



(a) Use Case diagram (b) Activity diagram

Fig. 1. Use Case and Activity diagrams of the application

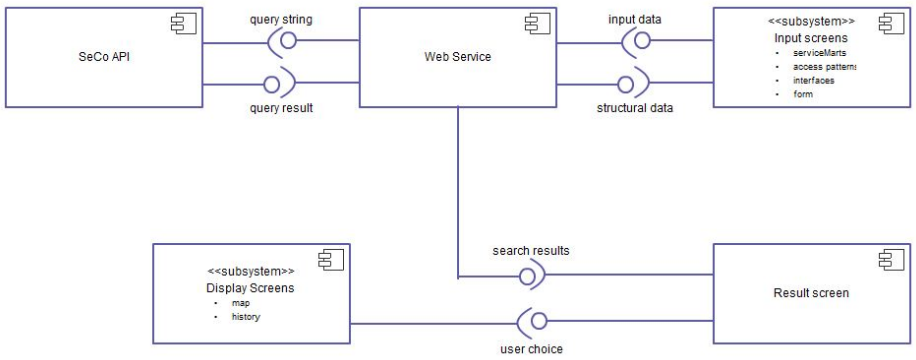


Fig. 2. Component diagram of the application

3.4 Implementation

Web-Based Mobile Application Web search, as the name suggests, conventionally carried out in web pages through a browser instead of standalone applications. This allows users to visit web pages linked among the results in the same context and without switching to the browser. We follow the convention in order to ensure that the user searches in a natural way on their mobile browsers.

Our application is developed as a web application optimized for mobile devices and browsers instead of multiple device-specific native applications. This choice is also motivated by the new opportunities provided by HTML5 and client-side technologies such as JavaScript, AJAX, and CSS. These technologies allow web-based applications to make use of most features of mobile devices like GPS adapter or Camera. Moreover, web-based mobile applications are cross-device which redeems developers from the complexity of developing native applications for each device.

A domain specific language for developing mobile web applications called mobl [12] is chosen to speed up building the application. Mobl is a statically typed language that integrates all aspects of the application: data modeling, user interfaces, application logic, styling and web services. Mobl projects generate static HTML5, JavaScript, and CSS files as output and they are supported by most devices and browsers.

Multi-domain and Exploratory Search Notions of Service Marts, Access Patterns and Service Interfaces in Search Computing are explained in sub-section 2.2. Same structure is used in the application in a top-down approach. Search process starts by identifying the initial domain by choosing a Service Mart from the list. Next steps are choosing an Access Pattern for that selected Service Mart and a Service Interface for that Access Pattern in order.

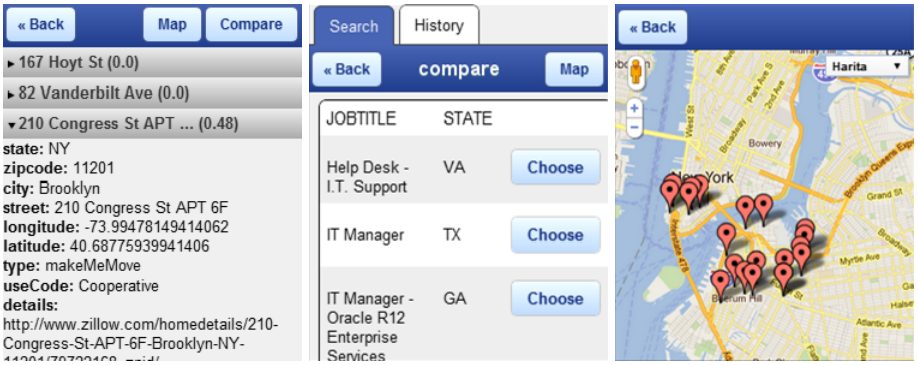
After the user successfully completes the search of the first domain by entering his/her search terms and selecting a result from the result set, exploration and multi-domain support begins. The user is presented with the domains related with the initial domain as defined in Connection Patterns. It is possible to add these domains to the query by following the same steps as the initial domain. However, results of the additional domain will also depend on the results selected from the existing domains and the pairwise Connection Pattern definitions between existing domains and the additional domain. In this way, user may combine as many domains as he/she desires and improve his/her query one domain at a time.

To illustrate this mechanism with a sample query, let us assume that the user wants to answer question: "Where can I find a Cinema in Paris that has Titanic on display with a good, nearby Chinese restaurant". One way to build up the query is to start with the movie domain. Movies can be searched by title using the related Access Pattern and Search Interface. Once Titanic is found, user may add Cinema domain through a Connection Pattern. This Connection Pattern would allow to list cinemas that has Titanic on display. User should also enter Paris as the city input to filter results. After the user decides which cinema to go, he/she may add another domain for the restaurants possibly connected to cinema by distance. Chinese kitchen can be used an input, while rating should be the ranking criteria to display "good" restaurants only. Final step of the search would be to choose the best restaurant from the list. One should note that, these example steps are only one to find best solution. Same or similar results can be obtained by starting with the restaurant domain for instance.

Another critical point of exploratory search is to assist user to formulate query which is achieved by pre-defined input attributes. Instead of presenting a generic free-text field, every Access Pattern has its own input parameters. An Access Pattern to search for restaurants by location should typically have at least street, city and country input attributes and they are displayed as separate input fields.

Presentation of Results Screen size and resolution limitations preclude usage of wide tables to display all output attributes of results. Number of these attributes, hence columns of the table, can easily exceed 10 for most domains. Three result views, that can be seen in Figure 3, are offered and implemented to tackle this issue.

Before describing the views, it is necessary to explain the proposed term "main output". Main output is defined for each domain and points to the most important output attribute of the domain that can more or less uniquely identify the result. Although, identification numbers exist, they do not contain any useful information for the user and are hardly user-friendly. As a result, main outputs are intuitively selected for each domain to meet these needs. Some examples are: title for news domain, street address for real estate domain, name for restaurant domain, etc.



(a) Accordion view (b) Compare view (c) Map view

Fig. 3. Screen captures from the application for three result views

Default view for all domains is the accordion view which can be seen from Figure 3(a). It allocates one line space for all results and displays only their main output and score initially. When the user wants to know more about a result, it is possible to tap on the record to slide down values of all output attributes. Accordion view lets the user to see maximum number of results in a small screen with the possibility of accessing details without screen change. It can also be used for multi-domain data in the list called history where the results selected from different domains are displayed.

While accordion view gives all details about one result at a time, it does not let user display details of multiple results at the same time. Such functionality is especially useful when the user needs to compare results according to certain criteria, for instance to compare the prices of a real estate result set. Comparison view is available for all domains as a secondary presentation method as seen in Figure 3(b). When the user switches to comparison view, he/she is first presented with a list containing the names of all output attributes for that domain. User chooses which attribute to use in comparison and the second screen is displayed. Second screen consists of a 2-column table for main attribute and selected comparison attribute. Each record of the result set occupies one row of the table. Comparison view cannot be used for multi-domain data as the output attributes of different domains would not always fit.

Map view (Figure 3(c)) is only available for domains with location data i.e. those with latitude and longitude information in their output attributes. In map view, each result is represented with a marker on a map initially centered and zoomed to make maximum number of results visible. If the user device provides GPS location information, another marker for the user position is added. It is possible to scroll, zoom in and out in map view using multi-touch gestures like swipe, pinch or double tap. Similar to the accordion view, map view can also be used for multi-domain data in history list as long as at least one of the domains

containing location data. Naturally, those domains without location data will not be displayed in history using map view.

3.5 A Typical Scenario

In the previous sections and sub sections, various examples are given to illustrate certain aspects of the search process. Here instead, we will demonstrate the whole process by listing the steps to answer first sample query given in the introduction section: "good physic conference October 2012 Milan reasonable 5-star hotel". Reader may find it useful to follow the flow from the statechart diagram given in Figure 4

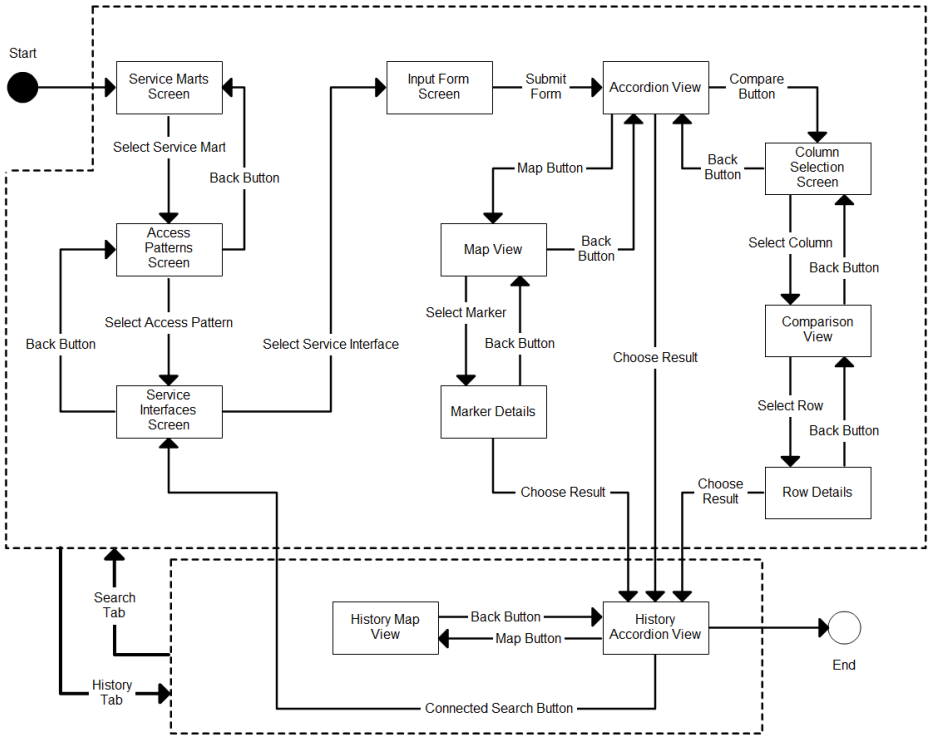


Fig. 4. Statechart diagram of the application

It would be sensible to begin with the most significant domain of the query: conference, although starting with the hotel domain would yield more or less the same solution. User selects the conference domain from the list of Service Marts and a list of Access Patterns is displayed. Let us assume that there is an Access Pattern to query conferences by subject and date. Once the user selects that

Access Pattern, a third screen for Service Interfaces is given. User selects one of the conference search engines from the list and proceeds to the next screen.

Input form is where the user provides filtering data for the result set. It is unique for each Access Pattern. User enters "Physic" and "October" to the subject and date fields respectively and submits the form. Results for the first query are now displayed in accordion view ranked by rating. User may also visualize results in map view or compare by city in comparison view to find top conferences in Milan. When the user is certain about the conference, he/she chooses it by tapping the related button.

Every time user chooses a result, history tab is activated. In history tab, user may see an accordion list of previously selected results from each domain. Only one result from each domain can be chosen and the rows of the accordion list contain domain names. When a row is expanded, buttons for connected domains are listed in addition to the details of the result.

User taps on a button to initiate the connected search and add another domain. Let us assume that there is a Connection Pattern between conference and hotel domains by location. Once clicked, input form for the second search is displayed and connected fields (latitude, longitude, city, etc.) are automatically filled with the data coming from the selected conference. User enters star preference and price range and submits the form. Again, by facilitating three view modes, user chooses a hotel and history tab is displayed. In history tab, user may remove a domain, add a new connected domain, or visualize all the results on the map until he/she is content with the outcome. Our sample query concludes here, with the selected conference and hotel in the history list.

4 Conclusion and Future Work

A web-based mobile application that makes use of recent notions in web search is developed in order to suggest a course of action about how these notions can be practically applied for mobile devices. In particular, solutions for multi-domain search and exploratory search are explained with also taking note of user interface elements required for them to function properly.

Resulting application aids the user to develop a complex query, customize it according to his/her needs, explore the results from credited sources with ease and possibly associate them with one another. It increases the usability of search in mobile devices and exploits their strengths by channeling them to search process.

Although, efficacy and convenience of the application can be realized intuitively, empirical analysis can be conducted to compare it with traditional text-based search. Ease of use, user satisfaction, precision, recall, and reproducibility of the results are some of the criteria that can be measured by surveys or automated methods.

References

1. Bozzon, A., Brambilla, M., Ceri, S., Fraternali, P.: Liquid Query: Multi-Domain Exploratory Search on the Web. In: 19th International Conference on World Wide Web, pp. 161–170. ACM, New York (2010)
2. Bozzon, A., Brambilla, M., Comai, S.: A Characterization of the Layout Definition Problem for Web Search Results. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.): On the Move to Meaningful Internet Systems 2010. pp. 150–159. Springer, Heidelberg (2010)
3. Marchionini, G.: Exploratory search: from finding to understanding. Communications of the ACM. 49, 4, 41–46 (2006)
4. Bozzon, A., Brambilla, M., Ceri, S., Quarteroni, S.: A Framework for Integrating, Exploring, and Searching Location-Based Web Data. IEEE Internet Computing. 15, 6, 24–31 (2011)
5. The Search Computing Project, Politecnico di Milano, `http://www.search-computing.eu/`
6. Brambilla, M., Ceri, S.: Engineering Search Computing Applications: Vision and Challenges. In: 7th Joint Meeting of the European Software Engineering Cnference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, pp. 365–372. ACM, New York (2009)
7. Campi, A., Ceri, S., Maesani, A., Ronchi, S.: Designing Service Marts for Engineering Search Computing Applications. In: 10th International Conference on Web Engineering, pp. 50–65. Springer, Heidelberg (2010)
8. Hearst, M.A.: Search User Interfaces. Morgan Kaufmann, Cambridge University Press (2009)
9. Bozzon, A., Brambilla, M., Catarci, T., Ceri, S., Fraternali, P., Matera, M.: Visualization of Multi-domain Ranked Data. In: Brambilla, M., Ceri, S. (eds.): Search computing. pp. 53–69. Springer, Heidelberg (2011)
10. Church, K., Smyth, B., Keane., M.T.: Evaluating Interfaces for Intelligent Mobile Search. In: International Cross-disciplinary Workshop on Web Accessibility , pp. 69–78. ACM, New York (2006)
11. Akin I.Z., Aral, A.: Development of Mobile Search Applications over Structured Web Data through Domain Specific Modeling Languages. M.Sc. Thesis, Politecnico di Milano (2011)
12. Mobl, Delft University of Technology, `http://www.mobl-lang.org/`