# Chapter 6
# Location-based Web Search

Dirk Ahlers • Susanne Boll

***Abstract.*** *In recent years, the relation of Web information to a physical location has gained much attention. However, Web content today often carries only an implicit relation to a location. In this chapter, we present a novel location-based search engine that automatically derives spatial context from unstructured Web resources and allows for location-based search: our focused crawler applies heuristics to crawl and analyze Web pages that have a high probability of carrying a spatial relation to a certain region or place; the location extractor identifies the actual location information from the pages; our indexer assigns a geo-context to the pages and makes them available for a later spatial Web search. We illustrate the usage of our spatial Web search for location-based applications that provide information not only right-in-time but also right-on-the-spot.*

## 6.1 Introduction

Even though the Web is said to be the information universe, this universe does not reveal where its information bits and pieces are located. Web information retrieval has long focused on recognizing textual content of Web pages and supporting keyword-based Web search. Just recently, also driven by the advent of mobile devices, the relationship of information to a physical location has gained a lot of attention on the Web. The geographic location of the user became a key element in providing relevant information "here and now". Many so-called Web 2.0 applications such as Flickr,[31] Plazes[32] or Upcoming[33] strongly relate their content to a physical location by means of manual tagging. Such manual annotation is not reasonable on the large scale of all Web pages but is not necessarily needed. Web pages such as home pages of businesses, restaurants, agencies, museums etc., but also reviews, link lists or classifieds directories, already contain location-related information. The relation to a physical location is not semantically captured but is implicitly part of the content as an address or a place name. Even though such pages represent only a fraction of all Web pages, their relation to a location is a yet unused asset for an interesting set of location-based applications.

In this chapter, we present our approach to automatically derive the spatial context from Web pages and contribute to the challenge of location-based Web information retrieval. Our technical foundations for spatial Web search comprise the methods and components for a spatial search engine – a focused crawler, a location extractor, a geocoder and a spatial indexer. The challenge is to reliably identify location-bearing pages, precisely extract the desired information and assign a geo-context to them. Even though our approach will not be able to spatially index all Web pages, it still allows for extracting geospatial context from unstructured Web resources so that a large set of Web pages can be automatically geotagged and employed in a variety of location-based applications.

The structure of this chapter is as follows. We briefly present related work in Section 6.2 before we introduce the reader to the challenges and potential of location-based Web search in Section 6.3. The architecture of our location-based Web search engine is introduced in Section 6.4, and key concepts are presented in Section 6.5. We present our demonstrator applications illustrating spatial search and discuss our experimental results in Section 6.6 before we come to a conclusion in Section 6.7.

## 6.2 Related Work

The related work can be grouped into three different fields: efforts to standardize the description of location information, existing location-tagged content, and commercial spatial search engines and recent research efforts for automatic extraction and indexing of location information from Web pages.

Today, several standards for description and exchange of location data on the Web exist with various powers of expression. These range from simple coordinate-oriented ones specifying latitude and longitude such as vCard (Dawson and Howes 1998), Microformats[34] or W3C Geo (W3C 2003) to more powerful formats able to express additional concepts like lines, boxes, polygons etc. such as Dublin Core Metadata,[18] the Geography Markup Language (GML)[28] or the KML format[35] of Google Earth.[2]

While older formats use simple text formats, recent formats are based on RDF or XML vocabulary, thus allowing them to be integrated into any XML document. Semantic approaches are under way to integrate spatial entities and their relations into OWL (W3C 2006b). The description of a location is typically accomplished in two ways: specification of a globally unique coordinate tuple (i.e., longitude, latitude and optional height, usually in the WGS84 frame of reference) or a named hierarchical description.

Existing geo-referenced data on the Web is mostly manually annotated or tagged. Services like geourl.org (Hansen 2006) parse specific HTML metadata specifying a coordinate enabling location-to-URL mapping; plazes.com and Placeopedia[36] are community-driven efforts to add location to content. Photo sharing sites such as Flickr[31] and Mappr[37] allow geotagging of images. Additionally, Web directories such as dmoz.org[38] organize Web links according to geographical classification. Hierarchies of places and place names are provided by so-called gazetteers, for instance (Getty Trust 2006).

Spatial search today is already provided by services such as Google Maps,[39] Yahoo! Maps[40] or MSN Live Loval.[41] Most of these rely heavily on classifieds directories and perform only little actual search to gather their points of interest (POIs), so their spatial ability is not directly coupled to the Web. For visualization on a map, addresses have to be converted to coordinates by geocoders such as the free US-geocoding service.[42]

Research towards extracting and assigning geographic meaning to plain non-tagged Web pages has gained attention in recent years. Graf et al. (2006) give a broad overview of state-of-the-art in this field. In the following, we select only those papers that are very recent and/or strongly relate to our field of work: learning geographical aspects of Web resources' contents as well as using third-party search engines for this task are covered by Ding et al. (2000). Markowetz et al. (2005) describe various challenges in identifying geographical entities as does McCurley (2001). The challenges associated with focused Web search in general are outlined in the works of Chakrabarti et al. (1999), Diligenti et al. (2000) and Tang et al. (2004). The most

recent work by Gao et al. (2006) addresses the use of geographic features for an improved multimachine crawl strategy.

Complementing existing approaches, we focus on extracting and indexing exact geographic points, in contrast to other work in the field that addresses broader geographic areas and entities.

## 6.3 Enabling Location-based Web Search

While most current search engines are very efficient for keyword-based queries, querying for Web pages relating to a certain location is not yet widely available. At the same time, location has a high significance for the user. A study in 2004 (Sanderson and Kohler 2004) finds that as much as 20 percent of Web queries have a geographic relation, with 15 percent directly mentioning a specific place.

The goal of a geographic search engine must be to best identify those pages that have a relationship to a geographic location, analyze and process it, and make it accessible for spatial search, which can then answer queries for relevant information at or near a certain location. The first step towards such a search engine is to find a source of spatial information. The physical infrastructure of the Internet reveals little about the geographic aspects of its contents. Estimates based on IP address or DNS entry of the servers can reveal the location of parts of network infrastructure but are seldom related to the information stored on it, especially for large hosts with thousands of domains per server. Exceptions may be dedicated Web servers for large companies; e.g., The New York Times as found by McCurley (2001) or Markowetz et al. (2005).

Therefore, a geographical search has to rely almost entirely on the contents of the information sources for information discovery; other techniques such as link graph analysis are outside the scope of this chapter. Fortunately, plenty of Web pages already contain viable location information, but not in a semantically structured way. According to "experiments with a fairly large partial Web crawl", (McCurley 2001) found that "approximately 4.5% of all Web pages contain a recognizable US zip code, 8.5% contain a recognizable phone number, and 9.5% contain at least one of these". Generally, location information can range from a brief mention of a region or a precise reference to a specific place.

Using Web information retrieval methods, we aim to analyze Web pages' unstructured contents to identify and extract geographic entities (geoparsing). If we can identify these, we are able to assign a geographic location to the Web page it was found on (geocoding). Previous work in the field of geographic information retrieval has often dealt with the extraction of regional or local coverage. Based on our research background in mobile applications and pedestrian navigation (Baldzer et al. 2004), we present an approach that aims for high-precision spatial information. We focus on the geographic entity of an individual address of a building identified by its house number.

Since the Web is a very large body of documents, it is not possible to visit and geoparse every single page. The challenge is to identify and predict those pages containing relevant geographic data within the growing and increasingly dynamic Web. We therefore developed different strategies to narrow the amount of pages we have to analyze. With our approach we create not only a geographically aware search engine, but also a truly regionalized crawling strategy. This makes it possible to create a specialized local search engine that can scan a confined region reasonably fast, missing only a few relevant pages, and so quickly create regional search coverage.

## 6.4 Overall Architecture of Our Spatial Search Engine

We designed and developed a search engine that enables geospatial search on the Web. We follow the general architecture of a search engine, but offer specific alterations and additions for the geographical focus. The typical components of a search engine can roughly be described as follows:

- A *crawler* discovers and downloads Web pages. The crawler takes a URL, downloads the page, analyzes the content and repeats this by following outgoing links.
- An *indexer* tokenizes the page, identifies relevant tokens and makes them accessible to search by storing them in an index.
- A *front end* allows the user to search the index with a submitted query. It handles query processing and presentation.

Our search engine is illustrated in Figure 6.1, especially highlighting the components that enable the geographically focused crawling and location-based indexing and search.

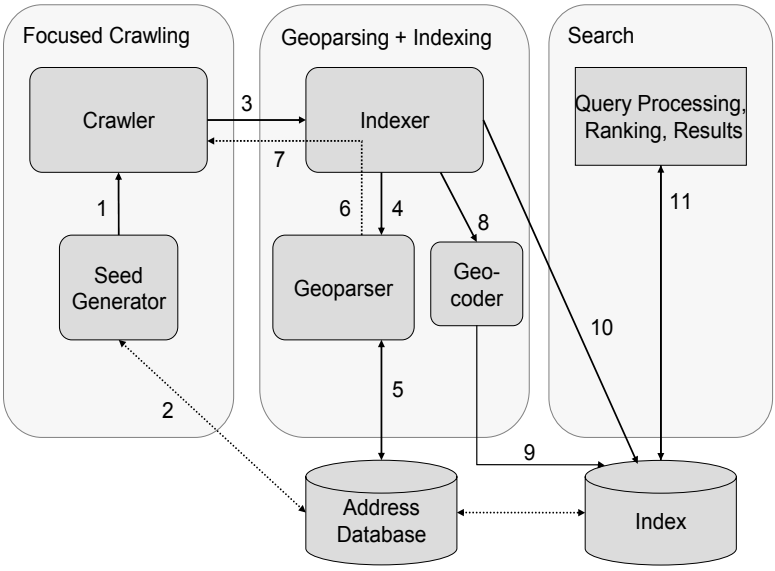


**Figure 6.1: Architecture of the spatial search engine**

To enable an efficient geographically focused crawling, a seed generator feeds (1) the focused crawler. These seeds are location-oriented so the crawler will start with pages that have a high probability of a relation to a physical location. For creating the seeds, an address database can be used (2). The crawler hands its downloaded pages over to the indexer (3). This relays the page to the geoparser (4), which tries to extract geographic information. This extraction process is supported also by an address database (5). Whenever relevant location-related information is found, this is indicated (6) to the indexer for later geocoding and insertion into a spatial index. At

the same time the information about a found location on a Web page is fed back (7) to the crawler to support the focused crawling. If the geoparser identifies one or multiple addresses on a page, each address is then geocoded, i.e., the hierarchical textual description is mapped to a coordinate of latitude and longitude (8) by a geocoder using a commercially available address reference table. The coordinate is then stored (9) along with the textual address and the Web page's URL. The page itself is also indexed (10). Both textual and spatial index can then be used for the known keyword-based search, now in combination with spatial search queries (11).

## 6.5 Central Concepts of the Geographical Search Engine

The two main concepts of our geographical search engine are geographically focused crawling and geoparsing. These two components serve to retrieve pages with geographic information and extract it.

### 6.5.1 Focused Crawling

While a usual crawler will simply crawl the Web breadth-first, retrieving all linked pages, a focused crawler is designed to retrieve only those pages related to a certain topic. The goal is to quickly build an index of most of the relevant pages on a topic while keeping the processing of nonrelevant pages to a minimum. The driving observation behind this idea is that links between Web pages are often set rather purposefully by their authors to link to pages containing similar information. Thus, for a given topic, strongly correlated subgraphs within the Web exist, comprising a large amount of all available Web pages on that topic.

To efficiently stay on-topic, the crawler is started with a set of pages from this subgraph (seeds). During the crawl it is steered and controlled by analyzing downloaded pages and only further processing pages that match the given topic as determined by a classifier. Observed in Diligenti et al. (2000) and Chakrabarti et al. (1999), a strong direct graph cohesion does not always exist. Highly relevant pages may be separated by nonrelevant ones. Therefore, a certain amount of nonrelevant pages, so-called bridge pages, has to be accepted to maintain a broad view of the relevant pages.

#### 6.5.1.1 Geographically Focused Crawling

A focused crawler specializing on a spatial topic poses some special challenges: while classic topics for focused crawling ("database systems", "health", etc.) are usually strongly interlinked, this is not necessarily true for the spatial topic. Pages with an address in a certain region do not necessarily directly link to each other. Most links are rather set between similar topics and not between similar locations. Still, a sufficient number of links exists between regional pages, but with a lower density. Therefore, the specific information we are interested in is not found in dense clusters, but rather in weakly connected networks. This means the crawler has to crawl a far greater radius, spanning many bridge pages to properly cover the spatial topic and reach relevant pages.

Starting with a given region like Oldenburg that is to be crawled, our classifier determines pages to be on-topic if an address of the given region is found. To deal with bridge pages, we assign a score to pages with addresses. For each followed link, the score is restored if an address found; otherwise the score is decreased. If the score falls below a certain threshold, the current crawl branch is pruned and no

more links are extracted. Our experiments with different values for the amount of bridge pages show that their impact varies strongly depending on seeds and discovered domains but that an overall limit of bridge pages of two to five pages yields promising results.

#### 6.5.1.2 Selection and Generation of Geo-Seeds

For geospatial search the seeds are pages that have a strong relation to the region of interest and are well linked to other on-topic pages. A well-centered seed such as the home page of a city in the targeted region or a list of local museums etc. considerably reduces the time it takes to reach other pages in that region. Two different seed types can be distinguished:

- Similar to other search engines, a first strategy to gain viable seeds is *directory-driven*. We take seeds mainly from dmoz,[38] a very comprehensive human-edited hierarchy of Web pages. The geographic hierarchy is examined, and pages located in the region of interest are retrieved. This ensures that a lot of relevant pages are already included in the seeds or are only very few links away.
- A second strategy for seed generation arises from the observation that current search engines are very good at keyword-oriented search. We shift the workload of the architecture from crawling to seed selection. We construct queries for each individual street within the desired area by using lists of all relevant cities, zip codes, and street names. The queries are sent to an already-available large index of Web pages from a major search engine and result in pages containing address parts for the region of interest. We call this keyword-query-driven approach *focused seed generation*.

Using the second strategy, the seeds themselves already contain location information. If only these seeds are analyzed and processed without further link extraction or crawling, the process can reach a coverage for a given region very fast, at the price of false dismissals that can occur for pages that are never downloaded. Additionally, it is able to use most index-backed search engines as data sources for aggregation. We call this rapid-result-oriented approach *Inverted Geo-Crawling*.

Geographically focused crawling can thus be realized using different strategies to retrieve relevant location-bearing Web pages. While the strategies cannot ensure that all retrieved pages are on-topic, they have a far higher yield than an unfocused crawl strategy.

### 6.5.2 Extraction of Geo-Information from Web Pages

The main challenge of geoparsing is to determine a Web page's geographic context and pinpoint an exact location. As we focus on the geographic entity of a specific address, our geoparser needs to identify zip code, city name, street name and house number. We target our approach to German addresses as our main area of interest. With reasonable effort, the geoparser can be adapted to other countries. In the following, we present our geoparser's methods for disambiguation and validation of address information from Web pages.

To reliably extract an address from a Web page, its individual parts have to be identified. The parts are not necessarily unambiguous; for disambiguation, individual parts have to be considered in relation to each other to ascertain a full address using various heuristics described in this section. Many geographic applications use a gazetteer for reconciliation with existing knowledge about geographical entities

such as places, regions, countries, cities. For improved accuracy, we take this strategy one step further and use a full database of address-related information, which contains zip codes for every possible city and also every city-zip combination for each street.

McCurley (2001) describes certain ambiguities particular to geographic entities that arise even with assisted search. *Geo/non-geo ambiguity* refers to the use of terms to name a place as well as a different concept, e.g., the German word "leer" means "empty", but is also a small town in East Frisia; "Münster" means a minster, but is also the name of several cities. The second example also illustrates *geo/geo-ambiguity* where different places share one name. These are cases where traditional keyword-based search most likely fails, since a single named entity can only be reliably located with additional location information. The appearance of, e.g., zip code and city name near each other generates a stronger geographic hint to a certain city. The same applies to street names. The zip code in itself is ambiguous as well: a German five-digit zip code could also be a product or phone number, a price, etc.

Starting with the zip code as the main supporting term, we initiate a coarse-to-fine term disambiguation. We draw upon an address database for reliable identification of terms by searching for city names in the close surrounding of a found zip code on a page. Once zip code and city are identified, we extend the disambiguation towards the street level by searching for street names for the city-zip pair. Finally, if a house number is found, the geoparser treats the address as valid and geocodes it.

City and street names on Web pages do not always match the names we have provided in the database. We therefore employ normalization and stemming methods to be immune against variations. For normalization, name additions or city districts are given a lower relevance to also match cities where this was omitted. Spelling variations are allowed to correct possible typos. For the detection of street names, they are subjected to stemming algorithms to reduce the street name designations (e.g., "Strasse" – street; "Allee" – avenue, etc.) to a single token as these are often abbreviated in various ways. Separation of name parts such as hyphenation, spaces, written as one word or mixture of this is identified. Again, spelling variations are considered.

This extraction method has the advantage that the extraction process is tied strongly to existing knowledge and only valid addresses with parts known to be correct are extracted. Some of our lessons learned while implementing the geoparser were used to improve the keyword-query-based seed generation described earlier.

## 6.6 Experimentation and Demonstration

Our proposed methods and strategies for crawling and geoparsing were implemented as prototypes to gain experimental results and support our design decisions. We discuss our results for different crawling and seed selection strategies and present the results for the location assessment.

### 6.6.1 Evaluation of Geographically Focused Crawling

Based on the two proposed seed selections, we ran several tests to show the validity of our approach. We present the results grouped by the method of seed selection as the seeds constitute the main input for the crawling strategies.

#### 6.6.1.1 Results of Query-based Seed Selection

We ran a crawl with query-based seeds in the regions of Oldenburg and Rügen. Oldenburg is an urban city, and Rügen, Germany's largest island in the Baltic Sea, is a rural area. These were chosen to assess influences of the crawled region from the results. We generated queries from both regions' street and city data to query a search engine and let both tests run independently. For our tests, we utilized the Google API,[43] which we chose due to the very large index size, a resulting high number of results and its public interface for queries. In this test, we only processed the retrieved pages with our geoparser; the crawler did not follow any further links (inverse geocrawling). Random manual sampling of the results was done to check the results for correct location assessment.

It took about four days to retrieve and analyze the data for the 1,379 streets of Oldenburg. The geocrawling resulted in about 24,000 addresses on 23,000 distinct pages. This means that some of the retrieved pages were directories with multiple addresses. Most of the retrieved pages contained only a single address. For 240 streets, no pages with an address were found. On average, each street was present in 17 addresses.

The results for Rügen were similar; here we started with 1,074 streets and found about 17,200 addresses on 21,100 pages in a little under four days. We found fewer directory pages, but some sites that featured the same address on each page. We found that 356 streets had no address associated. An average of 16 addresses per street was found.

Generally, the results for both regions are quite similar in structure but differ in quantity. We found a full address on 25 percent of all downloaded pages. About 75 percent of all raw results were discarded by our parser; thereof 90 percent because of missing cohesion (it is not currently possible to search for term nearness with the Google API, so a lot of pages were obtained with the search terms scattered about the page without forming a contiguous address) or because the address could not be found on the page at all. The remaining 10 percent were dropped because the page was no longer available, no house number for an address could be found, or other issues. Fewer than 10 percent of pages with otherwise correct addresses lacked a house number.

We could prove that fast coverage of a region is possible with the inverse geocrawling approach of query-based seed generation and that it is a reliable way to quickly build an index of a confined region, uncovering relevant pages in a short frame of time. We also showed that our approach of only allowing full addresses including house number is valid to discover high-quality addresses.

#### 6.6.1.2 Results of Directory-based Seed Selection

We tested our approach of geographically focused crawling with directory-based seeds from dmoz[38] for the large region of northern Germany. As opposed to the query-based seed selection, the seeds were mainly hub pages containing only very few addresses themselves. We therefore fed these seeds into a crawler with a generally unlimited crawl depth and high bridge page number, but some filters activated: a maximum number of documents per domain was in effect, as well as filters to exclude non-German domains, unpromising contents such as galleries or forums, etc. For a crawl of two weeks, the results show that of about 44,000 domains crawled, 20,500 contained at least one full address in the region with the address count at 3.8 per domain. For this domain-oriented analysis, we only counted unique addresses

and complete domains. For a rather broad crawl, we feel that finding addresses on about half of all visited domains is a good result. In a second step, we set up smaller crawls of Oldenburg with a small seed to directly examine the effect of different bridge page parameters. We found that when comparing an unfocused to a focused crawl for this region, we can retrieve up to 10 times as many addresses with a focused crawl in a given time.

Our geographically focused crawling was dependent on several factors: for small regions, the approach identified much more Web pages with addresses in a given time than a crawler without this focus. For larger geographic regions, however, the number of bridge pages had to be much higher to find enough addresses; even then, the ratio was smaller than for confined regions. This indicates that the crawl tree still dilutes quickly at some point, and gathers increasingly more pages that are off-topic. We currently work on larger crawls to give better estimates on the ratio of Web pages containing addresses on a larger scale.

### 6.6.2 Quality of Geo-Information Extraction

Some numbers on distribution and structure of address-bearing Web pages were already mentioned in relation to our crawls where the geoparser was used to extract addresses. Discussing quality measures for the address extraction itself, precision was found to be very high. Of the identified addresses, random sampling reveals almost no errors. The presented methods leave only very little room for misidentification and incorrect addresses are not recognized by our parser, so this is to be expected. Recall is difficult to measure as we cannot make reliable assumptions on the number of all relevant documents. We are aware, however, that there are certain omissions due to addresses that we cannot currently find. Our methods assume a strong cohesion between address parts by using maximum distances that are exceeded by certain pages, often by elaborate table structures that dilute the relationships between address parts. A typo in a zip code can invalidate a whole address. Finally, multiple typos or unusual abbreviations cannot always be matched. We already tuned the heuristics so that weakening them more would lead to erroneous addresses. Due to these effects, we estimate an omission rate of 5 to 10 percent.

We found that our database-backed approach still outperforms simple address matching as could be done by a general matching such as “street term + number + zip + city term”. For the city of Oldenburg alone, we found that 118 of 1,725 streets did not match any usual street name pattern, which was already extended to include local designations like “Kamp” (field). Some of these are decidedly un-street-like such as “Ellenbogen” (elbow), “Ewigkeit” (eternity), “Vogelstange” (bird perch) and might even be prone to misrecognition with additional supporting terms. Discovering these with a list of known names leads to better location coverage.

### 6.6.3 Applications

The spatially indexed Web pages gathered by our search engine enable some interesting applications. Generally, we install a spatial layer on top of the existing Web. This layer captures the semantic location information of the pages. Pages can be located at certain coordinates, and a search can be restricted to a desired area. We implemented two applications that illustrate the potential of a spatial search engine in two different application domains: first, a location-based search for locating Web pages on a map; second, a search engine for the enrichment of directory data of, e.g., yellow pages.

### 6.6.3.1 Localized Web Search

Our first prototype exclusively used query-based seeds for page discovery and directly processed these with our geoparser. It was built as a feasibility study of geo-extraction and -referencing. Using a commercially available geocoder from a related project, extracted addresses were mapped to geographic coordinates.


**Figure 6.2: Local search results for Oldenburg**

For our main demonstration we chose the city of Oldenburg. With our search engine, users can search the content of all pages with an address in this city. The screenshot in 2 shows the prototypical result page of a search for "pizza". The search box at the top of the page repeats the query; below is the map with the results as numbered icons. Only pages with an address in Oldenburg matching the keyword search are displayed. The zoom level of the map adjusts according to the distribution of the result. Below the map the results are listed for each icon number along with the page's title, URL and the identified address. It is immediately clear where the different pages and thus pizza services are located. Our prototype thus demonstrates a keyword-based Web search with spatial awareness.

### 6.6.3.2 Spatial Search for the Enrichment of Directory Data

While many local search applications rely on classifieds directories as data sources, the search engine for this project works the other way around. For our project partner, a major provider of yellow pages, we built a search engine that can automatically enrich existing directory entries and also discover promising new candidates. This will act as an additional source of data for the survey department, which until

now had to rely completely on manually gathered data. By automatically retrieving business Web sites, existing yellow page entries can be enhanced and sanitized and prospective new customers can be uncovered. The screenshot in Figure 6.3 shows the query interface and a detailed result of a search for the location of our institute.

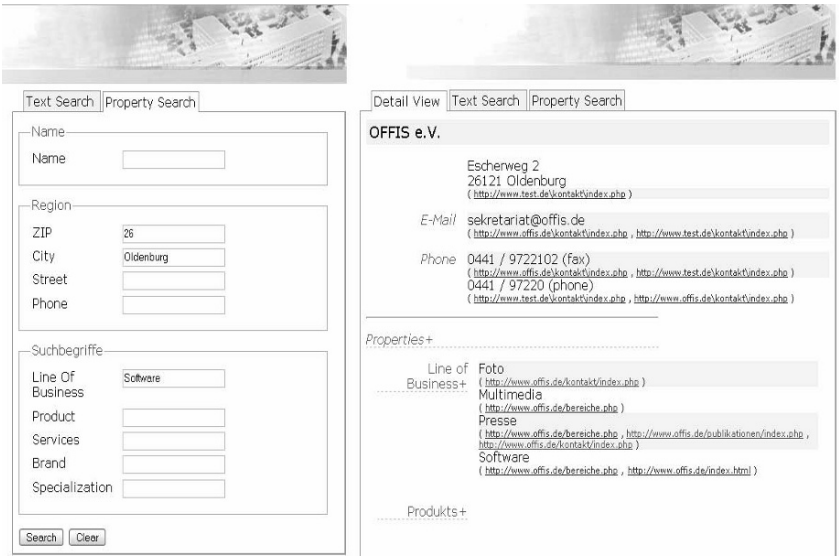


**Figure 6.3: Directory data query form and result**

We based this search engine on a standalone crawler for resource discovery. We implemented a general focused crawling to capture a large part of northern Germany with seeds selected from dmoz.[38] The geoparser is the initial component, which is extended with other parsers to derive additional information from the Web pages: we built an entity name extractor that can derive the person, company or organization that is referenced by an address. We also extract business specifications, such as commercial register entries, phone numbers, line of business, products, by abstracting from individual Web pages.

This automatic data acquisition is now used commercially to greatly improve quality and search time, since a much better and faster overview of companies is gained.

## 6.7 Conclusions

In recent years, location has gained much attention in a Web context. Rather than expecting mass Web content to be manually annotated with its location, the implicit location relation already present on plenty of Web pages that lay unused now forms a valuable asset for a range of commercially relevant applications.

In our approach, we developed central concepts and components for a geographic search engine: a focused spatial crawler and a geoparser that identify precise localized information and input it to a spatial index that complements the keyword-based index of today's search engines. We now have the ability to perform a keyword- and location-based search of common Web pages by filtering the index according to geographic properties. Our experimentation gives interesting results that

we will also refine in our future work. The data sets crawled for Oldenburg and Rügen gave an interesting insight into performance of the algorithms and precision of the found and spatially indexed Web sites in these regions.

We applied our spatial search engine in two different application domains. A localized search (profiting from our regionalized approach) can offer a spatial search on the Web and not only on previously selected and annotated pages. Based on keywords, the results show the spatial relationship of the found Web pages. The second application domain might not be so obvious on first sight. The localized search results are used for extending and sanitizing existing yellow page databases. This approach has been very successfully carried out with a research cooperation partner. Future work will mainly concern broadening the scope, adding more semantics and building stronger relationships and rankings.

*Acknowledgements.* We thank our colleagues Jörg Baldzer and Norbert Rump for their continuous support and collaboration on this project as well as our student Dorothea Eggers for her valuable work. Part of this research has been carried out as a subproject of the Niccimon project (Scheibner et al. 2006) and was supported by the State of Lower Saxony, Germany.