Towards a Search Driven System Architecture for Environmental Information Portals

Thorsten Schlachter, Clemens Düpmeier, Oliver Kusche, Christian Schmitt, Wolfgang Schillinger

Towards a Search Driven System Architecture for Environmental Information Portals

Thorsten Schlachter[1], Clemens Düpmeier[1], Oliver Kusche[1], Christian Schmitt[1], and Wolfgang Schillinger[2]

[1] Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
{thorsten.schlachter,clemens.duepmeier, oliver.kusche,christian.schmitt}@kit.edu
[2] Baden-Wuerttemberg State Institute for Environment, Measurements, and Nature Conservation, Karlsruhe, Germany
wolfgang.schillinger@lubw.bwl.de

Abstract. In order to merge data from different information systems in web portals, querying of this data has to be simple and with good performance. If no direct, high-performance query services are available, data access can be provided (and often accelerated) using external search indexes, which is well-proven for unstructured data by means of classical full text search engines. This article describes how structured data can be provided through search engines, too, and how this data then can be re-used by other applications, e.g., mobile apps or business applications, incidentally reducing their complexity and the number of required interfaces. Users of environmental portals and applications can benefit from an integrated view on unstructured as well as on structured data.

Keywords: environmental information portal · architecture

1 Motivation

Environmental portals offer a quick and comprehensive overview on available environmental information and data, which are often distributed over a variety of individual databases, systems, and business applications [1]. The information are of manifold types and ranges, i.e., structured measurement data, meta data, individual structured or semi-structured (environmental) objects, or reports on the state of the environment without an explicitly modelled information structure.

In web portals such a mix of structured, semi-structured[1] and unstructured data is typically made accessible via hierarchical navigation paths and/or a full text search.

For example, a search for 'nature reserves' in the environmental portal of the federal state of Baden-Württemberg in its present form links to about 3,000 relevant

[1] As semi-structured data can be transformed into a structured form [2], and both are treated analogously in the portal application, in the following only structured (as opposed to unstructured) data will be mentioned.

adfa, p. 1, 2011.

documents and also provides a direct entry into the map view of a business application. However, neither information on individual nature reserves nor a localized entry point into the map view are displayed – although all necessary data is present in databases as well as in a web-based business application, it is not available for the portal via services and interfaces.

From a user's perspective, the portal application should already provide an overview of all relevant nature reserves (in a desired area). Access to a complex business application should only be necessary for very specific tasks or advanced search requests. However, for this purpose all objects, e.g. nature reserves, must be made directly available for the portal application, which basically also applies for all other kinds of environmental information. This means that a vast number of information and environmental objects has to be made searchable in the portal. For the user, this search should be easy to use and as fast as possible.

1.1 Querying an Exploding Number of Data

With the aim to present fine-grained environmental information, the number of objects increases dramatically. Some traditional navigation approaches are therefore no longer appropriate, and the design of search interfaces becomes more important.

While full text search interfaces are usually instrumented to provide easy access to unstructured textual data, access to structured data is often implemented by querying the available data via complex search forms, data selectors, or map-based access mechanisms backed up by dedicated application logic mapped to underlying (relational) databases.

An advanced search approach based on simple full text search paradigms has to be implemented which integrates the classical (single-slot full text) search functionality with a search for structured data, and includes specialized approaches for displaying structured data such as map presentations (for the querying and representation of spatial data) or graphical and tabular presentations of measurement data.

Furthermore, search by voice input should be supported on mobile devices, which is easy as search terms are ultimately converted into textual queries anyway.

2 User Expectations and Objectives

A new search driven approach will change user behavior and raise expectations with respect to how information is accessed using mobile devices. For many users, accessing data via complex navigation paths is no longer acceptable. Especially on mobile devices, a context-based, e.g. location-based, representation of suitable information with a minimum number of user interactions is expected. Any further navigation is often done in an exploratory manner, based on the information already displayed. Every subsequent interaction can trigger a change in the search, i.e., the displayed information has to be updated if necessary. This generates great demands both on the flexibility of query interfaces and on their performance. Noticeable latencies in updating the display are unreasonable for reasons of ergonomics.

As many systems and services do not meet these requirements in practice (yet), it is desirable to prepare structured data in an optimized way for flexible and high-performance querying - an approach proven for unstructured data by means of full text search indexing. At this point, search indices maintaining the structure of data and providing advanced access functionalities come into play.

The principle of search driven websites and portals certainly also applies for structured data. Background services are used to maintain data in their original systems, e.g., relational databases, while keeping a structured search index up to date in parallel. The portal uses the query interface of the search engine for primary data access via search and takes care of data presentation.

Within a search driven portal, the primary information access is driven by search. All navigation items (including menus) as well as the contents are displayed as a result of one or several search queries. In this regard, search driven websites and portals can be characterized as being highly dynamic environments.

2.1 The Search Engine as Glue for Distributed Information

The described use of search engines can also contribute to solve further problems of the classical, service-oriented portal approach: as the landscape of environmental information systems is highly heterogeneous, the same applies for the nature of services and interfaces used. It is desired to significantly reduce the number of required interfaces within the portal software – however, for the synchronization of data with the search engine, they have to be implemented elsewhere. In turn, other applications, e.g. mobile apps, can also make use of the search engine and benefit from the availability of data by means of a single query interface.

By adding them to a structured search index, even such data can be made available for real time access that aren't available via an (accessible) service, or that are hidden behind a complex navigation or query form. Yet, even in the case of available data services, the performance they offer may often not be sufficient in order for them to be the basis for a search implementation with satisfactory response time. For example, in the context of environmental data, response times in the range of several seconds have been observed from some systems.

While importing or indexing data, these can be filtered or enriched by additional attributes, e.g., place names can be converted into explicit geographical coordinates using gazetteer services, which then can be used to implement a proximity search or bounding box queries. By integrating machine learning algorithms, even a more complex semantic enrichment of data is possible.

In summary, by using a search driven architecture for environmental information portals, the following objectives will be pursued:

- Use of data from their original source and concentration in a powerful search engine
- Comprehensive view of all available, relevant information, accessed via simple single-slot full text search queries
- Provision of additional aids for an exploratory search experience

- Both the discovery of object classes and of individual objects has to be possible
- All data is supposed to contain references to their original systems, as only the most important information is displayed in the portal or mobile application
- Reduction of the number and complexity of data interfaces (for portals and applications)
- Use of the search index by a variety of applications
- No or little extra load on original systems through additional search requests
- Querying large numbers of objects has to be as performant as a "simple" full text search
- The current state of the environment (measurements) has to be presented
- Context information (mobile devices, settings for personalization) has to be included into search

3 A Basic Architecture for Search Driven Environmental Information Portals

The presented architecture for a search driven environmental information portal is based on a JEE (Java Platform, Enterprise Edition) portal server (Liferay Portal [3]) as both a web frontend and an integration platform for underlying data services, but it also applies for a variety of other applications. The portal software provides a rich set of standard functionalities needed for an environmental portal, e.g., user authentication and authorization, support for corporate web design with responsive layouts [4], content and asset management, etc. In particular, however, its modular extensibility is important. New content and data presentation functionality can be easily added to a JEE portal using concepts like portlets and web widgets. The application logic of these extensions uses two search engines for primary data access.

3.1 Use of Specialized Search Engines

Since different search engines exist which each have their strengths in specific areas, several search engines are used in the portal. Unstructured data is indexed by a traditional full text search engine (Google Search Appliance [5]) using web crawlers and special connectors. Structured data items are stored and indexed as JSON objects in a separate structured index server based on NoSQL technology (Elasticsearch [6]). This second search engine can be easily accessed and managed through a REST interface.

The necessary tools for importing structured data into the Elasticsearch server as well as for ongoing/periodic synchronization of data are running outside the portal software and should be part of a workflow which also includes quality management for the data.

When data are not available by means of services or when access is limited, these data can be made available using generic cloud services, e.g., data from geo-referenced files can be uploaded to the Google Maps Engine, and subsequently can be used via data services or as map layers.

External search engines may complement the portal search by delivering special data, e.g., statistical data or descriptions [7] of administrative services [8]. The use of standardized interfaces like OpenSearch [9] can simplify the integration of external search engines.

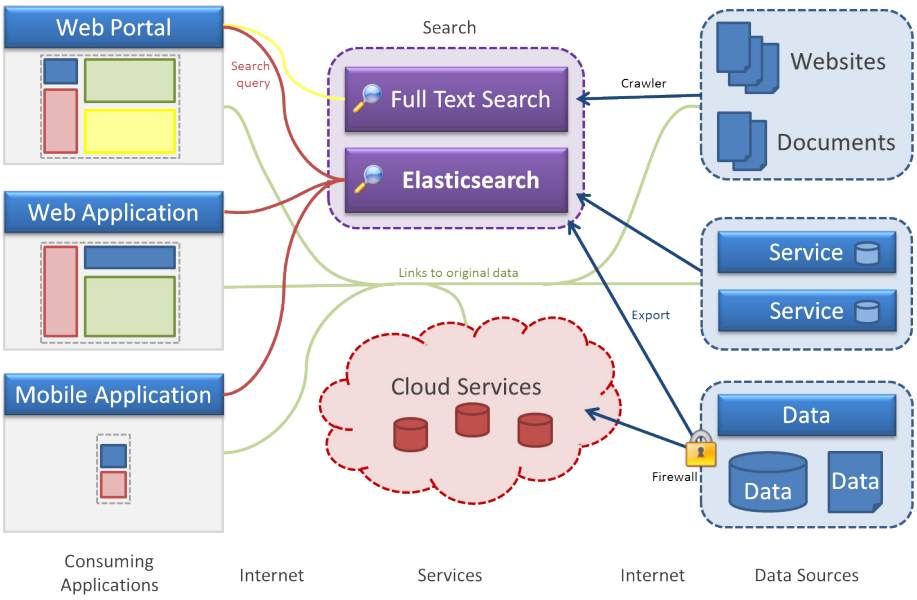

Fig. 1. Main components of a search driven architecture for web portals and (mobile) applications including two search engines and cloud services to complement original data services.

3.2 Creating the Search Experience

In addition to editorial content and navigation structures, the search engine is forming the core of the portal application for data access. Queries entered in the search slot will be preprocessed and semantically enriched, e.g., by use of gazetteer services for the identification of geographic coordinates and other location context, or by identifying a time context, and then forwarded to the search engines.

In particular, the enriched queries can drive a more precise search - not only on the basis of mere string comparisons, but allowing location-aware queries finding nearby objects by radius search. Using the enhanced query terms, the search engine returns weighted results according to (configurable) relevance criteria.

The presentation of search results can be performed by various components within the portal software. Currently this is realized via web widgets, i.e., encapsulated Javascript components, each implementing a certain type of data visualization. As soon as supported by relevant web browsers, these can be replaced by standardized “Web Components” [10]. Web widgets are independent from their container application,

and therefore can easily be re-used in other applications, e.g., in hybrid mobile apps [11].

For data access they use REST APIs, and for the seamless integration with the portal software every web widget is managed within a wrapper portlet. The individual widgets are kept as generic as possible, i.e., they can be configured using Mustache templates [12] for display, thus being adapted to the respective search results. For the re-use of generic components, it may be helpful to generate additional general attributes such as “title”, “abstract” and “link”.

3.3 Data Mashup: Beyond Simple Result Lists

Various presentation widgets can create or show menus, navigation structures, or individual objects. Each widget again can be a (complex) web application, e.g., a component to display objects on a map. Several portlets (including web widgets) can be placed on the result page, and thus form an information mashup, assembling information from different sources to a single integrated view (Fig. 3).

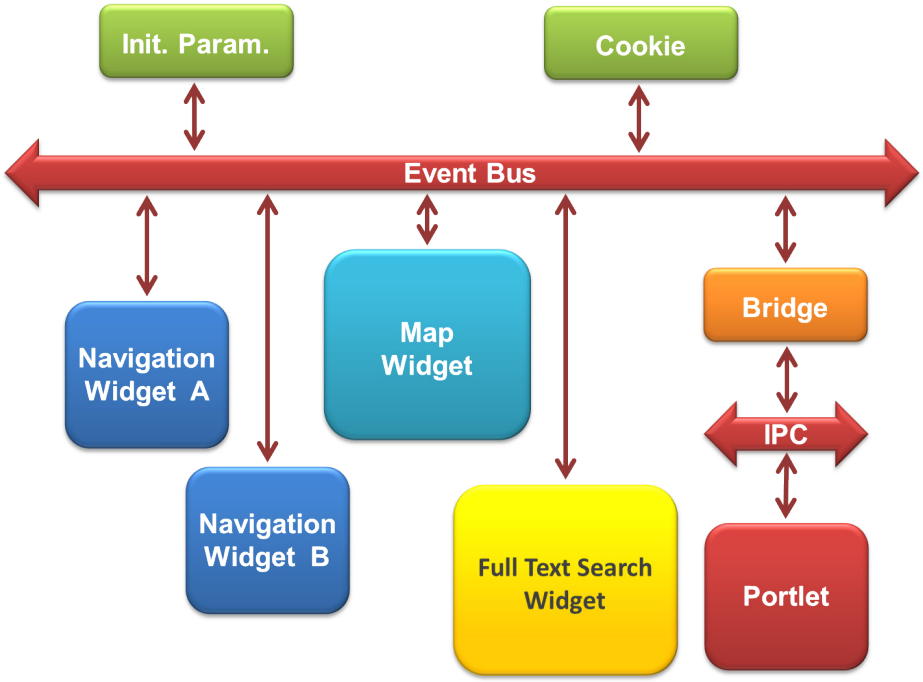

Fig. 2. Event bus connecting a set of components (widgets and portlets) on the result page. Other communication systems can be connected using bridge interfaces, e.g., the Liferay Inter-Portlet Communication (IPC).

In order to interconnect components (widgets) and to offer a rich user interface, these have to be able to communicate. An event bus implements the loose coupling of

individual components (Fig. 2). Each component can do both listen to events and send own events to the bus, and is solely responsible for the events to which it responds or not. In order to achieve a coherent interaction of components, only the set of event types has to be defined for the portal application.

Additional mechanisms are integrated in the event concept, e.g., for gathering personalization information from cookies, browser storage, or a web server, or to query sensor data (GPS or coarse-grained location) on a mobile device. URL parameters or search slot queries may be sent as events as well. If required, further adapters can be connected to the event bus for handling of external or system events, e.g., the portal software.

In addition to external data from various environmental information systems, data from the portal software itself can be made available via the (structured) search engine. For large datasets or complex requests, the load on the portal software and the underlying database can be reduced significantly by intelligent preprocessing and querying in the search engine. The next major release of the Liferay portal software (Liferay 7) will provide this option by default by means of an embedded Elasticsearch engine.

4 Case Study "State Environmental Portals" and Environmental App

The described search engine-based approach has been used to perform a prototype reimplementation of the state environmental portal (LUPO) [1] (Fig. 3). In addition to the weak structured and unstructured content (about 2,000,000 documents from more than 2,000 sites), many structured data, e.g., various types of protected areas, and locations of energy systems such as windmills, biogas plants, solar panels, and hydropower plants have been stored in the Elasticsearch index. Where possible, geographical coordinates were acquired or produced for all objects. This index forms the core of a significantly expanded functionality of the portal which now integrates the display of location based search results on maps and provides a location-based search.

Each query, whether manually entered to the search slot or clicked on using the tag cloud, first generates an empty result page. After preprocessing and semantic enrichment of keywords, both search engines and additional backend systems are queried in parallel and asynchronously. While the results for the full text search engine are shown in a classic hit list, the structured hits can trigger a variety of representations. For example, the discovery of objects of certain classes can force the display of corresponding map layers within the map component. Classes and individual objects can be shown for information and further navigation. Clicking on an object class can force the display of matching instances and switches the corresponding map layers on or off. The results already contain quantitative information, e.g., how many objects of a certain class were found for the current query. Clicking on a particular object can, for example, trigger the centering of the map to this object.

Coming back to the example of the "motivation" section, a search for "nature reserves" and a place name now displays the closest reserve objects in the left column.

The corresponding heading with the concept name “nature reserve” serves for further navigation and allows switching on and off the respective map layer. Clicking on a particular object (a single nature reserve) centers the map to the appropriate place. Another click on that area in the map then displays additional details and links to business applications.

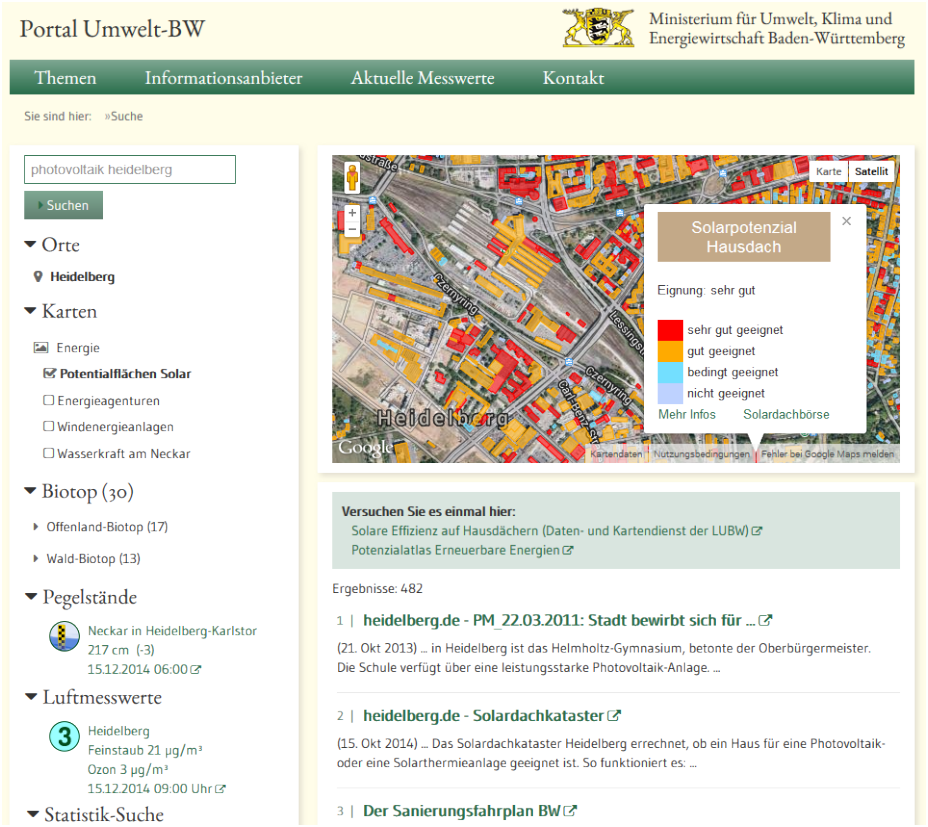

Fig. 3. Result page of a state environmental portal. Object classes and individual objects as well as measurement values are presented by multiple widgets on the left. They may trigger events toggling the display of map layers, centering of the map, or refining of search queries. Full text search results are presented below the map widget.

Current measured values are displayed either from the search engine or from separate cloud services to match the context.

Many components of the state environmental portals (data, services, search engines, and user interface) are also used in the mobile application “Meine Umwelt” (“My Environment”) [13]. It is a hybrid app whose core consists of a HTML5 application used in apps for Android, iOS and Windows Phone. The HTML5 application is organized into web widgets as well and utilizes the same services and search engines. This saves resources for both the development and the maintenance of components.

5 Conclusion

The procedure described can be summarized as follows: Modern web applications, and in particular mobile solutions, dynamically load data. In portal-like applications, a great variety of different data interfaces and data sources is used. In order to simplify the application and to reduce the load on the primary backend systems, data can therefore be administered in a separate search index:

- Let the engine do the hard work (preprocessing, indexing, querying)
- Existing data can be enriched
- Use structured data types to obtain individual ways to query and to present object classes
- Use generic formats and mechanisms, e.g., templates for presentation, so a variety of applications can be covered "out of the box"

What distinguishes the search driven approach in comparison to previous/alternative concepts?

- Reduction of interface diversity in the consuming applications
- Easy integration of data sets. However, it requires an operating concept for the update of the index
- (Quick!) retrieval of object classes and individual objects
- Fine-grained local search on a single object-level (as opposed to entire map layers)
- Combination of structured/semi-structured and unstructured data in the search
- Reduction of redundantly managed metadata
- Reusability of web widgets, e.g., for mobile and web applications
- Reusability of services and search indexes
- More interactivity for the user
- Prevention of a bottleneck on the server side (portal)
- Less load on the portal server by offloading and indexing content

The concept of search-driven portals and websites will stand or fall related to the relevance of their search index. Automatic updates of the search index, which have been only briefly mentioned in this paper, must not be underestimated. Basically, the operation of a search engine for information integration in principle means an added expense and also raises questions about data consistency, redundancy avoidance and operating cost.

However, the advantages of the search-driven approach clearly outweigh these concerns for the existing applications. The objectives are therefore fully met by the prototype. User feedback has been very positive and expectations are even being surpassed in many ways.

6 Outlook

Putting unstructured, semi-structured, and structured information into search engines can be considered as a first step towards a unified information access platform (UIA) [14], even if central aspects of UIA like advanced data processing or data analysis remain outside yet. However, the basic technologies create a huge potential for the generation of added value.

Most planned extensions do not affect the search-driven architecture as such, but rather the applications it supports. Especially with regard to user-experience and ergonomics, quite a number of improvements can be achieved. Due to the loose coupling of components and the use of an event-bus in the portal application, the integration of additional navigation aids and further functionality is very easy.

An improved linking of data in the search index is probably the most promising approach. Currently, many object classes in the index are data islands that are not or hardly connected to other classes. The use of shared concepts, such as in the thematic and spatial classification of data, promises a great potential, even for the generation of completely new applications. Corresponding relation or graph-oriented concepts based on Elasticsearch already exist and are waiting to be used.

References

1. Schlachter, T. et al.: „LUPO - Weiterentwicklung der Landesumweltportale“; in: Weissenbach, K. et al. (Eds.): Umweltinformationssystem Baden-Württemberg F+E-Vorhaben MAF-UIS Moderne anwendungsorientierte Forschung und Entwicklung für Umweltinformationssysteme, Phase II 2012/14; KIT Scientific Reports 7665; ISBN 978-3731502180; 2014; pp. 65-74
2. Abiteboul, S.; Buneman, P.; and Suciu, D.: “Data on the Web: From Relations to Semistructured Data and XML”; Morgan Kaufmann; ISBN 978-1558606227; 1999
3. Liferay Portal: http://www.liferay.com (visited Sep. 5th 2014)
4. Zillgens, C.: „Responsive Webdesign: Reaktionsfähige Websites gestalten und umsetzen“; Carl Hanser Verlag; ISBN 978-3446430150; 2012
5. Google Search Appliance: http://www.google.de/enterprise/search/ (visited Sep. 5th 2014)
6. Elasticsearch: http://www.elasticsearch.org (visited Sep. 5th 2014)
7. Statistisches Landesamt Baden-Württemberg: http://www.statistik-bw.de (visited Nov. 5th 2014)
8. Service-BW: http://service-bw.de/zfinder-bw-web/processes.do (visited Nov. 5th 2014)
9. OpenSearch: http://www.opensearch.org/Home (visited Dec. 15th 2014)
10. Web Components: http://www.w3.org/TR/components-intro/ (visited Sep. 5th 2014)
11. Multi-channel app development: http://en.wikipedia.org/wiki/Multi-channel_app_development (visited Sep. 5th 2014)
12. Mustache: http://mustache.github.io (visited Sep. 5th 2014)
13. Schlachter, T.; Düpmeier, C.; Weidemann, R.; Schillinger, W.; Bayer, N.: “My environment - a dashboard for environmental information on mobile devices”; International Symposium on Environmental Software Systems (ISESS 2013)
14. Probstein, S.: “Five Advantages of Unified Information Access (UIA)”; CIO; 2010; www.cio.com/article/2416284/ (visited Dec. 15th 2014)