

Semantic Descriptions in an Enterprise Search Solution

Uwe Crenze, Stefan Köhler, Kristian Hermsdorf, Gunnar Brand, Sebastian Kluge

interface projects GmbH, Tolkewitzer Straße 49, 01277 Dresden, Germany
{uwe, stefan, kris, gunnar, basti}@interface-projects.de

Abstract. Today customers want to use powerful search engines for their huge and increasing content repositories. Full-text-only products with simple result lists are not enough to satisfy this community. Different content sources require different analyzing and indexing strategies and a content-specific result set presentation. There is a lot of research in the field of using semantic web technologies for information retrieval. A wide range of useful standard vocabularies and powerful frameworks have been developed that can be used to gather, transform and store metadata. However, in practise we see a gap between the state of art of information retrieval and customer needs with a defined prise-performance relation. It is a challenge to index a large file server with heterogeneous content annotated with metadata from different vocabularies, to provide an ontology-based navigation, to produce semantic annotated search results, to use faceted browsers as powerful filtering mechanism and do that with an out-of-the-box solution, which is stable, has a good performance and provides a simple way to configure it. With this viewpoint we present in this paper the usage of RDF-based semantic descriptions in an enterprise search solution developed at interface:projects. This paper covers lessons learned from developing a metadata-focused information retrieval system called *inter:gator*¹. Especially we discuss the challenges and possible solutions in an enterprise (-wide) search scenario, and show the place where semantic descriptions matter in such a solution.

What is Enterprise Search?

First of all, enterprise search means federated search in different content sources. In our context, content is unstructured text from documents, where a document can be a text document, an email or a database record. On the other side we can interpret a JPEG-picture combined with IPTC-metadata as a semi-structured document.

Enterprise search can be a stand-alone IT-solution or is embedded in a content management system. In practice enterprise content management is more a vision as an IT-solution. So an enterprise search solution has to integrate several content repositories. The research field of *Semantic Desktop* works on the other end of the software stack – user interaction with content on the level of application clients, desktop and search tools. But *Semantic Desktop* depends on a semantic backend. At this point all aspects come together: powerful enterprise search needs semantic context information

¹ <http://www.intergator.de/>

from a metadata repository and *Semantic Desktop*, so for our work there are no differences between enterprise search and *Semantic Desktop*.

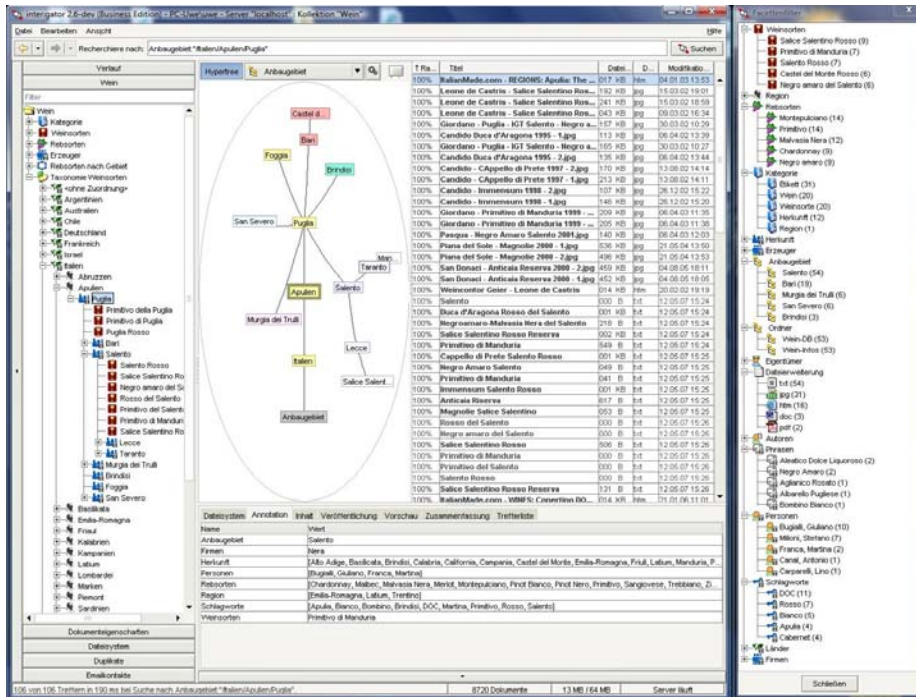


Fig. 1. inter:gator desktop client

Challenges for Search Solutions

The main challenge for indexing and search time is the amount of data. If we have 10.000 pages on a web server, we can do all analysis necessary for a rich search solution. But if we want to index a file server with >1.000.000 documents it is very strong to do more with it as full-text indexing (e.g. in such usage scenarios it is not reasonable to perform detailed linguistic and text structure analysis). Here are limitations for semantic notation of metadata also (see below).

Second challenge is the quality of data and text extraction tools. This is the ground of a universal search solution. No semantic analysis makes sense if the text and metadata extraction part is not a stable piece of software. It must run 7x24 hours without crashes and memory leaks (horrible content of many PDF-documents is one example for that).

Often different content sources require handling of different authentication schemas. Then security access for search results is the next challenge (think about different user management of databases and file servers).

Last but not least – complexity of the processing infrastructure and semantic framework which consist of crawlers, modeling tools, analyzers, visualization components and semantic integration parts). Additionally there are limitations for linguistic components in mixed-language scenarios like dealing with IT-content.

Full-Text Search vs. Metadata Search

For huge content repositories (also the internet) you can only do full-text search until we will have Web 3.0 (the *Semantic Web*). The question is: which functionality is possible for an enterprise search solution with few million documents? Semantic-driven search technologies need lots of metadata. Metadata is the key for context-aware searching and the visualization of search results according the semantic aspects of the result set (see fig. 1).

What users helps is a combined full-text and metadata search with powerful semantic-aware (metadata-aware) filtering and proposal functions.

Document Processing

After extracting the text documents are processed in a processing queue controlled by a workflow. Several processors are responsible for analyzing full-text and document properties to generate additional metadata like keywords, key phrases and to extract named entities (person names, companies, geographic items etc.). This can be done with dictionary and/or rule-based classification mechanisms. A widely used framework for this is *GATE*².

Metadata Repository and Information Model

The base of semantic search is a semantic description of the metadata set. On top of this description we can use additional information models expressed in RDF(S) or OWL. But in our experience it is necessary to separate property descriptions from the property store. In our case the *Jena*³ RDF-Store was the main bottle-neck to scale indexing and query performance. Possible solutions are a database-only (not RDF-based) property store or to store all properties in the full-text index also (as we do now with *lucene*⁴).

In the past, most of the knowledge management products use *topic maps* for semantic descriptions and visualize it as a semantic network. Now more and more solutions use a RDF(S) or OWL-based model, because of deep influences from the *Se-*

² <http://gate.ac.uk/>

³ <http://jena.sourceforge.net/>

⁴ <http://lucene.apache.org/>

semantic Web research. A lot of Java-based open source software (see footnotes) was developed based on the RDF technology.

OWL descriptions will be used by several components of an information retrieval system: system configuration, indexing engine and the search client graphical user interface (GUI). An important mission of an information model is to deliver a controlled vocabulary as dictionaries with expert terms or taxonomies or more complex domain ontologies.

Clustering and Classification

A main topic of this summer school is reasoning, so it is not necessary to talk about the benefit of model-based search query optimization by semantic reasoning, but there are other fields to use an information model also.

Clustering and classification are important techniques to make content accessible for users. Most of the established procedures have nothing to do with semantic descriptions. They are all about mathematical algorithms. But classification along Ontologies (or taxonomies) is an interesting aspect with many practical impacts.

Visualization and User Interaction

To present the search results better than a simple flat list is very important for satisfied search engine usage. The result set size and the number of different document properties determines the amount of data to be transferred from the search index to the client visualization components.

An innovative user interface is only usable if it has good performance and usability. In the most cases usability can be translated with "simple design". Very useful GUI elements for navigation and filtering are for example hierarchical faceted browsers.

For systematical investigations it is better to provide metadata catalogs as navigation elements and to present the user statistical views for significant associations between certain metadata. Analyzing user search terms and result sets allows to generate proposals for better search terms regarding the user input.

The search terms put in by the user are very important to discover the investigation context and to deliver proposals for result set grouping and search refinement. If we know the search context, it is possible to direct the user in his search process. But in an enterprise search solution out-of-the-box we have no hint about the search context. For this we would need a semantic description of the world. From the *Semantic Web* we know the effort of this.