Engineering a Semantic Desktop for Building Historians and Architects

René Witte[1], Petra Gerlach[2], Markus Joachim[3],
Thomas Kappler[1], Ralf Krestel[1], and Praharshana Perera[1]

[1] Institut für Programmstrukturen und Datenorganisation (IPD)
Universität Karlsruhe (TH), Germany
Email witte@ipd.uka.de
[2] Institut für Industrielle Bauproduktion (IFIB)
Universität Karlsruhe (TH), Germany
[3] Lehrstuhl für Denkmalpflege und Bauforschung
Universität Dortmund, Germany

Abstract. We analyse the requirements for an advanced semantic support of users—building historians and architects—of a multi-volume encyclopedia of architecture from the late 19^{th} century. Novel requirements include the integration of content retrieval, content development, and automated content analysis based on natural language processing.
We present a system architecture for the detected requirements and its current implementation. A complex scenario demonstrates how a desktop supporting semantic analysis can contribute to specific, relevant user tasks.

1 Introduction

Nowadays, information system users can access more content than ever before, faster than ever before. However, unlike the technology, the users themselves have not scaled up well. The challenge has shifted from finding information in the first place to actually locating useful knowledge within the retrieved content.

Consequently, research increasingly addresses questions of knowledge management and automated semantic analysis through a multitude of technologies [12], including ontologies and the semantic web, text mining and natural language analysis. Language technologies in particular promise to support users by automatically scanning, extracting, and transforming information from vast amounts of documents written in natural languages.

Even so, the question exactly how text mining tools can be incorporated into today's desktop environments, how the many individual analysis algorithms can contribute to a semantically richer understanding within a complex user scenario, has so far not been sufficiently addressed.

In this paper, we present a case study from a project delivering semantic analysis tools to end users, building historians and architects, for the analysis of a historic encyclopedia of architecture. A system architecture is developed upon a detailed analysis of the users' requirements. We discuss the current implementation and state first results from an ongoing evaluation.

2 Analyzing a Historical Encyclopedia of Architecture

Our ideas are perhaps best illustrated within the context of two related projects analysing a comprehensive multi-volume encyclopedia of architecture written in German in the late 19^{th} and early 20^{th} century.[4] In the following, we briefly outline the parties involved and motivate the requirements for an advanced semantic support of knowledge-intensive tasks, which are then presented in the next subsection.

The Encyclopedia. In the 19^{th} century the "Handbuch der Architektur" *(Handbook on Architecture)* was probably not the only but certainly the most comprehensive attempt to represent the entire, including present and past, building knowledge [6]. It is divided into four parts: Part I: Allgemeine Hochbaukunde *(general building knowledge)*, Part II: Baustile *(architectural styles)*, Part III: Hochbau-Konstruktionen *(building construction)*, and Part IV: Entwerfen, Anlage und Einrichtung der Gebäude *(design, conception, and interior of buildings)*.

Overall, it gives a detailed and comprehensive view within the fields of architectural history, architectural styles, construction, statics, building equipment, physics, design, building conception, and town planning.

But it is neither easy to get a general idea of the encyclopedia nor to find information on a certain topic. The encyclopedia has a complex and confusing structure: For each of its parts a different number of volumes—sometimes even split into several books—were published, all of them written by different authors. Some contain more than four hundred pages, others are much smaller, very few have an index. Furthermore, many volumes were reworked after a time and reprinted and an extensive supplement part was added. So referring to the complete work we are dealing with more than 140 individual publications and approximately at least 25 000 pages.

It is out of this complexity that the idea was born to support users—building historians and architects—in their work through state-of-the-art semantic analysis tools on top of classical database and information retrieval systems. However, in order to be able to offer the right tools we first needed to obtain an understanding on precisely what questions concern our users and how they carry out their related research.

User Groups: Building Historians and Architects. Two user groups are involved in the analysis within our projects: Building historians and architects. Those two parties have totally different perceptions of the "Handbuch der Architektur" and different expectations of its analysis. The handbook has got a kind of hybrid significance between its function as a research object and as a resource for practical use, between research and user knowledge.

An architect is planning, designing, and overseeing a building's construction. Although he is first of all associated with the construction of new buildings, more than 60% of building projects are related to the existing building stock, which

[4] Edited by Joseph Durm (⋆14.2.1837 Karlsruhe, Germany, †3.4.1919 ibidem) and three other architects since 1881.

means renovation, restoration, conversion, or extension of an existing building. For those projects he requires detailed knowledge about historic building construction and building materials or links to specialists skilled in this field. For him the gained information is not of scientific but of practical interest.

One of the specialists dealing with architecture from scientific motives is the building historian. All architecture is both the consequence of a cultural necessity and a document that keeps historical information over centuries. It is the task of architectural historians, building archaeologists, and art historians to decipher that information. Architectural history researches all historical aspects of design and construction regarding function, type, shape, material, design, and building processes. It is also considering the political, social, and economical aspects, the design process, the developments of different regions and times, the meaning of shape and its change throughout history. In order to "understand" an ancient building's construction and development, the building historian requires information about historical building techniques and materials. But he is also interested in the information sources themselves, in their history of origin, their development, and their time of writing. Literature research is one of his classical tools.

2.1 Requirements Analysis

We now examine the requirements for a semantic desktop support; first, from a user's perspective, and second, their technical consequents.

User Requirements. For the building historian the handbook itself is object and basis of his research. He puts a high value on a comprehensible documentation of information development, since the analysis and interpretation of the documentation process itself is also an important part of his scientific work. The original text, the original object is the most significant source of cognition for him. All amendments and notes added by different users have to be managed on separate annotation or discussion levels—this would be the forum for scientific controversy, which may result in new interpretations and cognition.

For the architect the computer-aided analysis and accessibility of the encyclopedia is a means to an end. It becomes a guideline offering basic knowledge of former building techniques and construction. The architect is interested in technical information, not in the process of cognition. He requires a clearly structured presentation of all available information on one concept. Besides refined queries ("semantic queries") he requires further linked information, for example web sites, thesauruses, DIN and EU standards, or planning tools.

Both user groups are primarily interested in the content of the encyclopedia, but also in the possibility of finding "unexpected information,"[5] as this would afford a new quality of reception. So far it is not possible to conceive this complex and multi-volume opus with thousands of pages at large: The partition of the handbook in topics, volumes, and books is making the retrieval of a particular

[5] Information delivered through a user's desktop is termed *unexpected* when it is relevant to the task at hand yet not explicitly requested.

concept quite complicated. Only the table of contents is available to give a rough orientation. But it's impossible to get any information about single concepts or terms. You can neither find an overall index nor—apart from a few exceptions—an index of single volumes. Because each of them comprises a huge amount of text, charts, and illustrations, it is unlikely to find the sought-for term coincidentally by running over the pages. Thus, this project's aim is to enable new possibilities of access by the integration of "semantic search engines" and automated analyses. An automated index generation alone would mean a substantial progress for further research work.

System Requirements. In [10] we previously examined the requirements for a system architecture supporting knowledge-intensive tasks, like the ones stated in our case study. Its most important conclusion is that such a system needs to integrate the classically separated areas of information retrieval, content development, and semantic analysis.

Information Retrieval. The typical workflow of a knowledge worker starts by retrieving relevant information. IR systems support the retrieval of documents from a collection based on a number of keywords through various search and ranking algorithms [1,5]. However, with a large number of relevant documents (or search terms that are too broad) this so-called "bag of words approach" easily results in too many potentially relevant documents, leading to a feeling of "information overload" by the user. Furthermore, the retrieval of documents is no end in itself: Users are concerned with the development of new content (like reports or research papers) and only perform a manual search because current systems are not intelligent enough to sense a user's need for information and proactively deliver relevant information based on his current context.

Thus, while also offering our users the classical full-text search and document retrieval functions, we must additionally examine a tighter integration with content development and analysis tools.

Content Development. New content is developed by our users through a number of tasks as outlined above: from questions and notes arising from the examination of a specific building through interdisciplinary discussions to formal research papers. At the same time, access to existing information, like the handbook, and previous results is needed, preferably within a unified interface.

As a model for this mostly interactive and iterative process we propose to employ a *Wiki* system [7], as they have proven to work surprisingly well for cooperative, decentralized content creation and editing. Traditionally, Wikis have been used to develop new material, but our approach here is to combine both existing and new content within the same architecture by integrating (and enhancing) one of the freely available Wiki engines.

Wiki systems allow us to satisfy another requirement, namely the users' need to be able to add their own information to a knowledge source; for example, a building historian might want to add a detailed analysis to a chapter of the encyclopedia, while an architect might want to annotate a section with experiences

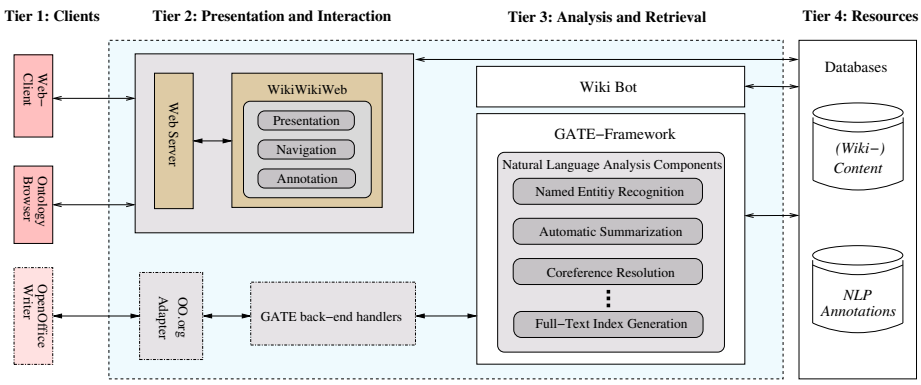

Fig. 1. Architecture integrating content development, retrieval, and analysis

gathered from the restoration of a specific building. Wiki systems typically offer built-in discussion and versioning facilities matching these requirements.

Semantic Analysis. Automated semantic analysis will be provided through tools from the area of natural language processing (NLP), like text mining and information extraction. Typical NLP tasks, which we will discuss in more detail below, are document classification and clustering, automatic summarization, named entity recognition and tracking, and co-reference resolution.

The aforementioned integration of information retrieval, content development, and analysis allows for new synergies between these technologies: content in the Wiki can be continually scanned by NLP pipelines, which add their findings as annotations to the documents for user inspection and internal databases for later cross-reference. When a user now starts to work on a new topic, e.g., by means of creating a new Wiki page, the system can analyse the topic and pro-actively search and propose relevant entities from the database to the user.

3 System Architecture

We now present the architecture we developed to support the detected requirements, as it is currently being implemented. It is based on the standard multi-tier information system design (Fig. 1). Its primary goal is to integrate document retrieval, automated semantic analysis, and content annotation as outlined above. We now discuss each of the four tiers in detail.

Tier 1: Clients. The first tier provides access to the system, typically for humans, but potentially also for other automated clients. A web browser is the standard tool for accessing the Wiki system. Additional "fat" clients, like an ontology browser, are also supported. The integration of the OpenOffice.org word processor is planned for a future version.

Tier 2: Presentation and Interaction. Tier 2 is responsible for information presentation and user interaction. In our architecture it has to deal with both content

development and visualization. In the implementation, most of the functionality here is provided through standard open source components, like the *Apache* web server and the *MediaWiki*[6] content management system.

Tier 3: Retrieval and Analysis. Tier 3 provides all the document analysis and retrieval functionalities outlined above. In addition to the search facilities offered by the Wiki system, a database of NLP annotations (e.g, named entities) can be searched through the *Lucene*[7] search engine.

Semantic analysis of texts through natural language processing (NLP) is based on the GATE framework, which we will discuss in Section 4.3.

The results of the automatic analyses are made visible in an asynchronous fashion through the Wiki system, either as individual pages, or as annotations to existing pages. Thus, automatically created analysis results become first-class citizens: Original content, human, and machine annotations constitute a combined view of the available knowledge, which forms the basis for the cyclic, iterative create-retrieve-analyse process outlined above.

Tier 4: Resources. Resources (documents) either come directly from the Web (or some other networked source, like emails), or a full-text database holding the Wiki content. The GATE framework provides the necessary resource handlers for accessing texts transparently across different (network) protocols.

4 Implementation

In this section we highlight some of the challenges we encountered when implementing the architecture discussed above, as well as their solutions.

4.1 Digitizing History

For our project most of the source material, especially the historical encyclopedia, arrived in non-digital form. As a first step, the documents had to be digitized using specialized book scanners, which were available through Universität Karlsruhe's main library. For automatic document processing, however, scanned page images are unusable. Unfortunately, due to the complexity of the encyclopedia's layout (including diagrams, formulas, tables, sketches, photos, and other formats) and the inconsistent quality of the 100-year old source material, automatic conversion via OCR tools proved to be too unreliable. As we did not want to engage in OCR research, a manual conversion of the scanned material into an electronic document was the fastest and most reliable option that preserved the original layout information, such as footnotes, chapter titles, figure captions, and margin notes. This task was outsourced to a Chinese company for cost reasons.

4.2 Information Storage and Retrieval Subsystem

The encyclopedia is made accessible via *MediaWiki* [9], which is a popular open source Wiki system best known for its use within the *Wikipedia*[8] projects. Media-

[6] http://www.mediawiki.org
[7] http://lucene.apache.org
[8] http://www.wikipedia.org

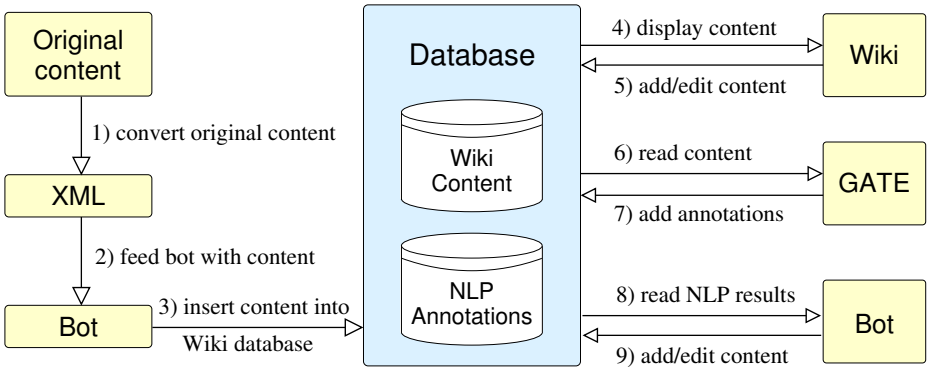

Fig. 2. Workflow between document storage, retrieval, and NLP analysis

Wiki stores the textual content in a MySQL database, the image files are stored as plain files on the server. It provides a PHP-based dynamic web interface for browsing, searching, and manual editing of the content.

The workflow between the Wiki and the NLP subsystems is shown in Fig. 2. The individual sub-components are loosely coupled through XML-based data exchange. Basically, three steps are necessary to populate the Wiki with both the encyclopedia text and the additional data generated by the NLP subsystem. These steps are performed by a custom software system written in Python.

Firstly (step (1) in Fig. 2), the original *Tustep*[9] markup of the digitized version of the encyclopedia is converted to XML. The resulting XML intends to be as semantically close to the original markup as possible; as such, it contains mostly layout information. It is then possible to use XSLT transformations to create XML that is suitable for being processed in the natural language processing (NLP) subsystem described below.

Secondly (2), the XML data is converted to the text markup used by MediaWiki. The data is parsed using the Python `xml.dom` library, creating a document tree according to the W3C DOM specification.[10] This allows for easy and flexible data transformation, e.g., changing an element node of the document tree such as `<page no="12">` to a text node containing the appropriate Wiki markup.

And thirdly (3), the created Wiki markup is added to the MediaWiki system using parts of the Python Wikipedia Robot Framework,[11] a library offering routines for tasks such as adding, deleting, and modifying pages of a Wiki or changing the time stamps of pages. Fig. 3 shows an example of the converted end result, as it can be accessed by a user.

While users can (4) view, (5) add, or modify content directly through the Wiki system, an interesting question was how to integrate the NLP subsystem, so that it can read information (like the encyclopedia, user notes, or other pages) from the Wiki as well and deliver newly discovered information back to the users.

[9] http://www.zdv.uni-tuebingen.de/tustep/tustep_eng.html
[10] http://www.w3.org/DOM/
[11] http://pywikipediabot.sf.net

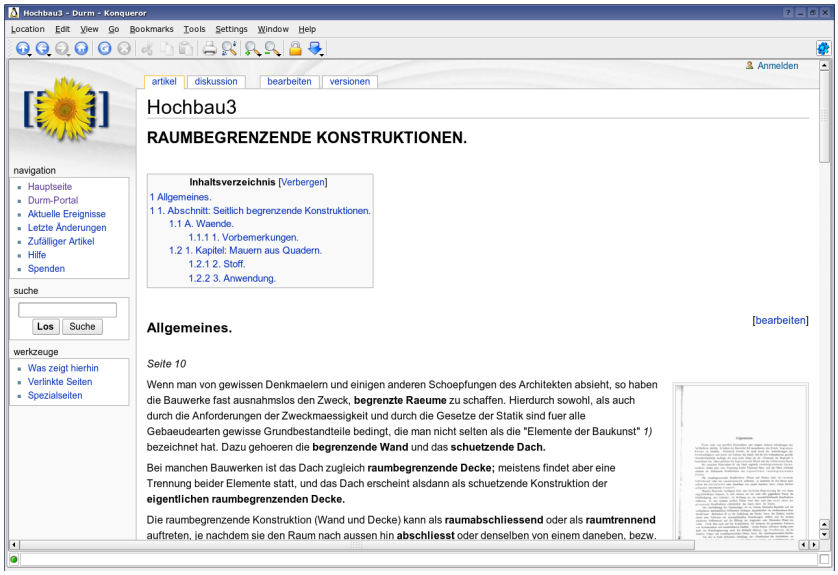

Fig. 3. Content from the encyclopedia accessible through *MediaWiki*

Our solution for this is twofold: for the automated analysis, we asynchronously run all NLP pipelines (described in Section 4.3) on new or changed content (6). The results are then (7) stored as annotations in a database.

The Wiki bot described above is also responsible for adding results from the natural language analysis to the Wiki. It asynchronously (8) reads new NLP annotations and (9) adds or edits content in the Wiki database, based on templates and namespaces. NLP results can appear in the Wiki in two forms: as new individual pages, or within the "discussion section" connected to each page through a special namespace convention within the MediaWiki system. Discussion pages were originally introduced to hold meta-information, like comments, additions, or questions, but we also use them for certain kinds of NLP results, like storing automatically created summaries for the corresponding main page. Other information generated by the NLP subsystem, such as the automatic index generation detailed in Section 4.3, are added to the Wiki as separate pages.

4.3 NLP Subsystem

The natural language analysis part is based on the GATE *(General Architecture for Text Engineering)* framework [4], one of the most widely used NLP tools. Since it has been designed as a component-based architecture, individual analysis components can be easily added, modified, or removed from the system.

A document is processed by a sequential *pipeline* of processing components. These pipelines typically start with basic preprocessing components, like tokenization, and build up to more complex analysis tasks. Each component can add (and read previous) results to the text in form of *annotations*, which form a graph over the document, comparable to the TIPSTER annotation model.

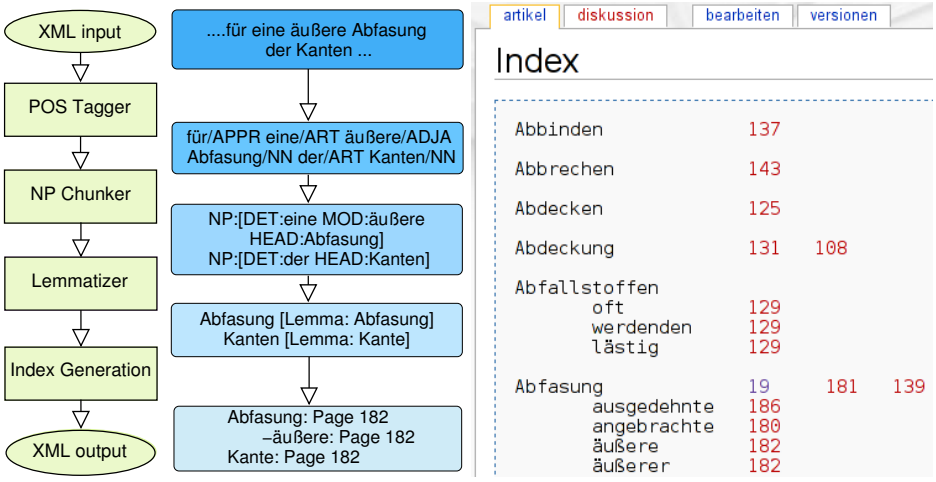

Fig. 4. NLP pipeline for the generation of a full-text index (left side) and its integration into the Wiki system (right side)

We now discuss some the NLP pipelines currently in use; however, it is important to note that new applications can easily be assembled from components and deployed within our architecture.

Automatic Index Generation. Many documents do not come with a classical full-text index, which significantly hinders access to the contained information. Examples include collections of scientific papers, emails, and within our project especially the historical encyclopedia.

In order to allow easier access to the contained information, we use our language tools to automatically create a full-text index from the source documents. This kind of index is targeted at human users and differs from classical indexing for information retrieval in that it is more linguistically motivated: only so-called *noun phrases* (NPs) are permitted within the index, as they form the grammatical base for named entities (NEs) identifying important concepts.

Index generation is implemented as a processing component in the NLP pipeline, which builds upon the information generated by other language components, particularly a part-of-speech (POS) tagger, an NP chunker, and a context-aware lemmatizer (see [8] for details on these steps).

For each noun phrase, we track its lemma (uninflected form), modifiers, and page number. Nouns that have the same lemma are merged together with all their information. Then, we create an inverted index with the lemma as the main column and their modifiers as sub-indexes (Fig. 4, left side).

The result of the index generation component is another XML file that can be inserted into the Wiki system through the Framework described above. Fig. 4 (right side) shows an excerpt of the generated index page for the encyclopedia.

Automatic Context-Based Summarization. Automatically generated summaries are condensed derivatives of a single or a collection of source text(s),

reducing content by selection and/or generalisation on what is important. Summaries serve an indicative purpose: they aim to help a time-constrained human reader with the decision whether he wants to read a certain document or not.

The state of the art in automatic summarization is exemplified by the yearly system competition organized by NIST within the Document Understanding Conference (DUC) [3]. Our summarization pipeline is based on the ERSS system that participated in the DUC competitions from 2003–2005, with some modifications for the German language. One of its main features is the use of fuzzy set theory to build coreference chains and create summaries [11], which enables the user to set thresholds that directly influence the granularity of the results. For more details on the system and its evaluation, we refer the reader to [2]. Summaries can take various forms:

Single-Document Summaries. A single-document summary can range from a short, headline-like 10-word keyword list to multiple sentences or paragraphs. We create these summaries for individual Wiki pages (e.g., holding a chapter of the handbook) and attach the result to its corresponding discussion page.

Multi-Document Summaries. For longer documents, made up of various sections or chapters, or whole document sets, we perform multi-document summarization. The results are stored as new Wiki pages and are typically used for content-based navigation through a document collection.

Focused and Context-Based Summaries. This most advanced form of multi-document summarization does not create summaries in a generic way but rather based on an explicit question or user context. This allows for the pro-active content generation outlined above: a user working on a set of questions, stated in a Wiki page (or, in future versions, simple by typing them into a word processor), implicitly creates a context that can be detected by the NLP subsystem and fed into the context-based summarization pipeline, delivering content from the database to the user that contains potentially relevant information. We show an example in Section 5.2.

Ontology-based Navigation and Named Entity Detection. Ontologies are a more recent addition to our system. We aim to evaluate their impact on the performance of the named entity (NE) recognition, as well as semantic navigation through a browser.

Named entities are instances of concepts. They are particular to an application domain, like *person* and *location* in the newspaper domain, *protein* and *organism* in the biology domain, or *building material* and *wall type* in the architecture domain.

The detection of named entities is important both for users searching for particular occurrences of a concept and higher-level NLP processing tasks. One way of detecting these NEs, supported by the GATE framework, is a markup of specific words, defined in *Gazetteer* lists, which can then be used together with other grammatical analysis results in so-called finite-state transducers defined through regular-expression-based grammars in the JAPE language.[12]

[12] For more details, please refer to the GATE user's guide: `http://gate.ac.uk/`.

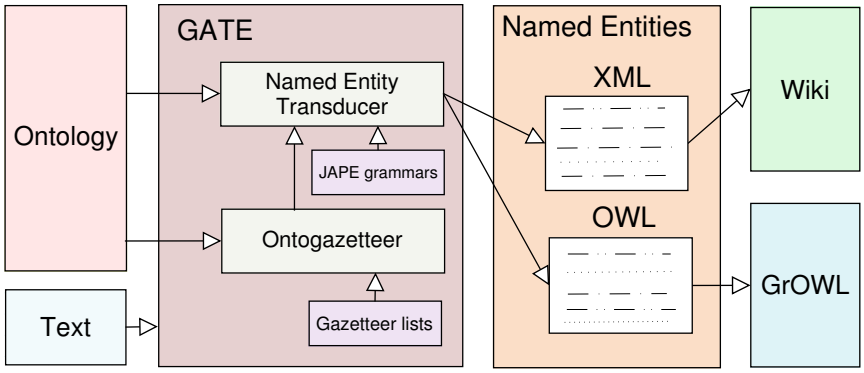

Fig. 5. Ontology-aware named entity detection through gazetteers and finite-state transducers delivering results in various output formats

The addition of ontologies (in DAML format) allows to locate entities within an ontology (currently, GATE only supports taxonomic relationships) through ontology extensions of the Gazetteer and JAPE components. The detected entities are then exported in an XML format for insertion into the Wiki and as an OWL RDF file (Fig. 5).

NE results are integrated into the Wiki similarly to the index system described above, linking entities to content pages. The additional OWL export allows for a graphical navigation of the content through an ontology browser like GrOWL.[13] The ontologies exported by the NLP subsystem contain sentences as another top-level concept, which allows to navigate from domain-specific terms directly to positions in the document mentioning a concept, as shown in Fig. 6.

5 Evaluation

We illustrate a complex example scenario where a building historian or architect would ask for support from the integrated system.

5.1 Scenario

The iterative analysis process oriented on the different requirements of the two user groups is currently being tested on the volume "Wände und Wandöffnungen"[14] *(walls and wall openings)*. It describes the construction of walls, windows, and doors according to the type of building material. The volume has 506 pages with 956 figures; it contains a total of 341 021 tokens including 81 741 noun phrases.

Both user groups are involved in a common scenario: The building historian is analysing a 19^{th} century building with regard to its worth of preservation in order to be able to identify and classify its historical, cultural, and technical

[13] http://seek.ecoinformatics.org/Wiki.jsp?page=Growl

[14] E. Marx: *Wände und Wandöffnungen*. Aus der Reihe: Handbuch der Architektur. Dritter Teil, 2. Band, Heft I, 2. Auflage, Stuttgart 1900.

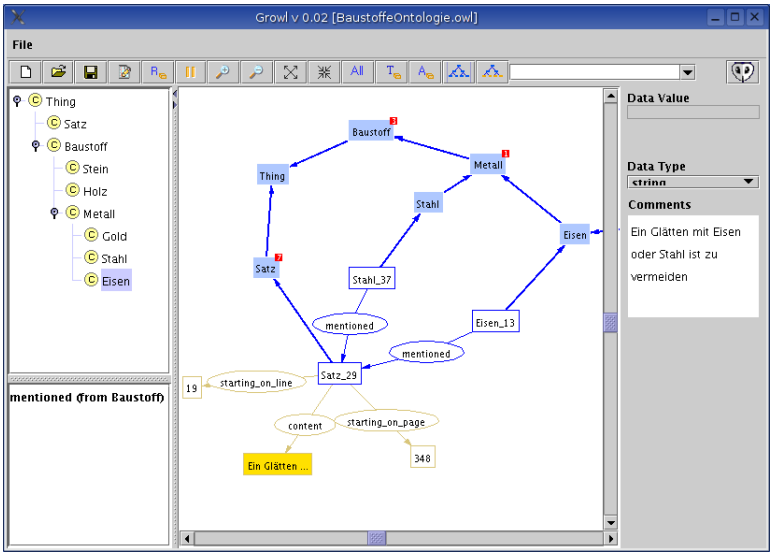

Fig. 6. Ontology excerpt on *building materials* visualized using GrOWL showing NLP-detected instances linked to concepts, anchored within the text

value. The quoins, the window lintels, jambs, and sills as well as door lintels and reveals are made of fine wrought parallelepipedal cut sandstones. The walls are laid of inferior and partly defective brickwork. Vestiges of clay can be found on the joint and corner zones of the brickwork. Therefore, a building historian could make the educated guess that the bricks had been rendered with at least one layer of external plaster. Following an inspection of the building together with a restorer, the historian is searching in building documents and other historical sources for references to the different construction phases. In order to analyse the findings it is necessary to become acquainted with plaster techniques and building materials. Appropriate definitions and linked information can be found in the encyclopedia and other sources. For example, he would like to determine the date of origin of each constructional element and whether it is original or has been replaced by other components. Was it built according to the state-of-the-art, does it feature particular details?

In addition, he would like to learn about the different techniques of plastering and the resulting surfaces as well as the necessary tools. To discuss his findings and exchange experiences he may need to communicate with other colleagues.

Even though he is dealing with the same building, the architect's aim is another. His job is to restore the building as carefully as possible. Consequently, he needs to become acquainted with suitable building techniques and materials, for example, information about the restoration of the brick bond. A comprehensive literature search may offer some valuable references to complement the conclusion resulting from the first building inspection and the documentation of the construction phases.

“Welche Art von Putz bietet Schutz vor Witterung?”
Ist das Dichten der Fugen für die Erhaltung der Mauerwerke, namentlich an den der Witterung ausgesetzten Stellen, von Wichtigkeit, so ist es nicht minder die Beschaffenheit der Steine selbst. Bei der früher allgemein üblichen Art der gleichzeitigen Ausführung von Verblendung und Hintermauerung war allerdings mannigfach Gelegenheit zur Beschmutzung und Beschädigung der Verblendsteine geboten. . . .

Fig. 7. Excerpt from a focused summary generated based on a question (shown on top), generated by the NLP subsystem through automatic summarization

5.2 Desktop Support

So far, we have been testing the desktop with the Wiki system and three integrated NLP tools within the project. We illustrate how our users ask for semantic support from the system within the stated scenario.

NLP Index. As the tested volume offers just a table of contents but no index itself, an automatically generated index is a very helpful and timesaving tool for further research: Now it is possible to get a detailed record on which pages contain relevant information about a certain term. And because the adjectives of the terms are indicated as well, information can be found and retrieved very quickly, e.g., the architect analysing the plain brickwork will search for all pages referring to the term “Wand” *(wall)* and in particular to “unverputzte Wand” *(unplastered wall).*

Summaries. Interesting information about a certain topic is often distributed across the different chapters of a volume. In this case the possibility to generate an automatic summary based on a context is another timesaving advantage. The summary provides a series of relevant sentences, e.g., to the question (Fig. 7): “Welche Art von Putz bietet Schutz vor Witterung?” *(Which kind of plaster would be suitable to protect brickwork against weather influences?).* An interesting properties of these context-based summaries is that they often provide “unexpected information,” relevant content that a user most likely would not have found directly.

The first sentence of the automatic summarization means: *The joint filling is important for the resistance of the brickwork, especially for those parts exposed to the weather, as well as the quality of the bricks.* This is interesting for our example because the architect can find in the handbook—following the link—some information about the quality of bricks. Now he may be able to realize that those bricks used for the walls of our 19^{th} century building are not intended for fare-faced masonry. After that he can examine the brickwork and will find the mentioned vestiges of clay.

The architect can now communicate his findings via the Wiki discussion page. After studying the same text passage the building historian identifies the kind of brickwork, possibly finding a parallel to another building in the neighborhood, researched one year ago. So far, he was not able to date the former building precisely because all building records have been lost during the war. But our

example building has a building date above the entrance door and therefore he is now able to date both of them.

Named Entity Recognition and Ontology-based Navigation. Browsing the content, either graphically or textually, through ontological concepts is another helpful tool for the users, especially if they are not familiar in detail with the subject field of the search, as it now becomes possible to approach it by switching to super- or subordinate concepts or instances in order to get an overview. For example, restoration of the windows requires information of their iron construction. Thus, a user can start his search with the concept "Eisen" *(iron)* in the ontology (see Fig. 6). He can now navigate to instances in the handbook that have been linked to "iron" through the NLP subsystem, finding content that mentions window and wall constructions using iron. Then he can switch directly to the indicated parts of the original text, or start a more precise query with the gained information.

5.3 Summary

The offered semantic desktop tools, tested so far on a single complete volume of the encyclopedia, turned out to be a real support for both our building historians and architects: Automatic indices, summaries, and ontology-based navigation can help them to find relevant, precisely structured and cross-linked information to certain, even complex topics in a quick and convenient fashion. The system's ability to cross-link, network, and combine content across the whole collection have the potential to guide the user to unexpected information, which he might not have realized even when completely reading the sources themselves.

In doing so the tools' time saving effects seems to be the biggest advantage: Both user groups can now concentrate on their research or building tasks—they do not need to deal with the time-consuming and difficult process of finding interesting and relevant information.

6 Conclusions and Future Work

In this paper, we showed how a user's desktop can integrate content retrieval, development, and NLP-based semantic analysis. The architecture is based on actual users' requirements and preliminary evaluations show the feasibility and usefulness of our approach. We believe our system also applies to other domains.

From a technical perspective, the biggest challenge is the lack of hooks in standard desktop components designed for use by humans enabling read-, write-, and navigate operations from automated components, requiring expensive workarounds. In our system, automated access to the Wiki system by the NLP subsystem requires the use of a bot, which was not originally designed for that purpose. We currently face similar problems integrating the OpenOffice.org word processor into the system. There is currently no way several desktop components can share a common semantic resource, like an ontology, or even delegate analysis tasks on behalf of a user. On a smaller scale, we are currently working on integrating a description logic (DL) reasoning system to allow semantic queries based on the automatically extracted entities.

However, one of the most interesting questions, from an information system engineer's standpoint, is a concern raised by our building historians: the apparent loss of knowledge throughout the years, which occurs when users of automated systems narrowly apply retrieved information without regard for its background, connections, or implications; or when they simply do not even find all available information because concepts and techniques have been lost over the years: As a result, a user might no longer be aware of existing knowledge because he lacks the proper terminology to actually retrieve it. While an analysis of this effect is still an ongoing consideration, we hope that the multitude of access paths offered by our integrated approach at least alleviates this problem.

Acknowledgments. The work presented here is funded by the German research foundation (DFG) through two related projects: "Josef Durm" (HA 3239/4-1, building history, Uta Hassler and KO 1488/7-1, architecture, Niklaus Kohler) and "Entstehungswissen" (LO296/18-1, informatics, Peter C. Lockemann).

References

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.
2. Sabine Bergler, René Witte, Zhuoyan Li, Michelle Khalifé, Yunyu Chen, Monia Doandes, and Alina Andreevskaia. Multi-ERSS and ERSS 2004. In *Workshop on Text Summarization*, Document Understanding Conference (DUC), Boston Park Plaza Hotel and Towers, Boston, USA, May 6–7 2004. NIST.
3. Document Understanding Conference. http://duc.nist.gov/.
4. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proc. of the 40th Anniversary Meeting of the ACL*, 2002. http://gate.ac.uk.
5. Reginald Ferber. *Information Retrieval*. dpunkt.verlag, 2003.
6. Ulrike Grammbitter. *Josef Durm (1837–1919). Eine Einführung in das architektonische Werk*, volume 9 of *tuduv-Studien: Reihe Kunstgeschichte*. tuduv-Verlagsgesellschaft, München, 1984. ISBN 3-88073-148-9.
7. Bo Leuf and Ward Cunningham. *The Wiki Way, Quick Collaboration on the Web*. Addison-Wesley, 2001.
8. Praharshana Perera and René Witte. A Self-Learning Context-Aware Lemmatizer for German. In *Human Language Technology Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Vancouver, B.C., Canada, October 6–8 2005.
9. Wikipedia, the free encyclopedia. Mediawiki. http://en.wikipedia.org/wiki/MediaWiki; accessed July 26, 2005.
10. René Witte. An Integration Architecture for User-Centric Document Creation, Retrieval, and Analysis. In *Proceedings of the VLDB Workshop on Information Integration on the Web (IIWeb)*, pages 141–144, Toronto, Canada, August 30 2004.
11. René Witte and Sabine Bergler. Fuzzy Coreference Resolution for Summarization. In *Proc. of 2003 Int. Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 43–50, Venice, Italy, June 23–24 2003. Università Ca' Foscari. http://rene-witte.net.
12. Ning Zhong, Jiming Liu, and Yiyu Yao, editors. *Web Intelligence*. Springer, 2003.