Converting a Historical Architecture Encyclopedia into a Semantic Knowledge Base

René Witte, *Concordia University*

Ralf Krestel, *L3S Research Center*

Thomas Kappler, *Swiss Institute of Bioinformatics*

Peter C. Lockemann, *Karlsruhe Institute of Technology*

Digitizing a historical document using ontologies and natural language processing techniques can transform it from arcane text to a useful knowledge base.

The *Handbook on Architecture* (*Handbuch der Architektur*) was perhaps one of the most ambitious publishing projects ever. Like a 19th-century Wikipedia, it attempted nothing less than a full account of all architectural knowledge available at the time, both past and present. It covers topics from Greek temples to contemporary hospitals and universities; from the design of individual construction elements such as window sills to large-scale town planning; from physics to design; from planning to construction. It also discusses architectural history and styles and a multitude of other topics, such as building conception, statics, and interior design.

Not surprisingly, this project took longer than planned. The encyclopedia's first volume was partly published in 1880, and over the next 63 years more than 100 architects worked on what would become more than 140 individual publications with over 25,000 total pages.

Somewhat tragically, the encyclopedia was quickly forgotten after its completion. The complete set is now available in only a few libraries. Its large scope, lack of a comprehensive index, and complex structure (with different aspects covered in separate parts) make it nearly impossible to gain an overview on a specific topic. Moreover, during the long publication phase, several volumes were edited and reprinted, and an extensive supplement was added, further complicating knowledge access.

1541-1672/10/$26.00 © 2010 IEEE
Published by the IEEE Computer Society

It is worthwhile to revisit this comprehensive encyclopedia, not only for historical reasons but also because much of the stored knowledge is still valid even by today's standards. For architects working in restoration, it can provide first-hand knowledge of the construction of an old building. The encyclopedia also contains knowledge that is not otherwise available on the Web. Digital technologies, together with semantic approaches including ontologies and text mining, can help us overcome this publication's knowledge-access challenges and even allow the use of the historical encyclopedia as a semantic knowledge base within modern software systems.

Semantic User Interfaces

The baseline for digital heritage document management is the generation of an online version that can display the scanned pages of the original document, possibly overlaid by a text-searchable version. This approach is highly focused on the original layout; it simply provides a "digital page" metaphor with suitable tools, emphasizing the digital preservation of heritage documents. This is certainly appropriate when the heritage documents are treated as artifacts. It is less useful when their content is the primary topic of investigation. In our case, we want to provide an interface that emphasizes the documents' encyclopedic nature in a way that allows architects and building historians to work with the heritage material. Such an interface must include facilities for collaboratively annotating the original documents—for example, to add additional levels of interpretation. For instance, the building materials available today might have very different qualities from those used 100 years ago, so even using the same construction process in a restoration project would lead to very different results. A historian might also discover that a certain description in the encyclopedia is an idealistic view that doesn't correspond to the way buildings were actually constructed at the time—giving further important clues to contemporary architects who are interested in the historical documents from a practical perspective.

From these observations we can derive several requirements for a user interface: First, it must have annotation capabilities that are clearly separate from the original content. Second, it must have full-text search functions because of the immense size of the encyclopedia, while still allowing direct access to the scanned page images. Third, the interface should provide navigation based on structure and topic—that is, not simply page-by-page navigation. On the basis of these requirements, we decided to develop a wiki-based user interface (see http://en.wikipedia.org/wiki/Wiki) for heritage document management.

A Wiki User Interface for Heritage Documents

We built our wiki interface to the historical encyclopedia using the MediaWiki framework (see http://www.mediawiki.org), which is perhaps best known for its use within Wikipedia. After scanning and digitizing the original publication (we describe the details of this process elsewhere[1]), we converted the encyclopedia into wiki pages on the basis of their structure: Each wiki page comprises a section (within a chapter) of the original. We generated additional pages for each chapter, linking the contained sections and containing content that sometimes appears before the first section heading in the printed original. To provide further structure to the wiki pages, we used the margin notes contained in the printed original as subsection headings. We transferred the typographical features of the original—such as italics, small caps, and footnotes—to the online version as faithfully as possible. We extracted figures (sometimes with captions, sometimes without) and added them as blocks, linking them to the scanned pages.

A public version of this wiki with a restricted set of features is available at http://durm.semanticsoftware.info/wiki. Figure 1 shows an example of the resulting user interface. Users have access to all the features delivered by the MediaWiki software, such as full-text search and automatically generated page content tables. An important feature is the additional generation of page indicators, also shown in Figure 1: these ensure that the content of a wiki page, which typically spans several book pages, is always associated with the printed original. This is essential for historians for citation purposes. Finally, we generated some additional pages—such as a hyperlinked list of all figures and a section and page index—that aren't available in the printed original.

The wiki interface also provides the annotation support stipulated earlier: In MediaWiki, each page is associated with a discussion page (the second tab in Figure 1), which lets users store comments and questions about the content or its application, thereby providing a collaborative annotation facility for the encyclopedia.

Adding Natural Language Processing Support

The concepts we have described thus far provide for a convenient and functional user interface to the historical documents, but by themselves do not significantly alleviate the knowledge access problem we discussed earlier. Two basic approaches—using ontologies and natural language processing (NLP) techniques—have proved

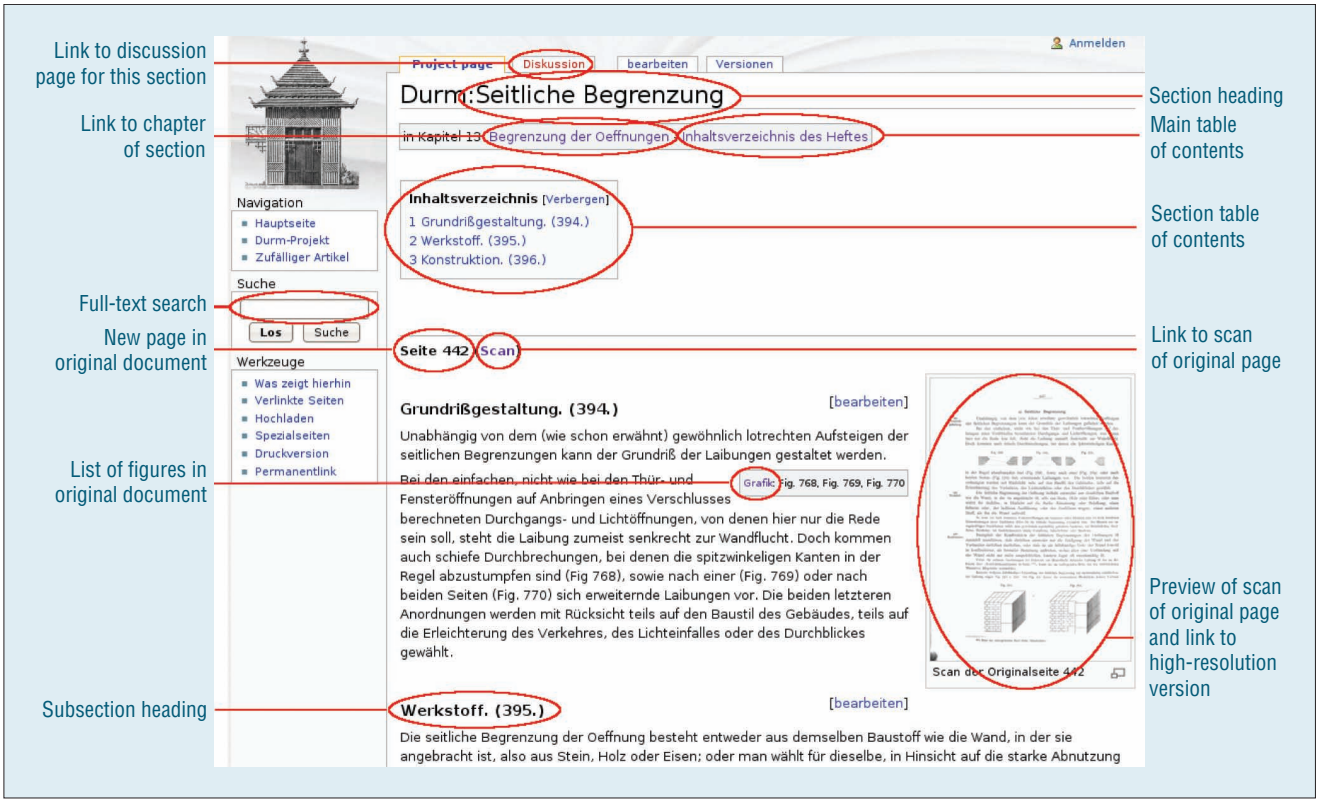

Figure 1. The wiki user interface for the historical encyclopedia, *Handbook on Architecture*.

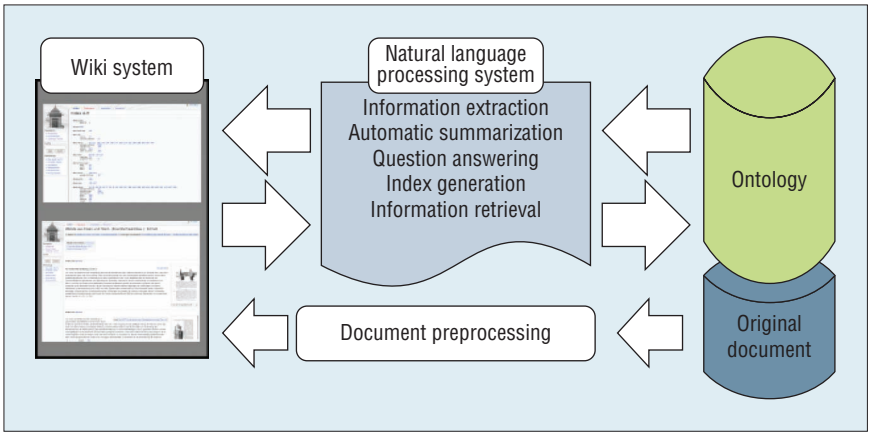

Figure 2. System architecture overview.

especially effective for this purpose. We will return to ontologies later; first, we show how our project employs language technology to support users. Other projects have already demonstrated how to successfully deploy NLP techniques for cultural heritage data: prominent examples include morphological analysis, information extraction, and automatic summarization.[2–4]

Our goal in using these technologies is to aid our users in managing the large amount of content in the encyclopedia and its complex structure. NLP allows automatic analysis of the encyclopedia, the addition of annotations, and the solution of several content-access tasks. We describe some of these analysis services in this article: basic processing and lemmatization, generation of a back-of-the-book index, summarization, and question answering. Our system integrates some of the analysis results directly into the wiki interface, using either new content pages or the discussion pages mentioned earlier. Figure 2 shows an overview of our system architecture—in particular, the connections between the original content, the wiki interface, and the analysis services.

Language Processing: From Basics to Lemmatization

NLP and text mining are computational approaches to dealing with natural-language documents—in our case, written in German. We implement these technologies in our system by assembling individual NLP components into analysis pipelines that can deliver the required results—for example, generate an index or answer a question. Some of these components implement basic processing steps used in every pipeline (such as tokenization);

others are particular to a specific task. Likewise, some components operate in a language-independent fashion, whereas others require language-specific resources or algorithms.

We implemented our NLP pipelines using the General Architecture for Text Engineering (GATE) framework.[5] GATE comes with several ready-to-deploy components, such as tokenization and sentence splitting, which can be immediately used as basic building blocks.

An important analysis step used by all our processing pipelines is the computation of a noun's *lemma*, or basic form—for example, deriving the base form *Haus* from the plural form *Häuser*. This is a difficult task in German because of the language's complex morphological structure, and it requires extensive resources that are unfortunately not freely available. Moreover, a particular characteristic of historical documents is that they contain many nouns that are no longer in use today and are therefore not contained in contemporary word lists or dictionaries. Additionally, word spellings have changed noticeably because of several grammatical reforms. Furthermore, because the documents are specific to architecture, they contain many words that aren't commonly found in the general domain. Because the manual development of a dictionary specifically for our encyclopedia would be an immense task, we developed an algorithm that can automatically learn the correct lemma of a noun using a combination of rule-based and statistical techniques by looking at a word's context—that is, a complete sentence.[6] This algorithm is independent of any domain, so we can apply it to our encyclopedia of architecture, and to any other domain-specific document where existing resources would fail. Our *Durm Lemmatizer* and all its resources are freely available under an open source license (see http://www.semanticsoftware.info/durm-german-lemmatizer).

Building an Index Using NLP

With the digital version of the encyclopedia, users can now use full-text search and information retrieval techniques to locate relevant knowledge. However, these approaches don't support knowledge discovery, in that the user must provide the search terms for the query. Especially in the case of historical documents, the terms used might have shifted over time. In the simplest case, a particular spelling might have changed over the centuries, but more importantly, terms common a century ago might no longer be in use. Obviously, if a user isn't aware of these changes, he won't find all relevant information. How can we facilitate knowledge discovery in this case?

To facilitate content access—especially for practitioners and laypersons—automatic summarization provides an instrument to condense information.

A common technique for finding content is the classical back-of-the-book index. Because the encyclopedia does not come with such an index, we investigated its automatic generation. This kind of index is different from a simple word list. For our project, we decided to index noun phrases (NPs) in the following manner: Our language-processing pipeline uses the TreeTagger (see http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger) to determine the part of speech for each word (noun, verb, adjective, and so on). Using this information, the MuNPEx chunker (Multilingual Noun Phrase Extractor, http://www.semanticsoftware.info/munpex) groups words into NPs, which consist of a head noun, a (possibly empty) list of adjectives, and an optional determiner. The Durm Lemmatizer, described earlier, lemmatizes the head noun. We can now build an inverted two-level index in the following fashion: The list of lemmatized head nouns makes up the first level of the index. Each recorded occurrence is linked to the page number where it was found. The adjectives of each occurrence are then used to create a second-level entry for each noun. Figure 3 shows an example of the process, the creation of an entry for the NP *eine äußere Abfasung* (Figure 3a), which is then merged with other occurrences of "*Abfasung*," shown in the generated index (Figure 3b). We then transform this index into the markup used within the wiki interface, with hyperlinks inserted that let a user go directly from the index to the page containing the entry.

Summarizing Content and Answering Questions

The vast amount of information in the encyclopedia makes it impossible to obtain all relevant information about a particular topic by browsing through the whole document by hand. To facilitate content access—especially for practitioners and laypersons—automatic summarization provides an instrument to condense information in various ways.[7] These automatically generated summaries let the reader decide whether or

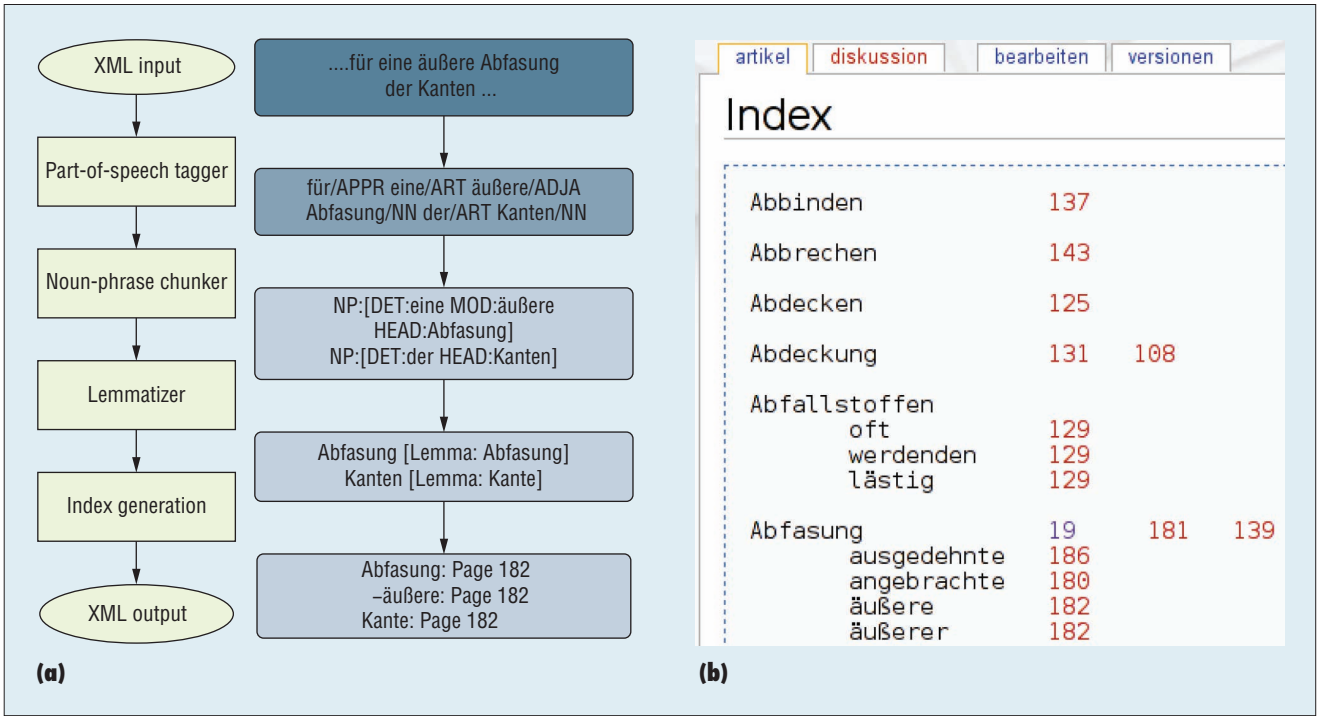

Figure 3. Building an index for a heritage document: (a) natural language processing (NLP) pipeline for automatically generating a back-of-the-book style index and (b) its integration into the wiki system. Page numbers are hyperlinks to wiki pages.

"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....

Figure 4. Summarizing content. In addition to summarizing individual chapters or sections, the NLP subsystem can generate an answer to an explicit question (top) by extracting and then assembling sentences that might answer the question. This excerpt of a focused summary is a response to the question "Which kind of plaster would be suitable to protect brickwork against weather influences?"

not to read the entire document (wiki page), thus saving time and effort.

Our system offers three types of summaries, generated by extracting sentences from the original text on the basis of the contained NPs.[8] Single-document summaries can range from a short, headline-like 10-keyword list to multiple sentences or paragraphs. We create these summaries for individual wiki pages (each, for example, holding a chapter of the handbook) and attach the result to the corresponding discussion page. To summarize longer parts, made up of multiple sections or chapters, or whole document sets, we perform multidocument summarization. The results are stored as new wiki pages and are typically used for content-based navigation through a document collection. The most advanced form of multidocument summarization doesn't create summaries in a generic way but rather on the basis of an explicit question. The system extracts sentences possibly answering the question from the document and assembles them into a summary, which it then presents to the user. This provides a quick overview of the information in the corpus related to the question. Figure 4 shows an example answering the question at the top, "Which kind of plaster would be suitable to protect brickwork against weather influences?" An interesting property of these context-based summaries is that they often provide "unexpected information"—relevant content that a user most likely would not have found directly.

From Historical Books to a Digital Knowledge Base

The technologies we have described thus far all aim at directly supporting users in knowledge discovery and processing. They are less useful for other systems that want to use the knowledge "stored" in the encyclopedia, such as software used by an architect. But we also want to allow such tools to access the encyclopedia. For example, a construction tool used in the restoration of an old building could display relevant knowledge about construction elements from

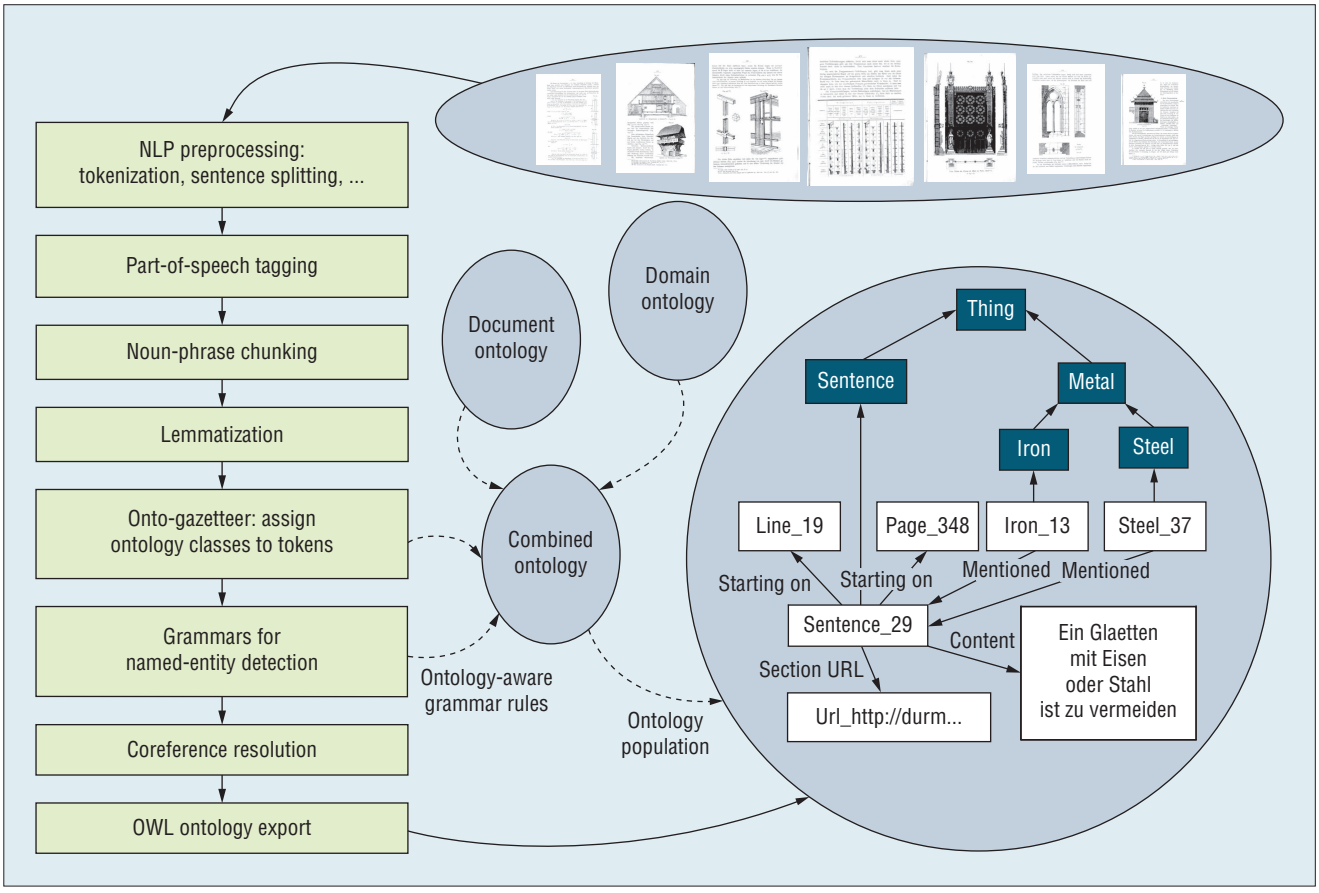

Figure 5. Ontology population example. The original documents are analyzed through a pipeline of natural language processing (NLP) components, shown on the left. Results of the analysis are exported into an ontology, which then allows the connection of content, such as a sentence, with domain concepts, such as building materials.

the historical encyclopedia alongside modern construction knowledge such as catalogs and standards. For this, we need to build a knowledge base, a formal representation of the historical knowledge that can be accessed through well-defined interfaces.

This requires the use of a representation formalism that can handle the semantics of the application domain (in our case, architecture), as well as the results of our automated analyses. In our approach, we use formal OWL ontologies based on description logics (http://www.w3.org/2004/OWL), which are a World Wide Web Consortium (W3C) standard used within the Semantic Web.[9] Ontologies can be used to structure, browse, and search data,[3,10] as well as to merge or integrate knowledge from different sources.[11,12]

Ontology Design for Cultural Heritage Data

An ontology allows the structuring of a domain of discourse using concepts and the relations between them. In our project, we have to capture two subdomains: (heritage) document management and the content domain, which in our example is architecture.

For the first subdomain, we created an ontology that models concepts for document management. In particular, it contains concepts such as `Sentence`, `NounPhrase`, and `PageNumber`. This ontology allows the description of source documents and automated text-analysis results. For example, an NP detected through our analysis pipeline (Figure 5) can be captured by the ontology and connected with the precise occurrence in the source document—both the printed version, through page numbers, and the digital version, using URLs and start/end offsets.

The second ontology is specific to the domain covered by the content of the heritage documents—that is, architecture. This ontology contains concepts such as `Wall` and `BuildingMaterial`. Although our approach to heritage document management itself is domain independent, it relies on the knowledge encoded in the domain ontology. In our case, the end users delivered this ontology, but for many domains OWL ontologies are freely available on the Web. These can at least provide a starting point for heritage data management.

We then combine the two ontologies to build the conceptual foundation for the heritage knowledge base, as Figure 5 shows. We can now connect architectural concepts with document-specific ones—for example, sentences

that mention construction elements of a certain material. However, before we can query the ontology in this way, we must first populate it.

Automatic Ontology Population

So far, the ontology contains only concepts, indicating what kind of knowledge can potentially be found in the documents. This alone does not help an end user in actually locating knowledge in the encyclopedia. Before this becomes possible, each concept must be populated with instances (OWL individuals). For example, every sentence in the document becomes an individual of the concept `Sentence`, with its corresponding information such as start and end position (page, URL) stored as OWL ObjectProperties. Figure 5 depicts a few example instances as unfilled boxes (giving a greatly simplified subset of the ontology). The populated knowledge base now allows the connection of each concrete occurrence of a certain concept (such as a construction element or material) with the location where it can be found in the encyclopedia.

Obviously, creating all these instances by hand is not a feasible approach—the concept `NounPhrase` alone requires the creation of 81,741 individuals. Our system achieves automatic ontology population through an NLP pipeline, which is essentially an extension of the analysis process described earlier for index generation and automatic summarization (Figure 5, left side).

Stated briefly, ontology population works as follows: in the first steps, the system preprocesses and lemmatizes documents as described earlier. Then, ontology-specific processing steps start matching detected lemmas with the ontology classes. An onto-gazetteer assigns an ontology class to each token according to mappings between ontology classes and tokens. These mappings are generated automatically on the basis of several heuristics, but can also be manually enhanced if the automatic process doesn't provide sufficient coverage. For example, for the ontology class "plaster," the system automatically creates a corresponding mapping, allowing this ontology class to be assigned to each occurrence of this word in the documents. In addition, our architects defined "mortar" and "grout" as also mapping to this ontology class. The next step in the pipeline applies grammars written in the JAPE language[5] as a cascade of finite-state transducers. These grammars can access the ontology and create named entities on the basis of the annotations created by the onto-gazetteer and the NP chunker. This ensures that entities correspond to the proper grammatical roles—for example, a verb tagged by the onto-gazetteer based on simple string matching cannot become a named entity. Only when a text segment is both recognized by the onto-gazetteer and confirmed by the grammar rules is it linked to its corresponding ontology class ("plaster" in the example just given) and therefore becomes an instance of this class. The coreference resolution step now compares detected named entities with one another and connects those that refer to the same referent in *coreference chains*, which are essentially equivalence classes of entities. In a final step, the OWL exporter connects the named entities with the document model described earlier, creating the connections between the domain concepts and the document concepts (such as sentences, noun phrases, and other document-specific information). For example, coreference chains are exported using the OWL language construct `owl:sameAs`. The OWL exporter also creates additional information for each exported individual, such as the page number in the original encyclopedia and a URL to the wiki page where this instance can be found (see Figure 5). The end result of this step is a populated ontology that connects architectural concepts with the actual content in the historical encyclopedia.

> Throughout its development, the system was validated by end users from building history and architecture.

Of course, this automatic population step might reveal inconsistencies between the domain ontology and the detected instances: there might be concepts for which no instances could be detected, or, likewise, detected noun phrases that could not be matched to any ontology concept. This information can be extracted from the population subsystem and used for further, iterative refinement of both the domain ontology and the population resources (such as gazetteering lists and grammars).

Querying the Ontology

The populated ontology can now be queried, either by a human user or another software system. Essentially, ontology queries and reasoning support automated problem solving. For example, a historian might want to formulate hypotheses concerning the source material. Translated into an OWL query, the result can be used to confirm or refute the hypothesis.

Ontology queries can enable far more fine-grained knowledge retrieval than simple full-text search. For example, a full-text search cannot

```
SELECT DISTINCT ?type ?page ?sentence ?content
WHERE {
       ?sentence :contains ?x;
          :contains ?y;
          :content ?content .
       ?x :originalPageNumber ?page;
          :pageURL ?pageurl;
          rdf:type ?type .
       ?y rdf:type :Mauer .
       ?type rdfs:subClassOf :Materialien
}
ORDER BY ?page
```

Figure 6. SPARQL translation of the question "Which building materials are mentioned in the handbook together with the concept 'Mauer' (wall), and on which pages?"

retrieve all passages where some kind of building material is mentioned. The capability for answering such questions directly is one of the benefits of an ontological knowledge base. However, before such questions can be answered from the knowledge base, they must be translated into a suitable ontology query language, such as SPARQL (see http://www.w3.org/TR/rdf-sparql-query). For example, Figure 6 shows the SPARQL translation of the question "Which building materials are mentioned in the handbook together with the concept 'Mauer' (wall), and on which pages?"

Upon execution, this query returns a subgraph of the ontology containing the requested information—here, the specific building material used (stone, concrete, iron), the page number in the original where it was found (and the URL in the wiki if desired), the sentence number, and its complete content.

Although this is interesting for users who want to analyze the historical material, ontology queries can also be generated by other systems that need to access knowledge from the encyclopedia, thereby directly integrating its content as a knowledge base into another end-user application or a larger automated knowledge discovery workflow. For example, an architect might want to access the knowledge stored in the handbook while planning a particular building-restoration task. Here, construction elements displayed in a design tool (such as a window or window sill) can be directly connected with the ontological entities contained in the NLP-populated knowledge base. This allows an architect to view relevant content down to the level of an individual construction element using the named entities, while retaining the option to visit the full text through the provided wiki link. We envision that such semantic extensions to construction tools will eventually be integrated into commercial products.

Experiences

We have implemented our ideas and tested them with a single volume of the encyclopedia that describes walls and wall openings.[13] The volume has 506 pages and 956 figures; it contains a total of 341,021 tokens, including 81,741 noun phrases. The complete data set—scanned page images, layout-focused Tustep markup, custom XML format for NLP—is available online under an open content license.[1] The automatic ontology population process creates about 10,000 instances, distributed over roughly 600 classes, stored in a 35-Mbyte OWL file (numbers depend on the runtime configuration).

Many of the concepts we have described in this article have been evaluated individually, such as the self-learning German lemmatizer, which performs with an accuracy of about 95 percent,[6] and the summarization strategies, which have been evaluated extensively at the DUC summarization competition, where they proved to be highly competitive across a wide range of summarization tasks.[8]

Throughout its development, the system was validated by end users from building history and architecture, who analyzed the encyclopedia within their own research. From their perspective, the introduction of the automatic index generation was particularly effective, because it offered a familiar interface metaphor to browse the historical texts, which was highly appreciated. Moreover, the generated entries underline the importance of alternative access methods, because several terms can't be found on the Web except in the encyclopedia. One such example is *Kokekorb,* which (when we originally wrote this article) Google would only find in the online version of our wiki. Overall, the various developed access methodologies are somewhat complementary, so they can balance out one another's weaknesses when combined in a complete system as proposed here.

Ontology queries, while a powerful paradigm, are difficult to exploit for users unfamiliar with formal query languages. In this area, ongoing research on transforming natural-language questions into formal ontology queries, together with graphical query facilities, has the potential to revolutionize knowledge discovery in the heritage domain.

Historical documents are a fascinating source of knowledge, preserving information over centuries. But they can also play a role in contemporary systems and tools when transformed into a semantic knowledge base. Although we developed

THE AUTHORS

René Witte is an assistant professor at Concordia University in Montréal, where he established the Semantic Software Lab. His research focuses on semantic computing, including the development of foundations in natural language processing, software engineering, semantic desktops, knowledge management, and ontologies, as well as applications in areas such as building engineering, language engineering, biomedical research, information system engineering, and social science. He has a Dr.-Ing. degree in informatics from Universität Karlsruhe. Contact him at witte@semanticsoftware.info.

Ralf Krestel is a junior researcher at the L3S Research Center in Hannover, Germany. His research interests include text mining, natural language processing, and Semantic Web technologies. He has a Dipl.-Inform. degree in computer science from Universität Karlsruhe. Contact him at krestel@l3s.de.

Thomas Kappler is a Web and infrastructure developer for the UniProt Knowledge Base bioinformatics resource at the Swiss Institute of Bioinformatics in Geneva. His research interests include text mining, ontologies, and software engineering. He has a Dipl.-Inform. degree in computer science from Universität Karlsruhe. Contact him at tkappler@googlemail.com.

Peter C. Lockemann is Professor Emeritus of Informatics at Karlsruhe Institute of Technology (formerly Universität Karlsruhe). He is one of the founders of FZI, a computer science research center dedicated to technology transfer, where he is still active as a director. His research interests include distributed and cooperative information systems and their applications in software engineering, transport and energy logistics, and ecology. He has a Dr.-Ing. degree in electrical engineering from Technische Universität München and an honorary doctorate from Universität Frankfurt. Contact him at Lockemann@kit.edu.

and deployed the technologies we described in this article with the historical encyclopedia of architecture in mind, many of the ideas also apply to other kinds of historical, or even contemporary, documents. For example, lexicographers could use a tool for the development of lexical entries that directly integrates and queries relevant documents.

One important insight of our work is that targeted text analysis support, already available today, can easily be integrated into common desktop tools to support users for their task at hand. While NLP techniques are far from perfect or comprehensive, they can already deliver knowledge discovery support that goes significantly beyond the currently used approach of full-text search and information retrieval.

Acknowledgments

Praharshana Perera contributed to the automatic index generation and the Durm Lemmatizer. Qiangqiang Li contributed to the ontology population pipeline. Thomas Gitzinger contributed to the index generation.

References

1. R. Witte et al., "A Semantic Wiki Approach to Cultural Heritage Data Management," *Proc. Workshop Language Technology for Cultural Heritage Data* (LaTeCH 08), 2008, pp. 61–68.
2. S. Fujisawa, "Automatic Creation and Enhancement of Metadata for Cultural Heritage," *Bull. IEEE Tech. Committee on Digital Libraries* (TCDL), vol. 3, no. 3, 2007; http://www.ieee-tcdl.org/Bulletin/v3n3/fujisawa/fujisawa.html.
3. E.C. Mavrikas, N. Nicoloyannis, and E. Kavakli, "Cultural Heritage Information on the Semantic Web," *Engineering Knowledge in the Age of the Semantic Web*, E. Motta et al., eds., LNCS 3257, Springer, 2004, pp. 477–478.
4. J.A. Rydberg-Cox, "Cultural Heritage Language Technologies: Building an Infrastructure for Collaborative Digital Libraries in the Humanities," *Ariadne*, no. 34; http://www.ariadne.ac.uk/issue34.
5. H. Cunningham et al., "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications," *Proc. 40th Anniversary Meeting ACL* (ACL 02), Assoc. for Computational Linguistics, 2002, pp. 168–175.
6. P. Perera and R. Witte, "A Self-Learning Context-Aware Lemmatizer for German," *Proc. Human Language Technology Conf. and Conf. Empirical Methods in Natural Language Processing* (HLT/EMNLP 05), Assoc. for Computational Linguistics, 2005, pp. 636–643.
7. I. Mani, *Automatic Summarization*, John Benjamins, 2001.
8. R. Witte and S. Bergler, "Fuzzy Clustering for Topic Analysis and Summarization of Document Collections," *Proc. 20th Canadian Conf. Artificial Intelligence* (Canadian AI 07), LNAI 4509, Springer, 2007, pp. 476–488.
9. G. Antoniou and F. van Harmelen, *A Semantic Web Primer*, 2nd ed., MIT Press, 2008.
10. M. Généreux, "Cultural Heritage Digital Resources: From Extraction to Querying," *Proc. Workshop Language Technology for Cultural Heritage Data* (LaTeCH 07), 2007, pp. 41–48.
11. M. Doerr, "The CIDOC Conceptual Reference Module: An Ontological Approach to Semantic Interoperability of Metadata," *AI Magazine*, vol. 24, no. 3, 2003, pp. 75–92.
12. P. Sinclair et al., "eCHASE: Exploiting Cultural Heritage Using the Semantic Web," *Proc. 4th Int'l Semantic Web Conf.* (ISWC 05), Springer, 2005, pp. 6–10.
13. E. Marx, *Wände und Wandöffnungen, aus der Reihe: Handbuch der Architektur* [*Walls and Wall Openings, part of the Handbook on Architecture*], 2nd ed., book 1, vol. 2, part 3, 1900 (in German).

Get the building blocks you need.

Take your career to the next level in software development, systems design, and engineering with:

- Article collections from the IEEE Computer Society
- Materials from Harvard Business School Publishing
- Computer discounts
- Online courses and certifications

Our experts. Your future.

www.computer.org/buildyourcareer