Realizing thesaurus based uses cases with the PoolParty Suite

HELMUT NAGY, TASSILO PELLEGRINI, THOMAS SCHANDL, ANDREAS BLUMAUER
Semantic Web Company GmbH
Vienna, Austria
h.nagy@semantic-web.at, t.pellegrini@semanticweb.at, t.schandl@semantic-web.at, a.blumauer@semantic-web.at

CHRISTIAN MADER
Faculty of Computer Science
Universität Wien
Vienna, Austria
christian.mader@univie.ac.at

Abstract [Resum] [Resumen]

Thesauri have been an important tool in Information Retrieval for decades and still are. They have the potential to greatly improve the information management of large organizations, but they are still underused in content management systems, search engines or tagging systems. In this article we want to describe these use cases for thesauri, in which way thesauri have to be structured in order to be suite different use cases and how Semantic Web technologies in the thesaurus management system PoolParty can help to realize them.

The PoolParty suite consists of a web-based thesaurus management system, a concept extractor and a semantic search server, which is completely built on top of W3C's semantic web standards.

1 Introduction

The PoolParty Thesaurus Manager (PPTM) is a tool to create and maintain multilingual SKOS (Simple Knowledge Organization System) thesauri, aiming to be easy to use for people without a Semantic Web background or special technical skills.

The PoolParty Extractor (PPX) offers an API providing text mining algorithms based on semantic knowledge models. With the PoolParty Extractor you can analyze documents in an automated fashion, extracting meaningful phrases, named entities categories or other metadata. Different data or metadata schemas can be mapped to a SKOS thesaurus that is used as a unified semantic knowledge model.

The PoolParty Search Server and Semantic Indexer (PPS) is a search application based on semantic technologies. Semantic technologies in a search engine provide a better understanding of what users are searching for and lead to better search results than what can be achieved by conventional search engines.

Before presenting the PoolParty suite in detail we look at conceptual assumptions about the interaction between the structural specificities of a thesaurus and the quality of a thesaurus-based application output. This is motivated by the fact that hardly any literature exists that discusses thesaurus modeling requirements with respect to the following thesaurus-specific application areas: classifying, indexing, autocomplete, query expansion, recommendations and glossaries. By looking
at these application areas the authors compare the structural attributes of SKOS and discuss their functional relevance. Taking these assumptions into account can significantly support application-oriented thesaurus modeling.

2 Knowledge organization with SKOS thesauri

Thesauri can be used to support various application scenarios like autocomplete, faceted search and browsing, recommendations or glossaries. Herein thesauri usually perform the function of harmonizing terminologies, controlling vocabularies and/or support the user in browsing through a concept space (Soergel, 2002). Despite a long research tradition in thesaurus quality assurance little attention has so far been paid to the interaction between the structural specificities of a thesaurus and the quality of output with respect to differing application scenarios supported by the thesaurus. Although several initiatives exist that focus on thesaurus and metadata quality in terms of expressivity and structural soundness (Kless, 2010; Stvilia, 2007; Park, 2006), existing ISO standards like (ISO 2788, 1986; ISO 5964, 1985; ANSI/NISO Z39.19, 2005; ISO 25964-1, 2011) and basic thesaurus and organization system literature (Broughton, 2006; Gaus, 2005), these approaches do not take the envisioned application into account, thus being of limited relevance for applied thesaurus modeling.

This article takes a look at the structural specificities of thesauri and their relevance in improving the output quality of a specific application. It is based on the assumptions that the structural attributes of a thesaurus are of varying relevance for specific application scenarios and that the modeling principle of a thesaurus has a direct effect on the quality of a thesaurus-based application.

The next section gives an overview over related work in the domain of thesauri for web-based applications. This analysis starts off with general look at thesaurus quality criteria but then takes a specific look at the W3C recommendation SKOS which has been widely accepted as a reference model for thesaurus-based applications on the (Semantic) Web. After that we collect and define the basic applications scenarios that are typically supported by a thesaurus and we give an overview of the structural attributes provided by SKOS and present our take on how these attributes influence the various application scenarios. In the following section the implications for the development of thesauri for different applications scenarios are discussed.

3 Structural specificities of thesauri for SKOS-based applications

Since its first release in 2004 the W3C recommendation SKOS (Simple Knowledge Organization System) has been utilized by several Semantic Web applications as a lightweight model to support interoperability at the terminological and schematic level (See: Avesani, 2005; Kules, 2006; Sah, 2007; Abel, 2008; Davies, 2008; Tordai, 2009; Golub, 2009; Echarte, 2009). "SKOS provides a vocabulary to define the basic structure and content of semi-formal knowledge organizations such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies and other similar controlled vocabularies. Since it is designed on RDF, SKOS allows these semi-structured concepts to be published on the Web, linked to data available on the Web and also incorporated with other concept schemes." (Sacco, 2010). Its comparably low ontological (semantic) complexity makes SKOS an ideal standard to be utilized for collaborative knowledge organization purposes especially within the context of socially generated classification schemes (See: Orlandi, 2010; Waitelonis, 2010; Sah, 2010). With the Linked Data initiative gaining momentum in the past years, SKOS (Simple Knowledge Organization System) has emerged as a common 'standard' (currently a W3C recommendation) for expressing knowledge organization systems (KOS) such as thesauri or taxonomies. SKOS features a concept-oriented approach, with a concept being "An idea or notion; a unit of thought." (as defined in the SKOS definition itself) that can be represented with an URI. A term-oriented approach proposed e.g., in ISO2788 and ISO5964 and other older thesaurus standards on the other hand treats lexical entries (terms) as the most basic units. A detailed comparison of the two approaches can be found in the appendix of the SKOS Primer (Isaac, 2008). Most term-based standards (ISO 2788 1986, ISO 5964 1985) were developed in a pre-web era and are revised at the moment in the upcoming new ISO standard (ISO 25964-1) where the term-oriented approach has also been changed towards a concept based approach: "The traditional aim of a thesaurus is to guide the indexer and the searcher to choose the same term for the same concept... The concepts are represented by terms, and for each concept, one of the possible representations is selected as the preferred term..." (Sacco, 2010).
Another sign for the importance of having controlled vocabularies in web-oriented formats like SKOS is that more and more existing vocabularies are offering SKOS versions of their vocabularies beside the classic formats provided up until now. Transformations have been made for thesauri like Agrovoc (Morshed, 2010), Eurovoc (Rodríguez, 2008), GEMET (Miles, 2004) and STW Thesaurus for Economic (Neubert, 2009) but also for other types of controlled vocabularies like subject headings.

Despite the broad uptake of SKOS, research in the interaction between the SKOS modeling paradigm and the quality of the application output is comparably scarce.

Wang et al. (2009) have conducted an experiment on the precision and relevance of automatic artwork recommendations with respect to the underlying semantic properties. They found that resources hierarchically related using SKOS broader and narrower properties leads to the largest number of recommendations in their described use-case. And recently Kless and Milton (2010) have developed a measurement construct to evaluate the intrinsic quality of thesauri mainly based on the framework for information quality developed by Stvilia et al. (2007) and the measurements constructs defined by Soergel (1994).

We will concentrate on thesauri as the type of controlled vocabulary that offers the highest level of expressivity with a focus on a concept-oriented thesaurus model. In the following we will try to show that different applications scenarios demand different structural specificities of a thesaurus.

4 Thesaurus-based application scenarios. An overview

"Today's thesauri are mostly electronic tools, having moved on from the paper-based era when thesaurus standards were first developed. They are built and maintained with the support of software and need to integrate with other software, such as search engines and content management systems. [...] Whereas in the past thesauri were designed for information professionals trained in indexing and searching, today there is a demand for vocabularies that untrained users will find to be intuitive, and for vocabularies that enable inferencing by machines." (ISO 25964-1, 2011). This introduction to the new ISO standard already states that scope and use of thesauri has changed since the shift from a pre-web to the web era. Broughton (2006) states that the main application scenarios for thesauri are indexing, providing metadata, search (query formulation and expansion) and browsing and navigation. Soergel (2002) defines use cases for thesauri in the context of digital libraries. According to his notion they support learning and assimilating information, assist researchers and practitioners with problem clarification, support information retrieval (support searching, meaningful information display, indexing, facilitate combination/access of multiple databases) and support document processing after retrieval. For Semantic Web applications especially the information retrieval and document processing aspects on top of controlled vocabularies are of great importance. With reference to the use cases proposed by Soergel (2002) and Broughton (2006) we provide a short description of the practical applicability of thesauri for the following application scenarios.

Filtering / Classifying

From an end-user perspective, the ability to browse classifications is useful to get an impression about the level of detail and type of the data held in an information system. Thus it enables a user to search the system even if no search terms should or can be formulated. This use case has been identified as "browsing the classification structure" in Soergel (2002). Also Broughton (2006) states that "the thesaurus is often used as an aid to navigation or browsing through the systematic display". These are examples using the "classification structure" from a user's perspective but this structure can also be used to classify documents automatically or provide a filter structure (facets) for narrowing down search results.

Indexing

Broughton (2006) states: "As it was first developed, the thesaurus was an indexing tool for large technical document collections." Indexing is maybe the most common application scenario for thesauri. A definition is given e.g., in NISO/ANSI Z39.19: "Indexing is the process of assigning preferred terms or headings to describe the concepts and other metadata associated with a content object." (ANSI/NISO Z39.19, 2005).

Autocompletion
Autocompletion supports the structured and context-sensitive formulation of a user's query string by mapping parts of the query string to descriptors or multiple natural language expressions within a knowledge base (Cafarella, 2011). A thesaurus can support an application providing a controlled vocabulary of terms suggested for input e.g., in a search field or as input in a form.

**Query Formulation / Expansion**

Moderated search provides knowledge-based support for end-users when vertically exploring a domain and/or when exercising federated search. This application has proven especially useful when combined with free-text search to enable structured browsing and formulating complex queries. Broughton (2006) distinguishes between query formulation and expansion, where the first means that additional search terms are provided to the user from the thesaurus and can be added to the search in the frontend while the second means that the search is enriched by the structure of the thesaurus automatically by the search engine without user interaction.

**Recommendation**

When browsing (or searching) an information system, recommended items help to broaden the user's view on the contained data. Often search terms are badly formulated or the existing structure doesn't fit the browsing needs of the user. A thesaurus can provide those recommendations via the knowledge model that is built around its concepts using synonyms and relations to recommend content or expand queries. Burke (2000) provides examples and experiments with knowledge-based recommendation systems. We could not find any direct relations to the use of thesauri for recommendation but we think that this application scenario is implicitly covered by (or at least an extension to) the scenarios mentioned before.

**Glossary**

Glossaries support the users of an information system in interpretation of the contained data. They can be a starting point to access/learn about a domain and also a reference point where a domain or the concepts of a domain are defined. Soergel (2002) defines "Support learning and assimilating information" and "Support meaningful information display" as functions of a thesaurus and we regard glossaries as the right tool to fulfill these functions.

**5 Differing structural attributes for application-specific thesauri**

The structure of a thesaurus influences the quality of the application output. With reference to the work of Klees and Milton (2010, p. 315), who defined general (intrinsic) quality criteria for thesauri, we discuss the relevance of structural SKOS elements for the application scenarios defined above. Table 1 shows a selection of structural elements related to SKOS.

<table>
<thead>
<tr>
<th>Thesaurus Components</th>
<th>Definition</th>
<th>Corresponding SKOS Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concepts</td>
<td>A concept indicates &quot;a unit of thought, an idea or a notion about a thing&quot;. Within SKOS a concept is an abstract entity (class) that exists independently of its labels or signifiers.</td>
<td>skos:Concept</td>
</tr>
<tr>
<td></td>
<td>Within SKOS labels signify/identify a concept with a natural language expression. A concept can be denoted by several labels (giving expression to synonyms), while one label can signify several concepts (giving expression to homonyms).</td>
<td>skos:prefLabel, skos:altLabel, skos:hiddenLabel</td>
</tr>
<tr>
<td>Equivalence relations</td>
<td>Using SKOS, synonyms can be expressed by linking a preferred Label (prefLabel) and an alternative Label (altLabel) or a hidden Label (hiddenLabel) to the same concept.</td>
<td>skos:prefLabel -&gt; skos:altLabel, skos:prefLabel -&gt; skos:hiddenLabel</td>
</tr>
<tr>
<td>Hierarchical relations</td>
<td>Within the same concept scheme, hierarchical</td>
<td>skos:broaden,</td>
</tr>
</tbody>
</table>
relations between concepts can be defined in SKOS by using the broader and narrower properties. Concepts in different concept schemes should be put into hierarchical relations by using the broadMatch and narrowMatch attributes.

**Associative relations**
These are relations between two concepts that are "related" to one another, without stating some kind of generalization. SKOS serves this purpose by defining the related and relatedMatch properties.

**Homonyms**
Since every "unit of thougth" is expressed as concept in SKOS, homonyms are just identical labels (i.e., they have the same string value) linked to the according concept.

**Polyhierarchies**
SKOS doesn't constrain hierarchy definitions. Every concept can be linked with an arbitrary number of broader/narrower concepts.

**Hierarchical depth**
The depth of a SKOS thesaurus can be expressed by the number of broader or narrower transitive steps originating from (or leading to) a skos:Concept that has been attributed to be a top-concept (skos:hasTopConcept, skos:topConceptOf).

**Documentation Elements**

**Definitions**
For clarification on a concept's meaning, SKOS provides the definition, scopeNote and example properties.

**Notes**
For general documentation purposes, SKOS defines the notes properties (editorial, change and history Note).

<table>
<thead>
<tr>
<th>Classifying / Filtering</th>
<th>Indexing</th>
<th>Autocompletion</th>
<th>Query Formulation / Expansion</th>
<th>Recommendation</th>
<th>Glossary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concepts</td>
<td>Quantity restricted by the scope of application</td>
<td>Quantity restricted by the scope of domain</td>
<td>Quantity restricted by the scope of domain</td>
<td>Quantity restricted by the scope of domain</td>
<td>Quantity restricted by the scope of domain</td>
</tr>
<tr>
<td>Labels</td>
<td>Quantity restricted by the scope of domain</td>
<td>Quantity restricted by the scope of domain</td>
<td>Quantity restricted by the scope of domain</td>
<td>Quantity restricted by the scope of domain</td>
<td>Quantity restricted by the scope of domain</td>
</tr>
<tr>
<td>Equivalence Relations</td>
<td>alt/hidden relevant</td>
<td>Especially, alt and hidden relevant</td>
<td>Especially, alt and hidden relevant</td>
<td>Especially, alt and hidden relevant</td>
<td>Especially, alt relevant</td>
</tr>
</tbody>
</table>

There are several other structural elements available in SKOS that are not taken in account in this paper. For an exhaustive coverage see the SKOS reference documentation (Miles, 2008) and the SKOS Primer (Isaac, 2008) for a complete coverage.

In the following we try to show that different applications areas stress different structural elements within a thesaurus.

Table 2 gives an overview of the different applications areas with respect to the requirements of the structural attributes created for the application types.
If the thesaurus structure provides necessary information for algorithmic processes, the importance of hierarchical and associative relations varies not just according to the application area, but also to the methodology applied to serve a specific application.

In the following we will go into more detail on the structural requirements defined for the different application scenarios.

6 Discussing structural aspects

Filtering / Classification

A thesaurus can be used to filter, browse or classify content by categories. As learning curves for complex classifications are steep, a static hierarchy with a defined scope (limited number of concepts) is preferable compared to a dynamic one. Hence the quantity of valid concepts and labels is restricted by the application. Equivalence relations are relevant for categorization as they increase the semantic consistency of a thesaurus, while polyhierarchies and homonyms should be avoided as they increase complexity. The hierarchical depth is restricted by the application. Depending on the completeness of the vocabulary, additional information to the subjects might be presented as part of a glossary (see below). Associative relations, definitions and notes are not relevant for classification purposes.

Indexing

A thesaurus can improve standard indexing functionalities for documents (statistical or linguistic) by providing domain knowledge for the extraction resulting in better indexing results. The higher the
domain specificity of a thesaurus, the better the indexing results will be. Hence the number of concepts and labels within a thesaurus is restricted by the scope of the domain. Equivalence relations are highly relevant for indexing documents as they increase the lexical explorativity of a document corpus, while the relevance of hierarchical and associative relations is not relevant for indexing purposes as they mainly play a role for retrieval of indexed content objects which is covered in our case in the recommendation scenario (see below). Indexing will go hand in hand with statistical and linguistic approaches for extracting terms. This can also support a semi-automatic thesaurus maintenance approach providing new terms by determining frequently extracted terms not found in the thesaurus and suggesting them as new concepts.

**Autocompletion**

A thesaurus can support autocomplete functionalities, the syntactic normalization of free text input by providing recommendations on top of a string analysis from the input field. Autocomplete supports the user not just in choosing existing terms from a predefined knowledge base (e.g., a thesaurus) but also helps the user to get an overview over the various contexts in which a term claims semantic validity. While the quantity of relevant concepts and labels is restricted by the scope of the domain, equivalence relations are one of the core elements within autocomplete functionalities, as they help the user to drill an arbitrary search term down to a corresponding concept. In contrast hierarchical and associative relations are of minor importance for autocomplete functionalities as information about the hierarchical depth of a thesaurus usually does not provide additional information for the construction of the search term. On the other hand information about polyhierarchies and homonyms are of major importance as they help the user to define the context in which the chosen concept demands validity.

**Query Formulation / Expansion**

A thesaurus as a search tool supports query formulation and query expansion. Query terms can be widened, narrowed or translated based on the terminological pool of the thesaurus and the corresponding semantic relations. In a moderated search alternative labels (equivalence relations) and related concepts (associative relations) are used to expand the search query. While equivalence relations are well suited to define the lexical entry point into a knowledge model, associative relations help to broaden the context, in which a search query demands validity. Hierarchical relations may also be used to show alternative search terms within a given context (path dependence) but are generally of minor importance for the query construction. For better navigation, results can be sorted according to their classification or filtered according to defined facets as a result of a previous classification (see above).

**Recommendation**

A thesaurus can provide recommendations that could improve retrieval of indexed content, autocomplete suggestions or query formulation/expansion (see above) by using the domain knowledge built in the thesaurus via relations. All relation types are relevant for providing recommendation but especially associative relations and hierarchical relations play an important role because they could be used to suggest alternative search queries or help to retrieve content that is not directly related to the search terms but related to the subject of the search (e.g., using broader or sibling terms in a hierarchy) or related to the scope of the search (e.g., using related terms).

**Glossary**

Glossaries can be beneficial for the user in various ways. Since the aim to completely describe the concepts of a domain all structural elements defined are relevant. A Glossary should provide a consistent and complete overview of a domain and by that could serve as a knowledge base or agreed reference of terminology for that domain. This implies also the need to clarify the meaning of concepts defined in a thesaurus by means of providing definitions, examples and scope notes. In this context, a thesaurus-based glossary can be seen as a source of metadata that can be, for example, used to provide context-sensitive help in information systems.

**7 Realizing these use cases with PoolParty**

With the PoolParty Thesaurus Manager you create controlled vocabularies and thesauri based on W3C standards. In its core PoolParty uses RDF to represent SKOS and other vocabularies like Dublin Core or FOAF, therefore an RDF triple store is used as its technological basis. Utilizing
semantic web technologies like RDF and especially SKOS allow thesauri to be represented in a standardized manner. While OWL would offer greater possibilities in creating knowledge models, it is deemed too complex for the average information worker.

Compared to other systems which still rely on relational databases PoolParty is ready to consume and to publish Linked Data out-of-the-box. Besides the possibility to publish any PoolParty based thesaurus via a Linked Data front-end, the system offers a SPARQL endpoint to execute queries over each thesaurus project. This technology can be used to integrate a thesaurus with other platforms (wikis, CMS, etc.) or search engines.

Collaborative thesaurus management

Thesauri in the age of the web most often should be engineered and maintained in a collaborative manner. PoolParty is fully web-based; administrators need only a web browser to do all typical CRUD operations like adding new concepts or relations. Its intuitive point and click interface enables working on concepts via drag and drop or autocompletion of concept labels.

PoolParty by default also publishes an HTML Wiki version of its thesauri, which provides an alternative way to browse and edit concepts, so that more people can be involved in the thesaurus development process. Through this feature anyone can get ready access to a thesaurus, and optionally also edit, add or delete labels of concepts. Search and autocomplete functions are available here as well. The Wiki's HTML source is also enriched with RDFa, thereby exposing all RDF metadata associated with a concept as linked data which can be picked up by RDF search engines and crawlers.

Linking concepts between different controlled vocabularies is another flexible way to build thesauri in decentralized structures. Based on the Linked Data principles thesauri can be maintained at different places but still can be connected to each other indicating that several concepts are similar or even identical to each other.

Technologies

PoolParty is written in Java and uses the SAIL API3, whereby it can be utilized with various triple stores, which allows for flexibility in terms of performance and scalability.

Thesaurus management itself (viewing, creating and editing SKOS concepts and their relationships) can be done in an AJAX Frontend based on Yahoo User Interface (YUI). Editing of labels can alternatively be done in a Wiki style HTML frontend.

For key-phrase extraction from documents PoolParty uses the PoolParty Extractor (PPX), which makes use of the SKOS thesauri to create an extraction model. The analyzed documents are locally stored and indexed in Solr along with extracted concepts and related concepts.

Thesaurus management and linked (open and closed) data

The rise of Linked Data indicated by the enormous growth of the Linked Open Data cloud is an important argument for many organizations to publish their own data at least partially as Linked Open Data. PoolParty's Linked Data frontend provides an easy-to-manage way to do exactly this by as it also offers options to customize the own publishing process. Since PoolParty is not only a system serving government organizations but also enterprises with metadata management solutions, PoolParty's Linked Data mechanisms can be used as a data integration technology also behind the corporate firewalls.

PoolParty not only publishes its thesauri as Linked Open Data (additionally to a SPARQL endpoint), but it also consumes LOD in order to expand thesauri with information from LOD sources. Concepts in the thesaurus can be linked to e.g. DBpedia via the DBpedia lookup service, which takes the label of a concept and returns possible matching candidates. The user can select the DBpedia resource that matches the concept from his thesaurus, thereby creating a SKOS mapping relation between the concept URI in PoolParty and the DBpedia URI. The same approach can be used to link to other SKOS thesauri available as Linked Data.

Other triples can also be retrieved from the target data source, e.g. the DBpedia abstract can become a SKOS: definition and geographical coordinates can be imported and be used to display the location of a concept on the map, where appropriate. The DBpedia category information may also be used to retrieve additional concepts of that category as siblings of the concept in focus, in order to populate the thesaurus.
To generate seed-thesauri for a certain domain the PoolParty team has developed a method to extract such structures from DBpedia automatically.

**PoolParty enterprise vocabulary and metadata management**

PoolParty is an enterprise ready system, which offers high reliability, usability, performance and mechanisms like failover which guarantees smooth workflows and protection from loss of data. Typical enterprise systems like Linux or Windows servers are supported. A constantly ongoing quality assurance process around the product including high quality documentation accompanies the overall development of PoolParty. Enterprise Vocabulary and Metadata Management is fully supported and open standards guarantee a high investment security. The integration of PoolParty thesauri with enterprise systems can be realized on top of standard APIs.

PoolParty supports the import of thesauri in SKOS (in serializations including RDF/XML, N-Triples or Turtle) or Zthes format.

**Text mining and semantic search**

PoolParty offers a variety of options to ease thesaurus management by means of text mining as well as solutions to make semantic search solutions possible. PoolParty can analyze different text formats like HTML, PDF or Word and can detect significant terms within a document either based on existing thesauri or to serve as a new candidate term to further expand a thesaurus. With PoolParty Thesaurus Management document repositories can be indexed and searched in a semantic way out-of-the-box.

The PoolParty product family consists of two other components which together with thesaurus management are the basis for enterprise semantic search solutions.

**Vertical search solutions: PoolParty product family**

PoolParty product family consists of three components: PoolParty Thesaurus Management (PPTM), PoolParty Extractor (PPX) and PoolParty Semantic Search (PPSS). Combined these elements form the basis for true semantic search and vertical search solutions. PoolParty can index unstructured, semi-structured and structured information and can integrate different sources on top of a semantic thesaurus.

PoolParty Semantic Search is shipped with a full blown Search API which can be used for integration with into existing enterprise platforms. The API supports categorized auto-complete, faceted search, full-text search and search assistants which are based on thesauri representing the background knowledge of the domain expert. PPSS can handle millions of documents and is very fast and ready for vertical search applications also in large companies. PPSS can also be used for the development of search assistants typically used within web shops or help desk and call center applications.

**8 Conclusion**

In this article we outlined conceptual assumptions on the structural requirements for various thesaurus-based applications. Our analysis indicates that some application types allow to create a single thesaurus to support different scenarios (e.g., Autocomplete and Query Formulation / Expansion), while other applications demand different thesauri or a defined subset of a thesaurus to support certain functions (e.g., Filtering / Classifying and Indexing). Another result that can be derived from the matrix is that the different application scenarios imply different complexity (e.g., Autocomplete vs. Glossary), hence differing in terms of effort and costs required for developing a vocabulary in sufficient quality. So two main aspects have to be taken into account when developing a thesaurus:

- What application scenarios should be supported?
- What structural elements are needed to support these scenarios?

By taking these aspects into account knowledge engineers can effectively plan the required functionalities of a thesaurus thus improving the efficiency of the thesaurus-based engineering effort.

The PoolParty product family offers a wide variety of options to create such thesauri according to
W3C standards and Linked Data best practices. The big three topics are: Semantic Search, Thesaurus Management and Linked Data. PoolParty uses its core Semantic Web technologies which are built around open standards and state-of-the-art technologies. Professional metadata management is the key for efficient information management in large organizations and on the web. PoolParty combines Semantic Web, text mining and collaborative knowledge engineering to make applications smarter.

Bibliography


