Querying standardized EHRs by a Search Ontology XML extension (SOX)

Stefan Kropf[1*], Alexandr Uciteli[1*], Peter Krücken[2], Kerstin Denecke[3], Heinrich Herre[1]

[1] Institute for Medical Informatics, Statistics and Epidemiology (IMISE), Leipzig University
[2] Institute of Pathology, Leipzig University
[3] Institute for Medical Informatics, Bern University of Applied Sciences
[*] Contributed equally

ABSTRACT

Motivation: The previously developed Search Ontology (SO) allows domain experts to formally specify domain concepts, search terms associated to a domain, and rules describing domain concepts. So far, Lucene search queries can be generated from information contained in the SO and can be used for querying literature data bases or PubMed. However, this is still insufficient, since these queries are not well suited for querying XML documents because they are not following their structure.
However, in the medical domain, many information items are coded in XML. Thus, querying structured XML documents is crucial for retrieving similar cases or for identifying potential study participants. For example, information items of patients with a similar tumor classification documented in a certain section of the respective pathology report need to be retrieved. This requires a precise definition of queries. In this paper, we introduce a concept for the generation of such queries using a Search Ontology XML extension to enable semantic searches on structured data.

Results: For a gain of precision, the paragraph of a document need to be specified, in which a specific information item expressed in a query is expected to appear. The Search Ontology XML Extension (SOX) connects search terms to certain sections in XML documents. The extension consists of a class which represents the XML structure and a relation between search terms and this XML structure. This enables an automatic generation of XPath expressions, which makes an efficient and precise search of structured pathology reports in XML databases possible. The combination of standardized Electronic Health Records with an ontology based query method promises a gain of precision, a high degree of interoperability and long term durability of both, XML documents and queries on XML documents.

*** Contact:** skropf@imise.uni-leipzig.de auciteli@imise.uni-leipzig.de

1 INTRODUCTION

Since untagged information in health information systems (HISs) is common, information access supported by automatic methods is difficult. It is still an open question how to accelerate the access to information captured in these systems or in Electronic Health Records (EHRs). On the one hand, content must be structured by automatic recognition processes. On the other hand, the structured data has to be queried in a structured way.

This paper will focus on the query side, by introducing a new solution of semantic meaningful queries on structured XML documents, defined by the Search Ontology (SO) XML Extension. The SO (Uciteli et al., 2014) has been developed to support full text search on unstructured documents. It allows an user to formally specify domain concepts, search terms associated to the domain, and rules describing domain concepts. In this way, it simplifies the definition of search rules. The SO can be used for information retrieval in any domain by extending it by the corresponding domain ontology.

In this work, we introduce an extension of the SO that enables the definition of queries on structured XML documents. Assuming that we have structured and standardized XML documents, then we can query certain parts of the XML document by XPath expressions. The development of such XPaths is time consuming for domain experts, but also for computer scientists. We suggest to use ontologies to support domain experts in modelling XML queries.

Out of the ontology based query models, XPaths can be generated automatically, which in turn can be applied to document corpora on XML database systems for searching similar cases or for the identification of potential study participants. Even though the approach is inherent independent from the underlying XML structure, we will demonstrate the approach on an example of querying standardized Electronic Health Records (EHRs) in the pathology domain.

To address the problem of creating structured queries for retrieving documents, previous work considered the unification of different XML structures on the conceptual level, on the one hand by the introduction of new query languages, e.g. CXPath (Camillo et al., 2003) or XSEarch (Cohen et al., 2004), or on the other hand by introducing conceptual ontologies (Cruz et al., 2004; Erdmann et al., 1999). In contrast to this unification approaches, the SOX approach, introduced in this paper, is strongly bound to the used XML structure. Indeed, this strong binding on a structure is only meaningful when standardized XML based EHRs are used.

2 METHODS

2.1 Overview

Figure 1 gives an overview on the basic approach presented in this paper.

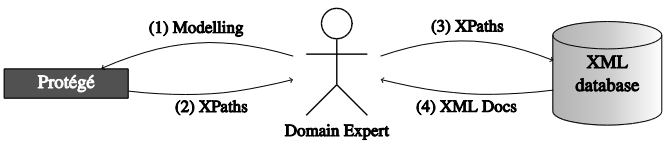

Fig. 1. (1) The domain expert models the queries by the usage of SOX in Protégé. (2) Using an extended version of the OntoQueryBuilder Plugin, Protége generates XPath expressions out of the ontology. (3) The expert applies the generated XPath expressions to an XML database, that (4) returns the relevant documents.

A domain expert is in the middle of the query formulation and retrieval process. He uses Protégé, the ontology editor of the Stanford University (Musen 2015), for modeling a query using the Search Ontology (section 2.2) and SOX (section 3.1) as shown on the left side in figure 1. By an adaption of the OntoQueryBuilder Plugin it will be possible to generate XPaths expressions. Additionally, the agent interacts with the XML database as shown on the right hand side of figure 1, After recognizing section boundaries, unstructured documents are stored on an XML database (section 2.3). Using the XPaths (sections 2.4, 3.2), the domain expert can retrieve relevant XML documents.

2.2 Search Ontology

The formulation of structured queries can be very time consuming, especially in safety-relevant domains like post market surveillance. A concept can be described in different ways, on the one hand by synonyms, on the other hand by complex phrases, which in turn consist of multiple terms. Because of that we distinguish `Simple_Terms` from `Composite_Terms`.

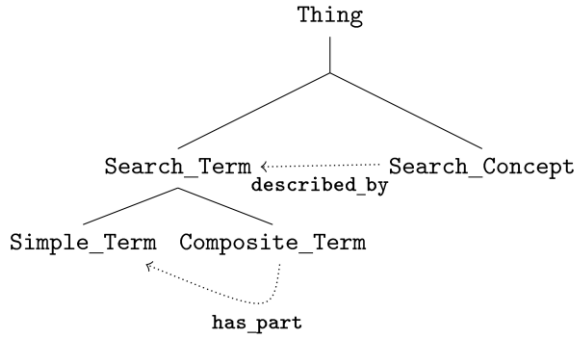

Fig. 2. Overview Search Ontology

`Composite_Terms`. are made up of `Simple_Terms`, related by the Object Property `has_part` and are constrained by the additional Data Property `max_distance`, which defines the word distance between two `Simple_Terms`, where `max_distance=0` represents that one word immediately follows after another word. Writing variations, synonyms of abbreviations of the `Simple_Terms` can be handled by the assignment of multiple labels to the concrete individual of a `Simple_Term`.

For instance, the complication of a medical device (e.g. occluder) is a reusable `Search_Concept`, which can be `described_by` several `Search_Terms`.

To such descriptions belongs among other things adjective phrases like *incomplete closure*. Instead of the adjective, other terms with the same semantic meaning could be used; the noun could be replaced by any term which represents the meaning of *closure*. Out of this definition, a query can be generated, which is in this example the disjunction of all combinations of adjectives and nouns (cf. second disjunction in Listing 1).

Listing 1. Lucene Query for an occlusion device complication; the expression was generated by the plugin OntoQueryBuilder.

```
("occlusion device" OR occluder) AND (("insufficient sealing"~2 OR "insufficient closure"~2 OR "incomplete sealing"~2 OR "incomplete closure"~2 OR "inadequate sealing"~2 OR "inadequate closure"~2))
```

The latter example indicates that the formulation of a query can become a complex task; the cross-product of only 10 adjectives with 10 nouns results in 100 adjective substantive combinations. Hence, we have to manage Concepts and Terms by an appropriate ontology, especially if we want to reuse concepts or if we want to generate cross-products of certain term combinations.

The SO is used in practice in the OntoVigilance project (OntoVigilance Homepage 2016), where semantic searches have to be managed within post market surveillance queries of medical devices. In brief, domain experts can manage their domain search ontology (DSO). By the usage of the developed plugin OntoQueryBuilder a Lucene query can be generated.

2.3 Standardized XML-based EHRs

In this paper, we will concentrate on the special domain of pathology, where a lot of semi-structured information occurs in terms of pathology reports. We consider this information semi-structured, because the pathologists structure their information by headers and keywords, but these structure is usually not technically implemented. In fact, pathology reports are based on certain section patterns and section-introducing keywords, like *material*, *macroscopy* or *microscopy*. We verified manually that documents originated from the Institute of Pathology of Leipzig, the sections introducing keywords like *Material*, *Makroskopie* or *Mikroskopie* were constantly used for section tagging. Therefore, the reports can be structured in sections by section boundary detection, which is not the main focus of this article. Consequently, legacy data can be transformed into a structured format. Suitable standards for long term persistence are EN 14822 a.k.a. HL7 RIM (EN 14822, 2006) and EN 13606 a.k.a. openEHR (EN 13606, 2012). Both standards are representable in XML, EN 14822 by the usage of CDA (Dolin et al., 2001) and EN 13606 by the usage of openEHR modeling tools (Kropf et al., 2015), which results in a standardized XML schema based on the openEHR XML schemas (Beale 2015).

In this work, we will use pathology reports, mapped to standardized EHRs by the usage of the openEHR archetypes openEHR-EHR-OBSERVATION.lab_test-histopathology.v1 for structuring pathology data and openEHR-EHR-CLUSTER.tnm_staging_7th.v1 for structuring the TNM classification (Sobin et al., 2011) data. Both latter archetypes are available at the Clinical Knowledge Manager (CKM) of openEHR [http://www.openehr.org/ckm/]. Consider the following snippet of an XML based pathology EHR (cf. Listing 2) where we demonstrate the challenges of querying the content. They are mainly due to the linguistic variability of natural language.

Listing 2. Simplified XML based pathology EHR snippet, containing a macroscopy and a TNM classification part. The snippet was cut to the necessary elements, which we want to address in the query in this paper, marked by a grey background: the macroscopy section (part of openEHR-EHR-OBSERVATION.lab_test-histopathology.v1) and the primary tumor classification (part of openEHR-EHR-CLUSTER.tnm_staging_7th.v1). The doubling of the value tag is a result of the EN 13606 reference model, in practice the two value tags have different namespace declarations.

```
<Pathology […]
       <Macroscopic_findings>
       <name>
         <value>Makroskopisch</value>
       </name>
       <Overall_macroscopic_description>
         <name>
               <value>Makroskopisch</value>
         </name>
         <value>
               <value>Ein ff. einfach fadenmarkiertes
keilförmiges Hautexzidat von 0,8 x 0,6 x 0,3 cm.[…]</value>
         </value>
       </Overall_macroscopic_description>
       </Macroscopic_findings>
       <Tumour__-__TNM_Cancer_staging_7th_Edition>
       <name>
         <value>Tumour - TNM Cancer staging 7th Edi-
tion</value>
       </name>
       <Primary_tumour__openBrkt_T_closeBrkt_>
         <name>
               <value>Primärtumor T</value>
         </name>
         <value>
               <value>pT2</value>
         </value>
       </Primary_tumour__openBrkt_T_closeBrkt_>
       […]
       </Tumour__-__TNM_Cancer_staging_7th_Edition>
 […]
 </Pathology>
```

The TNM structured classification string in the XML snippet is "pT2". The sentence which introduces the overall macroscopy description contains a noun Hautexzidat (HE) (en: excised skin material), marked by a bold font. Due to linguistic variability, this noun can vary, i.e. synonyms or abbreviations such as H.E. are used in practice. In front of the noun there is an underlined adjective keilförmig (en: cuneiform) for specifying the shape. Again, semantic and linguistic variants of the term exist (e.g. rundlich (en: roundish)). Furthermore, the order of the adjectives in the phrase could change: the order "Ein ff. rundliches fadenmarkiertes H.E." was found and is also valid.

2.4 XPath Queries

When EHRs are stored in structured XML, another query language is more suitable than classical free text retrieval methods such as Lucene (McCandless et al., 2010) or SOLR (Trey et al., 2014). XPath expressions are following the structure of the EHRs and are a W3C standardized method for addressing parts in XML documents (XML Path Language (XPath), 2015). An example XPath Query is shown in Listing 3 for querying T2 and phrases of HE from EHR documents similar to those in Listing 2.

Listing 3. Required XPath expressions for a search of EHRs which contains T2 as primary tumor classification in the first part and defined phrases of HE in the second part.

```
/Pathology/Tumour__-__TNM_Cancer_staging_7th_Edition/
Primary_tumour__openBrkt_T_closeBrkt_/
value[contains(value,'T2')]
```

```
/Pathology/Macroscopic_findings/Overall_macroscopic_description/
value[matches(value,'keilförmig(\w)* ([\w]*\s){0,2}Hautexzidat')]
or
/Pathology/Macroscopic_findings/Overall_macroscopic_description/
value[matches(value,'rundlich(\w)* ([\w]*\s){0,2}H.E.)]
or
/Pathology/Macroscopic_findings/Overall_macroscopic_description/
value[matches(value,'keilförmig(\w)* ([\w]*\s){0,2}H.E.)]
or
/Pathology/Macroscopic_findings/Overall_macroscopic_description/
value[matches(value,'rundlich(\w)* ([\w]*\s){0,2} Hautexzidat'')]
```

The first part of Listing 3 matches documents where the tumor class is T2. In the second part, each disjunction represents one adjective noun phrase. It consists of an adjective and a regular expression that reflects possible declension variations followed by the noun phrase representing Hautexzidat (HE). The expression `([\w]*\s){0,2}` implies that between the adjective and the noun a maximum of two words are allowed to match the pattern.

3 RESULTS

3.1 Extension of the Search Ontology

The output of the SO are Lucene queries, but they do not follow the structure of XML documents, thus, they are not applicable to XML documents. However, the SO delivers already a reusable framework which only has to be extended for enabling structured queries in XML. By extending the SO with the SOX, queries are automatically producible out of the ontology, which can be executed on XML documents. For this purpose, two elements were added to the SO, on the top level of the ontology the class `XML_Structure` and the Object Property `in`.

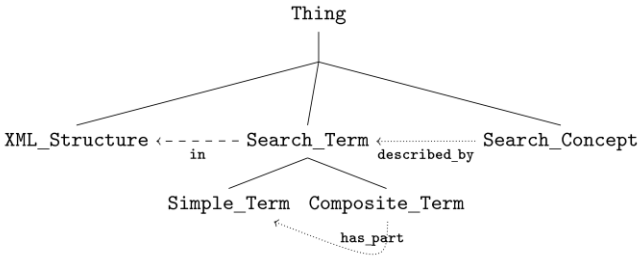

Fig. 3. The Search Ontology XML Extension introduces the top level class XML_Structure and the relation in (dashed arrow).

Figure 3 shows that `Seach_Concepts` are `described_by` `Search_Terms`, which belong to certain parts in the `XML_Structure`; in more detail, `Search_Terms` are linked to XML parts by the `in` relation. The subclass structure of

XML_Structure represents the XML document structure. Namespaces and tag names of the XML document are defined by XML_Structure class labels. Figure 5 illustrates the SO for the described use case, where the documents follow the structure of Listing 2 and XPaths of Listing 3 have to be generated as output.

The modelling of the SO (illustrated in fig. 5) has to be done manually by the domain expert. For querying HEs with different kinds of shapes the Search_Concept HE_Shapes was defined, described_by HE_Phrase, which consists of two Simple_Terms (HE_Form and HE_Term). In the example of this paper, individuals of HE_Form can be adjectives like *rundlich* or *keilförmig*; HE_Term has only one individual, the different writing variations (Hautexzidat, H.E.) can be handled by multiple label assignments. Figure 5 illustrates also the usage of the in relation for the specification of the position inside the XML document, for instance is the term T2_Term expected in the associated section Primary_Tumor. In a similar way, HE_Shapes are bound to the XML_Structure. The following figure 4 shows the class definition of HE_Shapes in Protégé.

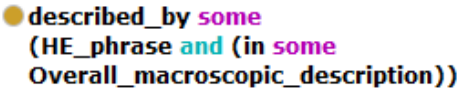

Fig. 4. Class definition of HE_Shapes

In summary, out of the DSO XML extension, it is possible to generate automatically the required XPath expressions.

3.2 Automatic XPath generation

For each Search_Concept one XPath can be generated automatically. This can be done by a scheduled adaption of the Lucene export plugin, the OntoQueryBuilder, which is already developed as part of the OntoVigilance (OntoVigilance Homepage 2016) project. The first part of Listing 3 can be generated out of the Search_Concept T2, in which description the term T2_Term is bound to the appropriate search location by the in relation. The second part of Listing 3 is producible out of the Search_Concept HE_Shapes, which is described_by the Composite_Term HE_Phrase and is expected in the Overall_macroscopic_description. This Composite_Term yields to a disjunction expression of all combinations of the labels of the HE_Form individuals with the labels of HE_Term individuals, which is in essence a kind of a cross product.

The generated XPaths can be used for structured queries and for the integration in other XML techniques (XSLT or XQuery).

4 DISCUSSION

With an example on querying structured EHRs, we introduced an extension of the Search Ontology to support querying structured XML documents. The SOX approach can simplify the managing of a big pool of XPath expressions in one overarching DSO in practice.

4.1 Standardized queries on standardized EHRs

Indeed, SPARQL queries on OWL based patient data would be more powerful than XPath expressions on XML, but a comprehensive and long term persistence storage of pathology data within semantic web technologies is only partially solved and still an open research question. Therefore, until there is no standardized domain ontology available, queries on standardized XML will be more stable and long term durable. To put it in brief, the first requirement is a layer of standardized EHRs and tools which work at this layer, like the introduced SOX. After that step a more powerful ontology layer is demandable.

We used the EN 13606 standardized XML in this work. However, another option would be EN 14822 or even any other proprietary XML format. When the community comes to an agreement, which EHR standard will be used in German Health Information Systems in future, not only the EHR would be interoperable, the usage of a standardized query language implies that queries could be interoperable too. Presupposed standardized EHRs would be used, it is imaginable that queries on such EHRs can be interoperable and therefore used in different hospitals. When openEHR is used, the depending SOX or the resulting queries could be stored in a repository and they could be linked to the belonging archetypes. Consequently, the

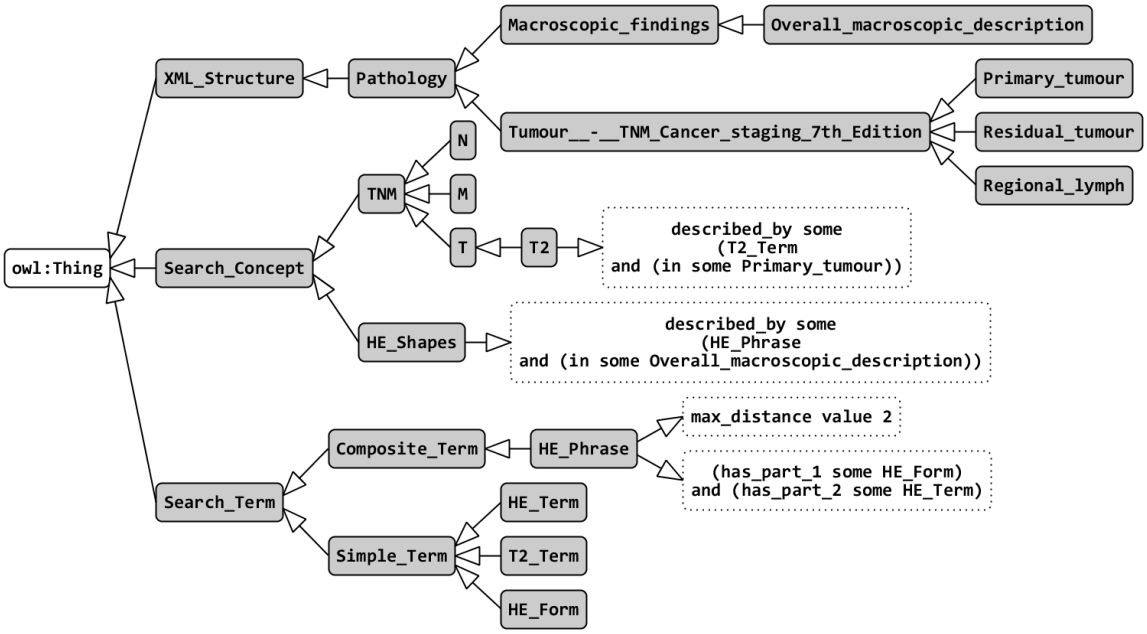

Fig. 5. Search Ontology for querying the concept primary tumor T2 and the concept HE_Shapes (manually defined)

query can be reused like the archetype itself, this would save development time and lead to quality intensification.

4.2 Recognition methods vs. querying methods

Research in Natural Language Processing (NLP) delivers methods (Hahn et al., 2002; Friedman et al., 1999) for the recognition of clinical information in medical documents. Nevertheless, it is unclear when a reliable NLP system will automatically recognize and annotate free text pathology reports or any other free textual clinical documents in daily practice. It is important to think about practical solutions on the recognition side, but also on the query side. The SOX delivers a practical solution on the query side by the connection of Search Terms to parts in XML documents.

Listing 3 already respects declension forms of adjectives which have a stable word stem. But for the development of a minimal SO, NLP methods like stemming are necessary too. Because of that, we have to think about the integration of such methods into ontological contemplations about querying structured information in the near future.

4.3 Future ontological work

XML elements are more than symbolic structures, they have to be considered in detail, last but not least they should have be bound to a top level ontology like the General Formal Ontology (GFO) (Herre 2010). In the SOX, the XML document structure was realized by `is_a` relations, because we wanted to model queries in tree structures in standard Protégé. Of course `has_part` relations would be semantically better. For this reason, we plan to develop a proper plugin for the modeling of `has_part` relations in tree structures in Protégé. In addition, an automatic conversion of XML documents into a SOX `XML_Structure` tree is demandable; this would accelerate the query development in Protégé. X2OWL can generate an OWL ontology from an XML data source (Ghawi et al., 2009) and is a good starting point.

5 CONCLUSION

When EHRs are persisted in standardized XML, it is possible to query them in a structured way. The introduced Search Ontology XML extension connects search terms to certain parts in XML documents and enables an ontology based definition of semantic searches. Out of this, XPath expressions can be generated for querying XML database systems. Our solution supports the reuse oriented specification of complex and powerful XPath expressions without deep syntactic knowledge about XPath. The approach is open for additional extensions; parts of the ontology can be reused and adapted easily for other use cases.

ACKNOWLEDGEMENT

Thanks to Claire Chalopin, Wolf Müller, Katrin Schierle, Lars Voitel, Christian Wittekind for their support, to the reviewers of ODLS for their constructive feedback, especially Dagmar Waltemath, and the organizers, among which we mention Frank Loebe and Daniel Schober, for arranging ODLS. This work was conducted using the Protégé resource, which is supported by grant GM10331601 from the National Institute of General Medical Sciences of the United States National Institutes of Health.

REFERENCES

Beale, T. (2015) openEHR reference-models XSDs release 1.0.2 https://github.com/openEHR/reference-models/tree/master/models/openEHR/Release-1.0.2/XSD [cited 2016-08-30]

Camillo, S. D., Carlos A. H. and dos Santos Mello, R. (2003) Querying heterogeneous XML sources through a conceptual schema. International Conference on Conceptual Modeling, pp. 186-199.

Cohen, S. et al. (2003) XSEarch: A semantic search engine for XML. Proceedings of the 29th international conference on Very large data bases. Volume 29, pp. 45-56, VLDB Endowment.

Cruz, I. R., Xiao, H. and Hsu, F. (2004) An ontology-based framework for XML semantic integration. Database Engineering and Applications Symposium. IDEAS'04. Proceedings. International, pp. 217-226, IEEE.

Dolin, R. H. et al. (2001) The HL7 Clinical Document Architecture. Journal of the American Medical Informatics Association 8(6), pp. 552-569.

EN 13606. (2012) Health informatics - Electronic health record communication [Norm].

EN 14822. (2006) Health informatics - General purpose information components [Norm].

Erdmann, M. and Studer, R. (1999) Ontologies as conceptual models for XML documents. Proceedings of the 12th International Workshop on Knowledge Acquisition, Modelling and Mangement (KAW'99), Banff, Canada.

Friedman, C. and Hripcsak, G. (1999) Natural language processing and its future in medicine. Academic Medicine 74(8), pp. 890-5.

Ghawi, R. and Cullot N. (2009) Building Ontologies from XML Data Sources. DEXA Workshops, pp. 480-4.

Grainger, T, Potter, T. and Seeley Y. (2014) Solr in action. Manning.

Hahn, U., Romacker M. and Schulz S. (2002) MEDSYNDIKATE—a natural language system for the extraction of medical information from findings reports. International journal of medical informatics 67(1), pp. 63-74.

Herre, H. (2010) General Formal Ontology (GFO): A foundational ontology for conceptual modelling. Theory and applications of ontology: computer applications. Springer Netherlands, pp. 297-345.

Kropf, S, Chalopin, C. and Denecke, K. (2015) Template and Model Driven Development of Standardized Electronic Health Records. Studies in health technology and informatics 216, pp. 30-4.

McCandless, M., Hatcher, E. and Gospodnetic, Otis. (2010) Lucene in Action: Covers Apache Lucene 3.0. Manning Publications Co.

Musen, M.A. The Protégé project: A look back and a look forward. AI Matters. Association of Computing Machinery Specific Interest Group in Artificial Intelligence, 1(4), 2015, pp. 4-12

OntoVigilance Homepage (2016) http://www.ontovigilance.org/ [cited 2016-06-16]

Sobin, L. H., Gospodarowicz, M. K. and Wittekind C., eds. (2011) TNM classification of malignant tumours. John Wiley & Sons.

Uciteli, Alexandr et al. (2014) Search Ontology, a new approach towards Semantic Search. GI-Jahrestagung, pp. 667-672.

XML Path Language (XPath) (2015) Version 1.0. W3C Recommendation. https://www.w3.org/TR/xpath/ [cited 2016-06-09]