

Extracting RDF knowledge from OMDoc

– work fulfilling the Guided Research requirement –

student: *Andrei Ioniță*
a.ionita@jacobs-university.de

supervisors:

Prof. Dr. Michael Kohlhase
m.kohlhase@jacobs-university.de

Christoph Lange
ch.lange@jacobs-university.de

Contents

1	Declaration	2
2	Acknowledgements	2
3	Executive Summary	3
4	Foundations	3
4.1	Overview	3
4.2	Semantic Web	4
4.3	Technology Enumeration	4
5	Use Case: Euclidean Geometry Atlas	5
5.1	Geometry research	6
5.2	Writing OMDoc documents	7
6	OMDoc Ontology Extension	7
6.1	Previous OMDoc Ontology status	8
6.2	Proposals for ontology extensions	8
7	Extracting RDF	10
7.1	Previous existing extractions	10
7.2	Stylesheet description	10

8	Integration to OMBase	11
8.1	JENA	11
8.2	Querying with SPARQL	12
9	Conclusion	12

1 Declaration

I hereby declare that the research submitted - ranging from this paper to periodical reports to programming language code - is my own work, completed under the supervision of Prof. Dr. Michael Kohlhase and Christoph Lange at Jacobs University Bremen. Any other contribution is specifically acknowledged.

Andrei Ioniță
 Bremen, May 7, 2008

2 Acknowledgements

I would like to briefly describe the work environment in which I have developed my research question, designed the resolving approach, started implementing it and finally drawing the conclusions.

I owe Prof. Dr. Michael Kohlhase the initial guidance that brought me to the field of Knowledge Representation and OMDoc particularly. Up to the end of my undergraduate studies he always found time to have discussions about my work and had the understanding of looking forward and make compromises between the goals and my achievements.

To Christoph Lange, soon to be Ph.D, I cannot feel more in debt, from the help and assistance that he offered. He has always been very responsive, managed to give me the necessary advice to carry on, even on the times in which my weekly contributions lacked that certain sharpness.

I also want to thank Florian Rabe, as the discussions with him helped me understand the logic behind OMDoc. The development of OMBase – although been kept independent from this guided research – had corresponded with this project that I only got to discover from the new OMDoc documents that he wrote.

On this note, I want to thank the entire KWARC research group for their understanding. As I have been part of this warm academic family, all

the projects were like brothers, needed care and the ensurance of interaction. Without Normen Müller's, Christine Müller's, Immanuel Normann's, Vyacheslav Zhodulev's, Milena Makaveeva's and Ștefan Anca's assistance and understanding I would not have gotten this far both knowledge- and communication-wise.

3 Executive Summary

Every time when reading a scientific text we arrange the information in our minds so that we first understand the topmost level of conceptualization. The neatly ornamented details (low level constructions or figures of speech) pale in our eyes as we are seeking the keywords that describe the subject matter at a first read. After spending some time on the paper we usually write down our ideas in the form of diagrams or as bullet points by summarizing the main characteristics. Following the example of the human mind that is able to identify relevant pieces of information according to a context, the intention in this work is to try and realize a system that achieves this purpose inside a mathematical framework. More precisely, this would yield both further machine-processable information, and a meaningful diagram in which one can distinguish the respective concepts.

For example, we would like to identify definitions and their corresponding examples, together with all the assertions (theorems, propositions, etc.) that relate to it. Obtaining such a graph with the concepts as nodes and their relations as edges would provide us with a first-level understanding of how the respective theory is structured. However, the resulting data will not be solely for human understanding, but would be furthermore processable for searching or can be used to integrate it into other theories. The diagrams that will be ultimately displayed may be included in presentation slides and stand as an e-learning tool.

4 Foundations

4.1 Overview

This introductory section will serve as a reference for the technologies used throughout this work. The descriptions will not be comprehensive, but emphasize aspects that are relevant in the following chapters. Although no extensive knowledge is required for their understanding, familiarity with fundamental principles of Computer Science is recommended.

4.2 Semantic Web

The World Wide Web is an immense pool of information, mostly available in HTML format. This structure does not make the machines to understand the data more than at the syntactic level, while ignoring what actually is meant by specific sequences of characters. Humans on the other hand, are interested in gaining access to a source of information that is appropriate to their wishes. Search engines nowadays are essentially limited to a syntactic search, due to the rudimentary layering of the pages across the Web. It is sought an enriching of these sources or at least the new ones, such that relationships between concepts can be specified and thus create an order in the information collection. Such a process is called *annotation* and constitutes an extensional feature of the Web, i.e. that of adding semantics. Of course, the task of adding information is by no means attributed to the machine side, but it is contributed by the users.

4.3 Technology Enumeration

Uniform Resource Identifier (URI)

URIs are identification codes for resources on the web. This is not limited to retrievable HTML pages – which is in fact the subset known as URLs (Uniform Resource Locators), as any abstract object or relationship can be specified by such a URI. The critical property that URIs ensure is that no two resources have the same identification, i.e. injectivity satisfied. Also, URI (more precisely the emerging IRIs (Internationalized Resource Identifiers)) assure a world-wide covering of information across the worlds' alphabets in the creation of the identification code, i.e. surjectivity accomplished. Thus URIs (or IRIs) establish a bijective correspondence between the pool of identifiable objects and their Web codes.

Extensible Markup Language (XML)

XML is the standard language for structuring information on the web. Benefiting from its extensional nature, it can be adapted to any application domain, and as a consequence is supported by a multitude of tools.

Open Mathematical Documents (OMDoc)

OMDoc[4] is an open markup language for mathematical documents. is currently an increasingly-used medium for storing both formal (formulæ) and informal (scientific language) content. The latest version released at the time of writing is 1.2. However, there have been advances concretized in conference papers that formalize the specification for the upcoming 1.8 version. OMDoc relies on OpenMath for describing mathematical objects,

by representing their meaning through basic XML elements. The data in OpenMath is grouped in Content Dictionaries that contain symbols that share a common field application.

Open Mathematical Database (OMBase)

OMBase[3] is a database that serves the management of OMDoc documents. It can be queried to return not only entire documents but also fragments of documents, i.e. theories, imports and symbols (wrapped in OpenMath elements to express a morphism, if necessary) . Also the database interface supports PUT and DELETE - HTTP methods that are implemented [2] (Representational State Transfer). Such a design aims at defining uniform resource names, through which the corresponding contents are accessible using URLs.

Document Ontology

A document ontology is a formal description of the structures of documents independently from their syntax. They contain a classification of concepts and relationships that build the semantics of a document format. Usually such a ontology is layered, starting with the low-level constructs and continuing with more refined elements. For each class, a parent is specified, a domain and a range of action, and also a distinction is made relative to the other construct on the respective level. Usually, an RDF ontology is written as a RDF schema vocabulary or as a OWL web ontology vocabulary.

Resource Description Framework (RDF)

Once the resources are identified via URIs, the RDF technology[7] is used to connect them in a form that adds semantics. The standard is to create sentences of triples, i.e. *subject, predicate and object, in this order*. To be able to perform the extraction, a vocabulary of types is specified using an RDF ontology description language, known as RDF Schema[1] (RDFS).

5 Use Case: Euclidean Geometry Atlas

Geometry is a mathematical field where numerous results with an individual character have been committed. This has conferred a special interest to the field, due to its requirement for ingenuity and its freedom in proving attempts. From the existence of multiple proofs for single results (which make a distinct feature from any other mathematical field) we notice a multivariety that leads to a nonlinear structure across its scope (as opposed to e.g., group theory).

Another point of attraction consists of classifying the proofs that enter geometry from different other fields. For example we often encounter a concept

that is expressible also in algebra, or in trigonometry. These also hold insightful representations, e.g., a volume of a tridimensional body corresponds to a determinat of a matrix in linear algebra.

Here are some extractions from a such a graph of concepts and relations:

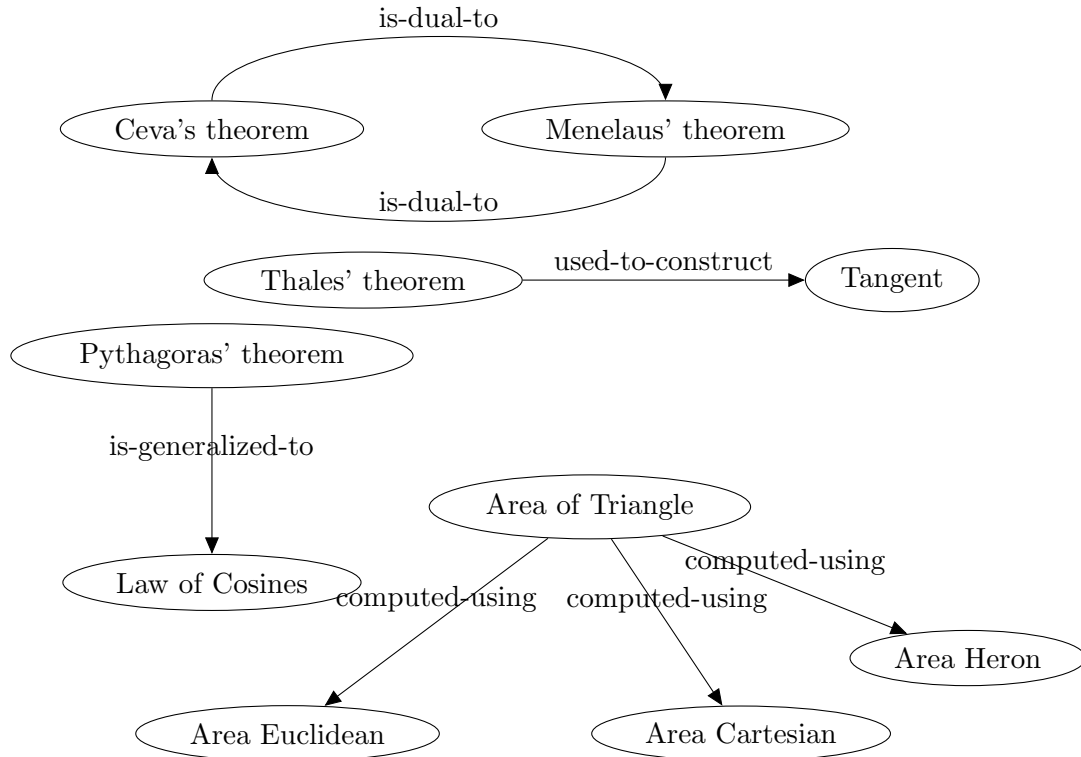


Figure 1: Different relationships between geometrical concepts

5.1 Geometry research

Beginning with Euclid's Elements as a basis for the whole geometrical theory, I intend to construct a succinct – yet modern – formalization, by highlighting the structural aspects of the field. The plan is to formalize the relationships between geometrical concepts by selecting consistent elements and properties that appear throughout the field. Also, by adding to external references to other theories, my work will not remain remote and would gain a meaning inside the whole mathematic scope.

I gathered information the following information¹.

- axioms (Euclid's 5 postulates, including the Parallel Postulate)
- main theorems (Pythagoras, Thales, Menelaus, Ceva, Ptolemy)
- concepts (congruence, similarity, types of triangles)
- shapes and their properties (triangles, types, lines inside; quadrilaterals, polygons)

5.2 Writing OMDoc documents

After gathering the necessary information, I wrote it in OMDoc format, using OpenMath elements. My main point of reference was the OMDoc 1.2 specification[4], but I was sure to modify the elements that changed their form in the recent Mathematical Theories paper[6].

6 OMDoc Ontology Extension

The OMDoc Ontology is currently developed in N3 notation, being structured as follows:

- a *base* file (containing base concepts and relations)
- a *statements* file, containing constitutive and non-constitutive statements (for more details, please consult the chapter about the OMDoc element specification in [4])
- a *proofs* file, designed by writing the necessary elements for proving statements types as in the statements file

Writing in N3 syntax is much more user-friendly than using the XML representation for creating the OWL documents. The latter is the format that is used for establishing the vocabulary that corresponds to the RDF extractions.

¹Wikipedia Geometry Portal <http://en.wikipedia.org/wiki/Portal:Geometry>

6.1 Previous OMDoc Ontology status

Christoph Lange has developed the main part of the OMDoc ontology. The relevant part for this research relates to the statements module. There, essential building blocks like the classes **odo:Statement**, **odo:Theory**, **odo:NonconstitutiveStatement** (elements that do not contain low-level OpenMath objects, e.g. Assertion), **odo:ConstitutiveStatement** (elements that directly contain OpenMath objects, e.g. Symbol, Definition, etc.). Also, properties as **odo:imports** and **odo:specifies** are defined.

6.2 Proposals for ontology extensions

A contribution that this project aimed for is enriching the ontology, in case there are some new requirements found in the application domain of study.

The examples in which geometrical theorems and their relations were modeled led to the appearance of new relationships that can be distinctively characterized. In addition, I consider that there are grounds to believe that these new elements can be consistently found across mathematical theories. A rather exhaustive list was brought to my attention by Christoph Lange, <http://www.ida.liu.se/labs/nlplab/ijcai-ws-03/papers/Tsovaltzi.pdf>. I will discuss some of them, for which simple examples can also be provided.

The **duality** relation between concepts can be intuitively seen as a one-to-one equivalence of symbols in different frameworks (e.g. in geometry: lines correspond to points). Abstractly defined, duality between concept A and B supposes that A is defined in the form $\phi \Rightarrow \psi$ and B is defined in the form $\psi \Rightarrow \phi$ with ϕ and ψ formulae. From this definition, we can speculate that duality can sometimes be particularized in a *converse* relation. It is left to future contributors to assess the line of separation between the concepts of duality and converse, and refine the definitions appropriately.

In OpenMath, **dual** could be the symbol introduced for this purpose. Here, we consider it belonging to the "proof_meta" content dictionary:

```
<assertion >
  <FMP>
    <OMOBJ>
      <OMS name="dual" cd="proof_meta"/>
        <!-- formula that acts as premise in the duality relation -->
        <!-- formula that acts as conclusion in the duality relation -->
      </OMOBJ>
    </FMP>
  </assertion >
```

In the OMDoc ontology, I have added the class **Duality**:

```
odo:Duality
  a owl:Class ;
  rdfs:subClassOf odo:NonconstitutiveStatement ;
```

Also, the property **isDualTo** is introduced, to alternatively put already defined concepts in relation to one another, instead of just making the point of **Duality** and specifying it as such.

```
odo:isDualTo
  a owl:ObjectProperty ;
  rdfs:domain odo:Assertion ;
  rdfs:range odo:Assertion ;
  rdfs:subPropertyOf odo:inRelationWith ;
```

Another concept that appeared in the Geometry use case was **Generalization**. Intuitively, a concept that contains more specifications than another, with regard to a precise criteria is less general than its counterpart. More formally, concept A is a generalization of concept B if and only if B is defined as $A \text{ and } \phi$, for some formula ϕ .

In the OMDoc ontology, the concept has been introduced as the class **Generalization**

```
odo:Generalization
  a owl:Class ;
  rdfs:subClassOf odo:NonconstitutiveStatement ;
```

Also, the relation **specializes** was introduced, having an emphasis on the concept (Assertion, more precisely) that contains more restrictions.

```
odo:specializes
  a owl:ObjectProperty ;
  rdfs:domain odo:Assertion ;
  rdfs:range odo:Assertion ;
```

Following the example from geometry given, the **Converse** class also was defined in the ontology. The relation can be formulated as: The converse of an assertion $\phi \Rightarrow \psi$ is $\psi \Rightarrow \phi$, for some non-empty formulae ϕ and ψ .

```
odo:Converse
  a owl:Class ;
  rdfs:subClassOf odo:Assertion ;
```

together with the **isConverseTo** relation

```
odo:isConverseOf
  a owl:ObjectProperty ;
  rdfs:domain odo:Assertion ;
  rdfs:range odo:Assertion ;
  rdfs:subProperty odo:inRelationWith;
```

7 Extracting RDF

7.1 Previous existing extractions

A basis for the RDF extraction constituted Christoph Lange's correspondent extraction used for the SWiM project. This was written in XSLT 2.0 and based on the Saxon XML model. However, Christoph only implemented a OpenMath element extraction, so my contribution would be complementary. Furthermore in its specification, the RXR representation was used for the RDF knowledge base. This differs from the standard RDF/XML notation in that it clearly specifies the subject, predicate and object in such named element tags. Despite being redundant, as many predicates are repeating and could be specified in a shorter notation (see RDF/XML for this purpose), RXR is very readable, and makes verification and processing also simpler. Another reference for my OMDoc extraction is Octavian Druță's RDF extraction, that was written however in XSLT 1.0.

7.2 Stylesheet description

The high-level file **extract-omdoc-rdf.xsl** contains templates that match the main elements in OMDoc, i.e. theories, imports, symbols, definitions, assertions, etc. (the list is for sure not exhaustive). For the relationships in which these elements are involved, the second low-level stylesheet is called, **extract-rdf.xsl**. Here there are templates for specifying the *type* of elements:

```
<xsl:template name="process-type">
  <xsl:param name="type"/>
  <xsl:param name="path-suffix"/>
  <xsl:sequence select="triple-uri(concat(base-uri(),
    concat('/', $path-suffix)), '&rdf:type', $type)"/>
</xsl:template>
```

The *path-suffix* specifies the path that the element points to, when in a XML database (see OMBase below). *triple-uri* is a function that returns the triple RDF statement.

For putting two elements in relation, *add-uri-property* or *add-literal-property* is used. Such an example in an OMDoc document would be a definition that has a reference for a symbol (which only defines the type).

```
<xsl:template name="add-uri-property">
  <xsl:param name="path-suffix"/>
  <xsl:param name="property"/>
  <xsl:param name="object" select="@xlink:href"/>
  <xsl:sequence select="if ($object) then
    triple-uri(concat(concat(base-uri(),' '),$path-suffix),
      $property, concat(concat(base-uri(),' '),$object)) else ()"/>
</xsl:template>
```

8 Integration to OMBase

Extracting RDF information is useful when all the collected information from documents in a field is centralized. As the documents that are subject to knowledge investigation would already be located in a database, a helpful feature would be to have an extraction mechanism to respond to user-friendly requests. We can imagine that one could use OMBase as a statistic bed for mathematical theories. Although it is early to affirm, the RDF information would not be stored permanently, and the user would have to initiate a knowledge collection from a selected pool of documents.

8.1 JENA

The Java Semantic Web Framework (JENA) provides a programmable environment for RDF under Java. Using its packages, requests to OMBase regarding knowledge can be directed to a processable module that would model the information appropriately. A simple manipulation of data can be seen in the following code snippet.

```
Model euclideanModel = ModelFactory.createDefaultModel();
Resource isoscelesProp =
  euclideanModel.createResource("\.../triangle/isosceles\");
isoscelesProp.addProperty(type, "\...#Symbol\");
```

8.2 Querying with SPARQL

The SPARQL Protocol and RDF Query Language (SPARQL)[5] is a query language that applies to RDF resources. JENA supports SPARQL and as queries would become available in OMBase, the Java framework will represent the processing center for user queries. An example of SPARQL code is listed in the following block:

```
PREFIX  odo: <http://www.mathweb.org/omdoc>
SELECT ?url
FROM    <triangle.rdf>
WHERE   {
    ?theory odo:homeTheory <../shapes> .
    ?theory odo:imports <../euclidean_basis> .
}
```

PREFIX defines the namespace used throughout the query. SELECT defines what would the return result would be (in this case an URL). FROM specifies the RDF knowledge base in the form of a file, while WHERE sets conditions on the seeked element.

9 Conclusion

The idea behind extending the OMDoc database with capabilities of displaying an organized set of information has come from the incentive of sharing knowledge. Actually, the information is present already, encapsulated in a mathematical, XML-based format. We can profit from the fact that this format has semantic enhancements, and we do not need any other human-aided addition to transport the data in a different display, at least not a content-oriented intervention. The Resource Description Framework via the Java Semnatic Framework is a comprehensive approach that captures the essential information and wraps it in a readily deliverable form. Finally, with SPARQL, one can achieve the – state-of-the-art – goal of the web, specifically of searching. Speaking of Semantic Web, this task is easily and cleanly obtainable, as long as the ontology of the language is consistent and comprehensive. Probably the latter issue is the most challenging. Nevertheless, producing such a system is a step forward in knowledge accesability, and in gaining mathematical insight.

References

- [1] D. Brickley and R. V. Guha. RDF vocabulary description language 1.0: RDF Schema. W3C Recommendation, World Wide Web Consortium, Feb. 2004. Available at <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>.
- [2] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
- [3] A. Ioniță. Developing a rest interface to a database for omdoc. <http://kwarc.info/projects/ombase/pubs/report2.pdf>, 2007.
- [4] M. Kohlhase. OMDOC – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer Verlag, 2006.
- [5] E. Prud’hommeaux and A. Seaborne. SPARQL query language for RDF. W3C Recommendation, World Wide Web Consortium, Jan. 2008. Available at <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>.
- [6] F. Rabe and M. Kohlhase. A web-scalable module system for mathematical theories. Manuscript, to be submitted to the Journal of Symbolic Computation, see <https://svn.kwarc.info/repos/kwarc/rabe/Papers/omdoc-spec/paper.pdf>, 2008.
- [7] Resource Description Framework (RDF). <http://www.w3.org/RDF/>, 2004.