The Thought Behind the Symbol: About the Automatic Interpretation and Representation of UDC Numbers†

Attila Piros

University of Debrecen, Hungary, Egyetem tér 1., H-4032, H-4031, Debrecen, Hungary,
<attila.y.piros@gmail.com>

Attila Piros is a PhD candidate at the Doctoral School of Mathematics and Computer Science at the University of Debrecen, Hungary. He holds a master's degree in teaching mathematics and library and information sciences from the University of Debrecen. His doctoral research deals with analysis of the UDC notations and their utilisation in computerized information retrieval. He has authored a number of research papers and has reported on his research at conferences, including the International UDC Seminar 2015. Attila has been working as a software developer and currently works on software solutions in market research.

Piros, Attila. 2017. "The Thought Behind the Symbol: About the Automatic Interpretation and Representation of UDC Numbers." *Knowledge Organization* 44(6): 416-424. 31 references.

Abstract: Analytico-synthetic and faceted classifications, such as Universal Decimal Classification (UDC) provide facilities to express pre-coordinated subject statements using syntactic relations. In this case, the relevance, in the process of UDC-based information retrieval, can be determined by extracting the meaning of the classmarks as precisely as is possible. The central question here is how the identification mentioned above can be supported by automatic means and an analysis of the structure of complex classmarks appears to be an obvious requirement. Many bibliographic sources contain complex UDC classmarks which are stored as simple text strings and on which it is very difficult to perform any meaningful information discovery. The paper presents results from a phase of ongoing research focused on developing a new platform-independent, machine-processable data format capable of representing the whole syntactic structure of the composite UDC numbers to support their further automatic processing. An algorithm that can produce the representation of the numbers in such a format directly from their designations has also been developed and implemented. The research also includes implementing conversion methods to provide outputs that can be employed by other software directly and, as a service, make them available for other software. The paper provides an overview of the solutions developed and implemented since 2015 and outlines future research plans.

Received: 3 August 2017; Accepted 3 August 2017

Keywords: UDC, data format, UDC numbers, XML, order, axuiliaries

† Above all I would like to acknowledge the support provided by my family during my work. I would also like to express my appreciation to Dr. István Boda for his valuable and constructive suggestions during my research work and to Dr. Aida Slavic for her heartfelt support and for providing information invaluable for my work. I extend my thanks to Daniel Benediktsson and Jonathan Wild who have been of great help in writing this paper.

1.0 Introduction

In their renowned work, Charles Kay Ogden and Ivor Armstrong Richards (1946, 10-12) modelled the relationship between linguistic symbols and the objects they represent as a triangle. Its points represent the object or reference, the thought regarding it and the designation (symbol) of the thought. In a bibliographic metadata schema in which subject descriptions are expressed using classification such as Universal Decimal Classification (UDC), one can consider "object" to mean the object of the description: a document, text, image, object of art or any other item that is indexed. The thought is the resume of the main subjects of the object as the indexer can express it using single statements. And finally, the symbol is the designation of the above sentences by simple or composite codes of the classification.

Analytico-synthetic classifications, such as UDC, express complex subject descriptions by building classmarks. In this case the effective decisions regarding the relevance or the identification of the object require extracting the thought from the symbol as precisely as possible. In information retrieval, a central question is how the identification mentioned above can be supported by automatic means. Among other things, analysing the structure of the codes is an obvious requirement of this.

The current research has focused on developing a new platform-independent, machine-processable data format that contains the whole syntactic structure of composite UDC numbers, to support their further automatic processing. An algorithm that can produce the representation of the numbers in such a format directly from their designations has also been developed and implemented. The research also involves implementing conversion methods to provide outputs that can be employed by other software directly and, as a service, make them available for other software. Future research plans include exploring application areas where the developed format and algorithms may be useful and proceed by testing the solution.

2.0 UDC as an analytico-synthetic classification

Universal Decimal Classification (UDC), an analytico-synthetic classification, featuring, in addition to the rich concept hierarchy, many ways in which knowledge fields express facets, also provides possibilities to describe complex subjects and the agglomeration of basic subjects (cf. Broughton 2015, 33-40; FID 1981). The concepts that are created by specifying basic subjects with facets are called compound subjects (Ranganathan 1967, 84). Facet combinations creating compound subject expressions are most prominent in UDC when it comes to use of common auxiliary facets denoting attributes of place, time, form, language, materials, persons, properties, relations, etc.. Since facets are attributes that typically also occur within a class, in UDC they are often expressed with special auxiliaries (Gnoli 2011, 19).

Complex subjects are concepts that are created by using phase relations, which join two or more subjects on the basis of some relation between them (Ranganathan 1967, 85). In UDC, complex subjects can be built by using the symbols denoting simple relation ":" order-fixing "::" and subgrouping "[]." If necessary, the type of the relationship may be specified by using common auxiliaries of phase relations, e.g., "-042," Table 1k.

The expression "agglomerated basic subject" was introduced in Neelameghan's "basic subject" (1973, 202-6) for concepts that are built by (quoted in Binwal 1988, 254-5) "collecting together of entities into larger masses without cohesion among the components." In UDC, coordination "+" and consecutive extension "/" can be used to express the two types of agglomeration identified by Neelameghan.

In an analytico-synthetic classification, the information carried by the syntactical relations needs to be revalued in the context of classification's use in online retrieval. It would be beneficial to have the exact identification of the basic classes of the facets, taking the phases of the relations into account, differentiating between compound, agglomerated and complex subjects or even taking the order of the elements of a compound or complex subject into account, etc. (cf. Robinson 2003). Such data would improve the effectiveness of the system, especially in raising the level of the precision during subject browsing and information retrieval. Although the hierarchical nature of the schedule provides excellent conditions for inclusive searching (cf. Soergel 1994), it also requires the precise identification of the elements and relations in the composite concepts.

3.0 The interpretation of UDC numbers

The complex nature of the UDC and inconsistencies that can be observed in the structure due to its hundred-year-old development history complicate the automatization of the classification. The UDC revisions of the past three decades have aimed at refactoring the schedules in a fully faceted fashion, based on a consistently applied facet analytical principle similar to the one applied in faceted classifications such as BC2 (McIlwaine 1998; McIlwaine and Mitchell 2006; McIlwaine and Williamson 2008; Gnoli 2009; Broughton 2010). It is assumed that a more systematic approach in structuring the schedule would result in a more consistent notational representation and would contribute to better handling of notation in an online environment.

The consensus regarding authority control as a satisfactory way to apply classifications in library OPACs and subject gateways is well-known: the 2015 UDC Seminar focused on this subject (Slavic and Cordeiro 2015). However, authority control is an expensive process that should be supported by automatic means, as far as possible. It is also a fact that tools and the means for supporting authority control are often not available, which sometimes renders computer based application of the classification almost impossible. Last, but not least, authority control may be rendered additionally complicated and difficult if one takes into account the notational syntax of faceted classification (cf. Tartaglia 2004), although the possibility to express those concepts is one of the main strengths of faceted and analytico-synthetic classifications.

An investigation into the ways of building indexes based on the elements of UDC notations and parsing them automatically started in the 1960s (see Rigby 1974). In the 1990s Gerhard Riesthuis developed algorithms and sample programs to identify the components of composite UDC numbers. He analysed and tested his algorithms on both classmarks in library catalogues and classmarks in the standard UDC scheme as held in the UDC Master Reference File (UDC MRF) database and found out that UDC classmarks can be parsed with one hundred percent accuracy if necessary data are pulled from UDC MRF. The actual objective of UDC number parsing in Riesthuis' research was enabling of translation of UDC notation to verbal expression stored in the UDC MRF database and thus enabling searching UDC using words (1997 and 1999). The results of this research are summarized in his PhD thesis (1998), which is, to date, the most comprehensive work on this subject.

More recently, similar research was conducted by Gábor Mándy in Hungary (2013). Mándy's focus was on UDC classmarks as they appear in library catalogues. He approached the subject by developing a set of algorithms that can decompose the UDC numbers as a chain of filters; every program receives the output of the previous one as input and is responsible for recognizing a specific auxiliary or operand. Finally, the chain will split the input number into its components. Recognizing the parts of complex or compound subjects is also possible if their notations contain valid tags from the UDC MRF database. The advantage of this approach is that those tags are also suitable for sorting numbers in the proper way, albeit it pre-supposes the application of MRF while building the classmarks.

As we can see, earlier research mainly investigated recognition of the elements of pre-coordinated numbers. This post-coordinated technique however can be improved further by taking the inner structure of the composite numbers into account, and the inner structure is determined not only by the numbers which are joined but the relations between them as well as their order.

3.1 Interpretation of UDC numbers preserving the meaning of pre-coordination

The research presented in this section focused on representing UDC numbers in a machine-readable and application-independent format while preserving the semantics expressed in syntactical relations of pre-coordinated classmarks. The research project involved the development of an algorithm for interpreting UDC numbers and their conversion into the aforementioned format (without the direct employment of the UDC Master Reference File) and the creation of an online service which fully implements the new algorithm. The format and the program mentioned above are available online at the following URL: http://piros.udc-interpreter.hu. The service can be used for processing UDC numbers via a user or application programming interface, while the XML schema definition can be downloaded and used freely, under a Creative Commons Attribution-NonCommercialNoDerivatives 4.0 International License (https://creativecommons.org/licenses/by-nc-nd/4.0).

3.2 XML Schema definition to describe UDC numbers

The most important requirement for the format designed for representing UDC numbers is describing the whole syntactic structure of a combined number, keeping all relevant information regarding its parts, the way they are connected to each other, their role in expressing the subject as a whole and the order of elements in the classmark when relevant. The second requirement is that it must be a standard format, which can be processed by various software applications and converted into other formats effortlessly. Taking into account the special characteristics of UDC and the abovementioned requirements, XML seemed to suit the purposes of the research. The main advantage of the chosen standard is the flexibility, the wide range of support and ease of use of the XML schema definition (XSD), which documents, facilitates processing and enables validation of the data format.

3.2.1 The underlying principles of the representation of UDC numbers

The precedence order of UDC symbols comes from conceptual definitions that govern the building of both compound and complex UDC numbers. For expressing composite subjects by relating main UDC numbers one has to use connecting symbols denoting either phase relations (combination using colon ":"), extension (span symbol slash "/") or co-ordination, addition (plus symbol "+"). Compound and complex subjects can be merged into an agglomerated subject by using addition. The extension symbol "/" can join neighbouring numbers which are descendants of the same class. These agglomerated subjects, such as the subjects enclosed by subgrouping, can be specified by facets like any table numbers.

The precedence order outlined above determines a tree for every precoordinated number in which the subjects of different types stand on different levels. For example, in the number 515.1+514:517 addition is represented on the first level, while relation is on the second. The lowest levels of the tree represent the main concepts, that may contain a main table number (interval, synthesis or subgrouping), possibly specified by special auxiliaries and zero or more common auxiliaries. Common and, sometimes, special auxiliaries may also contain further relations and auxiliaries, which must be handled under them. The leaves of the tree are always classes of the schedules or intervals built by using the extension symbol "/."

A further advantage of this approach is that the generated representation of faceted numbers will contain both the focus and the basic class of the facets without separating them (cf. Gnoli 2011). For example, in the

UDC number 27-475.5-23 ("sermons based on Holy Scripture") the relation between the basic class 27 and the focuses of the facets -475.5 and -23 will be saved in a clearly recognizable and reproducible way, independently from the order of the facets and possible other elements. Thus, the facets can be identified and retrieved without any significant noise or information loss.

3.2.2 The schema definition

Each of the trees described above can be described with XML. The elements of such XML can be defined in a schema definition that consequently defines a language to describe UDC numbers. The complex types of the XSD represent the branches and the leaves of the tree. The classes (table numbers) and intervals are complex types that contain two attributes for the opening and the (optional) closing numbers of the interval. The simple types serve for the purpose of validation, containing constraints for the table numbers. The following example illustrates an XML representation of a complicated precombined UDC number:

```
<ns:udc_concept
 xsi:schemaLocation="http://piros.udc-
   interpreter.hu/#xsd udc.xsd"
   xmlns:ns="http://piros.udc-interpreter.hu/#xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchem
   a-instance"   udc_edition="2017"
 notation="[515.1+514:517]-32(02.025.2)=161.1">
 <ns:description xml:lang="EN">
  Topology and analytical geometry (book in Russian,
   with illustrations)
 </ns:description>
 <ns:main_concept>
  <ns:main_table_subgrouping>
   <ns:main_table_addition>
    <ns:main_concept order="1">
     <ns:main_table_number number1="515.1"/>
    </ns:main_concept>
    <ns:main_table_relation order="2">
     <ns:main_concept order="1">
      <ns:main_table_number number1="514"/>
     </ns:main_concept>
     <ns:main_concept order="2">
      <ns:main_table_number number1="517"/>
     </ns:main_concept>
    </ns:main_table_relation>
   </ns:main_table_addition>
   <ns:special_auxiliary
  xsi:type="ns:special_auxiliary_hyphen" order="1">
    <ns:special_auxiliary_number
  xsi:type="ns:special_auxiliary_hyphen_number"
  number1="-32"/>
    </ns:special_auxiliary>
  </ns:main_table_subgrouping>
  <ns:common_auxiliary_independent
  xsi:type="ns:common_auxiliary_of_form" order="1">
   <ns:common_auxiliary_of_form_number number1="(02)">
    <ns:special_auxiliary
  xsi:type="ns:special_auxiliary_pointnought" order="1">
    <ns:special_auxiliary_number
  xsi:type="ns:special_auxiliary_pointnought_number" number1=".025.2"/>
    </ns:special_auxiliary>
   </ns:common_auxiliary_of_form_number>
  </ns:common_auxiliary_independent>
  <ns:common_auxiliary_independent
  xsi:type="ns:common_auxiliary_of_language" order="2">
   <ns:common_auxiliary_of_language_number
  number1="=161.1"/>
  </ns:common_auxiliary_independent>
 </ns:main_concept>
</ns:udc_concept>
```

3.3 Interpreter for parsing and XML formatting of UDC numbers

Following the design of the XML presentation format above, the next step was the development and implementation of a parser, i.e., an interpreter capable of converting UDC numbers into this format. The most important requirements for the interpreter were defined as follows: a) to respect the UDC number building rule, retaining all the information stored in a number (containing information regarding its parts and their whole syntactic context); b) to parse the numbers automatically, in a fully syntactic way as far as possible; and, c) to be available online for use by both humans and programs.

The interpreter is an automaton that recognizes the formal language determined by UDC numbers and auxiliary symbols. The inputs of the algorithm are the UDC number and the year of the UDC edition which was used to build it; the output is either an XML representation of the number or an error message which describes the problem if the number is not valid or cannot be interpreted within the given data or versioning constraints.

In spite of the proposition to construct an algorithm which is able to parse UDC numbers in a syntactic way as far as possible, there are special cases in which information regarding the exact places of their parts in the schedules determine the structures of the numbers and the access points to them; for example, recognizing synthesis or access points in 0/9 type parallel subdivisions and ranges requires being acquainted with the numbers (cf., Riesthuis 1999, 27-29).

3.3.1 The output formats

Although XML is a standard machine processable format, it cannot be expected to suit all scenarios of UDC processing and use. Thus, it was deemed logical and necessary to also provide conversion to other formats that can be employed directly, i.e., more straightforwardly, by different software applications. So, in addition to the raw XML format, the interpreter was adjusted to provide an HTML display of UDC numbers as well as to be able to provide the list of UDC number elements in other standard, machine readable formats such as JSON. JSON (JavaScript Object Serialization) is a self-describing and easy to understand, language-independent lightweight data-interchange format. While serializing data to JSON, the program takes an object hierarchy, such as UDC number and converts it to a string format for information exchange between applications which is able to express hierarchy in a simple way. The following example demonstrates a JSON-string compiled by the software to collect the parts of the given UDC number:

{"concept":"378.007.1","udc_edition":"1990",
"pref_labels":{"pref_label_1":{"pref_label":"",
"language":"EN"}},"udc_numbers":{"number_1":
{"notation":"378","filing":"3T7T8C","uri":"http://
udc data.info/025169","pref_labels":{"pref_label_1":
{"language":"EN","pref_label":"Higher education.
Universities. Academic study"}}},"number_2":
{"notation":".007.1","filing":"P0T0T7T1C","pref_
labels":{"pref_label_1":{"language":"EN","pref_
label":""}}}}}.

4.0 Recent research results

The goals and the basic principles of the research that was outlined in the previous section were also explained in more detail at the International UDC Seminar 2015 and in the last double-volume of *Extensions and Corrections to the UDC* (Piros 2015 and 2017). This section summarizes the results of the most recent research.

4.1 The Digital National Library of Portugal: a case study

At the end of 2015, it was decided to conduct a case study based on the The European Library (TEL) Open Dataset (http://www.theeuropeanlibrary.org). The Digital National Library of Portugal was chosen from the more than one hundred TEL contributor libraries, because of its medium size collection and the high number of UDC codes employed in it. After having downloaded the open data as a single RDF/XML file, the UDC numbers were retrieved from the subject triplets contained by its descriptions. After removing duplications from the list of the retrieved numbers, 13,741 unique notations were found. The final list was fed as a list of test cases to a tester application that called the interpreter service for each of them, processing the whole list in one batch. In this way, the service processed the whole collection within a few minutes.

Out of the complete set of 13,741 the program managed to convert successfully to XML a total of 13,604. The result of the investigation into the remaining 137 records revealed two bugs and five special practices to compose numbers that had not been implemented. The rest were caused by typing errors and indexing practices that did not match the exact rules of UDC. The XML validation identified further typos and irregular indexing practices. In addition to testing the performance of the service, the results of the study also provided excellent experience and feedback for further corrections and improvements to both the data format and the interpreter algorithm.

4.2 The new version of the XML Schema Definition

The first version of the XML Schema Definition (version 1.0), which was demonstrated at UDC Seminar 2015 (Piros 2015), was devised based on both printed and available online UDC editions. The UDC editions used in this research are Hungarian UDC editions published in 1990 (*OMIKK* and *OSZK-KMK* 1990) and 2005 (*OSZK KI* 2005), the *BSI UDC* printed standard edition published in 2005 (*BSI* 2005) and the *UDC Summary* (*UDCC* 2013).

The complete standard English UDC edition from the UDC Online Hub (http://www.udc-hub.com), which is available under an annual subscription licence, was included in the research at a later stage. This fact resulted in several inconsistencies that needed to be resolved to align the format to the most recent editions of the schedules. The possibility to investigate the whole of the schedules through such an easy-to-use interface and with the improved support that the portal provides for searching and browsing has rendered the interpretation of the rules, the quest for exceptional solutions and the comparison of the different versions very easy, which has sped the research. The new information taken from UDC Online English (http://www.udc-hub.com/index.php), the case studies managed and further research of the literature has provided a lot of useful experience that could have served as the base for the perfection of both the format and the interpreter.

After the corrections mentioned above were completed, a new version of the format was released. The new format is clearer, better documented and theoretically better established than the previous version. Furthermore, it includes several special and exceptional rules and practices that were not present in the previous version. In the future, further changes in the schema definition might be necessary only if the revisions of the schedules require it.

4.2.1 The main changes in the schema definition

The most important change in the new version of the schema definition is that special auxiliaries are handled in the same way as common auxiliaries, to enable reflecting even the most specific special auxiliary rules possible. In general, special auxiliaries can be described as follows:

```
<xsd:complexType name="special_auxiliary">
 <xsd:complexContent>
  <xsd:extension base="udc:special_auxiliary_root">
   <xsd:sequence>
    <xsd:element name="special_auxiliary_number"
  type="udc:special_auxiliary_number"/>
   </xsd:sequence>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

Another relevant modification in the new schema is that the citation order of the elements in the complex UDC classmarks can be stored. This includes not only the auxiliary numbers but the concepts joined by auxiliary signs as well. In the case of order-fixing the order has an important role. Otherwise it may be necessary to reproduce the original number from the representation and it might hold some syntagmatic meaning (cf. Robinson 2003).

A few constraints also had to be modified regarding some special cases. For instance, correcting the constraints for dates and ranges of time in common auxiliaries of time (Table 1g) was necessary, as is illustrated below.

```
<xsd:simpleType
    name="common_auxiliary_of_time_number_strin
    g">
  <xsd:restriction base="xsd:string">
   <xsd:pattern value="'\.\.\.'"/>
   <xsd:pattern value="'(-|\+)?[0-2]\d{0,3}'">
    <xsd:annotation>
     <xsd:documentation xml:lang="en">
     An optional -/+ sign and the (also optional) mil-
   lenium, century or decade
     </xsd:documentation>
    </xsd:annotation>
   </xsd:pattern>
   <xsd:pattern value="'(-|\+)?([0-
   2]\d{3}(\.\d{2}(\.\d{2}(\.\d{2}(\.\d{2}(\.\d{2
   })?)?)?)?)?)'">
    <xsd:annotation>
     <xsd:documentation xml:lang="en">
     Date and time. Optional minus/+ sign, the year
   and (optional) month, day, hour, minutes and sec-
   onds
     </xsd:documentation>
    </xsd:annotation>
   </xsd:pattern>
   <xsd:pattern value="'[3-
   9]((\.)?\d{1,4})(\.\d{1,4})*'"/>
  </xsd:restriction>
 </xsd:simpleType>
```

Further investigation revealed several issues in UDC number building that needed to be addressed. These included:

- Geographical place according to quadrants inside the common auxiliaries of place—classes "(161/164)."
- Spatial measurements and dimensions inside the common auxiliaries of place—classes under "(18)."
- Translations inside the common auxiliaries of language—facets under "=030.1/.9."
- Special auxiliary subdivision for language usage, dialects and variants inside the common auxiliaries of language "=...`276/`282."
- The relation inside common auxiliaries of ethnics "(=1:...)" is a commonly used solution in the Portugal Digital National Library. Since in the earlier UDC editions it was denoted by point "(=1.4/.9)," the XSD and the interpreter should handle both punctuation signs.

This and similar number building practices required refactoring the schema definition that was managed in version 2.0, which was introduced in *Extensions and Corrections to the UDC* (Piros 2017).

The most recent version (version 2.1) of the XSD contains only one theory-based modification. The 1.0 and 2.0 versions defined the main table numbers as attributes of a main concept and the auxiliaries as elements of it. This solution reflected the "the picture wall principle" (Ranganathan 1967, 425-426) rather than the way UDC approaches the common auxiliaries. The independent common auxiliaries can be used comparably to the main table numbers, in any place in the classmarks or even without a main table number (c.f. FID 1981; BSI 2005, xxvii-xxi). At the same time, when building numbers, UDC's facet formula suggests the correct order of common auxiliaries in the complex classmark. Thus, using main table numbers (intervals, synthesis or subgrouping) as elements on the same level as independent common auxiliaries better fits the rules of the classification:

```
<xsd:complexType name="main_concept">
  <xsd:sequence>
   <xsd:choice minOccurs="0" maxOccurs="1">
     <xsd:element name="main_table_number"
    type="udc:main_table_number"/>
     <xsd:element name="main_table_synthesis"
    type="udc:main_table_synthesis"/>
      <xsd:element name="main_table_subgrouping"
    type="udc:main_table_subgrouping" minOc-
    curs="1" maxOccurs="1"/>
   </xsd:choice>
   <xsd:element
    name="common_auxiliary_independent"
    type="udc:common_auxiliary_independent"
    minOccurs="0" maxOccurs="unbounded">
    </xsd:element>
  </xsd:sequence>
  <xsd:attribute name="order" type="xsd:int"
    use="optional"/>
</xsd:complexType>.
```

4.3 The evolution of the software

4.3.1 Aligning the software with XML

From the modifications of the schema definition, it obviously follows that the interpreter also needs to be refactored to support the changes and to provide its output in the new format. This implementation task took priority over other improvements.

4.3.2 Further output formats

In addition to the XML and KWOC, further machine-readable formats have been designed and implemented. Within the MARC group of formats, there are two prevalent formats created specifically to describe and exchange classification records. The MARC21 (earlier USMARC) Classification Format (Library of Congress 2000) was created mainly for managing *DDC* and L*CC* codes and is not suitable for the specific synthesizing rules of UDC. The development of the UNIMARC Classification Format started later, based on the experience of MARC21, declaratively for the purposes of UDC. However, the format has not been finished and is available in its draft form on the IFLA website (IFLA 2000) despite the proposals for its improvement (Slavic 2008). Since the UNIMARC format, especially after the proposed changes, support for UDC on an improved level implementing conversions to its published and improved versions were decided and have been implemented.

The pre-combined UDC numbers may also be represented as RDF. The triplets can be identified based on the XML and the URIs of the freely available UDC classes and auxiliaries. At the time of the preparation of this paper, the RDF schema and output are still under development. The exact output format is planned to be published together with the software component responsible for the number conversion.

4.3.3 Availability through a RESTful interface

The Representational State Transfer (REST) is an architectural style for distributed hypermedia systems, which (Fielding 2000, 105) "provides a set of architectural constraints that, when applied as a whole, emphasizes scalability of component interactions, generality of interfaces, independent deployment of components and intermediary components to reduce interaction latency." The service is currently accessed using simple HTTP calls. Clearly a RESTful architecture, if implemented, would provide a standard interface that can be accessed, interpreted and employed by other systems more easily. Thus, a part of the future development plan is to redesign the service having a more powerful architecture in mind. In general, it could be said that improving functionalities of the interpreter and its conversion methods is an important part of the current development plan.

4.4 The test set

The research project required the compilation of a test set in order to maintain the integrity of the software during the implementation phase and in order to review the UDC number building rules. The test set contains more than seven hundred test cases, grouped by their main purpose. There are tests for checking whether the different rules apply and for checking whether special cases during the application of auxiliary signs, consecutive extension, subgrouping, non-UDC notations and alphabetical specifications, etc. are handled well.

A test case contains a UDC number and XML that must be produced by the interpreter after having processed it. It can be used manually or automatically to compare the outputs of the interpretation to the expected results. The test set contains a list of UDC numbers with their identifiers and the XML files assigned to them. Although, in general, test cases need to contain valid UDC numbers, this is actually needed only if more than punctuation marks are required to determine the result of the process. Most UDC classmarks are collected from library catalogues, course books and articles and some are built specifically for the purpose of testing the interpreter.

In addition to maintaining the integrity of the software, such a test set provided useful examples of the application of the format and a good opportunity to review the rules of UDC regarding building complex numbers and to facilitate a better comprehension of their meanings. The whole test set, together with further tests for the other output formats is available online on the home page of the interpreter (http://interpreter-eto.rhcloud.com/#test).

5.0 Conclusion

Following the release of version 2.1 of the XSD and after the refactoring the service to make it available in a RESTful way (planned for the end of 2017), the first phase of the research will be completed. The future research will focus on the feasible applications of the outputs of the previous phase and on the analysis of accumulated experience.

One of the biggest advantages of storing a UDC number in an XML format is that this format, in contrast to the UDC notations themselves, is transparent and well supported by programming languages. Thus, its further automatic analysis and conversion would not require special algorithms and programming effort. This may improve the efficiency of methods applying information regarding the syntactic structure and the elements of UDC numbers, including quantitative studies (cf. Smiraglia et al. 2013), similarity measurements between composite subjects or to develop improved algorithms to search and browse complex subject descriptions inclusively.

The generated XML may also serve as the basis for the development of outputs and methods that can support implementing intelligent classification interfaces.

Generating the notation that contains proper tags to arrange the numbers in the correct filing order is already implemented. Building, for instance, a KWIC index based on these notations may render it possible to browse the composite numbers by their components, by taking their context into account. Permuting the elements by abiding by the rules of UDC is also possible, based on the representation that reveals the whole inner structure of the numbers; this may multiply the number of access points to the concepts. The final goal would be to change the notations of the numbers automatically and to display them in a form that fits the supposed target of the searching process and the cognitive status of the searcher during subject browsing.

In addition to applying the format and the program, investigating the experience accumulated while automatically representing and interpreting the concepts gives us a high chance of examining how the revisions and appropriate changes performed in recent years have helped in handling UDC notations.

References

Binwal, J. C. 1988. "Modes of Formation of Subjects and their Role in Information Retrieval." PhD diss., Karnatak University. http://shodhganga.inflibnet.ac.in/handle/10603/94558

British Standards Institution. 2005. *Systematic Tables.* Vol. 1 of *Universal Decimal Classification*, 3rd ed. London: British Standards Institution.

Broughton, Vanda. 2010. "Concepts and Terms in the Faceted Classification: The Case of UDC." *Knowledge Organization* 37: 270-9.

Broughton, Vanda. 2015. *Essential Classification*, 2nd ed. London: Facet Publishing.

Fielding, Roy Thomas. 2000. "Architectural Styles and the Design of Network-Based Software Architectures." PhD diss., University of California, Irvine.

Gnoli, Claudio. 2009. "The UDC Philosophy Revision: First Report." *Extensions and Corrections to the UDC* 31: 25-31. http://hdl.handle.net/10150/200633

Gnoli, Claudio. 2011. "Facets in UDC: A Review of Current Situation." *Extensions and Corrections to the UDC* 33: 19-36.

International Federation de Documentation. 1981. *Principles of the Universal Decimal Classification (UDC) and Rules for its Revision and Publication*, 5th ed. FID 598. The Hague: International Federation de Documentation.

International Federation of Library Associations and Institutions. 2000. "Concise UNIMARC Classification Format." http://www.ifla.org/archive/ubcim/p1996-1/concise.htm

Library of Congress. 2000. "MARC 21 Format for Classification Data (Update No. 24)." http://www.loc.gov/marc/classification/eccdhome.html

Mándy Gábor. 2013. "A posztkoordináció esélyei az ETO-ban." *Könyvtári figyelő* 59: 65-84.

McIlwaine, I. C. 1998. "The Universal Decimal Classification: Some Factors Concerning its Origins, Development and Influence." In *Historical Studies in Information Science*, ed. Trudi Bellardo Hahn and Michael Buckland. ASIS Monograph Series. Medford, NJ: Information Today, 94-106.

McIlwaine, I. C., and Mitchell, Joan S. 2006. "The New Ecumenism: Exploration of a DDC/UDC View of Religion." *Extensions and Corrections to the UDC* 28: 9-16.

McIlwaine, I. C., and Williamson, Nancy J. 2008. "Medicine and the UDC: The Process of Restructuring Class 61." *Extensions and Corrections to the UDC* 30: 9-16.

Ogden, C. K., and I. A. Richards. 1946. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*, 8th ed. New York: Harcourt, Brace & World. Inc.

Országos Széchényi Könyvtár Könyvtári Intézet. 2005. *Egyetemes tizedes osztályozás*. 1. kötet, *Táblázatok 1-2. Rész*. UDC Publication P057. Budapest: Országos Széchényi Könyvtár Könyvtári Intézet.

Országos Műszaki Információs Központ és Könyvtár and Országos Széchényi Könyvtár Könyvtártudományi és Módszertani Központ. 1990. *Egyetemes tizedes osztályozás*. 1. kötet, *Táblázatok,* rövidített kiadás. FID Publication 691. Budapest: OMIKK / OSZK-KMK.

Piros Attila. 2015. "Automatic Interpretation of Complex UDC Numbers: Towards Support for Library Systems." In *Classification & Authority Control: Expanding Resource Discovery: Proceedings of the International UDC Seminar, 29-30 October 2015, Lisbon, Portugal*, ed. Aida Slavic and Maria Inês Cordeiro. Würzburg: Ergon Verlag, 177-93.

Piros Attila. 2017. "Automatic Interpreter for Complex UDC Numbers." *Extensions and Corrections to the UDC* 36-37: forthcoming.

Ranganathan, S. R. 1967. *Prolegomena to Library Classification*, 3rd ed. New York: Asia Publishing House. http://hdl.handle.net/10150/106370

Riesthuis, Gerhard J. A. 1997. "Decomposition of Complex UDC Notations." *Extensions and Corrections to the UDC* 19: 13-9.

Riesthuis, Gerhard J. A. 1998. "Zoeken met woorden: hergebruik van onderwerpsontsluiting." PhD diss., Universiteit van Amsterdam.

Riesthuis, Gerhard J. A. 1999. "Searching with Words: Re-Use of Subject Indexing." *Extensions and Corrections to the UDC* 21: 24-32.

Rigby, Malcolm. 1974. *Computers and the UDC: A Decade of Progress 1963-1973*. FID 523. The Hague: International Federation for Documentation.

Robinson, Geoffrey. 2003. "Citation Order in UDC." *Extensions and Corrections to the UDC* 25:19-27.

Slavic, Aida. 2008. "Faceted classification: management and use." *Axiomathes* 18: 257-71. doi:10.1007/s10516-007-9030-z

Slavic, Aida, and Maria Inês Cordeiro, eds. 2015. *Classification & Authority Control: Expanding Resource Discovery: Proceedings of the International UDC Seminar, 29-30 October 2015, Lisbon, Portugal.* Würzburg: Ergon Verlag.

Smiraglia, Richard P., Andrea Scharnhorst, Almila Akdag Salah, and Cheng Gao. 2013. "UDC in Action." In *Classification and Visualization: Interfaces to Knowledge: Proceedings of the International UDC Seminar, 24-25 October 2013, The Hague, The Netherlands*, ed. Aida Slavic, Almila Akdag Salah, and Sylvie Davies. Würzburg: Ergon Verlag, 259-72.

Soergel, Dagobert. 1994. "Indexing Retrieval Performance: The Logical Evidence." *Journal of the American Society for Information Science* 45: 589-99. doi:10.1002/(SICI)1097-4571(199409)45:8<589::AID-ASI14>3.0.CO;2-E

Tartaglia, Stefano. 2004. "Authority Control and Subject Indexing Languages." *Cataloging & Classification Quarterly* 39, no. 1/2: 365-77. doi:10.1300/J104v39n01_05

UDC Consortium. 2013. *Multilingual Universal Decimal Classification Summary.* UDCC Publication no. 088. http://www.udcc.org/udcsummary/php/index.php