Publications Office of the European Union

infeurope

Initiative for Modelling the Legal Analysis Methodology

The structure of Excel workbook for bootrstraping the Legal Analysis Methodology descriptions

Deliverable WP 1.3

Eugeniu Costetchi

06 September 2021

Version 1.2

Disclaimer

Project acronym	LAM#3
Project title	Initiative for Modelling the Legal Analysis Methodology
Document reference	The structure of Excel workbook for bootrstraping the Legal Analysis Methodology descriptions
Author(s)	Eugeniu Costetchi
Editor(s)	Eugeniu Costetchi
Contractor	Infeurope S.A.
Framework contract	10688
Actual delivery date	06 September 2021
Delivery nature	Report (R)
Dissemination level	Public (PU)
Filename	wp1-3-excel-structure-update
Suggested readers	project partners, future users, legal practitioners, software architects

Abstract

The initiative for modeling the Legal Analysis Methodology (LAM) involves, among other things, a transformation step in which the LAM data represented in an Excel format is transformed into RDF format instantiating the LAM-SKOS-AP application profile. This document explains the structure of the Excel workbook serving as input data for the transformation script.

Contents

1 Introduction

The initiative for modelling the Legal Analysis Methodology (LAM) involves, among other things, a transformation step in which the LAM data represented in an Excel format is transformed into RDF format instantiating the LAM-SKOS-AP application profile.

This document explains the structure of the Excel workbook serving as input data for the transformation script. We recommend that the reader is already familiar with the context of LAM project and the general approach presented in Deliverable WP1.1[1].

2 General considerations

The Excel file is composed of worksheets, which are tabular structures organised by *rows* and *columns*. This tabular organisation is mapped to a formal conceptual structure. This mapping enables automatic transformation by a transformation script into a formal model. This section describes in broad lines the conventions and correspondences of the tabular organisation to the conceptual organisation.

2.1 Triple expressions

To aid the explanations we introduce the concept of a semantic triple and then show how it maps to the tabular structure. As its name indicates, a triple is a set of three elements (slots) that codifies a statement about semantic data in the form of *subject–predicate–object* expressions. For example the natural language expressions “Bob knows Frank” or “John likes Mary” conform to the triple structure. The components of a triple consist of a *subject* ("Bob" and "John"), a *predicate* ("know" and "like"), and an *object* ("Frank" and "Mary").

2.2 Worksheet structure

The Excel file contains eight worksheets: one of them define *properties*, two of them define *classes*, other three provide *helper classifications* and the last one contains *namespace prefix mappings*.

The *rows* generally correspond to descriptions of an identifiable element. In this case this is a LAM class or a LAM property. The *row identity* serves as *subject* in a triple statement. We assign an URI such as lamd:md_AU, lamd:c_31. The *columns* correspond to predicates (i.e. properties) in such statements. The *predicate* reference is indicated by the column name in the worksheet header. For example the column "DESCRIPTION" is mapped to the property skos:definition. The *cell value*, at the intersection of a row and a column, corresponds to the statement *object*.

In every worksheet, the first row is the header row which provides column names (titles). Each column name is used for deciding how all the values in that column should be interpreted and

[1] Costetchi E., 2019, Preliminary requirements specification for the Legal Analysis Methodology models

processed. In addition it is possible to relate a column to another one used for describing annotations (explained below).

2.3 Column conventions

The columns may be named according to one of the three conventions: (a) lower case plain text literal, (b) upper case reference keys defined in another worksheet, or (c) encoding of functional links between reference keys.

The columns headed by plain text labels, provided in lower case, usually serve a descriptive purpose and most of the time the transformation script uses them as such without additional processing. These columns are mostly found in the worksheet "LAM properties", for example "code", "controlled value property", "annotation1" etc.

The column names provided in upper case signify reference keys. They shall be interpreted by finding the row with the same code available in the "LAM properties" worksheet. These sort of columns are used in class definitions only, where the keys in the column header function as references to property definitions. Hence following the mapping to a triple statement described above. Examples of such columns can be found in "LAM classes" worksheet which contains headers such as "RJ_NEW", "CC", "IF", "EV" etc. All these values are reference codes to records in "LAM properties" worksheet.

The transformation script takes into consideration the property definition linked by the reference key and possibly additional relations or constraints when processing the column values.

In addition there are columns with headers that encode functional links between reference keys. The convention for such notations is *"KEY1(KEY2)"*, where *KEY1* acts as a *functor* applied to *KEY2*; we read it "KEY1 of KEY2" or, in the context of LAM model, "annotation of KEY2 with KEY1".

These notations are used for encoding annotations of values provided in another columns. For example the pair of columns "EV" and "ANN_COD(EV)" means that the column "EV" contributes to the description identified at the level of a row whereas the column "ANN_COD(EV)" further extends the description provided by the column "EV" in the form of an annotation.

The transformation script processes functionally related column pairs in a special manner tracking two levels of description identification, at the level of the row and at the level of column value, taking into consideration the definition linked to the reference keys, the link between the reference keys and the implied constraints and relationships.

2.4 Cell conventions

The worksheet cells, which are slots formed at the intersection of a row and a column, provide the values filling those slots. We distinguish few kinds of cell values that are each controlled by a set of conventions. The value types are as follows:

- Free text literal
- Short URI notation
- Controlled value

The *free text literals* are Unicode[2] strings which should be in Normal Form[3]. The intended meaning of *short URI notation* is specified by RFC 3986 on Uniform Resource Identifiers[4]. The expected form is short reference URI *"prefix:ID"* where the prefix (base URI) is formally defined in the document. The short URI form is preferred to absolute (resolved) form URI, the latter being discouraged from usage, nonetheless the transformation script is able to identify and process them as accordingly. Both, the free text literals and the short URI notations can be used as either (a) values of properties (denoted by the column header) or (b) as property constraint definitions. The interpretation depends of the column function described below.

The last type of values, the *controlled values*, refer to a convention of specifying cardinality constraints in the class definitions. This means that the cells with controlled values can not be interpreted as property values but serve only as property constraints. The conventions for cardinality constraints in LAM project are provided in Table 1.

Name	Cell value	Cardinality meaning	Alternative cell values
mandatory	Y	1..*	yes, y, according to text
mandatory unique	YU	1..1	
optional	<empty cell>	0..*	
optional unique	OU	0..1	
forbidden	N	0..*	no, n
irrelevant	<empty cell>	0..0	ignore

Table 1: Cardinality constraint conventions

The worksheet cells can contain commented values. It means that a cell can contain a value (literal, URI or controlled) and in addition a comment on that value. The value is separated from the comment by the pipe (|) character like this: "*value | comment*". The transformation script uses the pipe character for detecting commented values, and so this character should not be used for any other purpose.

The worksheet cells can contain multiple values. The new value separator is the new line character (CR/LF). This means that every new line of the cell will be interpreted as a new value for the property indicated by the column header.

[2]The Unicode Standard, Version 3, The Unicode Consortium, Addison-Wesley, 2000. `http://www.unicode.org/unicode/standard/versions/`

[3]Unicode Normalization Forms, Unicode Standard Annex #15, Mark Davis, Martin Dürst. http://www.unicode.org/unicode/reports/tr15/

[4]Berners-Lee, Fielding and Masinter (2005), RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax. `https://tools.ietf.org/html/rfc3986`

The basic structure of a worksheet cell has the following requirements/conventions:

Literal values - free text, without any markers present: (|)

Example	Counter-example
Council Common Position (CFSP number)	text with | pipe | separator inside

Cardinality values with comments - cell starts with the value found in the cardinality constraint convention controlled list (see Table 1), optionally followed by the pipe (|) marker and free text comment

Example	Counter-example
Y	XYZ | cardinality specifications that are not in the foreseen controlled list
O|Under Internal reference the reference to the procedure is doubled (as it is specifically under procedure)	YU | comment | with pipe (|) separators | inside |
N |	| no empty cardinality spec with a dangling comment

Multi line reference values with comments - cell contains one or more lines with one short URI values per line. The URIs generally refer to a specific concept in a controlled list. The last line can be followed by a pipe (|) marker and free text comment. The implicit cardinality interpretation if that the value is mandatory, which is equivalent to cardinality specification "Y" (1..*). Note that there is tolerance for extra spaces or blank lines, which are simply discarded by the parser.

Example	Counter-example
eurovoc:1452 eurovoc:4347 | this is an optional comment	eurovoc:1452, eurovoc:4347, | no comments | with | extra pipes|
eurovoc:1452 eurovoc:4347 | this is an optional comment tolerating empty new lines and spaces	 eurovoc:1452 | bad intermediary comment eurovoc:4347 | good final comment

Multi line reference to authority tables with comments - formally indistinguishable from the case above. Semantically, the difference is that the cell contains references to URIs

of classes or controlled lists rather than an individual concepts. In LAM data, this is the case only in the 'controlled values' column from LAM properties worksheet.

Example	Counter-example
at:country at:corporate-body	at:country country:FRA | no mixing of authority table and reference concept URIs

3 Worksheet conventions

3.1 LAM classes worksheet

The worksheet defining LAM classes plays central role in the LAM project as it defines the document classes used in the legal analysis methodology. It comprises of more than a hundred columns, which can be grouped according to the function they play in class definitions. We distinguish the following functions: *identification*, *description*, *mappings* to other classifications and *property constraints*. All the columns are headed with reference keys defined in the worksheet "LAM properties" described below.

The "UR" column provides a universal identifier (as the title suggests) for the row with values of the form *"prefix:ID"*. The prefix is defined in the prefix worksheet, described in Section **??**, and the ID part is automatically generated.

The *description* columns, containing examples, keywords, comments etc., represent human readable class descriptions. Their values essentially are simple text literals. The *mapping* columns provide correspondences between LAM classes and other classifications, in this case the CDM ontology, the Resource Type authority table, and CELEX classification. These mappings to other classifications are intended for manually or eventually automatically determining and/or validating the LAM class to which a legal document belongs.

The rest of the columns represent *property constraints*. In the context of class definition, property constraints mean that instances of the defined class must respect the specified constraint. The constraints are provided either as a literal value, one or more URI values with an optional comment or cardinality specification (see Table 1). In case of URI values, the constraints mean that the instances of the described LAM class must provide property statements with any of the provided values and not others. If there are multiple values, then the default interpretation is that of alternative values either of which should be found among those provided in the instance data.

In case of cardinality specifications, the interpretation is on the number of times a property is employed for a given instance. For example, mandatory properties must be employed once or multiple times, having the minimum cardinality set to one, while optional unique properties may be employed at most once with minimum cardinality set to zero and maximum to one. The cardinality constraints do not provide any indications about the range of values used of a

given property.

Some constraints headed by a function notation represent annotation constraints on a property. For example the column "EV" (date of end of validity) is annotated with "ANN_COD" (annotation: comment on date) column written as "ANN_COD(EV)". The values in this column represent cardinality constraints on the comment on date property. For example if there is a "O" value provided in "ANN_COD(EV)" column then, whenever there is an end of date property employed on an instance then, that value, may optionally be annotated with a comment on date.

The last column "CLASSIFICATION" provides an external super-class for the defined LAM class as originally specified in the LAM documentation. It is not yet formally decided the exact relation between the LAM class and the classification class, therefore, in the LAM-SKOS-AP model this relation is modelled through collection-item relation.

3.2 LAM property worksheet

The LAM property worksheet defines the meaning of the columns used in the class definition worksheet(s). As mentioned in the introduction above, the columns roughly correspond to predicates/properties in the LAM model (LAM-SKOS-AP) and are locally identified by a unique "Code" (usually in capital letters). The same codes are used as reference values in the column headers of the class definition worksheet indicating which property shall be used from the model for each column. The "Code" is used to generate the LAM property URI used in the formal statements.

The property definition worksheet is structured as follows. The "Label" column provides a human friendly property title; the "Definition" provides a human readable property meaning. "Analytical methodology" is a description of how the property contributes to the LAM practice. "Specific cases" and "Comments" provide examples, exceptions and additional comments related to property usage. Example values for these columns are provided in Table 2.

URI	Label	Property	Controlled value	Definition
lamd:EXAMPLE	English example	skos:example@en		English Example. This field used in the cataloguing methodology for information purposes.
lamd:CDM_CLASS	CDM class	lam:cdm_class		Class or subclass according to CDM.
lamd:FM	Type of act	cdm:resource-type	at:resource-type	Type of act is usually mentioned in the title.

Table 2: Example of human readable fields in LAM property definition

The "property" column specifies URI of the equivalent property formally defined in CDM ontology (other namespaces are also accepted). If there is a range constraint to, for example, a

controlled vocabulary then it is indicated in the "controlled value property".

The "property type" column indicate a formal constraint on how the property can be instantiated. Two options are available: "object property", which means that the range is always an URI and "data property", which means that the range is always a literal. This specifications corresponds to OWL2 semantics of owl:ObjectProperty and owl:DataProperty. If left unspecified, the fallback is the rdf:Property semantics, but this option is strongly discouraged.

As mentioned in Section 3.1, some CDM properties are annotated to provide extra information. There are more than 15 columns, each one representing a CDM annotation property that may be used for the defined property. The values in the columns provide the range constraints on the corresponding CDM property. These columns have the same pattern: "ANN_property_abbreviation". The translation pairs of mappings between the URI of the CDM property and the URI of the LAM property are provided in an auxiliary "mappings" worksheet. This worksheet plays a technical role for the transformation script.

The last column "CLASSIFICATION" provide a classification structure for the defined properties as originally specified in the LAM documentation.

3.3 CELEX classes worksheet

This worksheet aims at capturing the description of CELEX classes following the logic that has been used to allocate CELEX numbers since the setting-up of the EUR-Lex database (formerly known as CELEX). The CELEX classes are defined as a combination of DTS, DTT, DTA and OJ_ID columns (described below) and are structured on three levels:

1. DTS classes (CELEX sectors)
2. DTS*DTT (power product) classes. These classes corresponds to rows in the sector tables describing DTTs of the sectors.
3. DTS*DTT*OJ_ID (power product) classes. These classes correspond to cells in the sector tables describing DTTs for each of the three OJ_IDs of the sector.

This combination of DTS, DTT, DTA and OJ_ID columns that define CELEX classes is hard coded by the author.

DN	DTS	DTA	DTT	DTN
32019R0001	3	2019	R	0001
C2019/123	C	2019	<empty>	123
52014AE1723	5	2014	AE	1723

Table 3: Examples of CELEX number composition

$$DN = <DTS> <DTA> <DTT> <DTN> \quad (1)$$

The CELEX number anatomy is provided in Formula 1. Examples of CELEX numbers and how they are composed can be seen in Table 3. Where the column name acronyms mean the following:

- DN - the specific instance of CELEX number. Legal document metadata.
- DTS - Sector
- DTT - Document type
- DTA - The year
- DTN - The number

3.4 Classification worksheets

There are currently 3 classification worksheets: LAM property classification, LAM class classification and CELEX classes classification. These classifications are modelled in LAM-SKOS-AP as "skos:Collections". It is done so to postpone the decision on what role each of these classifications plays in the overall taxonomy. The structure of classification/collection worksheets is the following:

- URI - uniform resource identifier obtained by concatenating a prefix (worksheet specific) with the CODE column
- CODE - a mnemonic reference
- PARENT_COLLECTION - URI of the parent class/collection
- ORDER - a linear sorting of classes used, for example, when a human readable HTML/PDF document is generated.
- LABEL - a human readable label for the specific URI
- DESCRIPTION - a human readable description of the specific URI
- COMMENT - a comment for the specific URI

4 Common issues

- Distinguish between reference to a specific value and the entire controlled vocabulary (RDF realisation is not the same)

Examples
at:fd_340
at:fd_335
at:corporate-body
at:subdivision

- Cardinality specification combined with value specification. Should be either one or the other as the second one is treated by the parser as a plain literal value.

Examples
Y |eurovoc:4370 eurovoc:5455
Y | subject-matter:BCE subject-matter:INST
Y | 12016EN01

- Non-URI code references that are meant as URIs. The editors shall also pay attention when resolving the references as the current codes do not correspond to those in controlled lists. For example, in the treaties table, the TFEU concept does not exist. Instead, the editor shall refer to at_treaties:TFEU_2016. The same holds for TEU, which should likely be at_treaty:TEU_2016

Examples
Y | TFEU, TEU
Y | 12012M011
21994A0103(01)
agent, at:fd_50
Y|DG06
REP
LIE

- Non-standard multiple value specifications. The value separation shall be done at new line and no other separator such as comma (,) semi-colon (;), slash (/) and others.

Examples
at:fd_400:FIN/EXERC/BUDG |
fd_340:TRANS/PARL fd_340:TRANS/CONS
at:fd_365, at:fd_335, at:fd_330, at:fd_361, at:fd_340, at:fd_335, at:fd_350, at:place, at:country
at:corporate-body, at:fd_50

- Some values are provided as long URIs. They shall be switched to the short/compressed

form to normalise.

Examples
http://publications.europa.eu/resource/authority/country

- Correct the references; point to the controlled list instead of the dataset

Examples
http://publications.europa.eu/resource/dataset/eurovoc

- Re-asses the usage of [O]ptional and <blank> cells. We recommend that the optional property constraints shall be left as blank cells as they are ignored by the transformation script for brevity and reduce redundancy.

Examples
O | fd_400:P/EEE
O

- Some URIs are not valid. And in the cases specified below, it is not possible to provide valid values because not all of the ATTO tables comply with the EU-SKOS-AP standard. In such cases, the invalid URIs shall either be provided as comments, or replaced by the corresponding value in an authority list.

Examples
fd_330/EXERCICE
at:fd_400:FIN/EXERC/BUDG |

- Some literal values and comments contain non-standard characters. It is advisable to curate the text for non UTF-8 characters. Then preferably reduce to ASCII encoding if possible.

Examples
<non-visible-char>
…
«
°
—
©

- Double line breaks in literal values. For consistent an uniform formatting, especially in the generated HTML/PDF it is recommended to keep singe line breaks.

Examples
Amendment (MM) This link contains the CELEX number of earlier acts modified by the given act. It is a link from: amending act to the amended act (displayed under Amendment to - MS) amending proposal to amended proposal (displayed under Earlier related instruments - EA) This is rather more general relation. It should be used only if there is not any other more specific relation available.
Documents in EUR-Lex fall into one of 12 sectors. 1 Treaties 2 International agreements 3 Legal acts 4 Complementary legislation 5 Preparatory documents 6 EU case law 7 National transposition 8 References to national case-law concerning EU law 9 Parliamentary questions 0 Consolidated texts C Other documents published in the Official Journal C series E EFTA documents

- Authority tables (AT) shall be preferred to FD tables where possible. Some FD tables shall be avoided in the LAM data, especially FD_400 and FD_340. These tables seems to be designed to be used for the localisation purpose and not as reference table.

Examples
agent, at:fd_50