Publications Office of the European Union

infeurope

Initiative for Modeling the Legal Analysis Methodology

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

Deliverable WP 1.4

Eugeniu Costetchi

02 October 2019

Version 1.0

Disclaimer

Project acronym	LAM project
Project title	Initiative for Modeling 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	02 October 2019
Delivery nature	Report (R)
Dissemination level	Public (PU)
Filename	wp1-4-excel-structure
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 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. We recommend that the reader is already be familiar with the context of LAM project and the general approach presented in Deliverable WP1.1[1].

2 Worksheet structure

The excel file contains five worksheets: two of them define classes, two of them define properties and one defines namespace prefix mappings. The worksheet is composed of rows and columns. The rows roughly correspond to descriptions of an identifiable entity/element, the columns correspond to predicates (properties) in such descriptions while the cell values to predicate objects. In every worksheet, the first row is the header row. Each header cell denotes how the values in the corresponding columns should be interpreted and processed (predicate specification). Some denotations, as we will see below, signify/represent (a) plain labels, (b) reference keys described in another worksheet, (c) or encodings of functional links between reference keys. The rest of the non empty rows represent distinct definitions comprising value statements in all or most of the columns.

The columns that represent plain labels, provided lower case, serve a descriptive purpose and most of the time the transformation script uses them as such without much additional processing. Examples of such columns can be found in "LAM metadata worksheet" which contains headers such as "Code" "controlled value property" "annotation1" etc.

The columns that signify reference keys, provided in upper case, are richer in meaning, which is provided in another worksheet. These sort of columns are used in class definitions only, where the keys in the column header function as references to property definitions. The transformation script takes into consideration the definition linked to the reference key and any additional relations and constraints when processing the column values. Examples of such columns can be found in "Classes Complete" worksheet which contains headers such as "RJ_NEW", "CC", "IF", "EV" etc.

The columns that encode functional links between reference keys (using function notation), provided in upper case and round brackets, signify a second order descriptions. They 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 convention for such notations is *"KEY1(KEY2)"*, where *KEY1* acts as a functor

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

applied to *KEY2*; we read it "KEY1 of KEY2" or "annotation of KEY2 with KEY1" The transformation script processes such 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.

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 form 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 are 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	O	0..*	
optional unique	OU	0..1	
forbidden	N, <empty cell>	0..0	no, n

Table 1: Cardinality constraint conventions

[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 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.

2.1 LAM class definition

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 almost a hundred columns, which can be grouped according to meaning and 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 metadata" described below.

The URI 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 2.4, 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, URI or cardinality specification (see Table 1). In case of literal or URI values, the constraints mean that the instances of the class being defined must provide property statements with exactly same values. 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 three columns "Classification level 1" to "Classification level 3" provide a classification structure for the defined documents as originally specified in the LAM documentation.

2.2 LAM property definition

The LAM property definition worksheet defines the meaning to 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 and are locally identified by a unique "Code" (usually in capital letters). The same codes are used as a 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 in-

stantiated. 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 2.1, some CDM properties are annotated to provide extra information. The columns "annotation_1" to "annotation_7" specify which CDM annotation properties may be used for the defined property. Columns "controlled value_annotation_1" to "controlled value_annotation_7" provide range constraints on the corresponding property. The values in columns "annotation_11" to "annotation_71" are automatically generated and do not provide any additional information, but play a technical role for the transformation script, providing a mapping between the URI of the CDM property and the URI of the LAM property. The translation pairs are provided in an auxiliary "mappings" worksheet.

The last two columns "Classification level 1" and "Classification level 2" provide a classification structure for the defined properties as originally specified in the LAM documentation.

2.3 CELEX class and property definition

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.

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

Properties describing class at each level are as follows.

Level I classes:

- label
- code (=DTS)
- DTS
- definition
- scope note (optional)
- comments (optional)

Level II classes (same as above plus additionally):

- *code (=DTS*DTT)
- DTT
- author

Level III classes (same as above plus additionally):

- *code (=DTS*DTT*OJ_ID)
- OJ_ID (either *"OJC"*, *"OJL"* or *"EuroLex"*)
- DTA source of .. (as indicated in CELEXspecification documentation section 1 on general rules)
- DTN source of .. (as indicated in CELEXspecification documentation section 1 on general rules)

The CELEX property definition worksheet, just like the one for LAM properties, defines a set of properties used in CELEX class definition. They are primarily CELEX composition properties but in the current project a few auxiliary properties are used.

2.4 Namespace prefix definitions

This worksheet provides a mapping between the LAM property definitions and CDM ontology (or another namespace). This worksheet is auxiliary and has a technical role aiding the transformation script. The worksheet is composed of two columns, first the URI of the CDM ontology and the second one the URI of the LAM property.