**Title of session**: Controlled vocabularies management tool requirements specification

**Date**: 27 October 2016

**Participants**: Kerrin Borschewski, Taina Jääskeläinen, Sanda Ionescu, Anne Etheridge, Wendy Thomas, Joachim Wackerow

\*To be discussed further

X task assigned

# Introduction

A proposal has been submitted to Cessda for a tool to manage controlled vocabularies (CVs) for use by Cessda institutions. The tool will be referred to as the CV Manager Tool. As Cessda recommends using DDI CVs this requirement specification will include DDI Alliance requirements.

The CV Manager Tool will not manage suggestions for new or amended CVs. This tool only manages data entry for creation and maintenance of CVs once they have been approved by the DDI Controlled Vocabularies Group or by Cessda.

The storage container for the CV should allow for: persistent identifiers, version control and dating. This facilitates machine-actionability and human understanding of the CV.

The publication process for DDI CVs will be managed by the DDI Alliance separately from the tool. The Alliance will set requirements for tool outputs and will provide a repository for long term preservation of DDI vocabularies.

# Workflow diagram for publication and preservation of DDI CVs created with the CV Manager Tool

# C:\Users\AE\AppData\Local\Microsoft\Windows\INetCacheContent.Word\workflow.png

# Primary requirements

1. Online access regardless of location for example: data entry, browsing, creating exports
2. Input in required formats

* in SKOS
* input current CVs through transformation from Genericode to SKOS

1. Access to CV tool data for various purposes through exports or an API

* Output format requirements
  + SKOS mandatory as canonical format
    - XKOS, SKOS – lots of tools and well recognized
    - JSON-LD – for web brow
    - RDF – XML useful for preservation
  + human readable
    - display, web page, HTML
    - printable PDF
    - Word? Excel? – may not be necessary based on the quality of the HTML
  + Genericode
    - for current versions (for one-time input, but maybe existing CVs can be transformed to SKOS first, so not needed?)
    - for future versions? (may not be needed)
  + YAML? Consult Larry!!
* Custom metadata (CM) package (DDI 4) which contains the classes used to describe and populate information from a vocabulary that is external to the known DDI structure. It allows for the sharing and reuse of a structure of key, value pairs along with description of the keys and assignment of value type to the values. Cardinalities may also be specified. It also includes a set of classes to be used to describe a controlled vocabulary within DDI. <http://lion.ddialliance.org/package/custommetadata> (mandatory)
* API – technical expertise needed for specification
  + Cessda technical group input
  + Publish - push
  + Pull access – harvesting for catalogue creation or search support
  + Fully documented API

1. Publishing CVs is straightforward, and is straightforward in the RDF-space

* Push the button, automated output in binding
  + Applies only to CVs that have been marked as final
* Export objects should be validated for quality control e.g. SKOS needs to be validated either within tool or by external tools – technical format of the SKOS, not content
* One-step publishing system at least from the machine to DDI publication process
* Need to work with DDI Technical Committee to ensure that DDI Alliance can publish, has appropriate tooling (CV tool is not the publication tool for DDI CVs)
* Clear understanding of what CV tool does, what DDI Alliance does
* Output filtered for DDI to go to their repository
* Cessda Metadata Portfolio does not publish the DDI CVs, but references them

1. Guaranteed continuity and access rights

* Guaranteed continuity – Cessda policy decision
* Cessda to maintain the tool long term and provide access to relevant users
* DDI is responsible for the CV’s that it publishes from the system. The system should ensure preservation of internal administrative and versioning information that provides provenance and relationships between objects in the system. In the event that the CV Manager Tool is no longer supported, DDI should receive a copy of the content in its region of the system along with the administrative, provenance, and relational information is a format suitable for transfer to a new system.

1. Handles both source CVs and their language versions

* Ability to filter for specific languages for use by organisations
  + A tool to do this with (tool specific or DDI)
  + Language/country
  + Support UTF-8
  + Support non-latin scripts (Cyrillic, Arabic, Hebrew, Japanese)
* Support for translating into equivalent languages (similar to those used for translation in comparative survey question work), in similar ways to ELSST and DASISH tools
  + Policy discussion with Cessda and DDI – all terms in a CV must be translated before publication of the target language CV (if yes tool should enforce this rule)
  + Policy discussion with Cessda and DDI on whether all language versions need definitions translated, if not use source definition?
* Contains field for individual/group/organisation responsible for content

1. Handles versioning of both source CVs and their language versions

* Support levels of versioning (complex versioning re source and target language and links between them)
* Source version change – driven by change in source language
* Language/translation version change - driven by change in language coverage/content
* Flexibility to force DDI versioning rules whatever they may reasonably be
* Versioning approaches must be documented and the rule internalised by system
* System/tool should enforce versioning rules
* Should be able to replicate a previous version of a published CV
* Produce a version of a CV with complete documentation of differences
* Tool must enforce documentation of changes made
* Suggestion for versioning
  + TimeMethod\_3.4\_FI\_1.3
  + TimeMethod\_3.4\_DE\_2.1
  + TimeMethod\_3.4\_DE\_AT\_2. ( Austrian German, locale code)
  + TimeMethod\_4.0\_FI\_1.0
  + [CV name]\_[DDI CV source version]\_[Language code]\_[TargetLanguage CV version]
  + Target language version starts from 1.0 every time the source version changes.
* Issues to be discussed with both DDI and Cessda: language code enough or language country pair codes? It is probable that language codes will be enough but the tool must allow language country pairs just in case.
  + DDI – language code will be enough

1. Handles hierarchical CVs

* Currently support up to four levels, support up to five
* Not a thesaurus – relationships are hierarchical only

1. Allow groups and roles

* \*Access management for different roles and activities for example: administrator of system for rights management, administration of a source language, administrator of target language version, general browser(?)
* Support for different access roles for different regions of the system (DDI, Cessda)

1. Alerts of changes

* System triggered alerts to ‘users’ regarding version changes
* Allow watchers to see changes that have been made

1. Statistics

* \*Examples such as user statistics, language version completion stage - to be prioritised

1. Allows reference link to external sources (within definitions or term source etc.)

* to persistent identifiers primarily rather than URLs

1. Supports persistent identifier (PID) at CV level

* Tool does not control the formats or content of the persistent identifier of CVs.
* PIDs for DDI CVs are created in the publication process. The tool allows input of such PIDs as versioning is controlled by them.
* CVs can have more than one identifier
* per language version of CV
* choose persistent identifier system – may be dependent on CV owning organisation
  + DDI example: ddialliance.org/cv/aggregationmethod to overview page with all languages and versions
  + DDI example to specific language version: ddialliance.org/cv/aggregationmethod/de/version

1. Validation of content

* Custom metadata model in development
* \*Internal validation for quality control – standard spellchecker (EN-US for source as per DDI), grammar etc.
* Handling of phrases: multi-word terms (e.g., “child care” or “traffic regulation”) need to stay bound as a phrase for handling by applications.
* For source language and for target languages (nice to have?)
* Joachim’s rules for creating codes for terms (see Attachment 1)
* System/tool should enforce versioning rules
* Tool must enforce documentation of changes made

1. Attribution of CV

* Responsibility of updating CV
* Owning organisation – this could have an effect on the PID

# Possible fields within CV Manager Tool

**Controlled Vocabulary ( CV)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field label** | **Description of field** | **Format of field** | **Mandatory Y/N** |
| CV name | What the CV is called | Free text | Y |
| CV class DDI-4 | The class name of the CV in DDI 4. This is not translated. | Follows code naming rules (Attachment 1) | N |
| CV class DDI-L | The class name (Element Name) of the CV in DDI Lifecycle. This is not translated. | Follows code naming rules (Attachment 1) | N |
| CV class DDI-C | The class name (Element/Attribute Name) of the CV in DDI Codebook. This is not translated. | Follows code naming rules (Attachment 1) | N |
| Description | A definition of the CV | Free text |  |
| Language | Language used for CV name and CV description | ISO ?? | Y |
| Version | Created from version rules (TBA) |  |  |
| Persistent identifier | Machine created from version and Owning organisation |  |  |
| Owning organisation | probably need CV for this if it is used in creating PID |  |  |
| License | Default is Creative Commons Attribution-ShareAlike 4.0 International. |  |  |
| Copyright | Default is DDI Alliance? |  |  |
| Editor of CV |  |  |  |
| Date started |  |  |  |
| Date completed |  |  |  |
| CV published Y/N |  |  |  |
| Note | Note field for general information |  |  |
| Link to items for this CV |  |  |  |

**CV Item**

There can be one or more CV items per CV

|  |  |  |  |
| --- | --- | --- | --- |
| **Field label** | **Description of field** | **Format of field** | **Mandatory Y/N** |
| CV item term | What the CV item is called | Free text | Y |
| CV item description | A definition of the CV item | Free text |  |
| CV item code | The code to be used for the CV item | Follows code naming rules (Attachment 1) |  |
| Version | Created from version rules (TBA) |  |  |
|  |  |  |  |
|  |  |  |  |

**ATTACHMENT 1**

Code naming rules (Joachim Wackerow)

* Code terms should be self-describing and are based on the human-readable term in question.   
  A code term is a single English word or a concatenation of several English words. Each word starts with an upper case character.
* Only alphanumeric characters are allowed.
  + No abbreviations are allowed. Only a few exceptions are allowed in the case of acknowledged abbreviation code lists like the two-letter code for the US states.
  + Code terms should be not too long in respect to readability. There is no hard rule for that. It is reasonable not to use more than 5 words and 50 characters.
* Hierarchical CVs: One code exists for each level. The upper level codes get repeated. The concatenation of the codes of all levels are used as allowed value in DDI. The separator between the levels is a dot. Examples:
  + Longitudinal.Panel.Continuous
  + SelfAdministeredQuestionnaire.WebBased
* No slashes, replace with ‘Or’, example: EventOrProcess.
* The code is just a code. It should have a meaning for the purpose that developers can deal with them. Search interfaces should search on the list of captions and/or definitions i.e. the human-readable part.