Concept Constraint Definition Design

Timothy W. Cook, 2011-05-08

This document is a breakdown of the design concepts and the tooling needed in the Healthcare Knowledge Component Repository (HKCR)[1], to perform the conversion from XMind[2] format to XML Schema CCD format. The XMind file is a compressed file with several files needed by XMind. The only file we need to process the CCD is the *content.xml* file.

Background

The XMind tool was chosen for two major reasons. 1) The visual modelling approach is easier for domain experts to understand how their concepts are created with the building blocks of the reference model. 2) XMind creates a Universally Unique Identifier (UUID)[3], for every new node in it's map. So a domain expert may make as many copies of the various building blocks as they need and each one carries its own unique id.

The Multi-Level Health Information Modelling (MLHIM)[4] specifications define a reference model for healthcare data (not behavior). Though basic assumptions can be made about object oriented languages used for implementations, the actual behavior of an application is unique to that application and is not a concern for the data models developed as CCDs. The Multi-Level Health Information Modelling Reference Model (MLHIM-RM) is expressed in a set of XML Schemas[5]. These schema define a very broad data model. The CCDs then define constraints against that reference model to build a specific concept. While multiple concepts may be designed into one CCD it is considered best practice to design for one concept and connect/embed other CCDs into facilities called Slots. This approach has proven to provide a more flexible and conceptually clean product.

Governance

The open governance model used for CCDs allows for anyone interested in exploring and improving the process of CCD development is welcome to participate. The only requirement is to join the open group and mailing list on Launchpad[6]. This mailing list is where you can get all of your questions answered as well as participate in the overall MLHIM project and all sub-projects.

This open governance model means that anyone can create any type of CCD they desire. One “gentleman's agreement” rule at this point is to not overwrite anyone else's XMind filename. These XMind filenames are only relevant during the graphical design process. When HKCR is operation each modeler or group of modelers will have their own workspace to store their files. Also when HKCR does the conversion to XML Schema, it will create a UUID for each individual CCD.

For example the Blood Pressure XMind file is blood_pressure.xmind but after conversion the filename might be 3hios3rf8l8r1a13pccevqevc9.ccd. A best practices approach might be to use the primary author's initials or some other unique name such as an organization name in the XMind filename.

---

1 https://launchpad.net/hkcr
2 http://www.xmind.net/
3 http://en.wikipedia.org/wiki/Uuid
4 http://www.mlhim.org
5 http://www.w3schools.com/schema/default.asp
6 https://launchpad.net/mlhim

These files are not meant to be human-readable. The semantics of the CCD is contained in the MetaData and will be able to be read by any of the tools and applications using CCDs.

The CCD File

As previously stated, the CCD is an XML Schema that expresses constraints against the MLHIM-RM in order to represent a concept data model. In this section I will present some examples of how the schema is designed and how the constraints are represented. I will be using the Blood Pressure XMind content.xml file.

The top of the content.xml looks like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><xmap-content
xmlns="urn:xmind:xmap:xmlns:content:2.0"
xmlns:fo="http://www.w3.org/1999/XSL/Format"
xmlns:svg="http://www.w3.org/2000/svg" xmlns:xhtml="http://www.w3.org/1999/xhtml"
xmlns:xlink="http://www.w3.org/1999/xlink" version="2.0"><sheet
id="17cc6ffrnu597b4n4a15lmdqvu" theme="brainy.defaultGenre.classic"><topic
id="6684i3d42di49rc65t36al000m" style-id="2su1ecv0ofm5qq01vtpoq869ka"
timestamp="1304604152222"
xlink:href="xmind:#49ocl3g7gl1prvkr2jabn4qeqv"><title>CCD</title><
```

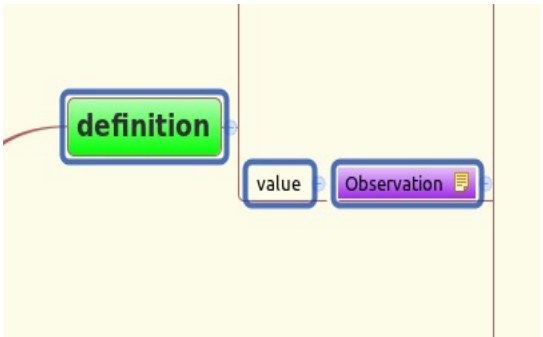
Note that the title of this topic is CCD and it's UUID is 6684i3d42di49rc65t36al000m. This UUID is consistent across all CCDs because it is the root topic of the root page from the template. Therefore it cannot be used as the CCD ID. HKCR will need to generate one during each conversion.

The header for the CCD schema file will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright 2011 Timothy W. Cook and Contributors. See http://www.mlhim.org-->
<!--Blood Pressure CCD - This is an example; of the expected output from HKCR-->
<xs:schema elementFormDefault="qualified" id="3hios3rf8l8r1a13pccevqevc9.ccd"
targetNamespace="http://www.hkcr.net/ccd/3hios3rf8l8r1a13pccevqevc9.ccd"
version="2.1.1" xmlns:mlhim2="http://www.mlhim.org/mlhim2/211"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
           <xs:include schemaLocation="mlhim2.xsd"></xs:include>
```

Note here the UUID has been assigned as 3hios3rf8l8r1a13pccevqevc9 and the target namespace is this UUID in the http://www.hkcr.net/ccd space. We also include the mlhim2.xsd schema which is a common schema tocollect all of the MLHIM2 schemas associated with this version number and found in the namespace for mlhim2.

Since the CCD and its definition attribute are the same across all CCDs we can use this to begin traversing down the tree. When we get to the Observation class:



The content.xml looks like this:

```
<topic id="2ok6l2hn4vi4jerakph75en3fa" style-id="0i5idp9uujf5ii92h91snghi4n"
timestamp="1304707578578"><title>Observation</title>
```

Here we have a topic ID of 2ok6l2hn4vi4jerakph75en3fa for this Observation. So the generated XML Schema (just for this type and also note that not all attributes/elements are included in these examples) will look like this:

```
<xs:complexType name="2ok6l2hn4vi4jerakph75en3fa">
<xs:annotation><xs:documentation>
Entry subtype for all clinical data in the past or present, i.e. which (by the time it
is recorded) has already occurred. OBSERVATION data is expressed using the class
HISTORY<T>, which guarantees that it is situated in time.
OBSERVATION is used for all notionally objective (i.e. measured in some way)
observations of phenomena, and patient-reported phenomena, e.g. pain.
Not used for recording opinion or future statements of any kind, including instruc-
tions, intentions, plans etc.
</xs:documentation></xs:annotation>
 <xs:complexContent>
 <xs:restriction base="mlhim2:Observation">

<xs:element name="data" maxOccurs="1" minOccurs="1" type="mlhim2:History">
<xs:annotation><xs:documentation>
The data of this observation, in the form of a history of values which may be of
any complexity.
</xs:documentation></xs:annotation>
</xs:element>


<xs:element name="protocol" maxOccurs="1" minOccurs="1" type="mlhim2:ItemTree">
<xs:annotation><xs:documentation>Description of the method (i.e. how) the
information in this entry was arrived at. For OBSERVATIONs, this is a description
of the method or instrument used. For EVALUATIONs, how the evaluation was arrived
at. For INSTRUCTIONs, how to execute the Instruction. This may take the form of
references to guidelines, including manually followed and executable; knowledge
references such as a paper in Medline; clinical reasons within a larger care
process.</xs:documentation></xs:annotation>

…. all of the additional elements of Observation would be here ….

</xs:restriction>
</xs:complexContent>
</xs:complexType>
```

Taking a look at this we can see that we create a new Complex Type with the UUID as the name. This

approach provides a way to establish unique paths throughout the data. But every instance document based on this schema will have the same paths so they can be consistently queried. The complexType 2ok6l2hn4vi4jerakph75en3fa is a restriction of the mlhim2:Observation Type. We have included in this example just two of the elements. The data element does not have any further restrictions since it is mandatory and it also must be of type mlhim2:History. However, the protocol element that is inherited from CareEntry is constrained. The type has been specified as an ItemTree whereas in CareEntry it was allowed to be any ItemStructure. Also, minOccurs has been set to "1" where it is "0" in CareEntry. Again this is the process for all of the components defined in the XMind file in order to create the schema CCD.

HKCR

Just a few notes on the work needed to create an open source conversion/repository.

Since this tool is first a content management system where domain experts can collaborate and develop the XMind CCD files. I have chosen the most popular and extensible content management system; Plone[7]. It is also chosen because it is written in Python. There is a tool that can be used parse the content.xml file. It is called mekk.xmind, also written in Python. Though HKCR will still need a significant amount of logic added. The mekk.xmind tool is a solid foundation.

7 Http://www.plone.org