Working with ValueLists and the WhereClause in define.xml

As of define.xml 2.0, it is mandatory to define under which circumstances value level metadata is used, using so-called "WhereClauses". For example, for VSORRES, the properties of the value itself will differ depending on what the value of VSTESTCD is. For example, when VSTESTCD=DIABP (diastolic blood pressure), we expect the value to be an integer between 80 and 120 (mm[Hg]) or so. In case VSTESTCD=WEIGHT, we probably expect a floating point number, and the unit either being kg or pounds. For some other tests, we might however expect a string, like "S", "M", "L" or "XL" in the case that the test is "FRMSIZE" (frame size).

In this short tutorial, we will describe a somewhat more complicated example which is also described in the define.xml 2.0 specification: we will describe that the weight of a subject is described in pounds in case the country of the subject is the USA, and in kilograms in case the country of the subject is France or Germany\(^1\). Such a condition is described in define.xml 2.0 using a so-called "WhereClause".

"WhereClause" did not exist in define.xml 1.0. Instead, some people used nested "ValueLists" instead. In define.xml 2.0 and higher, nested Valuelists are not necessary anymore and should not be used.

Now, important is that we assign the "WhereClauses" to the correct SDTM variable. Because in the mentioned example the information is about units used, the "WhereClauses" needs to be assigned to VSORRESU. In case that we state that mm[Hg] is an integer for VSTESTCD=SYSBP and DIABP, and that the value is of type "text" enumerated to "S", "M", "L" and "XL" for VSTESTCD="FRMSIZE" the "WhereClauses" needs to be assigned to VSORRES.

So what we need to express is:

- the value of VSORRES is an integer in case VSTESTCD=DIABP or SYSBP
- the value of VSORRES is of type text and enumerated to "S", "M", "L" and "XL" when VSTESTCD="FRMSIZE"
- the unit in VSORRESU is "pounds" when VSTESTCD=WEIGHT and DM.COUNTRY=USA
- the unit in VSORRESU is "kilogram" when VSTESTCD=WEIGHT and DM.COUNTRY=GER or DM.COUNTRY=FRA

\(^1\)In the define.xml 2.0 specification, Mexico and Canada are mentioned for the use of kilograms, but using France and Germany is more clear especially for non-US users.
Creating a ValueList for VSORRES

Let us first create a valuelist for VSORRES. The easiest is to start from an existing codelist, and use the menu "Insert – Create New ValueList from existing CodeList":

and look for "VSTESTCD",
In this case, we will not check "Create simple 'WhereClause' automatically, as we want to develop a more complicated "where".
This is leading to:

where we start from CL.VTESTCD as our metadata will depend on the value of VTESTCD. This leads to a new dialog with:
allowing us to assign metadata for each type of test. In this tutorial, we will limit ourselves to a few tests only, i.e. DIABP", "SYSBP", "HEIGHT", "WEIGHT" and "FRMSIZE", so we can remove all other rows. We will however only show the "WhereClause" for each of them, as they are all very similar, but with different values for the "check value" (see further).
After having removed the unnessary\(^2\) rows(using the "Remove row" button), we can start adding the data types:

\(^2\)This depends of course on your own study.
One can always click the "Validate" button to check whether the combinations of filled fields makes sense.

Also notice that we changed the OID into "VL.VSORRES" to reflect that this is a valuelist for the variable VSORRES. We also see that the last "Description" cell is colored red, meaning we do need provide a value there:

Now, we do also want to provide the reviewer the information what the origin of the data is. This can of course depend of the test code. In our case, we had a CRF with all vital signs on one page. So when clicking the cell "Origin" for "DIABP" the following dialog is displayed:

3"Description" replaces "def:Label" in define.xml v.2.0
where can select between⁴:

- **Assigned**: judgement from evaluator not being the investigator
- **Protocol**: prescribed by the protocol
- **Derived**: calculated using some algorithm
- **Electronic Data Transfer (eDT)**: e.g. ECG data, lab data
- **CRF**: case report form

In our case, we of course select "CRF". Some other fields become available:

---
⁴ In the case of Define-XML 2.1, the dialog will look a bit different.
We can either choose between a single page or page list (e.g.: 21 24 27) or a PDF named destination or a page range where we then need to provide the first and last page. In our case, the field that goes into VSORRES for VSTESTCD=DIABP can be found on page 25 of the annotated CRF. Clicking OK leads to:

<table>
<thead>
<tr>
<th>OID</th>
<th>Name</th>
<th>Data Type</th>
<th>Length</th>
<th>Sign Digits</th>
<th>Origin</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITDIABP</td>
<td>DIABP</td>
<td>integer</td>
<td>3</td>
<td></td>
<td>CRF</td>
</tr>
<tr>
<td>ITFRMSIZE</td>
<td>FRMSIZE</td>
<td>text</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITHEIGHT</td>
<td>HEIGHT</td>
<td>float</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ITSYSBP</td>
<td>SYSBP</td>
<td>integer</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITWEIGHT</td>
<td>WEIGHT</td>
<td>float</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

and we can now to the same for the other rows, leading to:

For "FRMSIZE" we need to indicate that the value is enumerated. This is usually done by a CodeList, so we click the cell "CodeList" for row "FRMSIZE",

leading to:
and scroll down, or use the "Search" button until we find the appropriate codelist. Notice that the allowed values are displayed in a tooltip, making it easier to find the correct one.

All we still need to do (mandatory) is to provide the "WhereClause" for each row, indicating under which conditions the metadata (data type, length, enumeration) need to be used:

<table>
<thead>
<tr>
<th>Method</th>
<th>CodeList</th>
<th>WhereClause</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WC.INI.DIABP</td>
<td>WC.INI.DIABP</td>
</tr>
<tr>
<td>CL.FRAMESIZE</td>
<td>WC.INI.FRMSIZE</td>
<td>WC.INI.HEIGHT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WC.INI.SYSBP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WC.INI.WEIGHT</td>
</tr>
</tbody>
</table>

In the "WhereClauses", we will need to define that the metadata for "DIABP" need to be used when VSTESTCD=DIABP (simple isn't it?). So clicking the cell WC.INI.DIABP (the system already provides a proposal for the identifier), the following dialog is displayed:
The only thing we need to state as condition is that: VSTESTCD=DIABP, which easily accomplished by:

using the comparator "EQ" which means "equals to", which is exactly what we want.

If in doubt, one can use the button "Show 'Where' clause", providing:
We can also add a comment, and for the comment, even a link to an external document. Adding a comment is especially important when the condition is complex (see further) so that one can a textual declaration of the condition. If the condition is very complex, one may also provide a link to a specific page or named destination of an external document by clicking the "External document for comment" button, leading to:

in this case to a named destination within a PDF document.

Of course we also need to create the "WhereClause" for each other row, and adapt the condition accordingly (for example for "WEIGHT", we must state VSTESTCD=WEIGHT).

One thing we should NOT forget, is to assign the new ValueList with all its "WhereClauses" to the SDTM variable "VSORRES". This is done by selecting VSORRES in the main table and then using the menu "Edit – SDTM Variable Properties":
and then use the checkbox "New ValueList OID", and selecting the newly created ValueList "VL.VSORRES"

In the define.xml, this will later look like:
The first picture showing how the ValueList VL.VSORRES is referenced by VSORRES, the second one showing the details and some (the first two from five) "WhereClause" definitions. The details are then found in the corresponding "ItemDefs":

```
<def:ValueListDef OID="VL.VSORRES">
   <def:ItemType OID="IT.DIABP" Mandatory="No" OrderNumber="1">
      <def:WhereClauseRef WhereClauseOID="WC.IT.DIABP"/>
   </def:ItemType>
   <def:ItemType OID="IT.FRMSIZE" Mandatory="No" OrderNumber="2">
      <def:WhereClauseRef WhereClauseOID="WC.IT.FRMSIZE"/>
   </def:ItemType>
   <def:ItemType OID="IT.HEIGHT" Mandatory="No" OrderNumber="3">
      <def:WhereClauseRef WhereClauseOID="WC.IT.HEIGHT"/>
   </def:ItemType>
   <def:ItemType OID="IT.SYSBP" Mandatory="No" OrderNumber="4">
      <def:WhereClauseRef WhereClauseOID="WC.IT.SYSBP"/>
   </def:ItemType>
   <def:ItemType OID="IT.WEIGHT" Mandatory="No" OrderNumber="5">
      <def:WhereClauseRef WhereClauseOID="WC.IT.WEIGHT"/>
   </def:ItemType>
</def:ValueListDef>
<def:WhereClauseDef OID="WC.IT.DIABP">
   <RangeCheck Comparator="EQ" SoftHard="Soft" def:ItemType OID="VS.VSTESTCD">
      <CheckValue>DIABP</CheckValue>
   </RangeCheck>
</def:WhereClauseDef>
<def:WhereClauseDef OID="WC.IT.FRMSIZE">
   <RangeCheck Comparator="EQ" SoftHard="Soft" def:ItemType OID="VS.VSTESTCD">
      <CheckValue>FRMSIZE</CheckValue>
   </RangeCheck>
</def:WhereClauseDef>
```
One nicely sees that it is stated that for DIABP, an integer is expected, whereas for FRMSIZE, a reference is made to the codelist CL.SIZE, listing the enumerations "S" (small), "M" (medium), and "L" (large) and "XL" (extra large):

```xml
<CodeList DataType="text" Name="Frame size" OID="CL.FRAMESIZE">
  <EnumeratedItem CodedValue="S"/>
  <EnumeratedItem CodedValue="M"/>
  <EnumeratedItem CodedValue="L"/>
  <EnumeratedItem CodedValue="XL"/>
</CodeList>
```
Creating a ValueList for VSORRESU

Where creating a valuelist for VSORRES (vital signs original result) was pretty simple, we will make us ourselves a lot more difficult for VSORRESU (vital signs original results units). We will define that in case the weight of the subject was captured at a site in the USA, then the unit in VSORRESU is "pounds" whereas when the weight was captured at a site in Germany or France, the unit used is "kilograms". We will also state that the unit used is "mmHg" when the test code is either DIABP or SYSBP.

Also here, we can start from an existing codelist and then transform that to a valuelist using the menu "Insert – Create new ValueList from existing CodeList":

Remark that the choice of the codelist may depend on which version of controlled terminology you use: CDISC has not been very consequent here. This time we select the "Unit for Vital Signs Results" codelist, which is then transformed to a ValueList:

Renaming it to "VL.VSORRESU" and removing all unnecessary rows:
Of course we can also alter the OID and Name of the items, e.g.:

<table>
<thead>
<tr>
<th>OID</th>
<th>Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT:WEIGHT.EUROPE</td>
<td>Weight Units</td>
<td></td>
</tr>
<tr>
<td>IT:WEIGHT.USA</td>
<td>Weight Units</td>
<td></td>
</tr>
<tr>
<td>IT:BLOODPRESSURE</td>
<td>Blood pressure Units</td>
<td></td>
</tr>
</tbody>
</table>

as the unit itself is given, the data type is "text" in all cases, and again, we can state that the origin is a specific page of the CRF. In this case, it could however also "protocol defined", as the source might simply have been the protocol stating "weight must be measured in pounds in the US, and in kilograms in France and Germany". We will assume the latter:

Also having assigned a short description. Now we still need to define under which conditions each of these should be used ("Where Clause"). For "blood pressure" we need to state that mmHg needs to be used when VSTESTCD=DIABP or VSTESTCD=DIABP. This is pretty easy to define:

We cannot simply state VSTESTCD=DIABP as we also need to state "or VSTESTCD=SYSBP". The simplest way we can do this is to select the comparator "IN", meaning "in the set of ..":

```sql
SELECT * FROM table WHERE VSTESTCD IN ('DIABP', 'SYSBP');
```
leading to a somewhat different screen:

On the right, we can now use items, like "DIABP" and "SYSBP" and selecting VS.VTESTCD in the middle ("ItemOID"). Using the "Add to list" button, this leads to:

where we have stated that "the condition for using mmHg is that the value of VTESTCD is in the set \{"SYSBP","DIABP"\}", which is exactly what we want.

Now there is one important thing that we forgot: "what is the value that is expected" in VSORRESU itself? The only allowed one is "mmHg", so we need to generate a CodeList with only one value which is "mmHg".
We will leave the ValueList definitions for a moment, click OK until we are back in the main screen, and generate a new CodeList only containing "mmHg". To do this use the menu "Insert – Create new CodeList from existing CodeList":

![Image of the create new CodeList dialog box]

and select CL.VSRESU for which we will make a sub-codelist:

![Image of the create new sub-codelist dialog box]

which then displays the full list of units available for vital signs.

We now need to remove all unnecessary rows until we only keep "mmHg"\(^5\), and give the new codelist the OID "CL.BLOODPRESSUREUNITS", as it keeps all the units (well, only one in this case) for measurements of blood pressure:

\(^5\)The corresponding international UCUM unit is however "mm[Hg]"
There will be other cases where the sub-codelist can contain more than one entry. Suppose for example that the US-based investigators always report "HEIGHT" in "inches", whereas some of their colleagues report in centimeters and others in meters. In Europe, these investigators have tickboxes for "m" and "cm" on the CRF, from which they can choose. As such, there is no way that we can define a "WhereClause" for distinguishing between "cm" and "m" as it depends on the investigators habit (and may of the mood of the day...).

We can now go back to the ValueList VL.VSORRESU that we were editing and now assign the CodeList CL.BLOODPRESSUREUNITS to the item "Blood pressure units" (IT.BLOODPRESSURE):

The "Where Clause" was already added stating that the condition is VSTESTC in "IN" the set of {"DIABP", "SYSBP"}.

We now still need to repeat this for the use of weight units in the USA (IT.WEIGHTUSA) and in Europe (IT.WEIGHTEUROPE"). The exact values for the OIDs do not care, as long as we choose the correct attributes such as data type, origin, codelist, and then add a well-defined "WhereClause".

For the item "IT.WEIGHTEUROPE" we once again need to create a codelist which only contains a single entry "kg" (kilograms), e.g.:

---

6 But choosing something meaningful may help in doing selections later, especially when there are many of them. OIDs are fully arbitrary ...
and a similar one but with different contents for "Weight units in the USA":

and then assign them to their corresponding items in the valuelist:

All we now still need to do is to assign a "WhereClause" for both. Let's start with "Weight units in Europe". The first condition is of course that VSTESTCD=WEIGHT, the second being that the country is either Germany or France. The "WhereClause" editor than looks like:
having two checks:

VSTESTCD=WEIGHT

and:

DM.COUNTRY has either the value "GER" (Germany) or "FRA" (France).

One can again use the button "Show 'Where' clause":

The "WhereClause" for weight units in the USA is somewhat simpler:
defining two conditions:
VSTESTCD=WEIGHT
and:
DM.COUNTRY=USA

In the underlying define.xml, this is reflected by:

```xml
<def:ValueListDef OID="VL.Y9000RESULT">
  <ItemRef ItemID="IT.WEIGHT.EUROPE" Mandatory="No" OrderNumber="1"/>
  <ItemRef ItemID="IT.WEIGHT.USA" Mandatory="No" OrderNumber="2"/>
  <ItemRef ItemID="IT.BLOODPRESSURE" Mandatory="No" OrderNumber="3"/>
  <def:WhereClauseDef OID="WC.IT.BMI" def:CommentOID="COM.WC.IT.BMI">
    <RangeCheck Comparator="EQ" SoftHard="Soft" def:ItemOID="VS.VSTESTCD">
      <CheckValue>BMI</CheckValue>
    </RangeCheck>
  </def:WhereClauseDef>
</def:ValueListDef>
```

and e.g. the "WhereClause" for units for weight in Germany and France:
and the corresponding comment:

Not a bad idea to check whether we really assigned our newly developed valuelist to VSORRESU:
It looks as we didn't, so we do it now using the checkbox "New ValueList OID":

which appears in the define.xml as:

```xml
<ItemDefDataType="text" Length="6" Name="VSORESU" OID="VL.VSORESU">
  <Description>
    <TranslatedText xml:lang="en">Original Units</TranslatedText>
  </Description>
  <def:ValueListRef ValueListOID="VL.VSORESU" />
</ItemDef>
```

and in the HTML view (use "View – View define.ml in browser"):

### Value Level Metadata

#### Value Level Metadata - Vital Signs [VSORESU]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Where</th>
<th>Type</th>
<th>Length / Display Format</th>
<th>Controlled Terms or Format</th>
<th>Origin</th>
<th>Derivation/Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIABP</td>
<td>VTESTSCD EQ DIABP</td>
<td>integer</td>
<td>3</td>
<td></td>
<td>CRF Page 25</td>
<td></td>
</tr>
<tr>
<td>FRMSIZE</td>
<td>VTESTSCD EQ FRMSIZE</td>
<td>text</td>
<td>0</td>
<td>[“S”, “M”, “L”, “XL”]</td>
<td>CRF Page 25</td>
<td></td>
</tr>
<tr>
<td>WEIGHT</td>
<td>VTESTSCD EQ WEIGHT</td>
<td>float</td>
<td>5</td>
<td></td>
<td>CRF Page 25</td>
<td></td>
</tr>
<tr>
<td>SYSBP</td>
<td>VTESTSCD EQ SYSBP</td>
<td>integer</td>
<td>3</td>
<td></td>
<td>CRF Page 25</td>
<td></td>
</tr>
</tbody>
</table>

This finalizes our tutorial about the use of "ValueList" and the "Where Clause". Once again, when using "Where Clause" in define.xml 2.0/2.1, you are not allowed to use nested value levels.
We also showed how easy it is using SDTM-ETL to define the "Where Clauses" in a user-friendly way using a graphical, easy-to-use dialog, and without any necessity to do any XML editing or to "fight" with "black box" worksheets and software.