HeD Template Definition Guide

The HeD editor relies on the notion of "template" (or "primitive" or "clause") to simplify the authoring process, providing common expression patterns which domain experts should be familiar with. The templates are internally defined using a dedicated ontology, but their initial definition can be provided using a simple spreadsheet. The editor will be able to consume the spreadsheet and generate the internal ontologies automatically.

The spreadsheet format is derived from the HeD UC-II template spreadsheet, and preserves the compatibility with the same. Like in the original specification, multiple rows are actually required to define a template, namely one for each attribute involved.
The columns which are consumed by the editor are as follows:

Template descriptors (needs to be repeated on each row belonging to the same template):

- **A** (Sequence ID): this column is used internally to establish a priority between templates, which results in the display order. The sequence number needs not be consecutive, but will be used to sort templates
- **E** (Template Category): categories are used to classify and group templates. A full ontology of admissible categories will be released later. Categories include the special values *Trigger*, *Condition* and *Action*. If a template is assigned to at least one of those categories, it will appear in the corresponding tab in the editor and will be used to create trigger, condition or action sentence expressions, respectively.
- **F** (Template Name): the name that will be used to display the template in the lists. It will also be used as the initial default value for any expression tentatively generated using the templates

Attribute constraint descriptors. An attribute constraint follows the structure:

attribute (*operation reference_value*)+

and is defined using the following columns:

- **H** (Attribute Constraint Label): the user-friendly name that will be used to denote the attribute in the editor UI
- **I** (Domain Model Class): the class which defines the attribute. At the moment, all attributes in a template must belong to the same class.
- **J** (Domain Model Property [chain]): the name of the attribute which will be constrained by the template. It must be a valid attribute of the parent class, as per the domain model which is being used (e.g. vMR)
- **K** (Cardinality): defined in the form (min..max). Setting the min cardinality to 0 will denote optional constraints, any other value will indicate a mandatory constraint. Setting the max cardinality to 1 will denote single constraints, while any other value will denote multiple constraints. Optional constraints can be omitted when the template is instantiated, while multiple constraints can be repeated. See the User Guide for additional details
- **M** (Default Value): allows to specify default values for the literal or the expression involved in the constraint. The defaults are specified as "key=value" pairs, separated by ";". Valid keys are the names of the elements of a literal of the appropriate datatype, in addition to the fixed values "*operation*" and "*expression*" (the latter used to get the reference value from a named expression).

- **O** (Data type): defines the type of the reference value involved in the constraint. It must be coherent with the default value (column M) and will influence the set of admissible operations. The values in the column follow the convention of the ISO datatypes (e.g. ST, INT, PQ, …), but they are mapped internally to the HeD literal expressions.
- **P** (Restrictions): additional restrictions other than default values can be specified here. At the moment, the only supported restriction applies to CD attributes. A pair "[CodeSystem]:[Code]" will force any actual value chosen by the user to be a subconcept of the one indicated here, according to the inheritance relationships defined in the CodeSystem. More restrictions will be added in the future.

Example 1 : "Patient's age greater than THRESHOLD"
In this case, only one attribute is involved: the attribute "age" (J) of the class "EvaluatedPerson" (I). The datatype is PQ – Physical Quantity (O). The template can involve several defaults (M): the operation is >= and it is reasonable to assume that the age will be specified in years. So, the default value string should be "operation=GreaterOrEqual;unit=years". The user will still be able to change the values during the authoring process, or keep the defaults if appropriate. Column (K) should finally be set to "1..1" to ensure that a single value is provided.

Example 2 : "Patient with Diagnosis of SPECIFIC ACUTE HEART PROBLEM"

In this second example, we can map the template to the domain class "Problem" (I) and its attribute "problemCode" (J). Assuming that the user will want to pick a specific coded value, one can set the datatype to "CD" (O). "Equal" is the always the default operation, so there is no need to specify that, but one might want to restrict the selection of the codes to SNOMED-CT. To do so, column M should be set to "codeSystem=SNOMED" (or whatever ID is used in the CTS-II server to denote SNOMED-CT). To further ensure that the chosen code belong to the appropriate position in the taxonomy, column P should be set to "SNOMED:127337006". Columns (E), (F), (H) can be used to control the labels displayed on the UI.

In the future, we will try to provide a template authoring tool to facilitate the creation of the templates.