Restrictions of Classes for the Purpose of a Functional View

Knutholmen Sprint | 23–27 May 2016

**Goal:** How to Define a Subset of the Model Library for a Functional View?

**Authors:** Achim Wackerow and Oliver Hopt

# Introduction

A Functional View like Codebook is a subset of the Model Library. On an abstract level, this means it is a subset of the definitions of the Model Library which are useful for the Codebook Functional View. On the UML level, this is a subset of classes. In the first place, it would also be good to have a subset of properties of a class which are used for a Functional View. But there are a couple or reasons which seem to stay against this.

1. UML seems to have no mechanism to define a subset of properties of classes. The latest version of UML, 2.5, may have some ways to address this. This should be more explored. Dan Gillman will ask people from OMG (Elisa Kendall), if there are possibilities in UML 2.5.
2. It would be conceivable to transform the used parts (for a specific Functional View) of the Model Library to another (restricted) model. The subsetting of classes and properties could be defined as transformation rules.
3. Independent of the way of subsetting the Model Library (1. or 2.), the result would be a model per Functional View. A restricted set of properties of a class would result in a new class (like InstanceVariable\_FV\_Codebook derived from Instance\_Variable). This means a lot of additional models and classes would be created by this approach. This seems to be hard to manage on the model level and would result in a huge amount of classes on the implementation level (bindings).
4. Another way was mentioned on a high level to address defining classes for Functional Views: starting from simple classes and extending classes for different purposes (Functional Views). This approach is loaned from object-oriented program languages and would turn the current setup in the model library upside down. Anyway, it would result again in a lot of models and classes which seem to be hard to manage. This problem is not addressed by this approach.

A major requirement is that instances are interoperable between Functional Views, i.e. that the common set of metadata of multiple Functional Views can be processed by software applications which are using these Functional Views.

Similar applies to relationships from classes used by a specific Functional View to classes not used by this Functional View.

One way to go forward would be to use the current approach of using a selected set of classes of the Model Library for a Functional View with the full set of properties of each class. The documentation could describe which subset of each class is used for a specific Functional View. For automatic generation of this documentation it would be required to define in a formal way which properties of each class are used for a Functional View and which not. This could be used as configuration for a creation/transformation process of the documentation files (Markdown). This approach is in the spirit of the document on “How to Create a Functional View”.

Proposed additions to Drupal: The “Edit View” of a Functional View provides the indication of the referenced classes. A separate page per referenced class could provide the indication of used properties of the specific class.

The captured information on used properties per class can be exported in a configuration file which could be consumed by the creation/transformation process of the documentation of a specific Functional View. The generated documentation should only mention the used classes and used properties per class.

# Implementation in Drupal

In Drupal, there will be a need to change the form for views. The list of all classes with checkboxes would have to be replaced by a mechanism quite similar to the adding of relationships to classes. The following screen fake shows it in its final state. The lists of used properties and relationships should be filled after choosing the class to be added. If possible, it would be nice to show the properties/relationships cardinalities together with it name.

