Communications Document: DDI Moving Forward Modelling Topics

# I. Overview

This document is intended to help the content modeling teams to consistently model some parts of the views they are producing. DDI 4 now has some abstract models covering specific areas, intended to be extended and specialized in particular views. This includes the “Collection” model, which will be used as the basis for grouping metadata items for various purposes. Another area is in the description of process flows, where a very generic process model has been created.

We will look at examples in both these areas, to show how the views should intersect with the abstract constructions held in the DDI 4 Object Library.

# II. Identification

# III. Names, Labels, and Descriptions

# IV. Collections

# V. Processes

The process model in DDI 4 provides a number of generically useful objects: the Process Step object can be used to describe processes at any level, and is subclassed into a set of objects which can be used to describe logical flows. The Control Construct object is a specialization of the Process Step for this purpose, and it has in turn a number of specific objects which extend Control Construct: Sequence, Repeat While, Repeat Until, Loop, If Then Else, and Act. Of these, Act is the use of a metadata item within the flow, but not as an input or output (Act covers questions, statements, instructions, etc.)

Each of the Control Constructs can also act as aggregations of other Control Constructs, although that is not shown in the diagram. This provides a wealth of extension points for use in other models within DDI 4.



Process steps have Inputs and Outputs, and are themselves composed of other Process Steps. There is a specialized type of Process Step which is the Service which performs a Process Step. This specialized type has an Agent, which is the individual, organization, or machine which is performing the Service.

The DDI 4 process model is a generic, abstract one, which can be used for many purposes: to describe how a process *should* be performed, to provide an historical description of a process as it *was* performed, etc.

The additional semantics around what the process model is to be used for are added when the Process Step or one of its sub-types is extended for use in a view. For example, when describing a Simple Instrument, the Control Construct object is extended for explaining the flow logic within a data collection process. This is not an historical description, but instead is providing the pattern in which the data collection is to be conducted.

In order to incorporate the Process Step into their description of the flow of data collection, the Simple Instrument uses an object specific to data collection as an extension of Control Construct – Instrument Component. This extended object provides the semantics needed to explain how the abstract process model will be used.



Here, we see the extension of Control Construct into Instrument Component, which is itself extended by Capture, an abstract class which is extended by the subclasses Measurement (for data coming from a device) and Question (for data coming from a questionnaire).