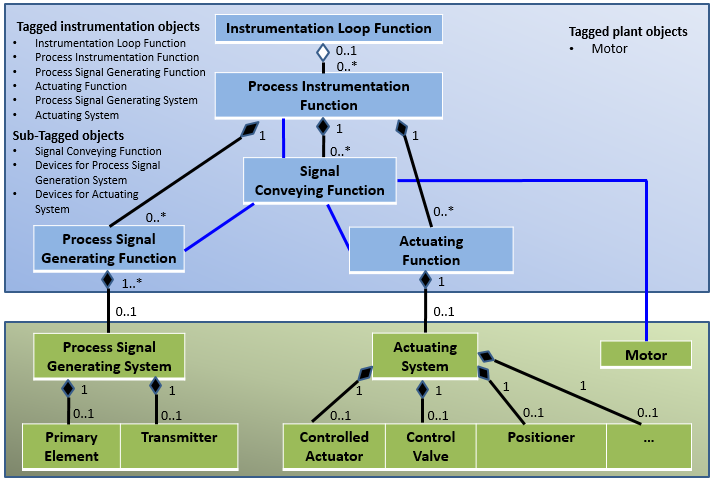
Extensions and Modifications of Proteus Schema for IIMM Instrumentation Model

|  |  |  |  |
| --- | --- | --- | --- |
| 0.1 | 2015-10-27 | Manfred Theißen, AixCAPE | initial version |
| 0.2 | 2016-01-18 | Manfred Theißen, AixCAPE | update for RDL2 terminology |
| 0.3 | 2016-01-28 | Manfred Theißen, AixCAPE | * option 4 for SignalLine chosen by IIMM (cf. version 0.2) |
| 0.4 | 2016-02-03 | Manfred Theißen, AixCAPE | * add ActuatingSystemComponent and ProcessSignalGeneratingSystemComponent * emphasize that the 4 logical start/end association types are new |

This document describes a proposal for extending Proteus schema 3.6.0 to 4.0.1 in order to support the new IIMM instrumentation model.



# New Element: <InstrumentationLoopFunction>

* top-level element in <PlantModel>
* <ProcessInstrumentationFunctions> in the <InstrumentationLoopFunction> are assigned via <Association> elements (cf. InstrumentLoop in Proteus 3.6.0)

Example:

<InstrumentationLoopFunction ID = "PCL1314" TagName = "1314">

<Association Type = "is a collection including" ItemID = "PCR\_1314L"/>

<Association Type = "is a collection including" ItemID = "PCR\_P1314C"/>

</InstrumentationLoopFunction>

# New Element: <ProcessInstrumentationFunction>

* top-level element in <PlantModel>, even if part of a <InstrumentationLoopFunction>
* <ProcessInstrumentationFunctions> are optionally assigned to a <ProcesControlLoop> via <Association> elements (cf. InstrumentLoop in Proteus 3.6.0; redundancy accepted for now for consistency with Proteus 3.6.0)
* ProcessControlFunction:
  + subclass of ProcessInstrumentationFunction
  + no schema class required (RDL reference sufficient)

Example:

<ProcessInstrumentationFunction ID = "PCR\_1314L" TagName = "P1314L">

<!--TODO: Graphics: bubble with inner deco -->

<!--TODO: Ports: probably need classifications -->

<Association Type = "is part of" ItemID = "PCL1314"/>

...

</ProcessInstrumentationFunction>

# New Elements: <ActuatingFunction>, <ProcessSignalGeneratingFunction>

* children of <ProcessInstrumentationFunction>
* <ActuatingSystem>/<ProcessSignalGeneratingSystem> are assigned via <Association>  
  (Multiplicity <ProcessSignalGeneratingSystem>-<ProcessSignalGeneratingFunction> is 1..\*. Hence, assigning <ProcessSignalGeneratingSystems> as XML children will not work in general.)

# SignalConveyingFunction

* The existing SignalLine with is replaced with a “generic” element (e.g., InformationFlow, but probably there are better names).
* RDL references are used to distinguish SignalConveyingFunction, SignalLine, etc.
* This results only in minor compatibiliy issues: Old files are invalid, but can be adapted easily.
* New association types required: has logical start, is logical start of, has logical end, is logical end of

Example:

< InformationFlow

ID="SL\_P1314C\_1"  
 ComponentClassURI="http://some.rdl.org/SignalConveyingFunction" >

<Association Type = "has logical start" ItemID = "PCR\_P1314C"/>

<Association Type = "has logical end" ItemID = "PCF\_U1313"/>

...

Future task:

Find a better name for “InformationFlow”

# New Elements: <ActuatingSystem>, <ProcessSignalGeneratingSystem>

* Top-level elements in <PlantModel>
* Assignment to <ActuatingFunction>/<ProcessSignalGeneratingFunction> via <Association> with type “fulfills”/”is fulfilled by”
* Assignment of parts via <Association> with type “is an assembly including” (optionally with RDF reference, e.g., to identify the “DetectingElement” or the “Transmitter” independent from the type of the referenced object)

<ProcessSignalGeneratingSystem ID = "SA\_PT1314C " TagName = "PT1314C">

<Association Type = "fulfills" ItemID = "SPCR\_PT1314C"/>

<Association Type = "is an assembly including" ItemID = "SPCR\_PT1314C"/>

</ProcessSignalGeneratingSystem>

# New Elements: <ActuatingSystemComponent>, <ProcessSignalGeneratingSystemComponent>

* Children of ActuatingSystem, ProcessSignalGeneratingSystem, respectively
* need “generic” approach (note ‘…’ in class diagram!)
* if applicable, a ActuatingSystemComponent/ProcessSignalGeneratingSystemComponent can have an Association (refers to/is referenced by) that links to an object that is already part of some other hierarchy in Proteus (e.g., a PipingComponent): In this way, the ActuatingSystemComponent/ProcessSignalGeneratingSystemComponent acts as a kind of RelationClass. Note that the relation class is required in general, because the relation itself can have attributes! Example: A PipingComponent acting as a ControlValve can have an attribute that is relevant only with respect to the role of the PipingComponent as a ControlValve.

# Graphics

* In principle, graphical representations for all objects above are optional; thus, PlantItem is an adequate XSD type for all of them
* Details of graphics depend on the underlying standard; they should be covered by the existing schema elements