

## Arrowhead Profile - Eclipse Papyrus Plugin

### Code to Models Consistency Update

#### About

- An Arrowhead profile has been built in the Eclipse Papyrus workspace<sup>1</sup> for (1) the modeling of physical local clouds and deployed entities and (2) the logical definitions of System Design Descriptions (SysDD) and Interface Design Descriptions (IDD)<sup>2</sup>. A plugin for the automatic generation of provider and consumer system's code from a UML model file based on this profile has been developed<sup>3</sup>.
- As an extension of this plugin, I have implemented an analysis of the generated and potentially edited code for the detection of newly defined or updated physical systems or definitions. The extension parses the model and code, when an addition or update has been detected, it modifies the UML file defining the papyrus model for maintaining a consistency between the model and the code<sup>4</sup>.
- Some assumptions of this experiment include:
  1. The potentially edited code can only have more physical systems or definitions, i.e. the removal has not been considered.
  2. The newly added systems (if any) are considered to be registered in the Service Registry.
  3. Although the definition of the IDD includes a differentiation between consumer and provider interfaces, it has not been considered.
  4. The RESTful operations defined in the interfaces include GET and POST.
  5. Communication protocol has only been defined as HTTP and encoding type as JSON.
  6. In the naming convention of the generated code, the name of the physical deployed entity is assumed to be defined in the application.properties file where as the name of the SysDD is given by the project folder that contains the code.
  7. The main file of the provider or consumer system is assumed to define the consuming services, where as the service controller file is assumed to define the providing services.
  8. A system that provides and consumes services is considered as a provider.

#### Results

- The extension identified a discrepancy in the generated code from the ModelstoCode repository where the *controlSystem* SysDD had two already defined interfaces added (*WindowMotorService* & *WindowOpeningService*) and the *controlSystem2* SysDD had one already defined interface added (*SystemConfigService*). This discrepancy was taken as the test basis for the UML updating step. The output of the identification can be seen in the last section.

---

<sup>1</sup><https://www.eclipse.org/papyrus/>

<sup>2</sup><https://github.com/eclipse-arrowhead/profile-library-sysml>

<sup>3</sup><https://github.com/CristinaPaniagua/ModelstoCode>

<sup>4</sup><https://github.com/fernand0labra/sysml-code-to-models>

- The original UML definition of both SysDDs can be seen respectively in figures 1 and 3.
- After the processing steps and UML update the resulting SysDDs can be seen in figures 2 and 4. One detail worth mentioning is that the ControlSystem SysDD defines two TempService ports but it should only define one, as a port is a connection to the interface, and the interfaces defines the various operations.

## Implementation Issues

- When creating or updating the ports that a certain SysDD has, the connections defined for a specific deployed entity are erased. This information could not be rebuilt or parsed as it has to be obtained from the Service Registry itself and that behavior is not yet implemented.
- The edition of the UML file does not imply its update in the visual environment as this is defined in the .notation XML file. When a new PackageableElement containing a physical system or logical definitions is added, the reference of the identifier in the notation cannot be found and therefore is not displayed. However, the internal logic can be accessed through the workspace.
- Papyrus Workspace does not include the generation of the .di and .notation files from the UML definition, so it would be necessary to update both of them in order to display the new added physical systems or definitions.

## Other Information

- The output for the model and workspace parsing, as well as that of the identification of discrepancies can be found under */results/*.
- The documentation of the code can be accessed under */doc/index.html*.
- An update on the ModelsToCode repository is needed in order to perform a more extended testing e.g. addition and update of other elements besides the SysDD. An important aspect to be discussed is the naming convention of the generated folders, files and code.
- The Arrowhead profile example used in the papyrus workspace is the studio4education project.

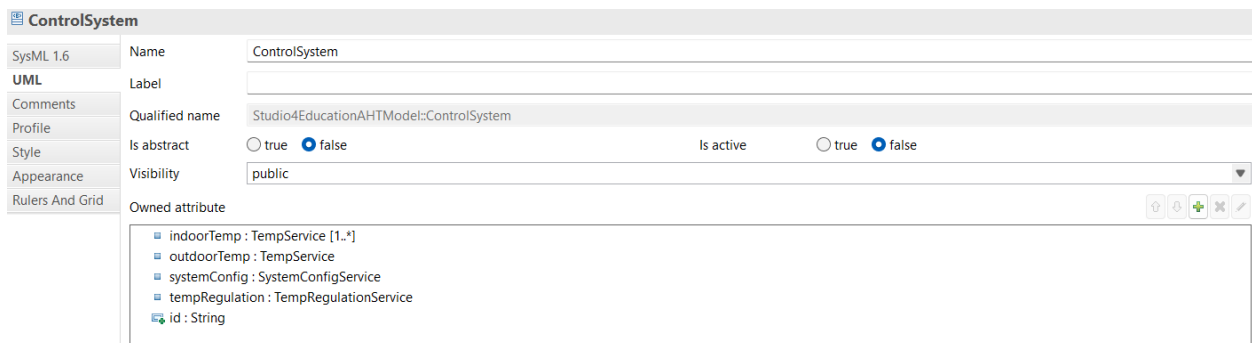


Figure 1: The ControlSystem SysDD previous to the consistency update

ControlSystem			
SysML 1.6	Name	ControlSystem	
UML	Label		
Comments	Qualified name	Studio4EducationAHTModel::ControlSystem	
Profile	Is abstract	<input type="radio"/> true <input checked="" type="radio"/> false	Is active <input type="radio"/> true <input checked="" type="radio"/> false
Style	Visibility	public	
Appearance			
Rulers And Grid	Owned attribute	<div> <input type="checkbox"/> systemConfigService : SystemConfigService             <input checked="" type="checkbox"/> id : String             <input type="checkbox"/> tempRegulationService : TempRegulationService             <input type="checkbox"/> tempService : TempService             <input type="checkbox"/> windowMotorService : WindowMotorService             <input type="checkbox"/> windowOpeningService : WindowOpeningService         </div>	

Figure 2: The ControlSystem SysDD latter to the consistency update

ControlSystem2			
SysML 1.6	Name	ControlSystem2	
UML	Label		
Comments	Qualified name	Studio4EducationAHTModel::ControlSystem2	
Profile	Is abstract	<input type="radio"/> true <input checked="" type="radio"/> false	Is active <input type="radio"/> true <input checked="" type="radio"/> false
Style	Visibility	protected	
Appearance			
Rulers And Grid	Owned attribute	<div> <input checked="" type="checkbox"/> id : String             <input type="checkbox"/> indoorTemp : TempService             <input type="checkbox"/> windowOpeningService : WindowOpeningService             <input type="checkbox"/> windowMotorService : WindowMotorService         </div>	

Figure 3: The ControlSystem2 SysDD previous to the consistency update

ControlSystem2			
SysML 1.6	Name	ControlSystem2	
UML	Label		
Comments	Qualified name	Studio4EducationAHTModel::ControlSystem2	
Profile	Is abstract	<input type="radio"/> true <input checked="" type="radio"/> false	Is active <input type="radio"/> true <input checked="" type="radio"/> false
Style	Visibility	protected	
Appearance			
Rulers And Grid	Owned attribute	<div> <input type="checkbox"/> systemConfigService : SystemConfigService             <input checked="" type="checkbox"/> id : String             <input type="checkbox"/> tempService : TempService             <input type="checkbox"/> windowMotorService : WindowMotorService             <input type="checkbox"/> windowOpeningService : WindowOpeningService         </div>	

Figure 4: The ControlSystem2 SysDD latter to the consistency update

## Identification of Discrepancies at the SysDD Level

updateSystemDesignDescription      ControlSystem

    WindowMotorService

        Role: Consumer

        Protocol: HTTP

        Encoding: JSON

        Operations:

            openWindow

                Method: Get

                Response Type: String

            closeWindow

                Method: Get

                Response Type: String

updateSystemDesignDescription      ControlSystem

    WindowOpeningService

        Role: Consumer

        Protocol: HTTP

        Encoding: JSON

        Operations:

            getWindowStatus

                Method: Get

                Response Type: String

updateSystemDesignDescription      ControlSystem2

    SystemConfigService

        Role: Consumer

        Protocol: HTTP

        Encoding: JSON

        Operations:

            setSystemMode

                Method: Post

                Request Type: SetSystemModeRequestDTO

                Response Type: String