# *Developers guide on how to further develop the code for the CIM 2 MODelica Transformation Tool*

1. Deploy project into Eclipse IDE (current version Eclipse Neon 4.6.3, JAVA 1.8) (we suppose Eclipse and JAVA to be installed into the system)
2. Download the source code from the repository (<https://github.com/ALSETLab/cim2modelica)> into a desired folder
3. Open Eclipse and Import the project into the current workspace: File> Import…> Existing project into Workspace. Browse on the Select root directory field and select the project from the list of Projects:
4. Under the folder res.map there is a folder with XML and XSD files with mapping rules for the OpenIPSL components. To comply with the package organization of the Modelica Library, the folder res.map.openipsl is organized in the same way (e.g. the folder res.map.openipsl.controls.es contains mapping rules for Excitation System components). To add a new mapping component rule, follow the next steps:
5. Copy the files .dtd, .xml, and .xsd with the new name. Place the new files into the corresponding folder
6. Modify the .dtd changing the name the component <!ELEMENT and <!ATTRIBUTE tags(e.g. <!ELEMENT newComponentMap)
7. Modify the .xsd changing the name of the main component <xsd:element tag (e.g. <xs:element name="newComponentMap">
8. Modify the .xml changing the name of the main tag, and the values of the tag attributes (you can leave the tags rdf\_id and rdf\_resources empty)
9. Modify the .xml changing the values from the <attributeMap> tag. Add as many <attributeMap> tags as variables of the openipsl component need to map
10. Execute the MappingClassStructure.jar with one input: the name relative path of the .xsd file (e.g. MappingClassStructure.jar ./res/map/openipsl/controls/es/cim\_openipsl\_esdc1a.xsd
11. If developing under Eclipse, create a New Configuration to run JAVA code. Indicate the main class and the .xsd file as attribute:
12. Run Configurations> Java Application> New Launch Configuration
13. Main Class: cim2model.MapStructureGenerator
14. Arguments> Program Arguments: ./res/map/openipsl/controls/es/cim\_openipsl\_esdc1a.xsd
15. Under the package cim2model.cim.map, the java classes SpecificComponentMap.java, AttributeMap.java and ObjectFactory.java have been created. To have a good organization of the classes, under the software architecture, follow this final steps:
16. Move the new *SpecificComponent*Map.java into the corresponding package. Modify the header of the class to extend from the *ComponentMap* class, that is under the cim2model.cim.map package. If no additional attribute is defined for the new *SpecificComponent* mapping rules, leave the new class empty.

@XmlRootElement(name = "eSDC1AMap")

public class ESDC1AMap extends ComponentMap {

}

1. Within the same package, create a new function in the available class *SpecificComponent*MapFactory.java file. Name the new function with the corresponding *SpecificComponent* name.
2. Modifiy this new function to create and return a class of the new *SpecificComponent* type. This functions are responsible to *unmarshall* the data from the .XML mapping rule files to the java class of the same name. (The process of *unmarshall* is specific from the JAXB library, for loading XML schemas into its equivalent JAVA class). This is used later to store the values from the CIM Model.
3. Add new methods to use this new mapping rules and build the corresponding OpenIPSL component based on this rule.
   1. Within the cim2model.cim.map.ModelDesigner class create a new method create\_SpecificComponentModelicaMap
   2. Within the cim2model.modelica.ModelBuilder class create a new method create\_SpecificComponent
   3. In case of new machine component or new controls add a new case on the corresponding method of the main class: *public static OpenIPSLMachine factory\_Machine()* for machine components or *public static voide factory\_Plant()* for excitation systems, turbine governors or stabilizers
   4. In case of other components, go to the available cases and substitute the name of the mapping file.

Note: Currently working on the automation of functionalities described on point VI.c and VI.d

# B- Required Metadata

# B1 Current executable software version

# *Ancillary data table required for sub version of the executable software: (x.1, x.2 etc.) kindly replace examples in right column with the correct information about your executables, and leave the left columns as they are*

# *Table 1 – Software metadata*

|  |  |  |
| --- | --- | --- |
| **Nr** | **(executable) Software metadata description** | ***Please fill in this column*** |
| S1 | Current software version | *for example 1.1, 2.4 etc.* |
| S2 | Permanent link to executables of this version | *example : https://github.com/combogenomics/DuctApe/releases/tag/DuctApe-0.16.4* |
| S3 | Legal Software License | *List one of the approved licenses* |
| S4 | Computing platform / Operating System | *for example Android, BSD, iOS, Linux, OS X, Microsoft Windows, Unix-like , IBM z/OS, distributed / web based etc.* |
| S5 | Installation requirements & dependencies |  |
| S6 | If available Link to user manual - if formally published include a reference to the publication in the reference list | *Example http://mozart.github.io/documentation/ or* |
| S6 | Support email for questions |  |

# B2 Current code version

# *Ancillary data table required for subversion of the codebase. Kindly replace examples in right column with the correct information about your current code, and leave the left columns as they are*

# *Table 2 – Code metadata*

|  |  |  |
| --- | --- | --- |
| **Nr** | **Code metadata description** | ***Please fill in this column*** |
| C1 | Current Code version | *For example v42* |
| C2 | Permanent link to code / repository used of this code version | *For example : https://github.com/mozart/mozart2* |
| C3 | Legal Code License | *List one of the approved licenses* |
| C4 | Code Versioning system used | *For example svn, git, mercurial, etc. put none if none* |
| C5 | Software Code Language used | *For example c++, python , r, etc.* |
| C6 | Compilation requirements, Operating environments & dependencies |  |
| C7 | If available Link to developer documentation / manual | *For example : http://mozart.github.io/documentation/* |
| C8 | Support email for questions |  |