

```
if (firstElement instanceof IProject)
{...}
```

Strange
Start

Like how the `IVdmProject` is derived from the `IProject` class, the `IVdmProject` class likewise contain a method for creating an instance of the `IVdmModel` class which contain information about the abstract structure of the VDM model. This model is then used when examining if the model is syntax and parse correct and therefore fit for the UML transformation. The code snippet below shows how the `IVdmModel` class is instantiated and used to check the correctness of the model.

```
final IVdmModel model = vdmProject.getModel();
if (model.isParseCorrect())
{
    if (model == null || !model.isTypeCorrect())
    {...}
```

Considering the goal of migrating the UML connection to VDM VSCode, the handlers use of the project classes poses a problem, since their usefulness is dependant on the Eclipse IDE. This stems from the fact that the input to ...?

One way to solve this could be to extract information about the current VS Code project, create an intermediate Eclipse project, and then perform the transformation like `Vdm2UmlCommand`. This however runs into the problem of having to depend on the IDE side as discussed in 5.

dealing?

To circumvent this, one might also simply remove the notion of an `IProject` and its derived types. It would then require finding new ways of providing the same functionality as these classes provide. An overview of the Eclipse project classes and the methods used in the old `Vdm2Uml` handler, is illustrated below.

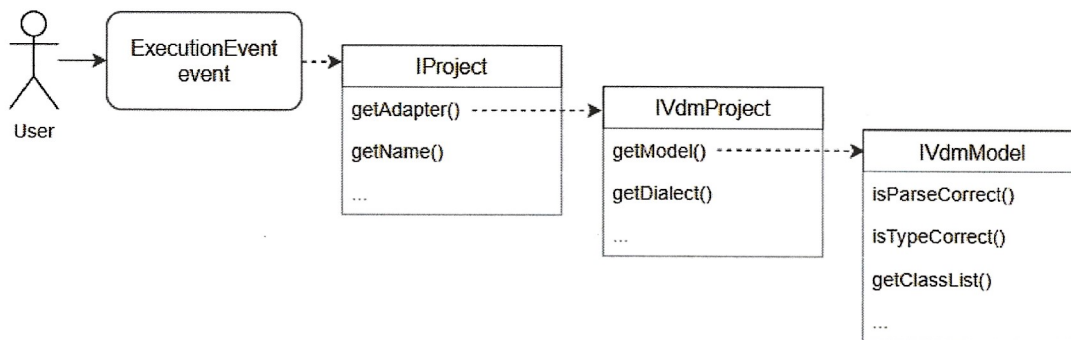


Figure 10: Overview of `IProject` its derived classes, and the methods used in the old `Vdm2Uml` handler