

CONSUMING an EMF MODEL in UR ECLIPSE UI

PART 2 – Using the Editing Domain and Command Stack

Having learnt how to use the EMF Model and its Item Providers to display the model on a TreeViewer in the previous article, I felt that Part 2 should discuss the Editing Domain and the Command Stack.

EMF claims to support unlimited Undo and Redo Operations. The background behind that claim is the support of Editing Domain and Command Stack. These are 2 powerful features of EMF not discussed too often in the EMF Articles.

In this article, I document the basic understanding of Editing Domain and Command Stack. Editing Domain and Command Stack are implementation features of Command Pattern.

Every Command Pattern needs to know under which scope it needs to operate. We could treat Editor or a View as a Scope. Any changes happening within the Editor/View should be tracked and supported with UNDO and REDO. This definition of scope in EMF is called as Editing Domain.

Every Command being performed needs to be dumped into a Command Stack, so that during UNDO and REDO, these commands can be popped out and re-executed to perform respective operations.

```
BasicCommandStack commandStack = new BasicCommandStack();
AdapterFactoryEditingDomain editingDomain = new AdapterFactoryEditingDomain(
    new CdiagramItemProviderAdapterFactory(), commandStack);
```

Fig 1. Snippet of Defining Command Stack and Editing Domain

Now having initialized the Command Stack and the Editing Domain, all the Operations on model should be done via a Command and the command should be executed within the Command Stack of the Editing Domain.

EMF provides a set of Commands / Operations for e.g. ADDCommand, RemoveCommand, SetCommand etc. Below documented is a sample use of AddCommand.

```
Action createUClass = new Action("Create Class") {
    @Override
    public void run() {
        UClass uClass = CdiagramFactory.eINSTANCE.createUClass();
        uClass.setName("Sample Class");
        AddCommand addCommand = new AddCommand(ed, cDiagram.getClasses(), uClass);
        ed.getCommandStack().execute(addCommand);
    }
};
```

Fig 2. Snippet for Add Command

Below documented is a sample use of UndoCommand

```
Action undoAction = new Action("Undo Action") {  
    public void run() {  
        ed.getCommandStack().undo();  
    }  
};
```

Fig.3 Snipped for Undo Command

Similarly you would use RemoveCommand, CopyCommand, SetCommand etc.

Please download the example code snippet from our website www.ancitconsulting.com