Creating Heavyweight instances

1. **How to create CrossCuttingConcern**

To create a package like this “**Package** **Persistence;**” you have to create this following source code:

1. **How to create Aspect**

To create an aspect like this “**public aspect Logging{}**” you have to create this following source code:

If your aspect is inserted in a package you have to bound it the corresponding package, like this:

If you want to add some of its properties, you just have choose the property and set the value, as you can see in the source code below:

1. **How to create Pointcut with one Joinpoint**

To create a pointcut like this “**public** **pointcut** **figure():** **call** **(move());**”you have to create this following source code:

1. **How to create Pointcut with more than one Joinpoint**

To create a pointcut like this:

“**public pointcut move(): execution (getID()) || call (getName());**” You have to create this following source code:

First you have to create the “*CompositePointCutUnit”* and then you have to set the property “*name”* and “*compositionType”* . Notice that you have to set the name ONLY in the “*CompositePointCutUnit”* element.

The next step is to create the joinpoints and for this you have to create the respective element that you want to represent. In this example we have two joinpoints, so the following source code represents these two joinpoints.

To set correctly a joinpoint you have to set the properties “*Operation*” and “*Composite*”, the “*Operation”* is the method affected by the joinpoint and the “*Composite”* is the “*CompositePointCutUnit”* that the joinpoints belongs.

1. **How to create Advice**

To create an advice like this: “**before() : move() {// Some JAVA souce code}**” You have to create this following source code:

The main properties that you have to set while creating an Advice are “*pointCut”* (which specifies the pointcut that it belongs) and the “*adviceExecutionType”* (that specifies the behavior of the advice).

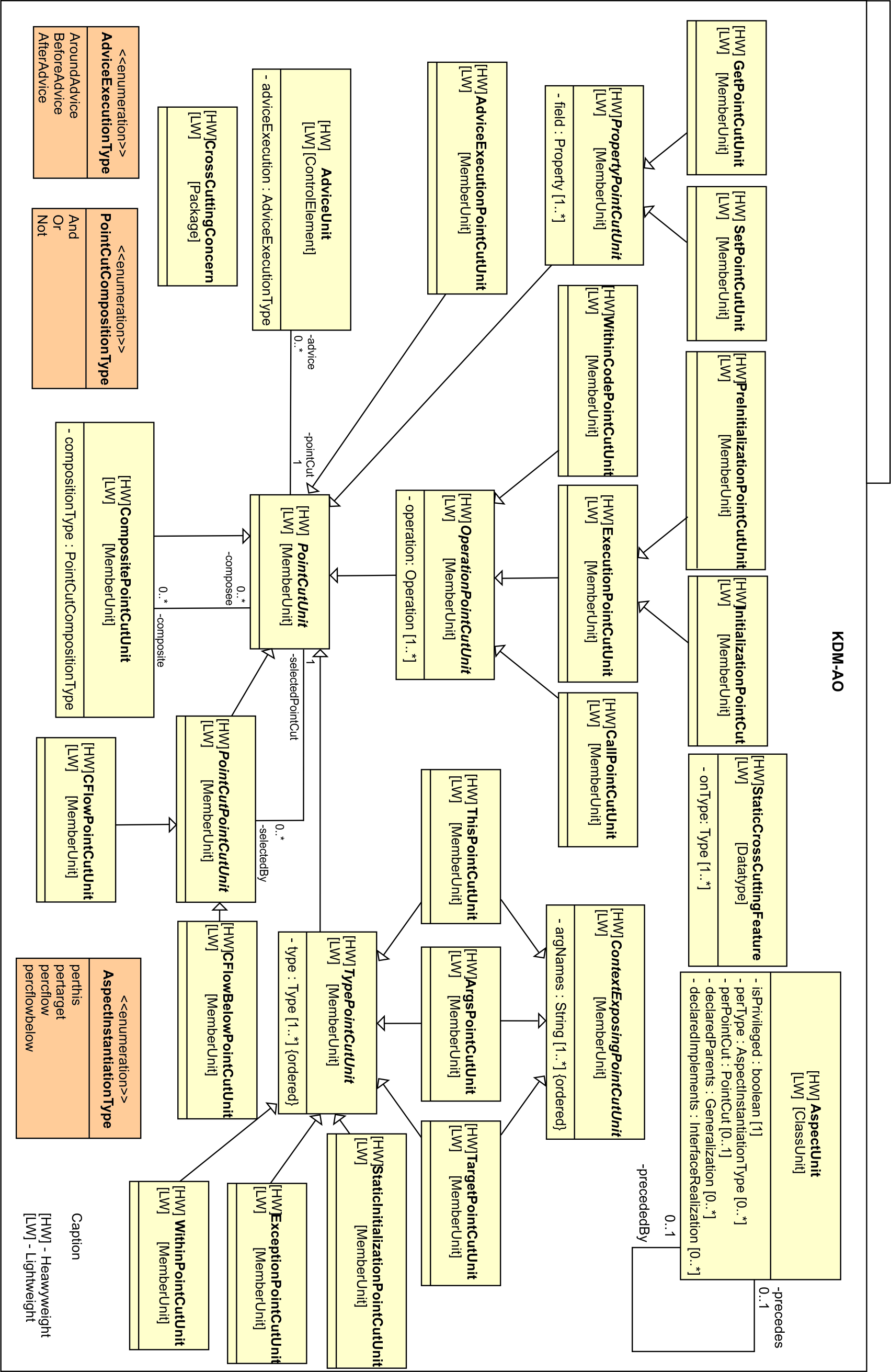
1. **How to create Inter-type declaration**

To create an advice like this:

“**public PersistentRoot.name;**

**public void PersistentRoot.move();**” You have to create this following source code:

Notice that you have to create a “*StaticCrossCuttingFeature”* element and set the “*OnType*” property. After creating the “*StorableUnit”* and the “*MethodUnit”* you have to bound them into the “*StaticCrossCuttingFeature”*.



|  |  |
| --- | --- |
| **AspectJ Profile Elements** | **KDM-AO elements** |
| Aspect | AspectUnit |
| PointCut | PointCutUnit |
| CompositePointCut | CompositePointCutUnit |
| OperationPointCut | OperationPointCutUnit |
| WithinCodePointCut | WithinCodePointCutUnit |
| ExecutionPointCut | ExecutionPointCutUnit |
| CallPointCut | CallPointCutUnit |
| PreInitializationPointCut | PreInitializationPointCutUnit |
| InitializationPointCut | InitializationPointCutUnit |
| PropertyPointCut | PropertyPointCutUnit |
| GetPointCut | GetPointCutUnit |
| SetPointCut | SetPointCutUnit |
| AdviceExecutionPointCut | AdviceExecutionPointCutUnit |
| PointCutPointCut | PointCutPointCutUnit |
| CFlowPointCut | CFlowPointCutUnit |
| CFlowBelowPointCut | CFlowBelowPointCutUnit |
| TypePointCut | TypePointCutUnit |
| WithinPointCut | WithinPointCutUnit |
| ExceptionPointCut | ExceptionPointCutUnit |
| StaticInitializationPointCut | StaticInitializationPointCutUnit |
| TargetPointCut | TargetPointCutUnit |
| ArgsPointCut | ArgsPointCutUnit |
| ThisPointCut | ThisPointCutUnit |
| ContextExposingPointCuit | ContextExposingPointCuitUnit |
| Package | CrossCuttingConcern |
| AdviceExecutionPointCut | setAdviceExecution |
| PointCutCompositionType | setPointCutCompositionType |
| AspectInstantiationType | setAspectInstantiationType |
| Intertype Declaration | StaticCrossCuttingFeature |
| Advice | AdviceUnit |