copy

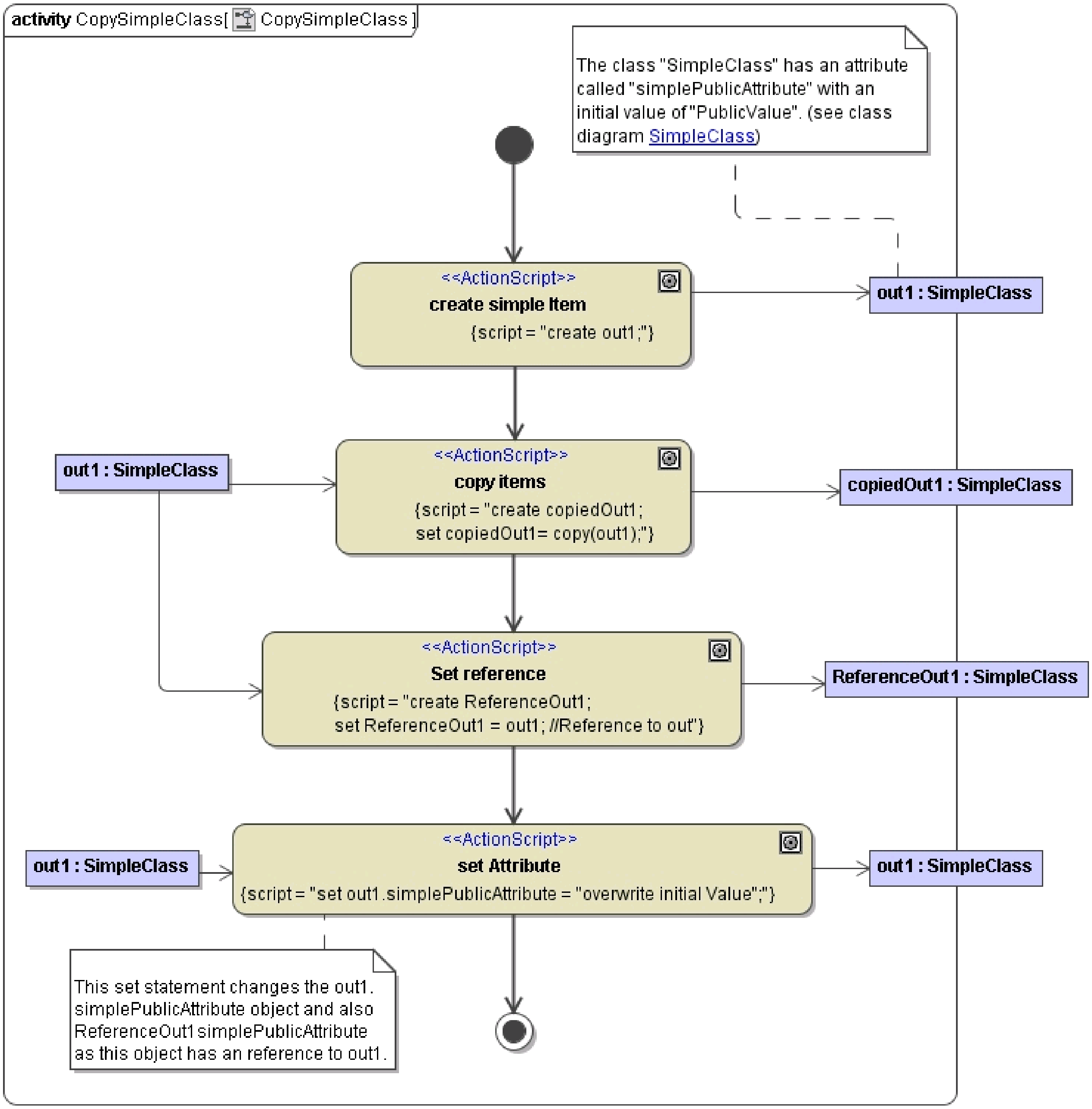
Syntax	`set aClonedObject = anObject.copy()`	
Semantics	An independent copy of the object will be created. The Bridge distinguishes between referencing and copying object nodes. If you have two objects of same type, for example `item1` and `item2`, you can write `set item2 = item1` thus assigning `item2` a reference to `item1`. This means, if you later on modify any attribute values of `item1`, object `item2` will also have these changed values. Contrary to this situation, if you write `set item2 = item1.copy()`, changes in the state of `item1` will not effect `item2` (see The Essential Thing: Object References).	
Substitutables	`anObject`	Any object.
Examples	`set cloneOfA = a.copy();`	

Example File (Builder project Basic Modeling/Data):

⤓	<your example path>\Basic Modeling\Data\uml\copy.xml

The following activity diagram illustrates the difference between copying and referencing items.

Figure: Copy Object Example


activity CopySimpleClass[CopySimpleClass]
The class "SimpleClass" has an attribute called "simplePublicAttribute" with an initial value of "PublicValue". (see class diagram SimpleClass)
<<ActionScript>>
create simple Item
{script = "create out1;"}
out1 : SimpleClass
out1 : SimpleClass
<<ActionScript>>
copy items
{script = "create copiedOut1;
set copiedOut1= copy(out1);"}
copiedOut1 : SimpleClass
<<ActionScript>>
Set reference
{script = "create ReferenceOut1;
set ReferenceOut1 = out1; //Reference to out"}
ReferenceOut1 : SimpleClass
out1 : SimpleClass
<<ActionScript>>
set Attribute
{script = "set out1.simplePublicAttribute = "overwrite initial Value";"}
out1 : SimpleClass
This set statement changes the out1. simplePublicAttribute object and also ReferenceOut1 simplePublicAttribute as this object has an reference to out1.

The first step creates an item called **out1** and uses the initial value, which is set on the class definition for the attribute **simplePublicAttribute**. The second step creates a copy of **out1** which has also this initial value for **simplePublicAttribute**. The third step creates a reference to **out1** called **ReferenceOut1** having also this initial value.

At this point all items have the same value for the attribute **simplePublicAttribute**. The fourth step executes a set assignment statement:

```
set out1.simplePublicAttribute = "overwrite initial Value";
```

This statement modifies **out1** and **ReferenceOut1**, but not **copiedOut1**.

```
set out1.simplePublicAttribute = "overwrite initial Value";
```