

copy

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

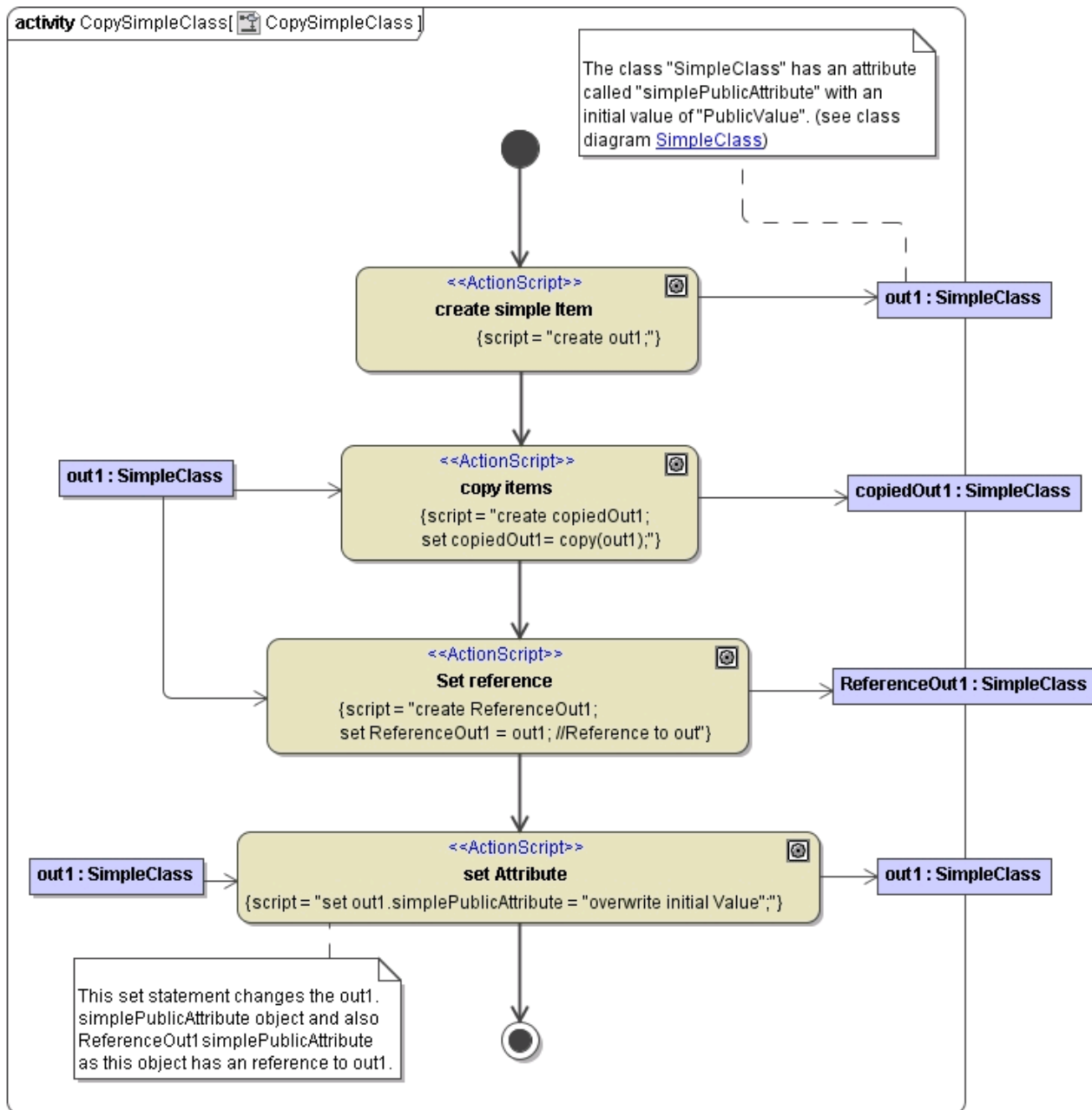
Example File (Builder project Basic Modeling/Data):



<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



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";
```