Scripting with the Boo.Lang.Interpreter API

Using the Boo.Lang.Interpreter API is quite straightforward. You can expose values to the scripting environment through the **SetValue** call and you can read values from the scripting environment with the **GetValue** call. The Eval call allows you to execute arbitraly complex source code:

```
import Boo.Lang.Interpreter from Boo.Lang.Interpreter

interpreter = InteractiveInterpreter()
interpreter.SetValue("Message", "ping")
interpreter.Eval("""
print Message
Message = 'pong'
""")
print interpreter.GetValue("Message")
```

You can also use InteractiveInterpreter as an expression evaluator by setting **RememberLastValue**:

```
import Boo.Lang.Interpreter from Boo.Lang.Interpreter

interpreter = InteractiveInterpreter(RememberLastValue: true)
interpreter.Eval("3**2")
print interpreter.LastValue
```

A nice trick is to provide code completion in a GUI application by leveraging **InteractiveInterpreter.SuggestCodeCompletion**:

```
entity = interpreter.SuggestCodeCompletion("print 'zeng'.__codecomplete__")
```

entity is a reference to the Boo.Lang.Compiler.TypeSystem.IEntity object found as the target of the **codecomplete** member.

Exposing code from an arbitrary assembly (let's say the current executing one) to the interactive interpreter instance is done through the **References** collection:

```
import Boo.Lang.Interpreter from Boo.Lang.Interpreter

def foo():
    print "bar"

interpreter = InteractiveInterpreter()
interpreter.References.Add(System.Reflection.Assembly.GetExecutingAssembly())
interpreter.Eval("foo()")
```

A more complete example of a script which callbacks to the precompiled code:

```
import Boo.Lang.Interpreter from Boo.Lang.Interpreter

class Sensor:
   event Triggered as callable()

   def Trigger():
       Triggered()

class Car:
   def Stop():
       print 'stopped'

   public RedLightSensor = Sensor()


car = Car()

interpreter = InteractiveInterpreter()
interpreter.SetValue('car', car)
interpreter.Eval("""
car.RedLightSensor.Triggered += do:
   car.Stop()
""")

car.RedLightSensor.Trigger()
```