

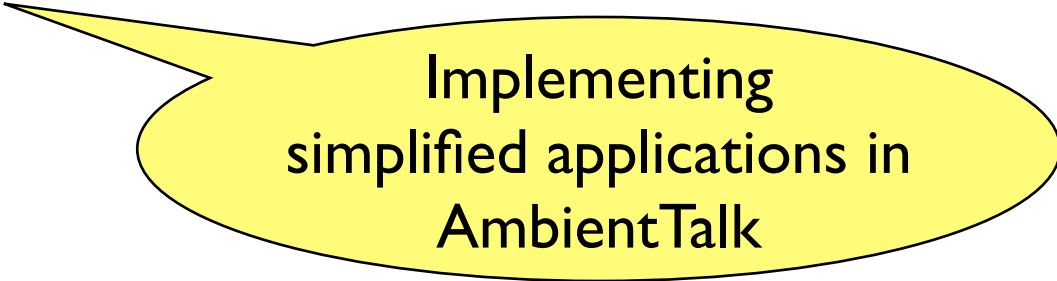
First Steps in AmbientTalk

Elisa Gonzalez Boix
egonzale@vub.ac.be



Lab Sessions - Goals

- Get you familiar with concurrent and distributed programming abstractions.
- Get you ready for the project.



Implementing
simplified applications in
AmbientTalk

Project

- Design and develop a more complex distributed application in AmbientTalk.
- Evaluated mostly on good distributed design.
- What is important to remember?
 - Individual!
 - Quality and structure of the code!
 - Test cases and report!



Material

<http://soft.vub.ac.be/amop/>

- Language reference
- Tutorial
- Lab sessions material

<http://code.google.com/p/ambienttalk>

- AmbientTalk IDE for Eclipse (IdeAT)
- iat command line parameters





Unit Testing



Unitest Framework

- ~ JUnit, SUnit

```
def UnitTest := /.at.unit.test;
```

```
def myUnitTest := extend: UnitTest.new("my UnitTest") with: {  
  def testSomething() {
```

```
    self.assertEqual(3, 1+2);
```

Method defined in UnitTest object

```
  }  
}  
  
myUnitTest.runTest()
```

Asynchronous UnitTest

by default, parallel!

.new("my UnitTest", true)

```
def myUnitTest := extend: UnitTest.new("my UnitTest") with: {  
  def testAsyncOneCustomer()  
    def f := when: customer<-haveComputer() becomes:{ | id |  
      self.assertEquals(id, 1);  
    };  
    f;  
  };  
}
```

Return value of an asynchronous
unit test should be a future!

myUnitTest.runTest()

Unittest Framework

```
assertTrue(boolean)  
assertFalse(boolean)  
assertEquals(o1,o2)  
assertNotEquals(o1,o2)  
assertLessThan(o1,o2)  
assertGreaterThan(o1,o2)  
assertLessThanOrEqualTo(o1,o2)  
assertGreaterThanOrEqualTo(o1,o2)  
assertMatches(str, pattern)  
assertNotNil(val)
```

```
assert: exceptionType raisedIn: closure  
fail(reason)
```





<http://soft.vub.ac.be/amop>