

Mock Objects In Action

IndicThreads.com Conference On Java Technology

PUNE, INDIA

25

26

27

Nov. 2008

Paulo Caroli & Sneha Jha

ThoughtWorks

About us

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

About us

Paulo Caroli & Sneha Jha

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

About us

Hands-on developers

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

About us

Open Source

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

About us

ThoughtWorks

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

About us

TDD enthusiasts

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Agenda

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Agenda

- History
- Design for Testability
- Dependent Components
- Mock Objects in Action
 - Frameworks comparison
 - Code Watch
- Top Offenders
- Q&A

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

History

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Once upon a time we started writing Unit
Tests...

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Then we started using unit tests
frameworks...

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

And soon after we started creating (and re-creating) data and implementations to help drive our tests...

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Then we came across Mock Objects frameworks...

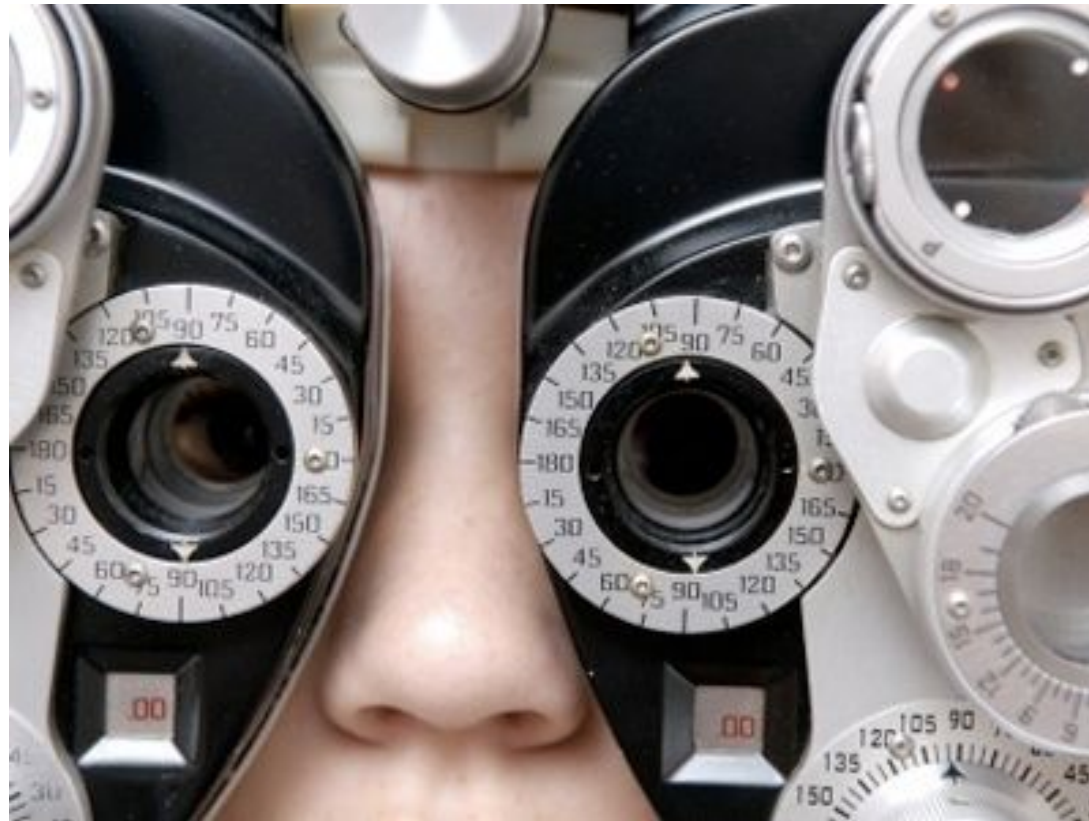
IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

But frameworks don't work by themselves!

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA


```
public class Greeting {  
    private Translator translator;  
    private final String HELLO = "Hello";  
    public Greeting() {  
        this.translator = new Translator();  
    }  
    public String sayHello(String language, String name) {  
        String message = HELLO;  
        // no need to call translator if the language is English  
        if (!language.equals("English")){  
            message = translator.translate("English", language, HELLO );  
        }  
        return message + " " + name;  
    }  
}
```

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Design for Testability

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

```
public class Greeting {  
    private Translator translator;  
    private final String HELLO = "Hello";  
    public Greeting(Translator translator) {  
        this.translator = translator;  
    }  
    public String sayHello(String language, String name) {  
        String message = HELLO;  
        // no need to call translator if the language is English  
        if (!language.equals("English")){  
            message = translator.translate("English", language, HELLO );  
        }  
        return message + " " + name;  
    }  
}
```

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Mock Objects

Mock Objects enable your unit test to mimic behavior and verify expectations on dependent components.

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

```
import static org.mockito.Mockito.*;
public class GreetingTest{

    @Test
    public void shouldGreetInSomeLanguage(){
        // setup
        Translator translator = mock(Translator.class);
        Greeting greeting = new Greeting(translator);
        when(translator.translate("English", "French",
            "Hello")).thenReturn("Bon jour");
        // execute
        assertEquals("Bon jour Madame",
            greeting.sayHello("French", "Madame"));
        // verify
        verify(translator).translate("English", "French", "Hello");
    }
}
```

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Mock Objects

Mock Objects enable your unit test to mimic behavior and verify expectations on dependent components.

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Mock Objects

Mock Objects enable your unit test to mimic behavior and verify expectations on **dependent components.**

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Dependent Components

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Dependent Components

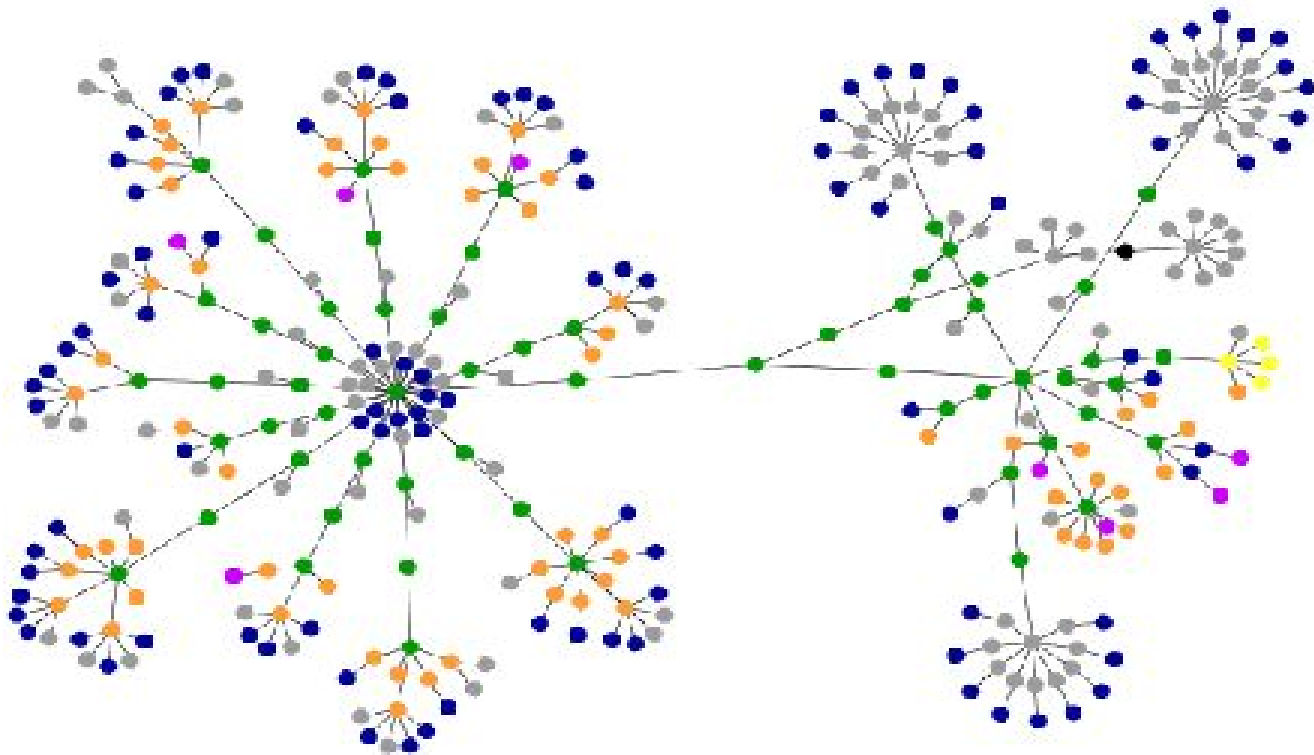


IndicThreads.com Conference On Java Technology 2008

© ThoughtWorks 2008

PUNE, INDIA

Object Graphs



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

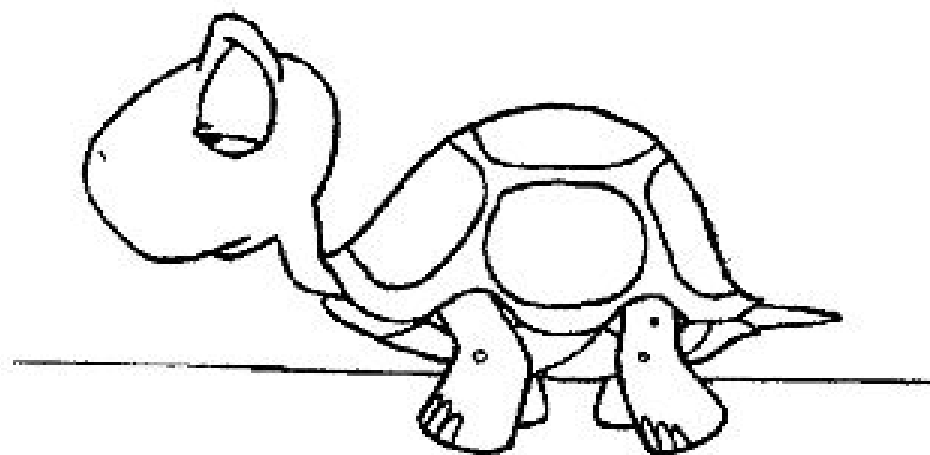
Third Party



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

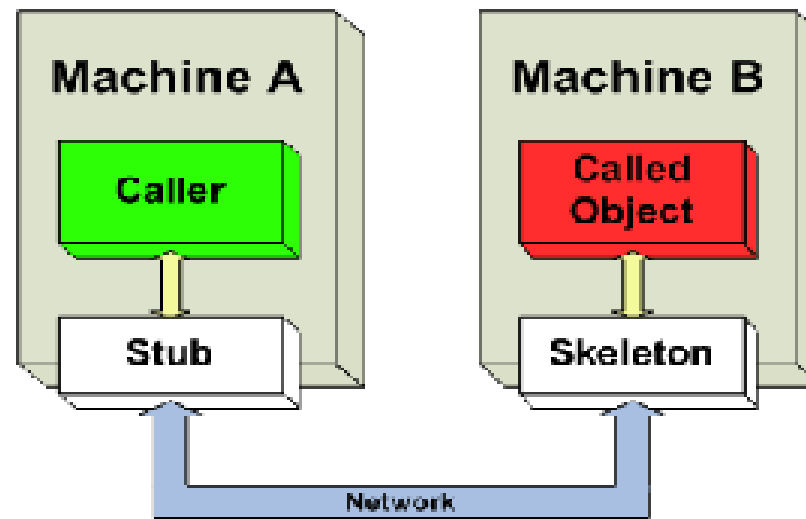
Very Slow



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

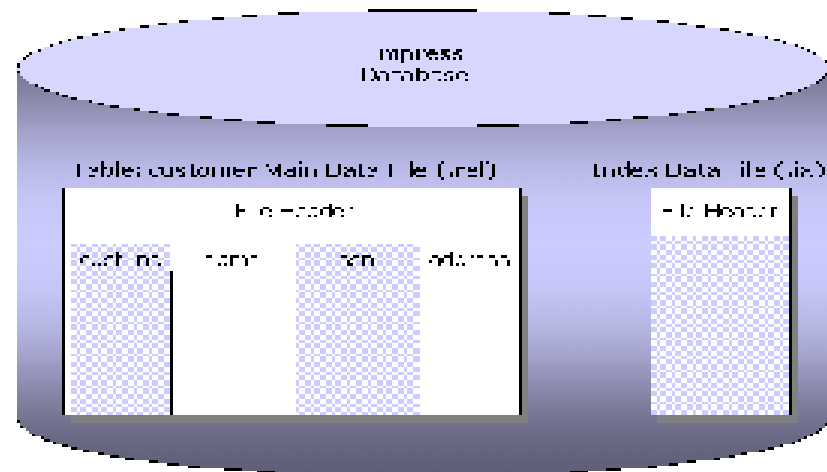
Remote Component



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

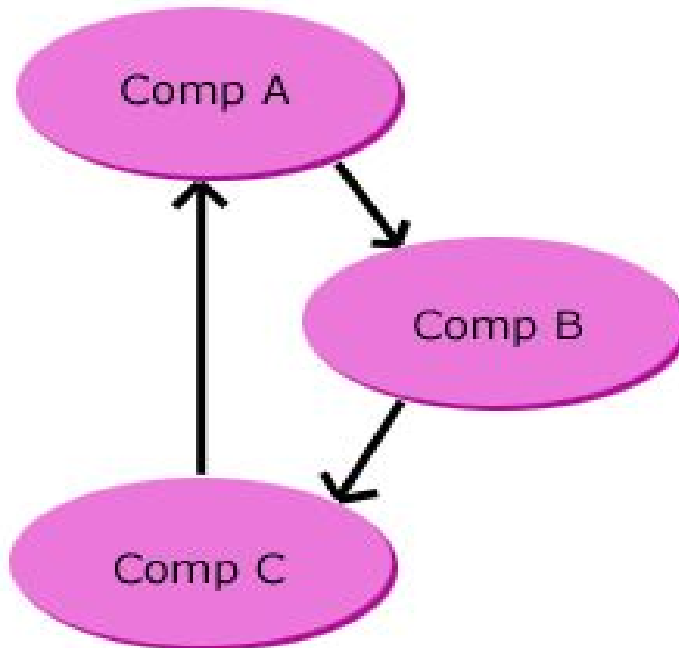
Database Component



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Circular Dependency



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

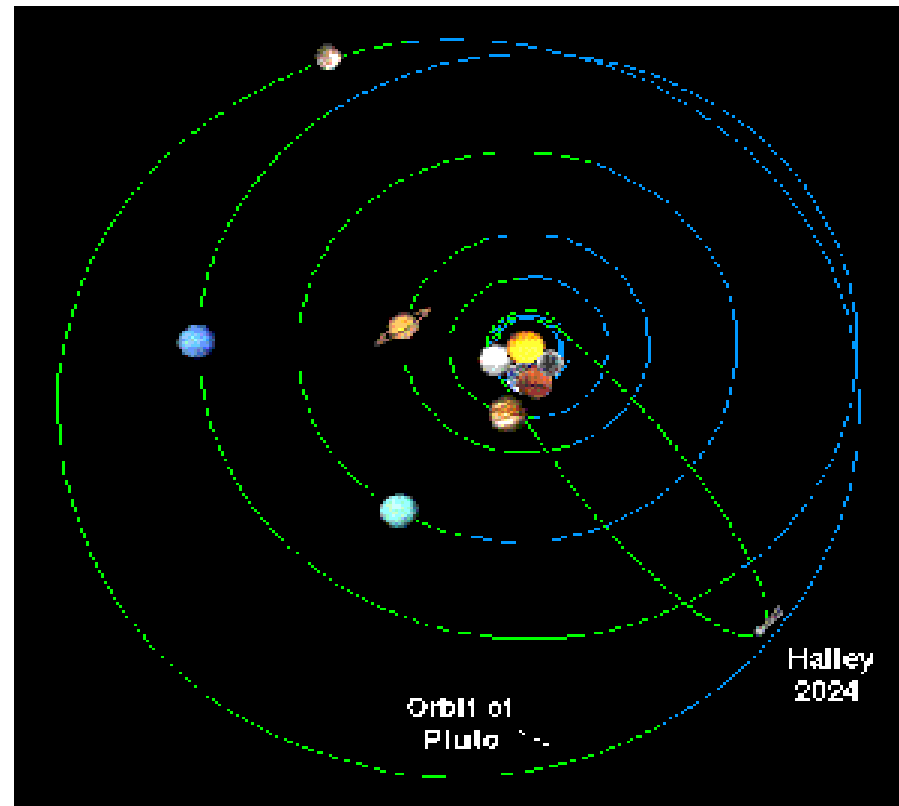
Complex Set-up



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Exceptional Behavior



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

My co-worker's component



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Mock Objects In Action

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Mock Objects in Action

1. Declare the mock object
2. Instantiate the mock object
3. Inject the mock object
4. Set expectations and behavior for the mock object
5. Verify expectations for the mock object

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Frameworks Comparisons

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Code Watch



IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Mock Objects Top Offenders

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Test readability

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Replacement for integration test

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Unnecessary Mock for simple Value Object

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Abuse of mocks

(instead: FakeDB, Dummy data, Stub,
Test Fixture)

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Mock context confusion

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Blaming the framework

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA

Mock Objects in Action

Thanks for attending!
Questions?

Paulo Caroli & Sneha Jha

pcaroli@thoughtworks.com
snehajh@thoughtworks.com

IndicThreads.com Conference On Java Technology 2008

PUNE, INDIA