



# How to Deal With Your IT Legacy? Reverse Engineering with MoDisco...



Hugo Brunelière  
Frédéric Madiot

INRIA / AtlanMod team  
Mia-Software

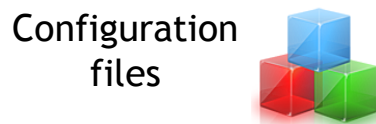
# Overview



a **Model-Driven** platform

for **Software Modernization** projects

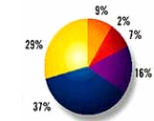
Support multiple  
Legacy Technologies



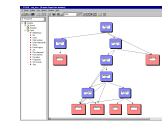
Support multiple  
Modernization Scenarios

- Quality Assurance
- Understanding
- Refactoring
- Technical Migration

Support multiple  
Outputs



Metrics



Models



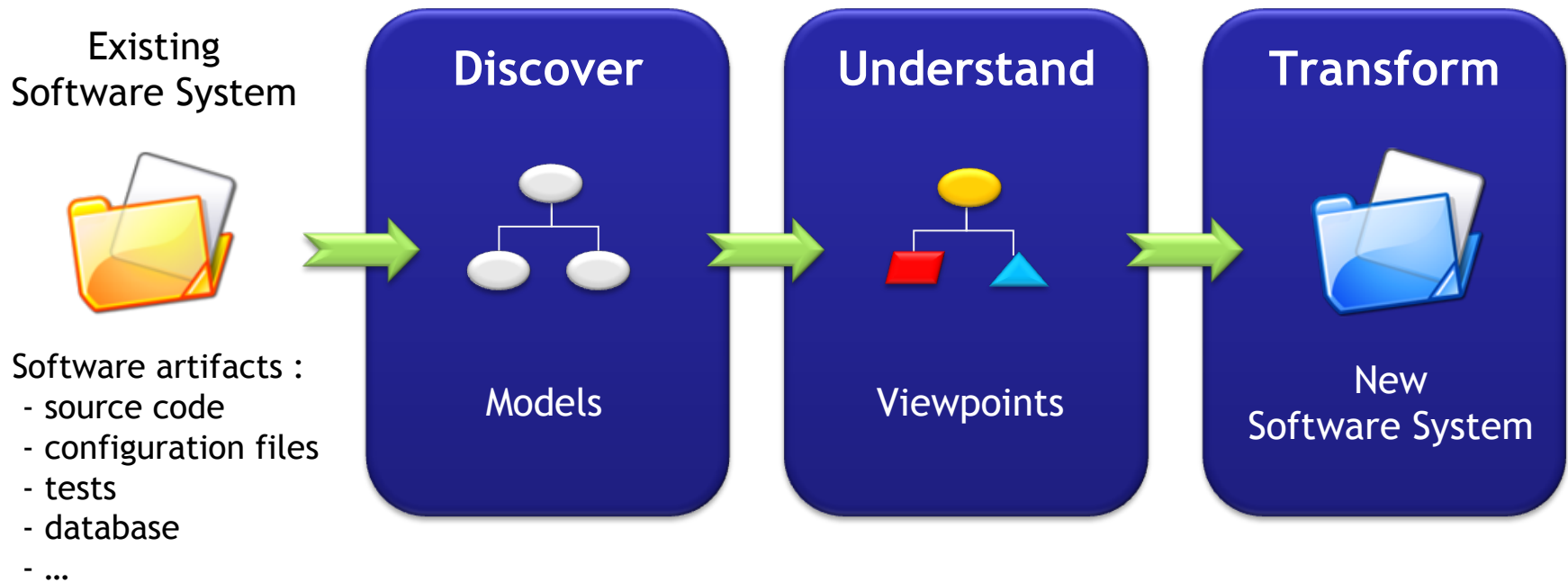
New system



Documentation

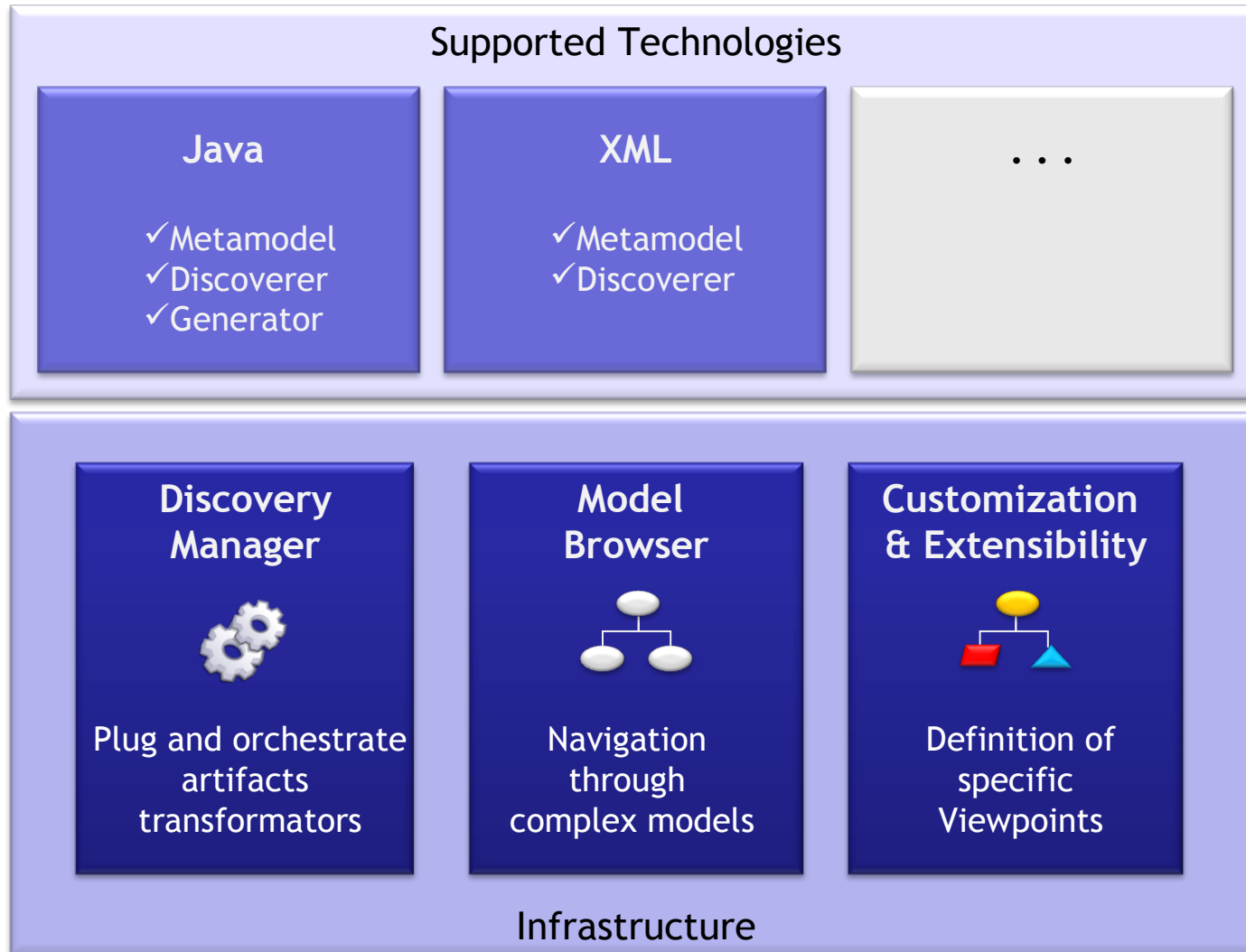
# Approach

**Models** to represent and manipulate  
**artifacts** of existing systems





# Architecture



# Legacy applications in Java ?



*From the moment one writes a line of code, it becomes legacy, and that legacy accumulates (Grady Booch)*

## - Samples



- LinkedIn > 1M lines of java code
- Texas Health (TIERS application) > 2.5M lines of java code
- CruiseControl > 5M lines of java code
- Eclipse (Galileo) > 24M lines of java code
- SAP (NetWeaver) > 24M lines of java code (255k classes)

## - Continuous evolution

- JDK versions x Frameworks x Design Patterns

# Complex models ?



## Java metamodel :

- 126 types
- 173 references

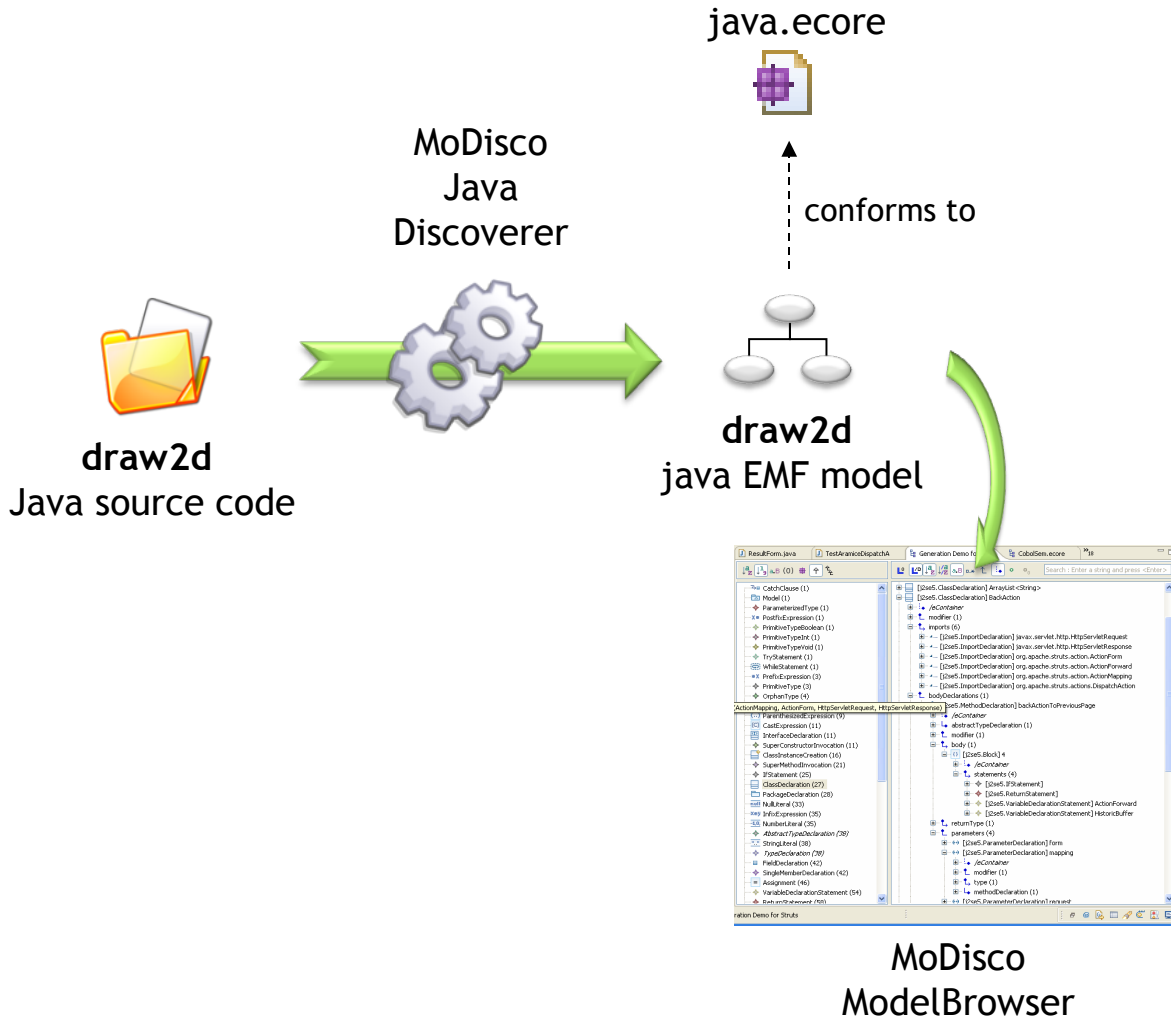
## Example :

Java model of **draw2d** :

- org.eclipse.draw2d
- org.eclipse.draw2d.tests

447 Java classes  $\Rightarrow$  **144 374** nodes !

# MoDisco Demo



Demo

# Find Patterns in your Models !



Define and dynamically plug  
your own  
**Facets & UI Customizations**

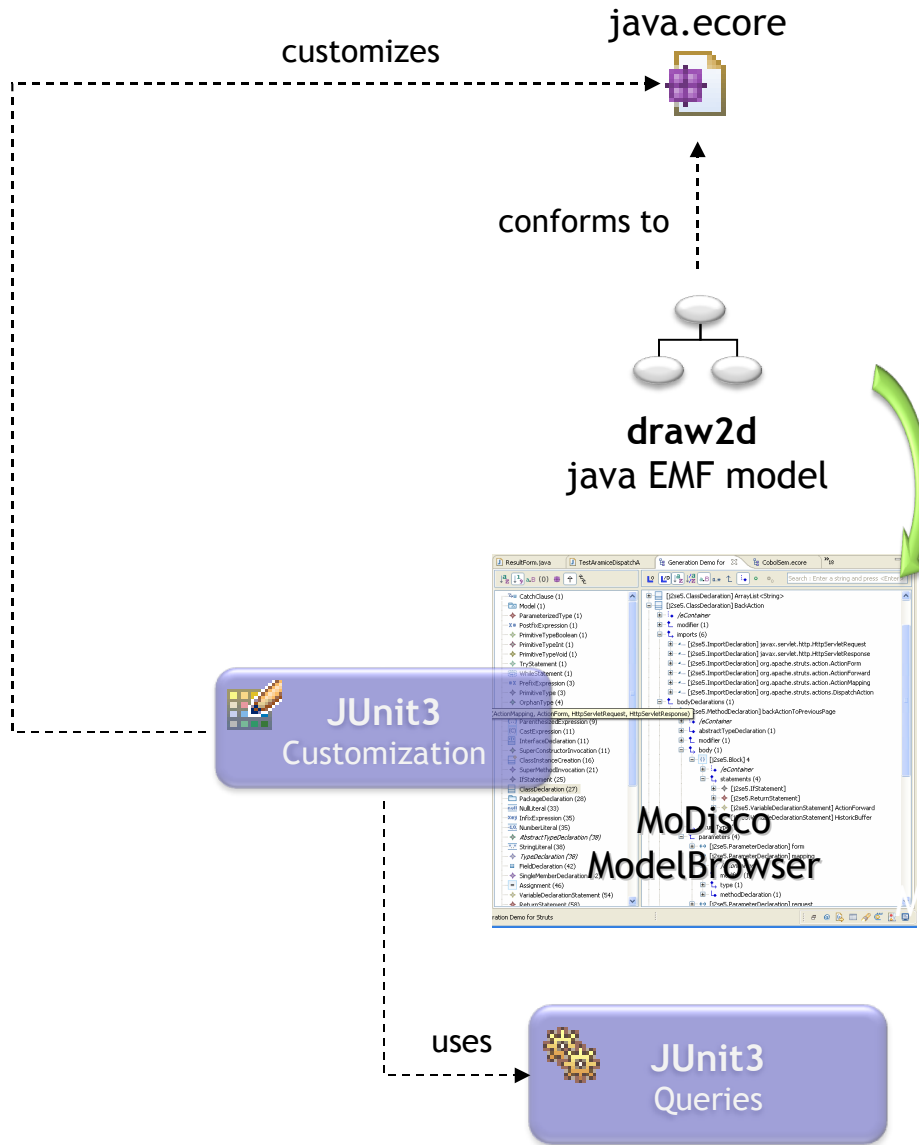
## Example

Decorate the Java model of  
**org.eclipse.draw2d.tests**  
To highlight **JUnit** patterns:

- Test suite
- Test case
- Test

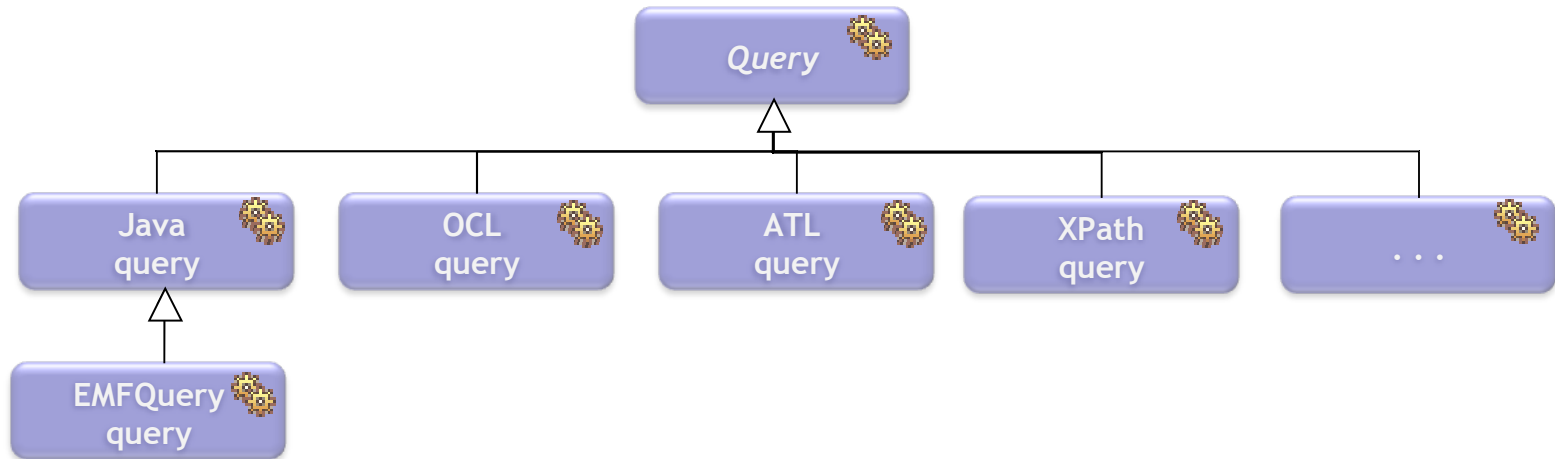


# MoDisco Demo

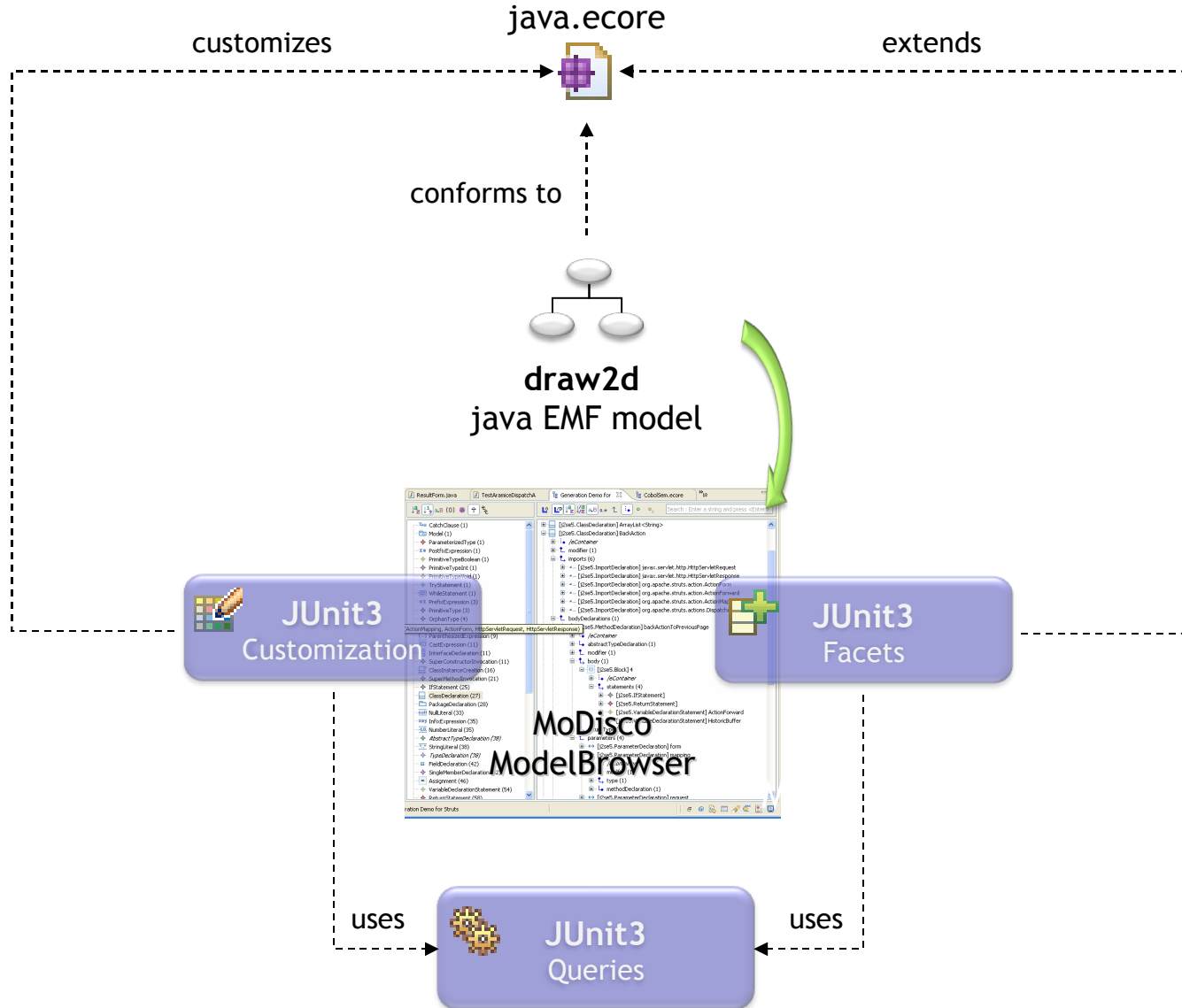


Demo

# MoDisco Demo

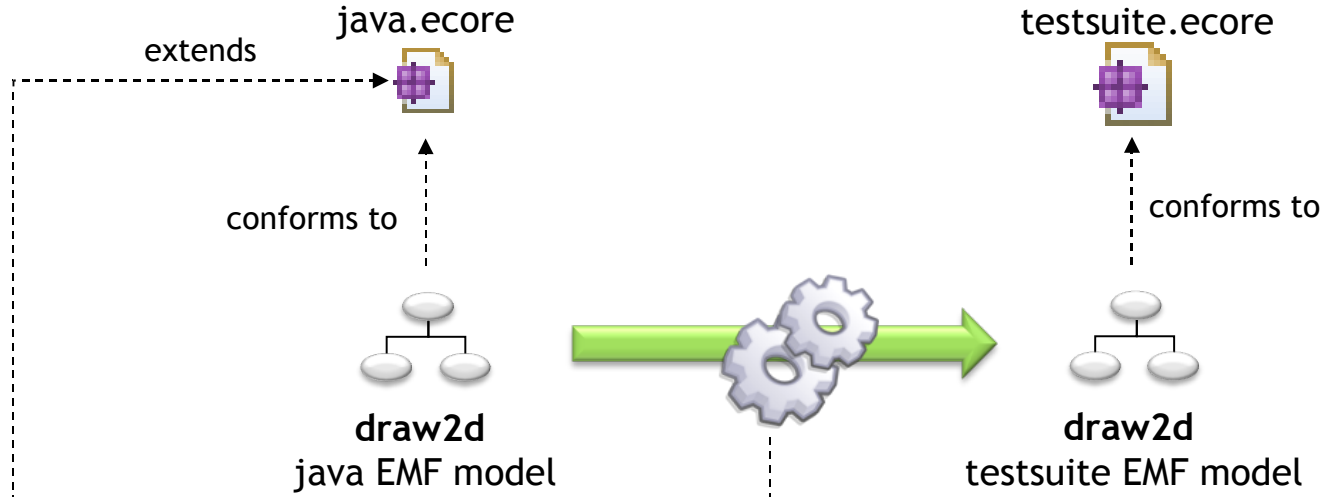


# MoDisco Demo



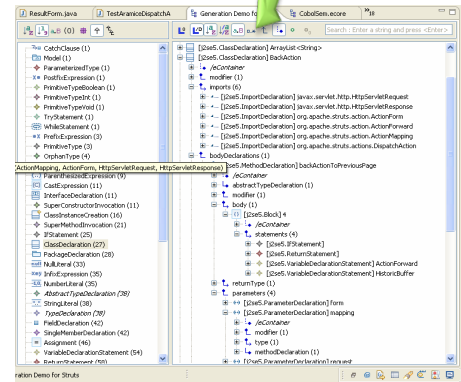
Demo

# MoDisco Demo



**JUnit3 Facets**

**JUnit3 Queries**

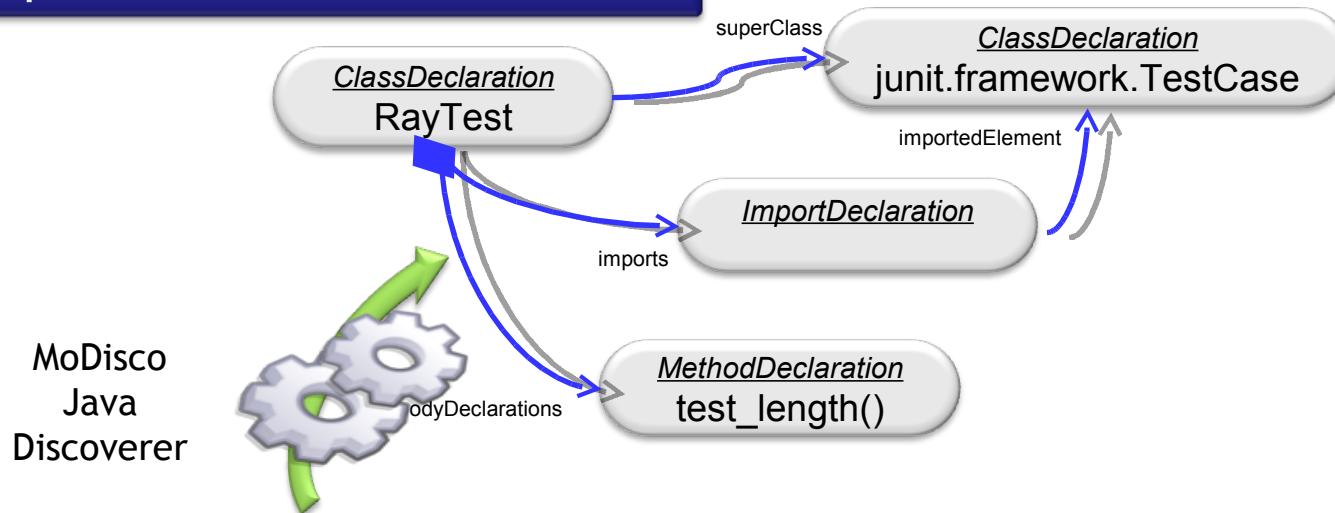


**Demo**

**MoDisco ModelBrowser**

# Transform and Regenerate ...

## Example : from JUnit3 to JUnit4



```

package org.eclipse.draw2d.test;

import junit.framework.TestCase;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest extends TestCase {

    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }
  
```



```

package org.eclipse.draw2d.test;

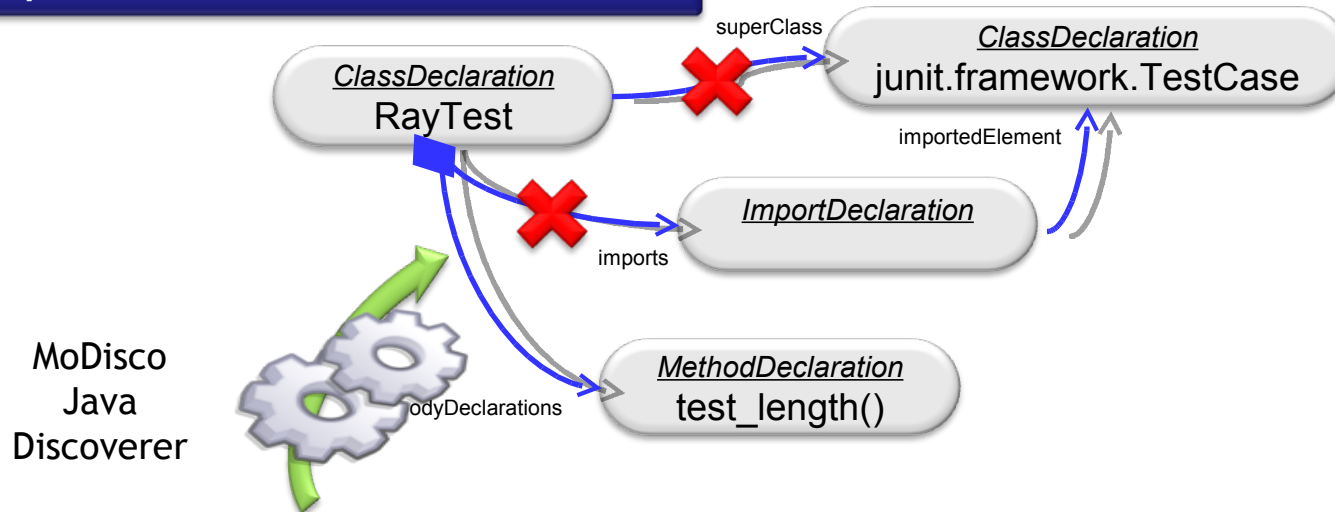
import org.junit.Test;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest {

    @Test
    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }
  
```

# Transform and Regenerate ...

## Example : from JUnit3 to JUnit4



```

package org.eclipse.draw2d.test;

import junit.framework.TestCase;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest extends TestCase {

    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }

```



```

package org.eclipse.draw2d.test;

import org.junit.Test;
import org.eclipse.draw2d.geometry.Ray;

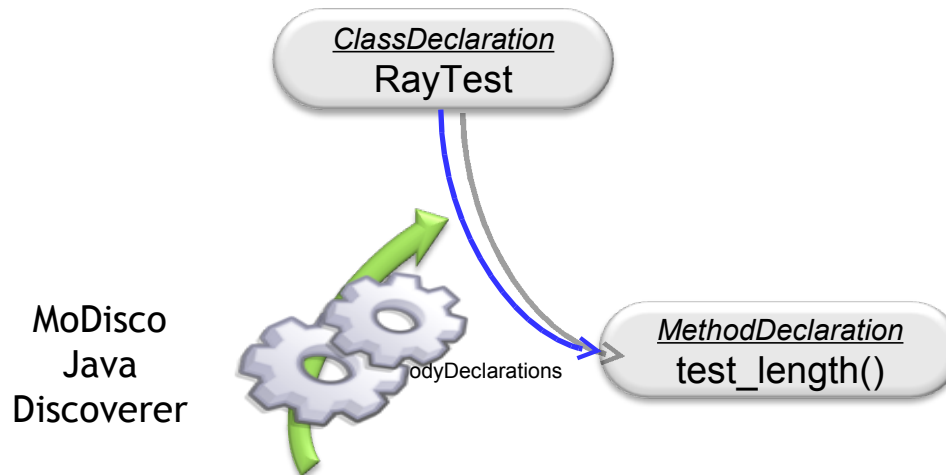
public class RayTest {

    @Test
    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }

```

# Transform and Regenerate ...

Example : from JUnit3 to JUnit4



```
package org.eclipse.draw2d.test;

import junit.framework.TestCase;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest extends TestCase {

    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }
}
```



```
package org.eclipse.draw2d.test;

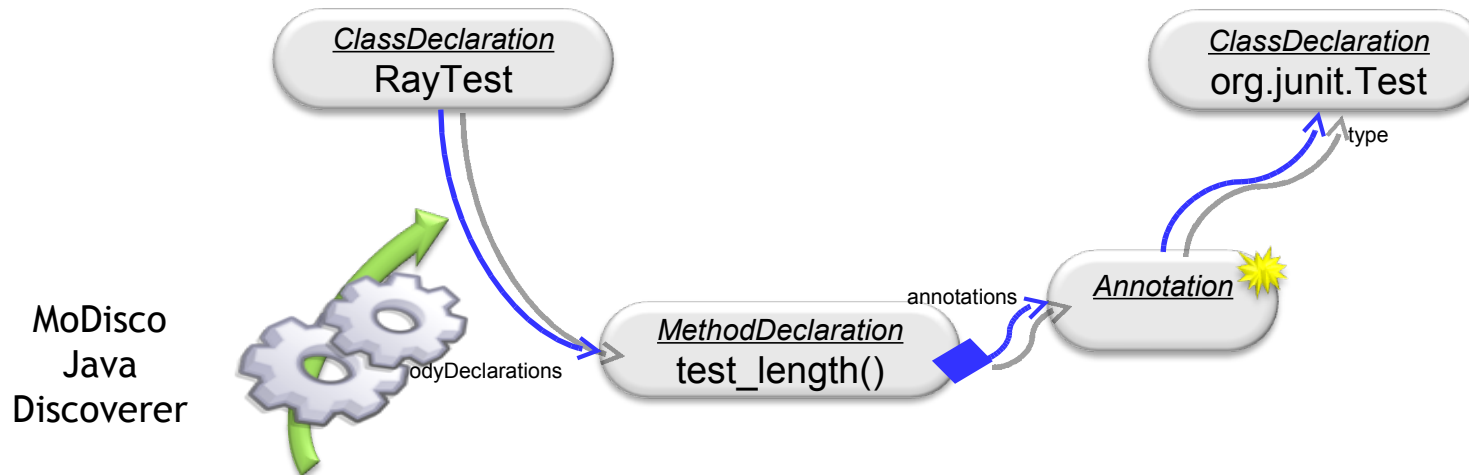
import org.junit.Test;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest {

    @Test
    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }
}
```

# Transform and Regenerate ...

## Example : from JUnit3 to JUnit4



```

package org.eclipse.draw2d.test;

import junit.framework.TestCase;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest extends TestCase {

    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }

```



```

package org.eclipse.draw2d.test;

import org.junit.Test;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest {

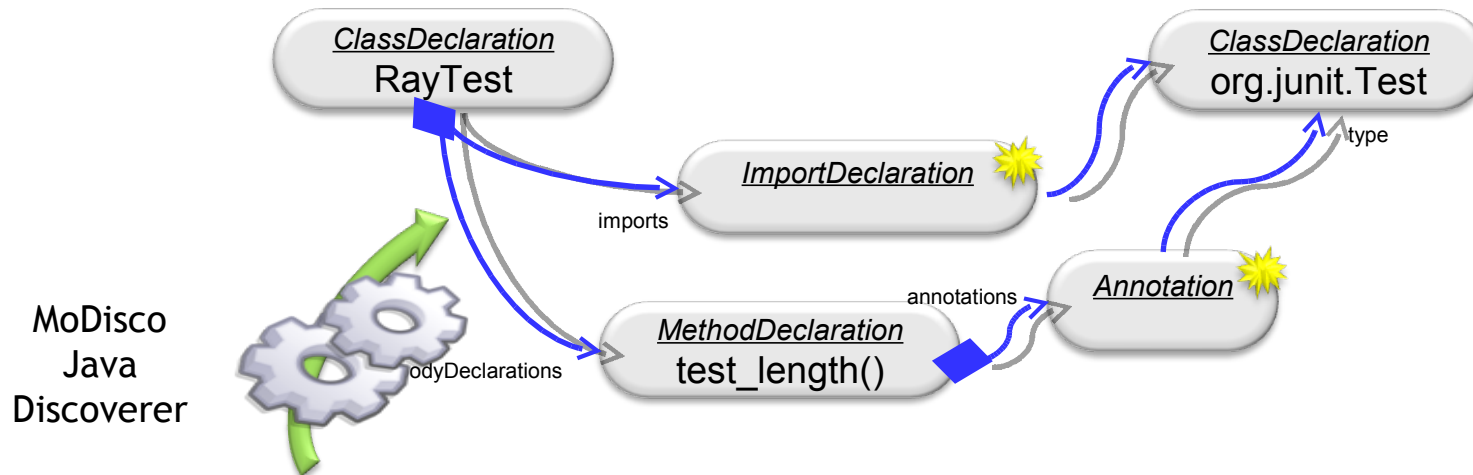
    @Test
    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }

```



# Transform and Regenerate ...

## Example : from JUnit3 to JUnit4



```

package org.eclipse.draw2d.test;

import junit.framework.TestCase;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest extends TestCase {

    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }

```



```

package org.eclipse.draw2d.test;

import org.junit.Test;
import org.eclipse.draw2d.geometry.Ray;

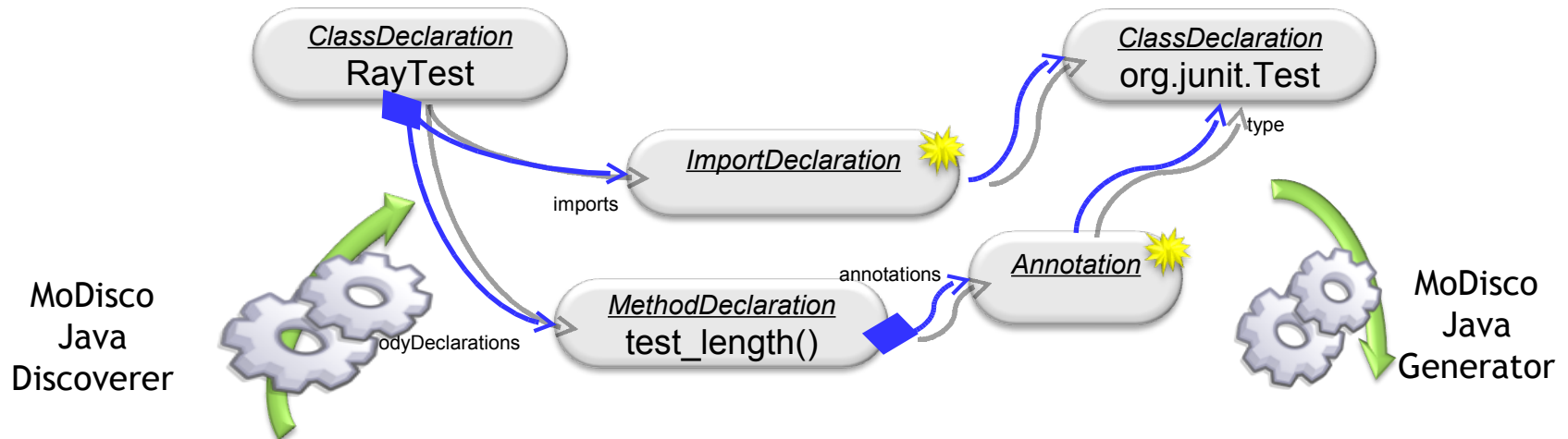
public class RayTest {

    @Test
    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }

```

# Transform and Regenerate ...

## Example : from JUnit3 to JUnit4



```

package org.eclipse.draw2d.test;

import junit.framework.TestCase;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest extends TestCase {

    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }

```



```

package org.eclipse.draw2d.test;

import org.junit.Test;
import org.eclipse.draw2d.geometry.Ray;

public class RayTest {

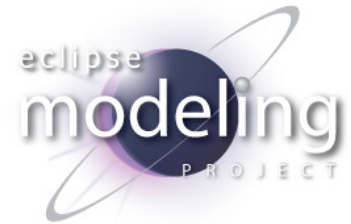
    @Test
    public void test_length() {
        testLengthValues(3, 4, 5);
        testLengthValues(0,
            Integer.MAX_VALUE,
            Integer.MAX_VALUE);
    }

```

# Who is involved in MoDisco ?



- The MoDisco team
  - 6 committers
    - 1 from INRIA
    - 5 from Mia-Software
  - Several regular active contributors
- Supported by the OMG (ADM TF)
  - Reference implementations of standards from the
    - Architecture Driven Modernization Task Force
      - KDM (Knowledge Discovery Metamodel)
      - SMM (Software Metrics Metamodel)
- Supported by the European Commission
  - Project initiated by INRIA in the context of
    - the MODELPLEX IST-FP6 European Project





## Questions ?

[www.eclipse.org/gmt/modisco](http://www.eclipse.org/gmt/modisco)

### MoDisco Downloads




MoDisco	Eclipse	Java
0.8	3.8 (Helios)	5.0
0.7	3.5 (Galileo)	5.0

### Latest Releases

>> [0.7.1 \(2009/10/12\)](#)  OK ✓

### 0.8.0 Stable Builds

>> [0.8.0M6 \(2010/03/16\)](#)  OK ✓

>> [0.8.0M5 \(2010/02/01\)](#)  OK ✓

>> [0.8.0M4 \(2010/01/19\)](#)  OK ✓