The Eclipse/GMT MoDisco Component

Overview

Outline

- Introduction
- Background
- Discovery Principles
- Motivating Examples
- Component Description
- Benefits of the Approach
- Organization
- Roadmap
- First Use Cases Descriptions

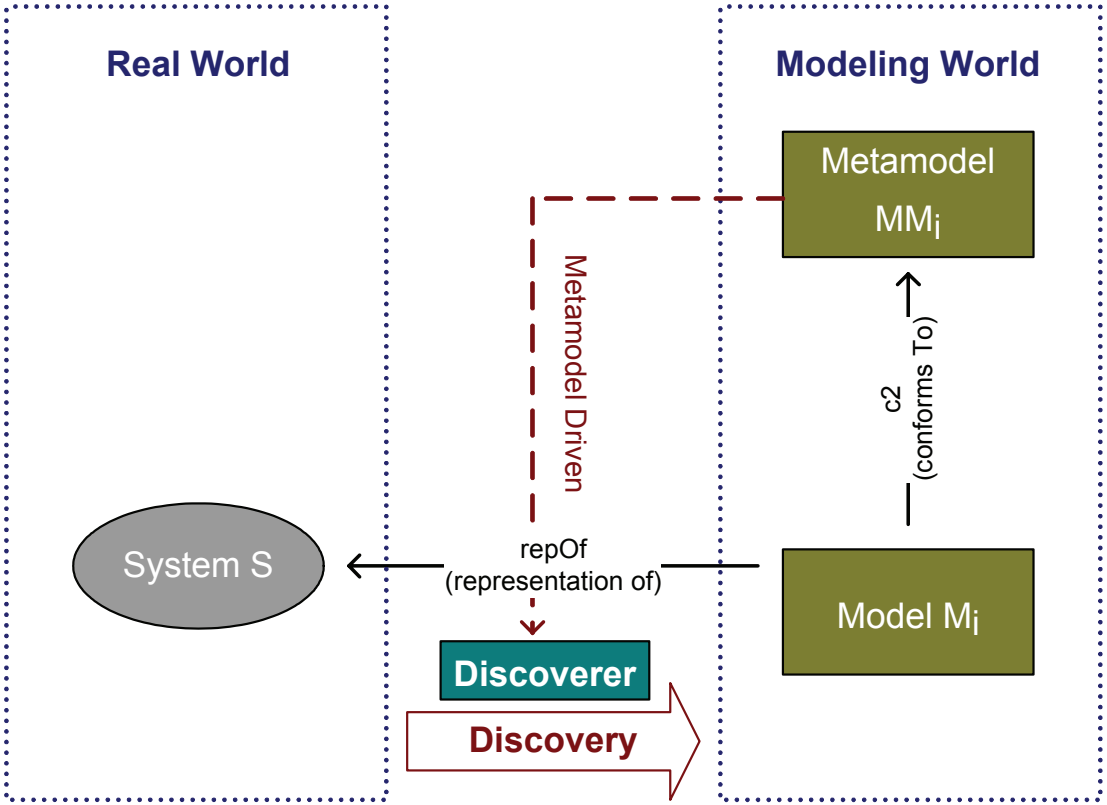
Introduction

- MoDisco for "**Model Discovery**"
- Eclipse GMT component for model-driven reverse-engineering (MDRE)
- Extraction of models from legacy systems
  - Different natures and technologies
- A Generic and extensible metamodel-driven approach to model discovery

Background

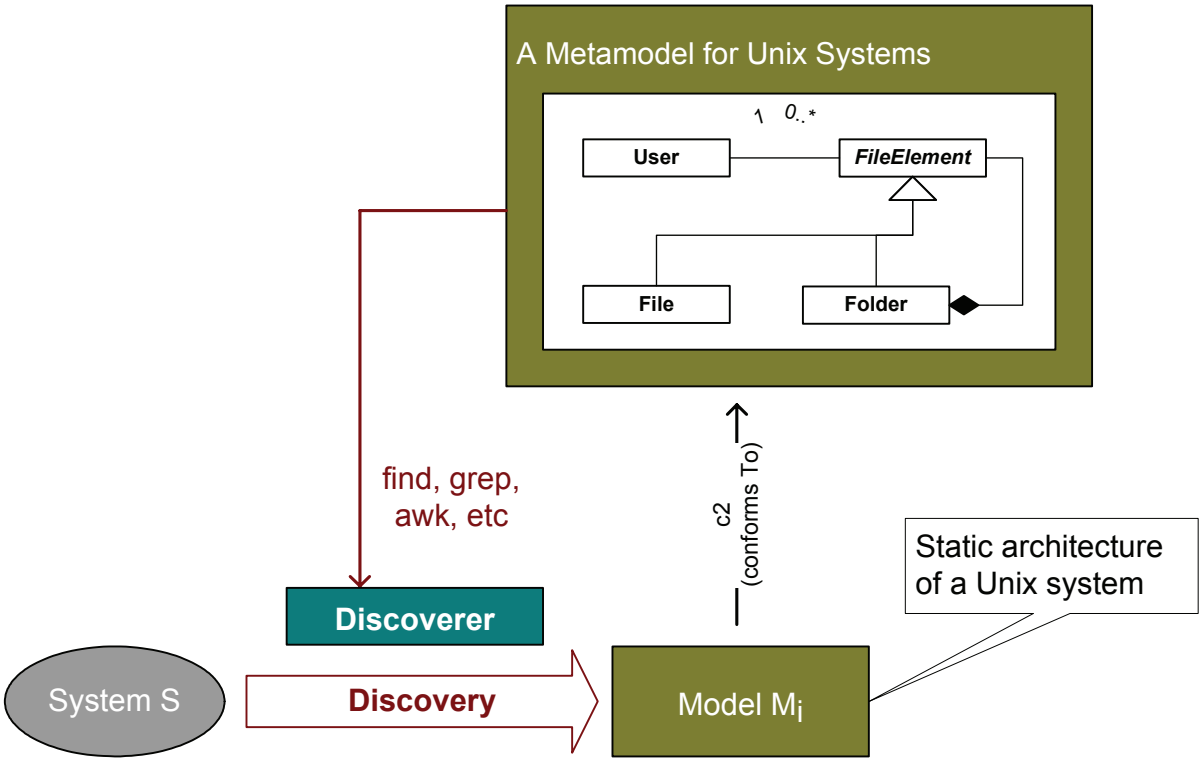
- Develop and manage systems becoming more and more complex
- Important issue:
  - **Reverse-engineering of legacy systems**
- **MoDisco** component's goal:
  - Provide an extensible base framework for performing **metamodel-driven reverse engineering** (MDRE)
- The key to success:
  - Adoption by leading industrials
  - Development of a wide user community in different application domains

Discovery Principles



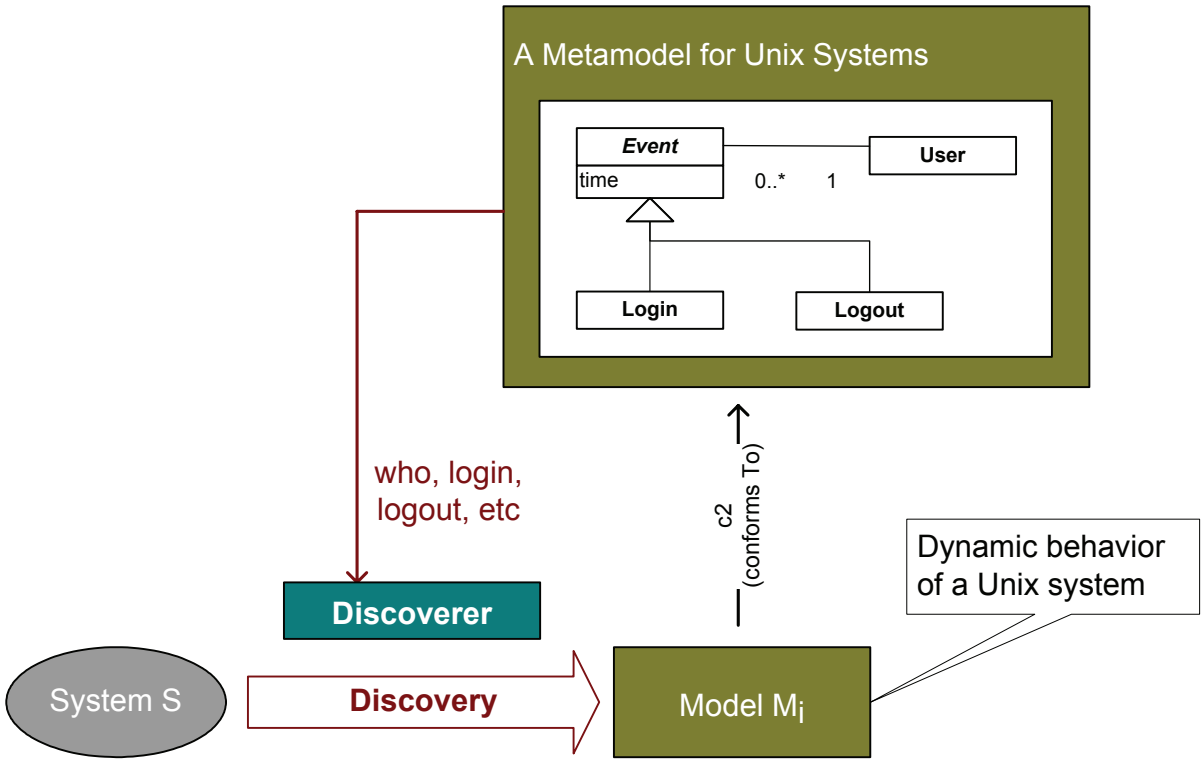
- Step 1:
  - Define the metamodel
- Step 2:
  - Create the "discoverer"
- Step 3:
  - Build the model

Motivating Examples (1/4)



- Example of the Unix file system
- Study of a **static** view on the system
  - Snapshot of the system at time *t*

Motivating Examples (2/4)



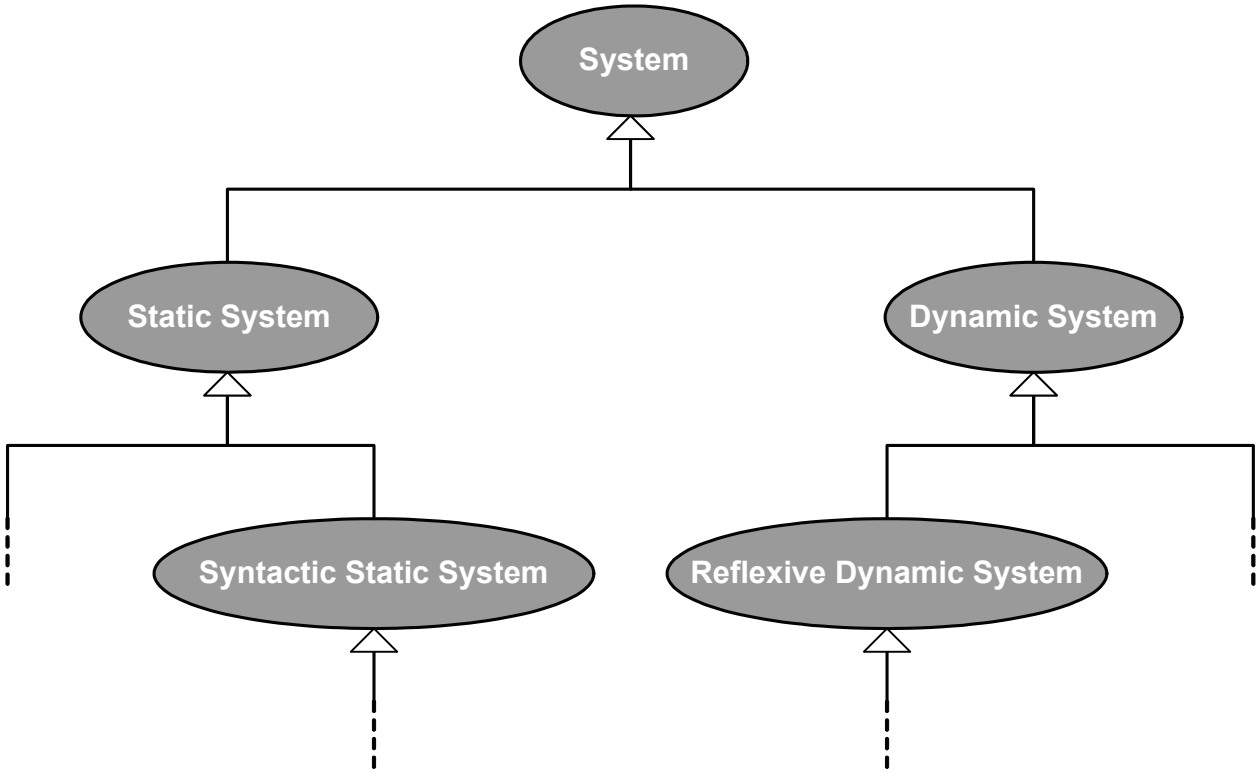
- Example of the Unix users' actions
- Study of the **dynamic** behavior of the system
  - Execution trace of the system

Motivating Examples (3/4)

- Conclusions:
  - The same general discovery process is applied in both examples
    - Only the nature of the "discoverers" is changing
  - Need of a system classification
    - A decision tree more than an absolute classification
      - Different points of view are possible on a same system
    - A support and methodology for facilitating the development of the "discoverers"
      - For instance, encouraging the use of the introspection capabilities in the case of a reflexive system

Motivating Examples (4/4)

- A possible system classification (basic very first version):

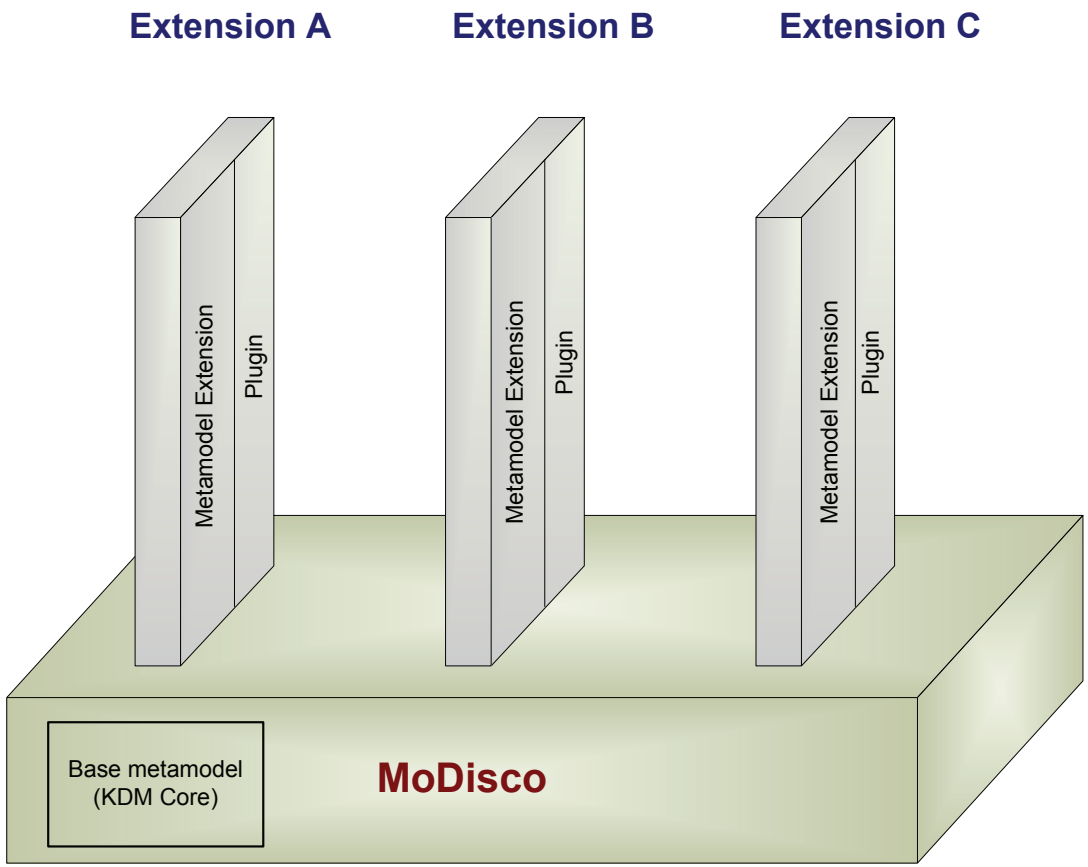


Component Description (1/2)

- A base generic and extensible framework
  - A core metamodel (based on the OMG™ KDM specification)
  - A metamodel extension's mechanism
  - Facilities for manipulating models
  - A methodology for designing extensions of this framework
    - An extension (or "blade") is a couple: extension of the core metamodel + plugin
    - Different extensions for different domains in various fields

Component Description (2/2)

- Overall vision of **MoDisco**:



Benefits of the Approach

- A unified model-based approach and a metamodel-driven methodology:
  - Work in the homogeneous world of the models
  - Match different requirements
    - Data integration, tools interoperability, systems migration, etc
  - Use models properties and facilities
    - Transformations, weavings, extractions, etc
- A possible wide user community

Organization (1/2)

- Creation of MoDisco supported by the **ModelPlex European Integrated Project** (FP6-IP #034081)

- Initial committers & contributors :
  - Hugo Bruneliere (INRIA)
  - Mikael Barbero (INRIA)

Organization (2/2)

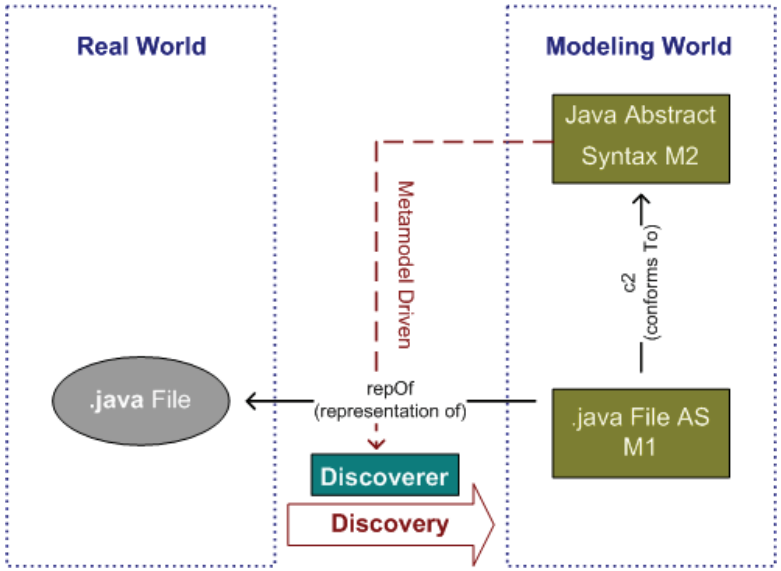
- Contributors and/or interested parties (industrials and academics):
  - INRIA
  - University of Nantes
  - MIA Software
  - Sodius
  - Obeo

Roadmap

- 1. Creation and initialization of the project (general description, web site, newsgroup, etc).
- **2. Elaboration of several use cases provided by different partners. A use case is usually composed of a specification and an implementation.**
- 3. Consolidification of the common toolbox and of the initial framework for building model discoverers. Improvement of the guidelines, methodological support and basic documentation.
- 4. Improvement of the framework as additional use cases are built and contributed.

First Use Cases Descriptions (1/3)

- Java Abstract Syntax

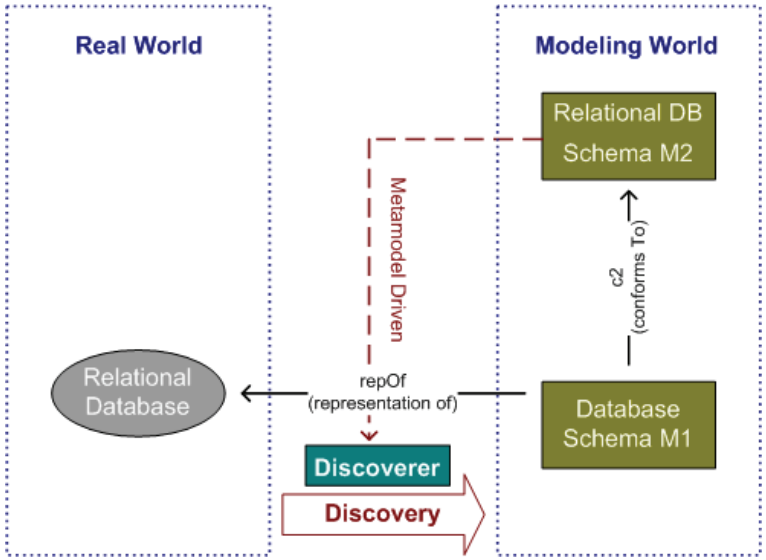


- Example of produced model (in XMI):

```
<statements xsi:type="java.ast:ExpressionStatement">
        <expression xsi:type="java.ast:MethodInvocation">
                <arguments xsi:type="java.ast:StringLiteral" escapedValue="&quot;Done !&quot;" literalValue="Done !"/>
                <expression xsi:type="java.ast:QualifiedName" fullyQualifiedName="System.out">
                        <name fullyQualifiedName="out" identifier="out"/>
                        <qualifier xsi:type="java.ast:SimpleName" fullyQualifiedName="System" identifier="System"/>
                </expression>
                <name fullyQualifiedName="println" identifier="println"/>
        </expression>
</statements>
```

First Use Cases Descriptions (1/3)

- Relational Database Schema

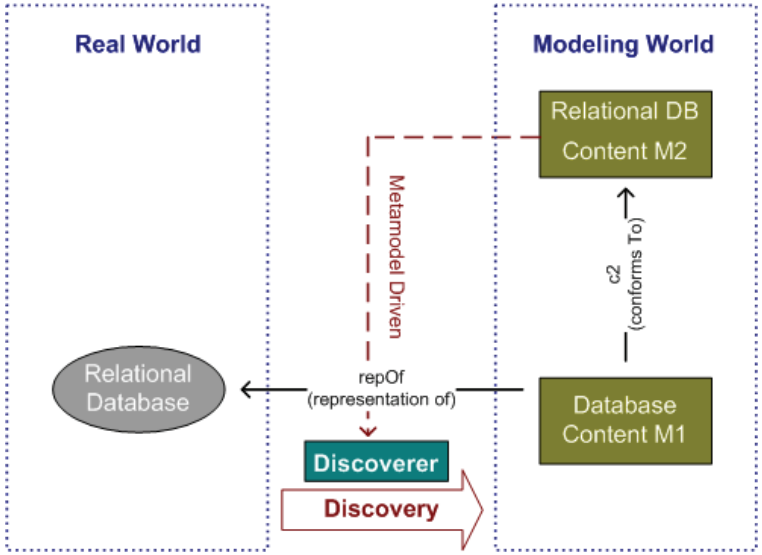


- Example of produced model (in Ecore):

```
<tables name="employees" key="#//@tables.1/@columns.0">
  <columns name="employeeNumber" dataType="int(11)" defaultValue="" keyOf="#//@tables.1"/>
  <columns name="lastName" dataType="varchar(50)" defaultValue=""/>
  <columns name="firstName" dataType="varchar(50)" defaultValue=""/>
  <columns name="extension" dataType="varchar(10)" defaultValue=""/>
  <columns name="email" dataType="varchar(100)" defaultValue=""/>
  <columns name="officeCode" dataType="varchar(10)" defaultValue=""/>
  <columns name="reportsTo" dataType="int(11)" null="true"/>
  <columns name="jobTitle" dataType="varchar(50)" defaultValue=""/>
</tables>
```

First Use Cases Descriptions (1/3)

- Relational Database Content



- Example of produced model (in Ecore):

```
<tables name="employees">
 – <tuples>
     <elements value="1002"/>
     <elements value="Murphy"/>
     <elements value="Diane"/>
     <elements value="x5800"/>
     <elements value="dmurphy@classicmodelcars.com"/>
     <elements value="1"/>
     <elements/>
     <elements value="President"/>
   </tuples>
```

End

- Thank you
  - Questions?
  - Comments?

- MoDisco website
  - http://www.eclipse.org/gmt/modisco
- MoDisco newsgroup
  - eclipse.modeling.gmt.modisco
- Contacts
  - Jean.Bezivin@univ-nantes.fr
  - Hugo.Bruneliere@univ-nantes.fr
  - Mikael.Barbero@univ-nantes.fr