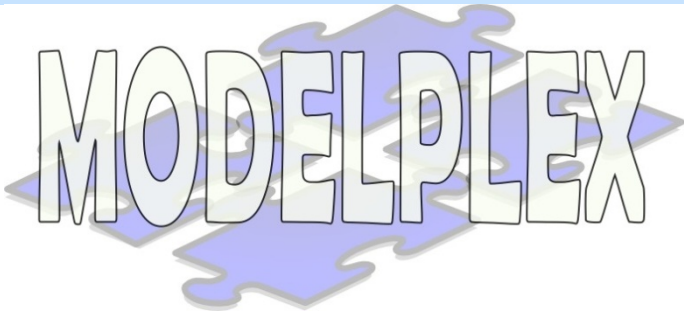
Knowledge Discovery: How to Reverse-Engineer Legacy Systems

Hugo Bruneliere, Frédéric Madiot

INRIA & MIA-Software

Context of this work



- The present courseware has been elaborated in the context of the MODELPLEX European IST FP6 project (http://www.modelplex.org/).
- Co-funded by the European Commission, the MODELPLEX project involves 21 partners from 8 different countries.
- MODELPLEX aims at defining and developing a coherent infrastructure specifically for the application of MDE to the development and subsequent management of complex systems within a variety of industrial domains.
- To achieve the goal of large-scale adoption of MDE, MODELPLEX promotes the idea of a collaborative development of courseware dedicated to this domain.
- The MDE courseware provided here with the status of open-source software is produced under the EPL 1.0 license.

Outline

- Knowledge Discovery Principles
  - Definition of Knowledge Discovery
  - Description of the overall process:
    Model Discovery + Model Understanding
- The Eclipse-GMT MoDisco Project
  - Presentation
  - Current toolbox & use case
  - The future platform
- Possible Applications
  - From source code
  - From database
  - From other kinds of systems
- A Concrete Application: Legacy System Interoperability Discovery
  - Global picture of the process
  - The implemented framework
  - First experiments on concrete material from industrial partners

Knowledge Discovery Principles

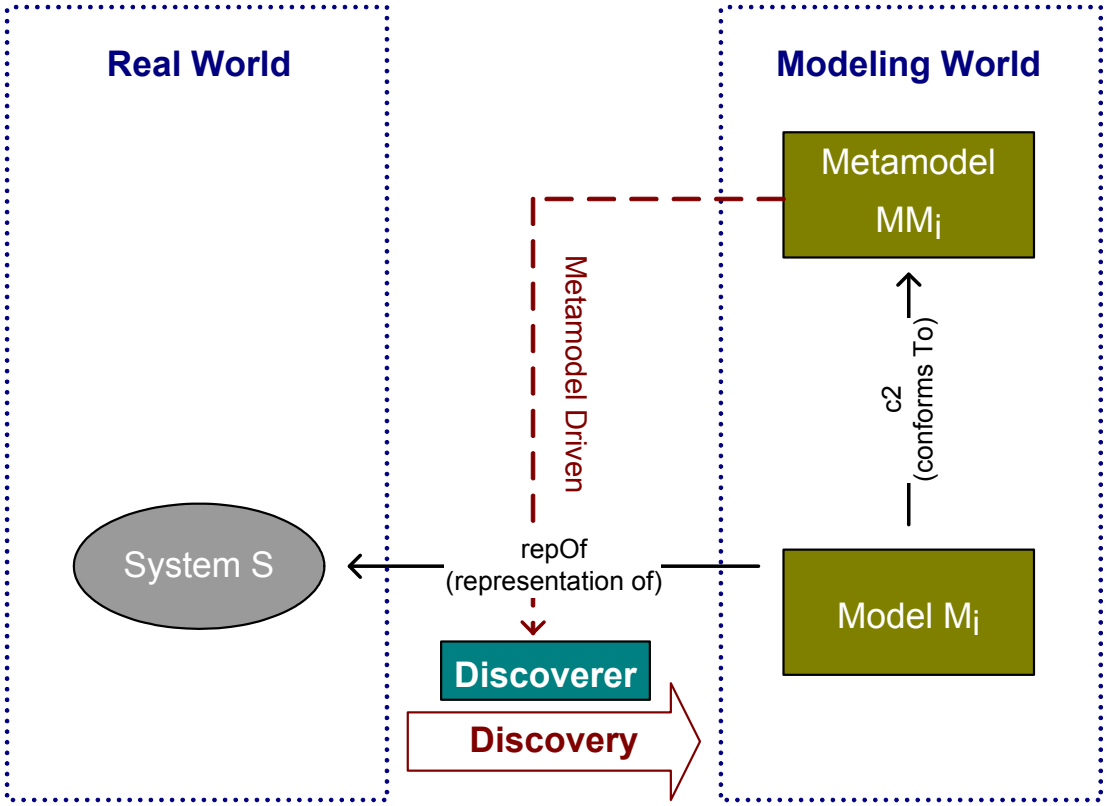
- Definition of knowledge discovery
  - Important issue:

    → **Reverse-engineering of legacy systems**
  - The objective is to apply MDE in order to bring practical solutions to this issue:
    - Extraction of models from legacy systems (applying a metamodel-driven approach)
    - Use of the information they stored

⇨ **Model Discovery**
or **Model-Driven Reverse Engineering** (MDRE)

Knowledge Discovery Principles

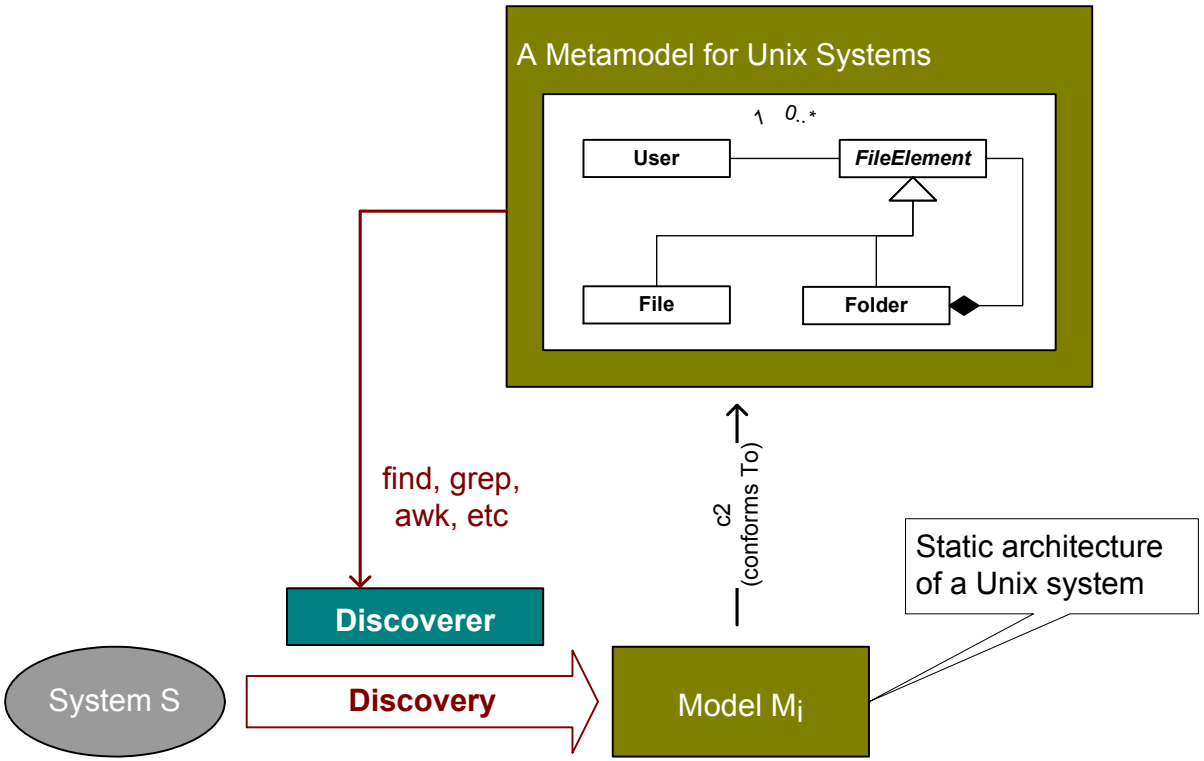
- Definition of knowledge discovery



- Step 1:
  - Define the metamodel
- Step 2:
  - Create the "discoverer"
- Step 3:
  - Run the discoverer to extract model M_i from system S

Knowledge Discovery Principles

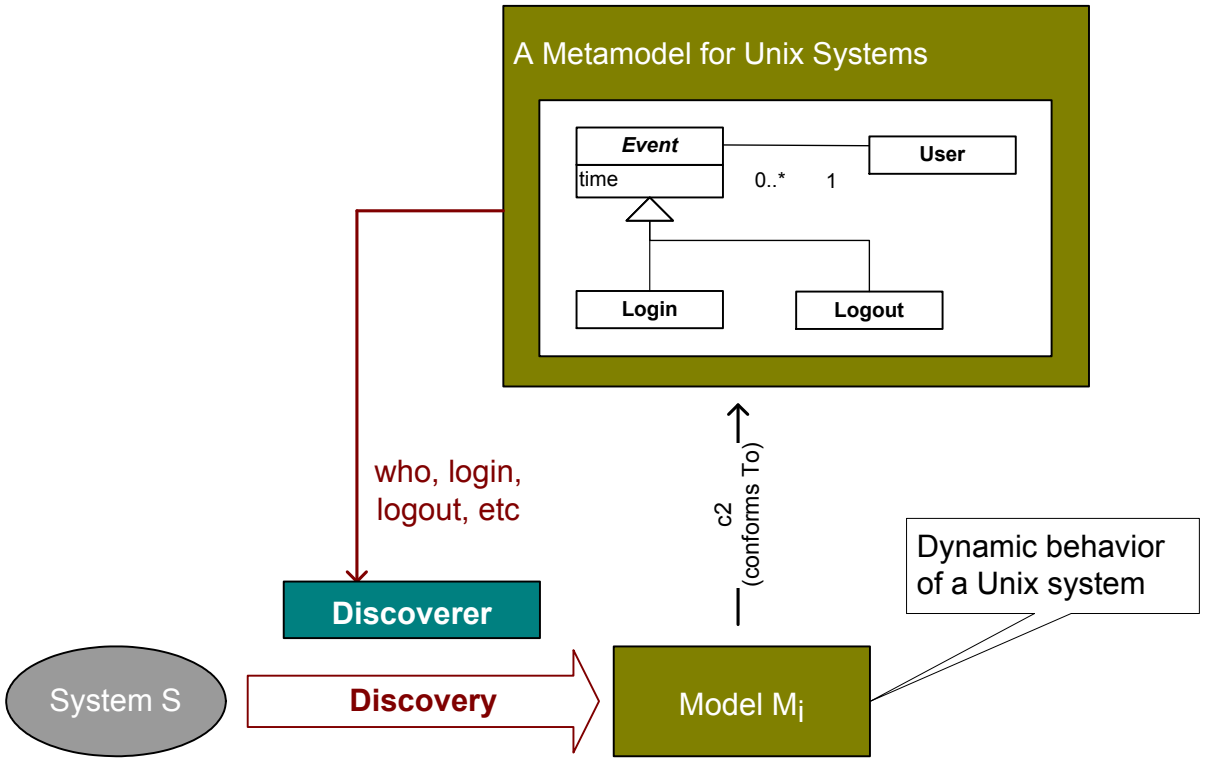
- Definition of knowledge discovery
  - Motivating Examples (1/4)



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

Knowledge Discovery Principles

- Definition of knowledge discovery
  - Motivating Examples (2/4)



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

Knowledge Discovery Principles

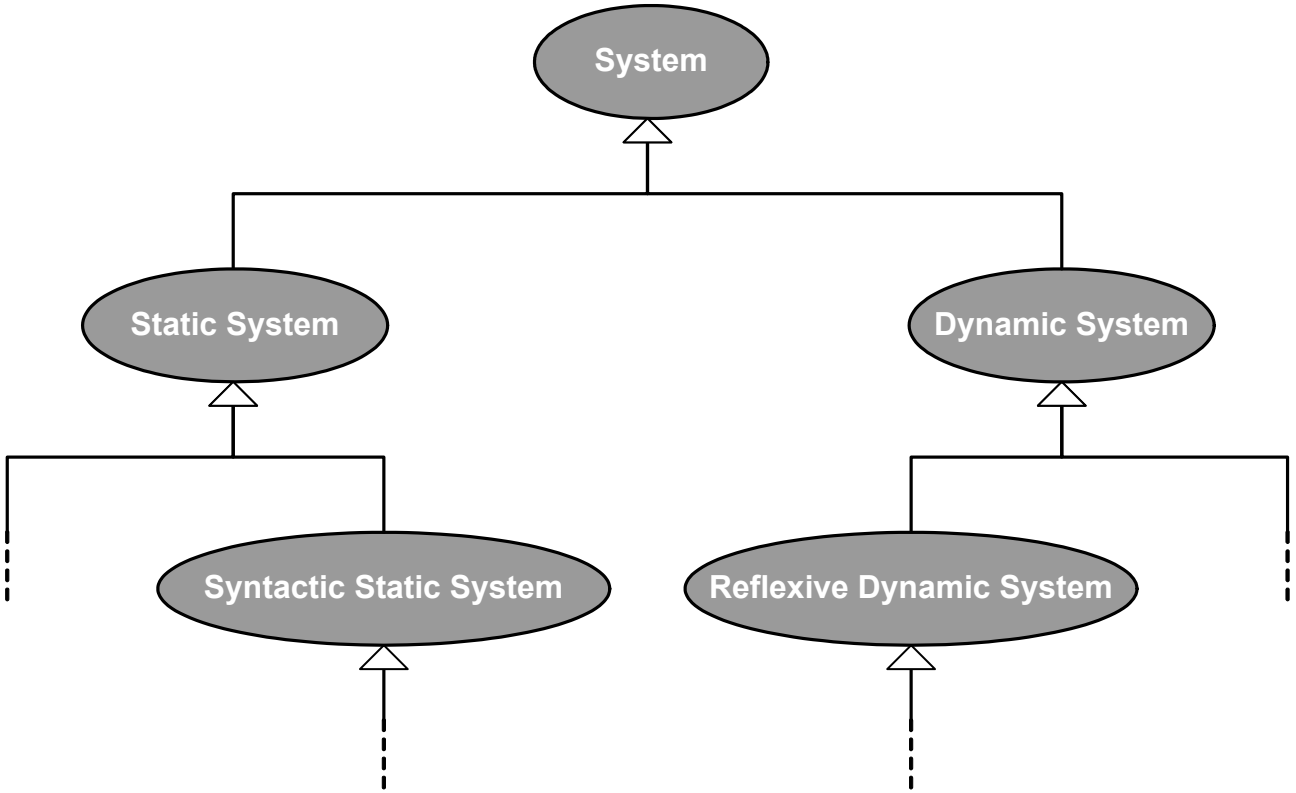
- Definition of knowledge discovery
  - Motivating Examples (3/4)

- Conclusions:

- The same general discovery process is applied in both examples
  - Only the nature of the "discoverers" is changing
- Need for a system classification
  - A decision tree more than an absolute classification
    - Several points of view are possible on the 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

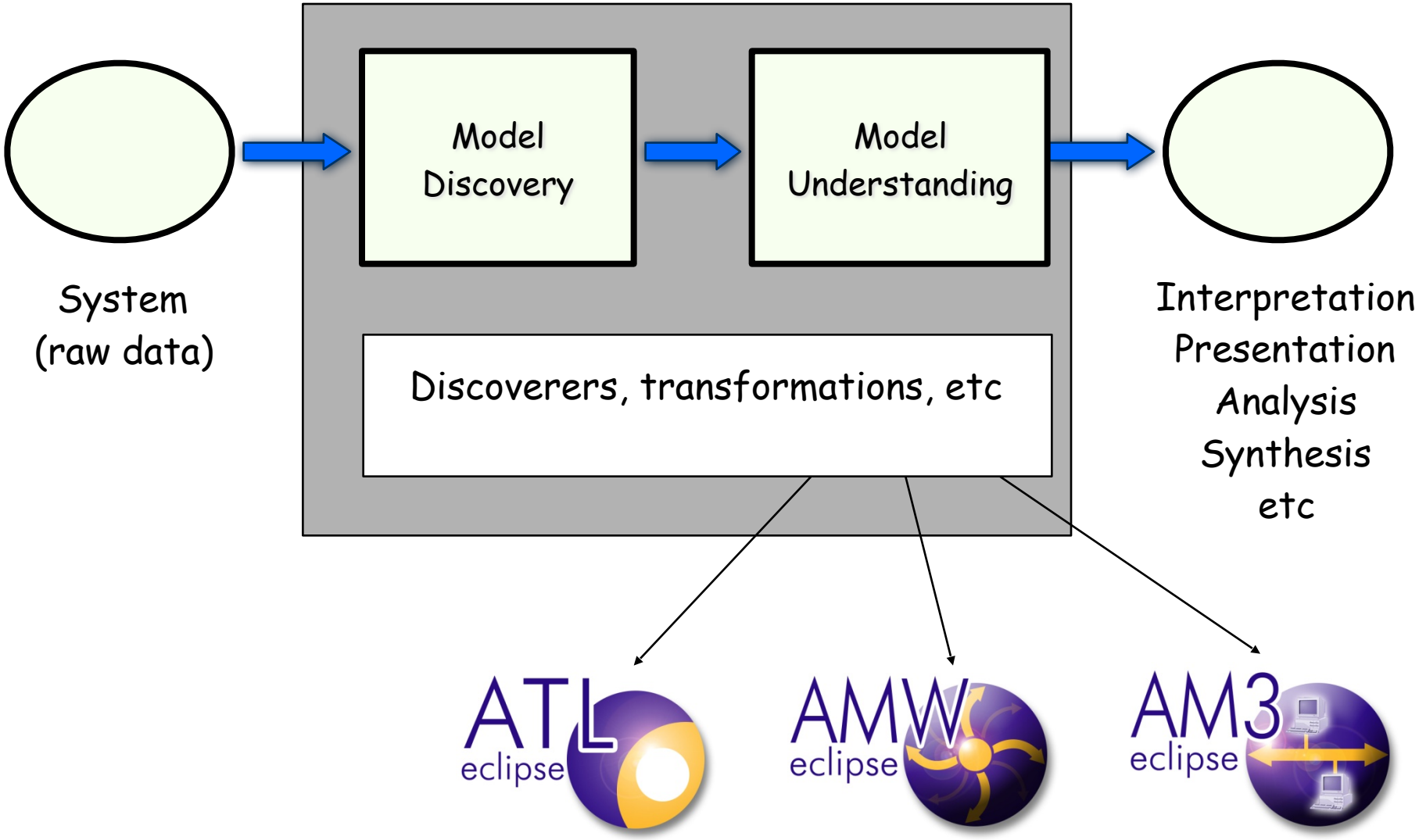
Knowledge Discovery Principles

- Definition of knowledge discovery
  - Motivating Examples (4/4)
    - A possible system classification (simplified version)



Knowledge Discovery Principles

- Description of the overall process



Knowledge Discovery Principles

- Description of the overall process
  - **Model Discovery** → build a view on a system
    - Retrieval of the data from an existing system according to a specific metamodel (expressing a viewpoint)
    - Injection of this data into a model (i.e. the view on the system) conforming to this metamodel
  - **Model Understanding** → extract additional knowledge from the model
    - Process the discovered model in order to:
      - Select parts of it
      - Reorganize it
      - Transform it to another model conforming to a different metamodel
      - Compute additional elements
      - Etc

The Eclipse-GMT MoDisco Project

- Website homepage: http://www.eclipse.org/gmt/modisco/

The Eclipse-GMT MoDisco Project

- Presentation
  - **MoDisco** component's goal:
    - Provide an extensible base framework for performing **metamodel-driven reverse engineering**
  - The key to success:
    - Adoption by leading industrials
    - Development of a wide user community in different application domains

The Eclipse-GMT MoDisco Project

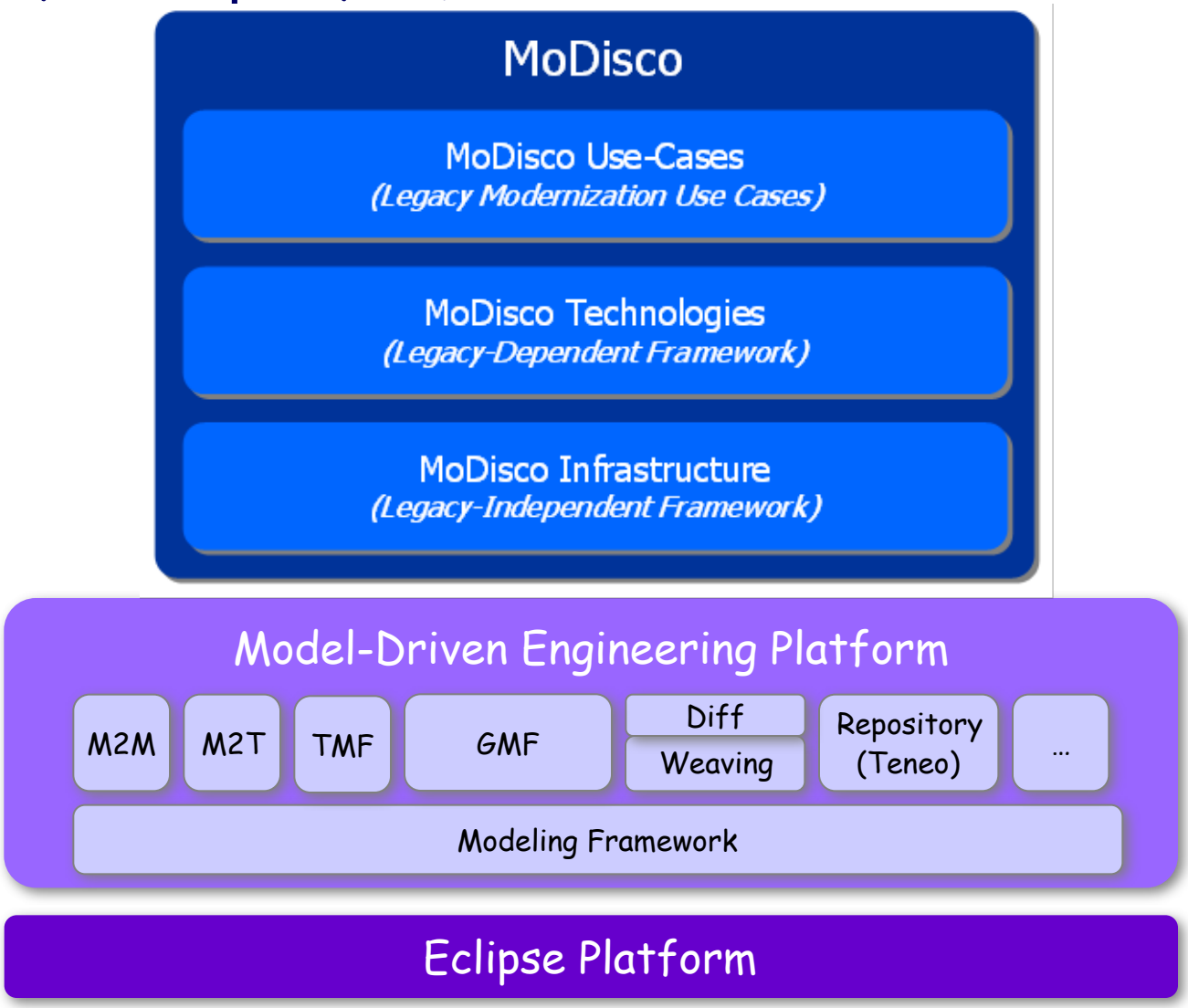
- Presentation
  - A unified model-based approach and a metamodel-driven methodology:
    - Work in the homogeneous world of the models
    - Match different requirements
      - Data integration, tools interoperability, systems migration, etc
    - Use models properties and facilities
      - Transformations,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

The Eclipse-GMT MoDisco Project

- Already available sample tools & use cases
  - Tool Box
    - Java Abstract Syntax Discovery Tool
    - Metrics Visualization Builder
    - ATL model-to-model transformation tool
    - AMW model-weaving tool
    - etc
  - Use Cases
    - Bugzilla Metrics
    - Eclipse/BIRT Project Sample Database
    - Performance-Annotated UML2 State Charts
    - etc

The Eclipse-GMT MoDisco Project

- The future platform



The Eclipse-GMT MoDisco Project

- The future platform
  - Model-Driven Engineering Platform
    - Modeling Framework :
      - Facilities to manage models and metamodels
      - Provides serialization capabilities
      - Provides navigation capabilities
      - Supports main modeling standards (MOF, eCore, XSD, DTD, DSLTools, KM3, …).
    - M2M (Model-to-Model transformations)
      - Supported ITM use-cases :
        - Knowledge Discovery (extraction of viewpoints)
        - Model Understanding (detection of patterns or anti-patterns)
        - Generalization (upgrading level of abstraction)
        - Specialization (downgrading level of abstraction)
        - Architecture Transformation
    - M2T (Model-to-Text transformations)
      - Supported ITM use-cases :
        - Refactoring (regeneration of code)
        - Export to existing tools (generation to proprietary interchange formats)
    - T2M (Text-to-Model transformations)
    - Visualization
      - Generation of modeling tools from the definition of the graphical representation
      - Supported ITM use-cases :
        - Graphical visualization of viewpoints on existing source code (control flow, dependencies, databases, …)
    - Weaving
      - Creation of links between model elements (possibly from different models)
      - Supported ITM use-cases :
        - Traçability during modernization process
        - Mapping between equivalent artifacts (ex : data migration)
        - Mapping between patterns participants (ex : MVC)
    - Diff (Comparison of models)
      - Supported ITM use-cases :
        - Integration of source modifications done during a migration
        - Measurement of transformation
    - Repository (Storage of models)
      - Supported ITM use-cases :
        - Modernizations of big existing applications

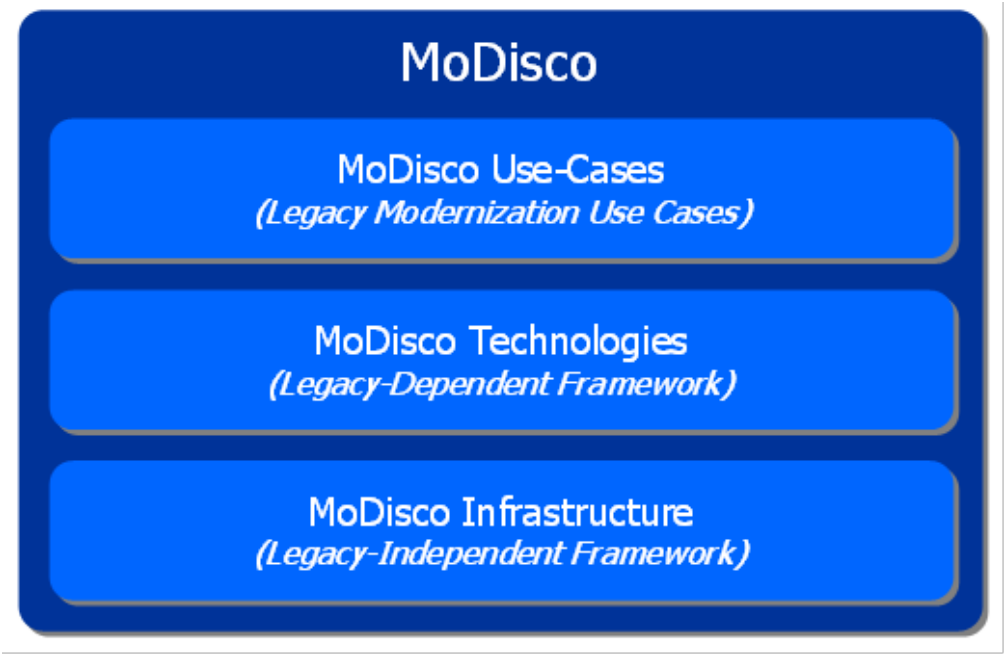
The Eclipse-GMT MoDisco Project

- The future platform

• Use-cases layer: components providing a solution for a specific modernization use-case.

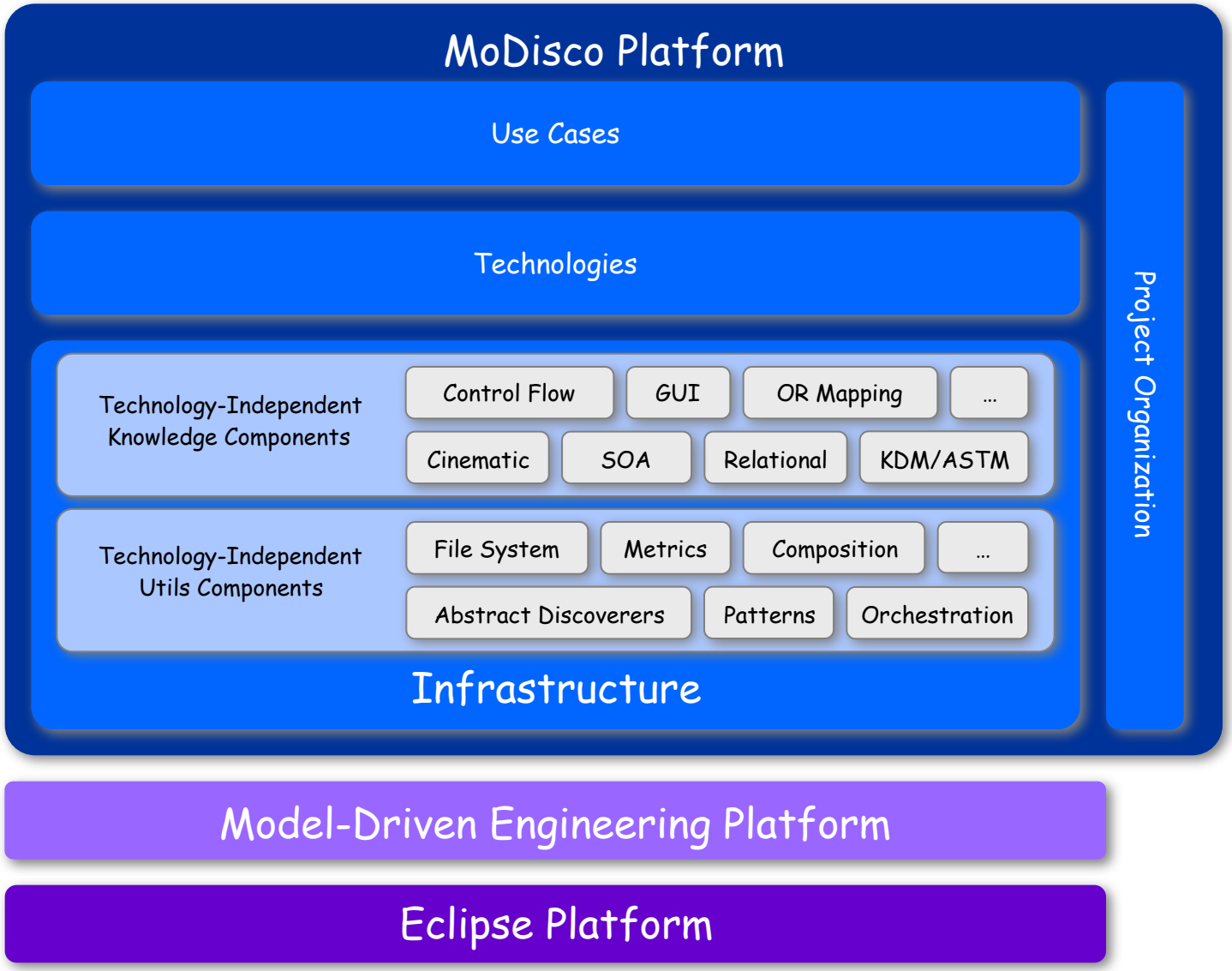
• Technologies layer: components dedicated to one legacy technology but independent from the modernization use case.

• Infrastructure layer: generic components independent from any legacy technology.



The Eclipse-GMT MoDisco Project

- The future platform



The Eclipse-GMT MoDisco Project

- The future platform
  - Technology-Independent Utils Components

- Set of metamodels providing utilities for manipulating the models of the existing systems, independently from the kind of knowledge we need to extract and the technology of the source artifacts

- Examples :
  - FileSystem : model of the physical representation of existing systems (disks, folders, files, source regions, …)
  - Composition of metamodels : metamodels composed of already defined metamodels (ex : Struts = Java + JSP + MVC, Hibernate = Java + Relational + ORMapping)
  - Abstract Discoverers : set of Java Interfaces that discoverers must implement (ex : FolderInjector, FileInjector, DataflowInjector, DatabaseInjector, …)
  - Metrics : model of metrics calculated from an existing application (lines of code, number of components, average complexity, number of defaults, …)
  - Patterns : model describing patterns (or anti-patterns) and model elements conforming to these patterns

The Eclipse-GMT MoDisco Project

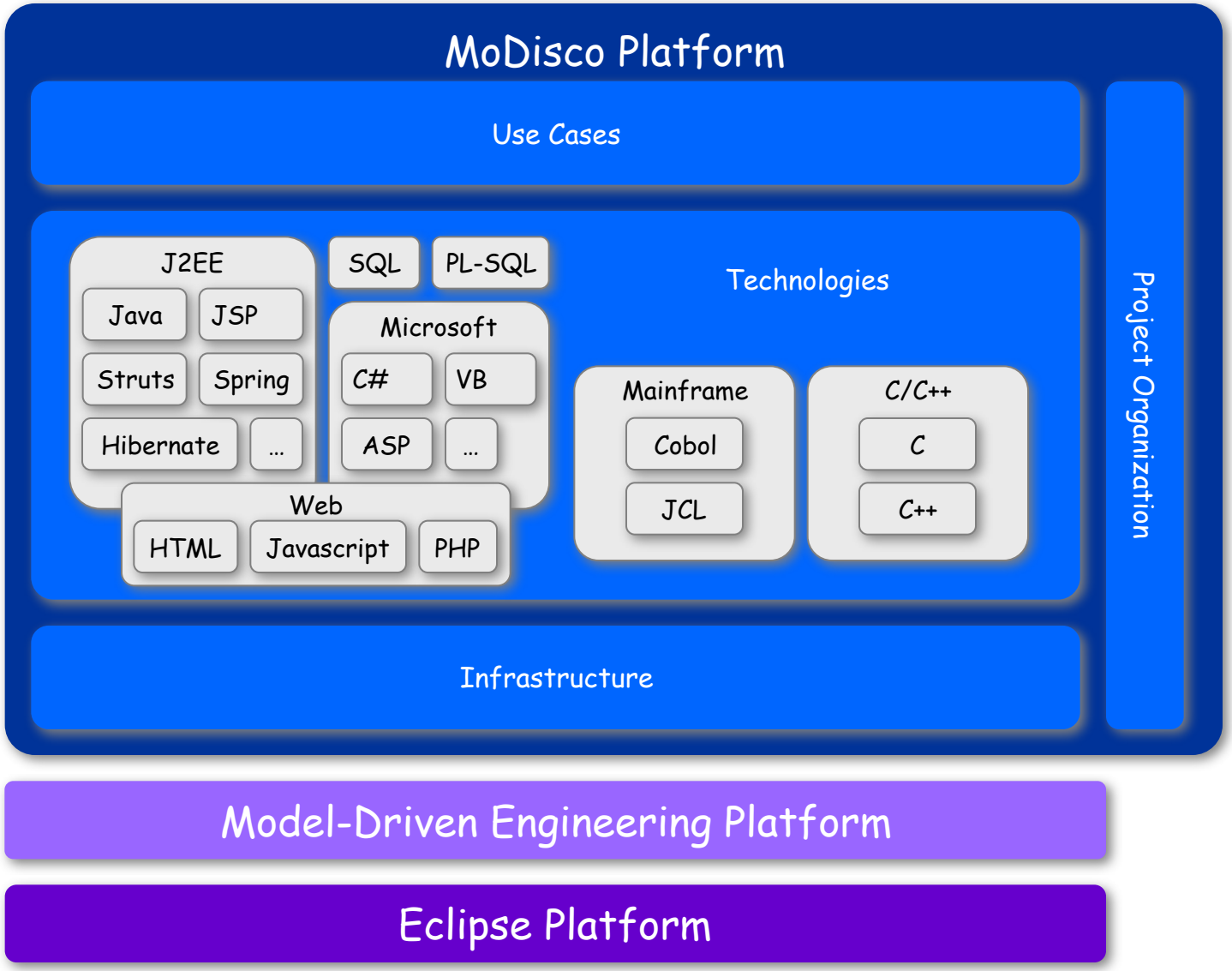
- The future platform
  - Technology-Independent Knowledge Components

- Set of metamodels defining the concepts dedicated to a kind of knowledge we need to extract out of an existing system, independently from the technology of the source artifacts

- Examples :
  - Control Flow : the execution paths of a program
  - GUI : the graphical interface of an application (screens, widgets, events, …)
  - Cinematic : the flow of screens and actions in a graphical interface
  - Relational : the structure of tables and columns in a relational database
  - ORMapping : the way objects are translated to raws into a relational database
  - SOA : the signature of services and their collaboration in a Service-Oriented architecture
  - KDM/ASTM : the OMG standard to describe existing systems independently from their implementation

The Eclipse-GMT MoDisco Project

- The future platform

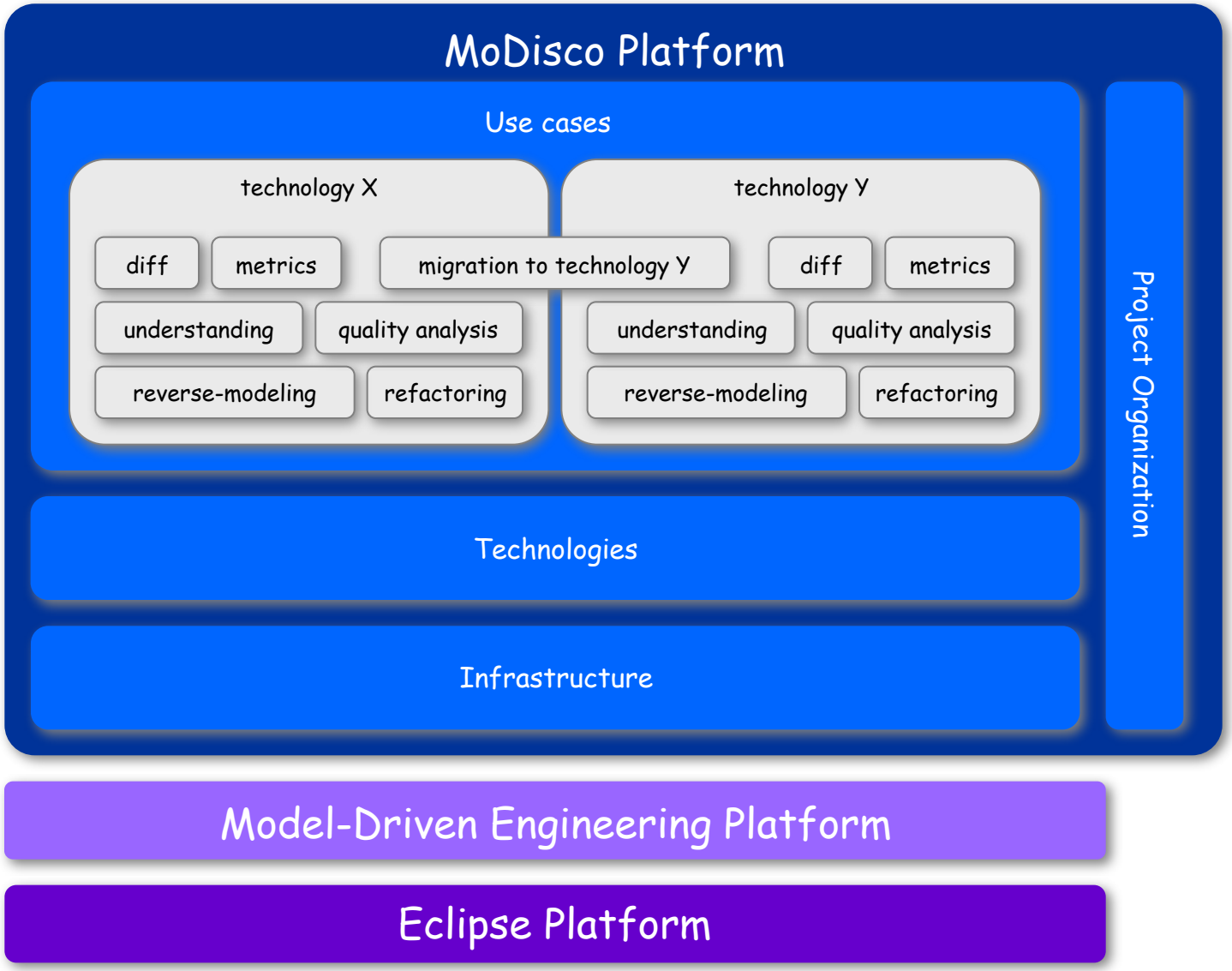


The Eclipse-GMT MoDisco Project

- The future platform
  - Technologies
    - Metamodel
      - Mapping :
        - Source concepts <-> Metamodel concepts
    - Discoverer(s)
      - Partial/Complete
      - Project/Archive
    - Sample(s)
      - Model
      - Source code
    - Model-browser extension
      - icons
      - derived links ?
      - source code association

The Eclipse-GMT MoDisco Project

- The future platform



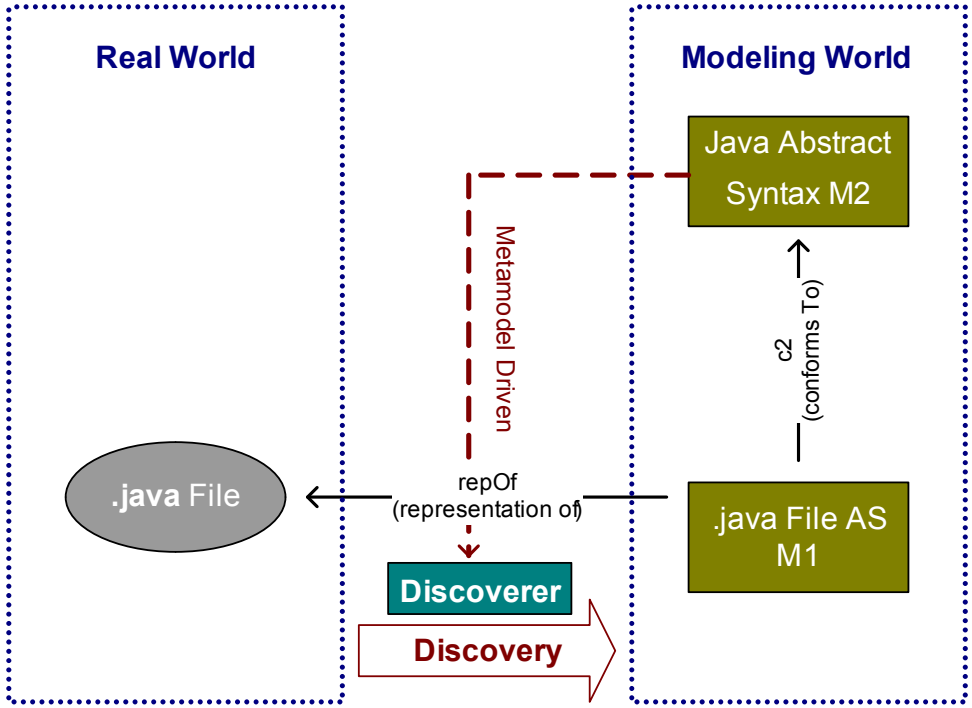
The Eclipse-GMT MoDisco Project

- The future platform
  - Use Cases
  - Set of "off-the-shelf" components for concrete use-cases
    - Work for one source technology
    - The launch of the component, its parametrization and the presentation of the results are integrated with the development environment of the source
  - Examples :
    - metrics : computation of metrics for a language
    - quality analysis : detection of patterns and anti-patterns for a language
    - understanding : utilities to help understanding an existing program
    - reverse-modeling : bridge to existing modeling tools (UML, DSL, …)
    - refactoring : automatic transformations on an existing program
    - diff : structural differences between two versions of the same program
    - migration : automatic transformation from a language to another
  - Use-cases reuse components from MoDisco's SSF and SIF

Possible Applications

- From Java source code (abstract syntax discovery)
  http://www.eclipse.org/gmt/modisco/toolBox/JavaAbstractSyntax/

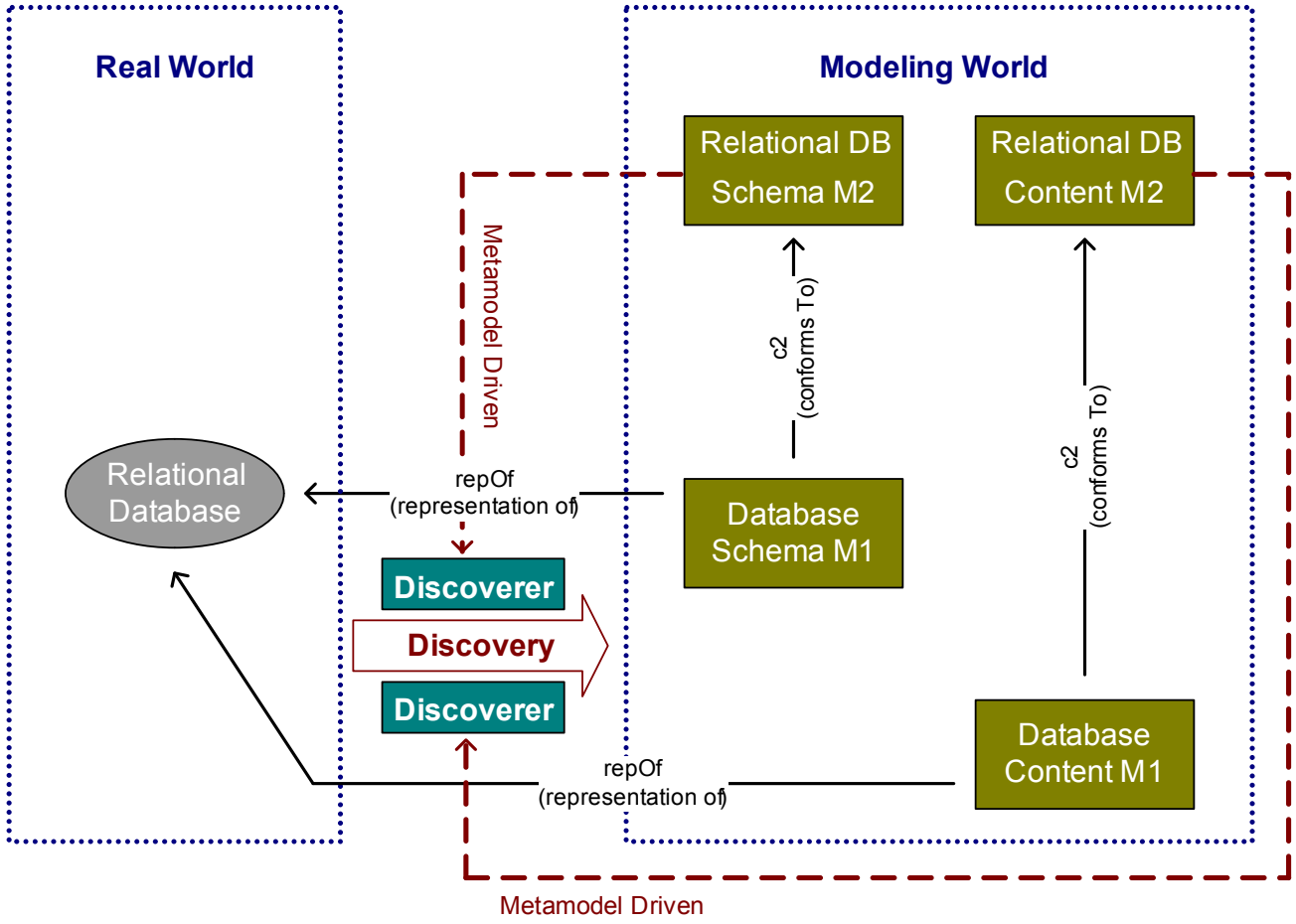
- Example of a produced model (excerpt) in XMI




```
<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>
```

Possible Applications

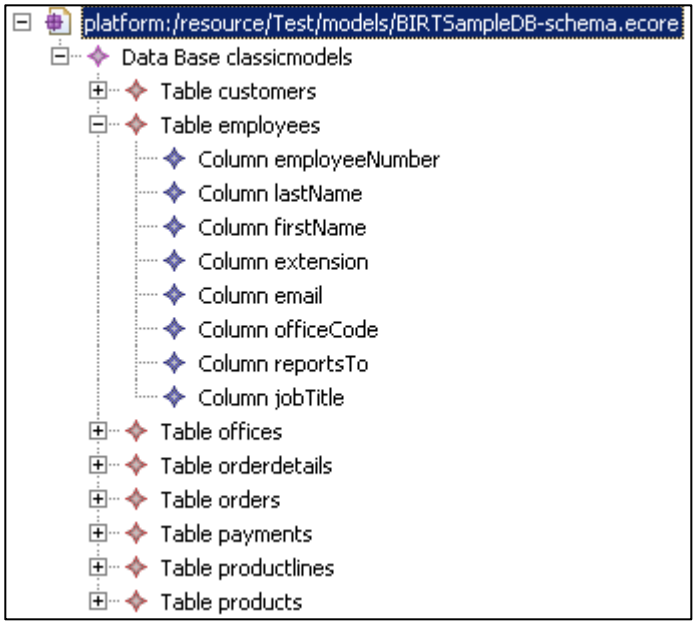
- From a MySQL database (schema + content discovery)
  http://www.eclipse.org/gmt/modisco/toolBox/RelationalDBInformation/



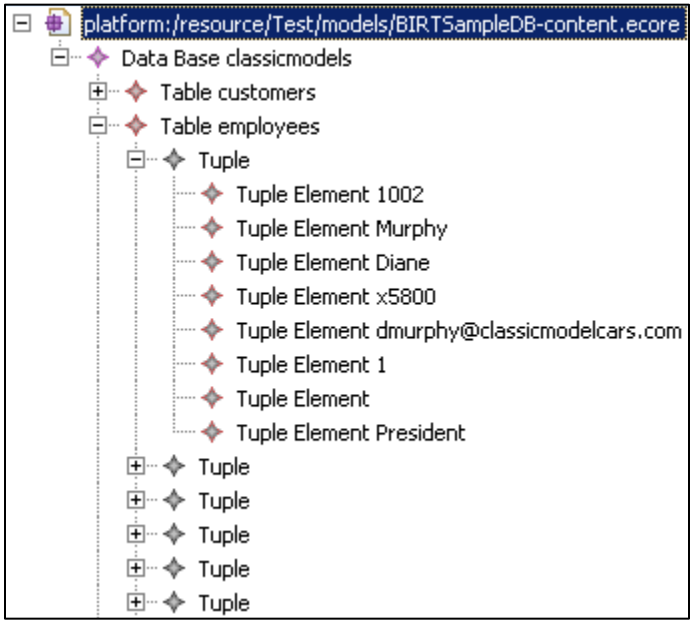
Possible Applications

- From a MySQL database (schema + content discovery)
  http://www.eclipse.org/gmt/modisco/toolBox/RelationalDBInformation/

- Excerpt of a "schema" model

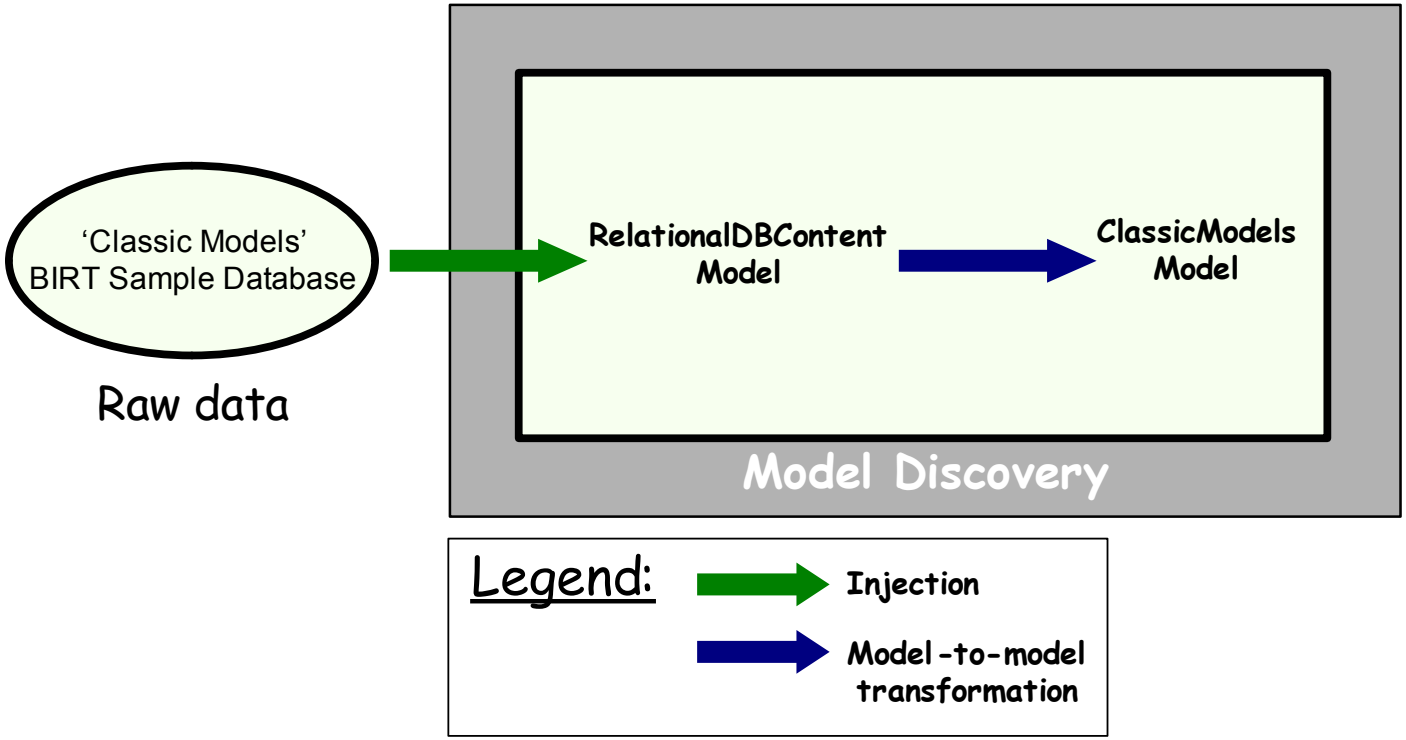


- Excerpt of a "content" model



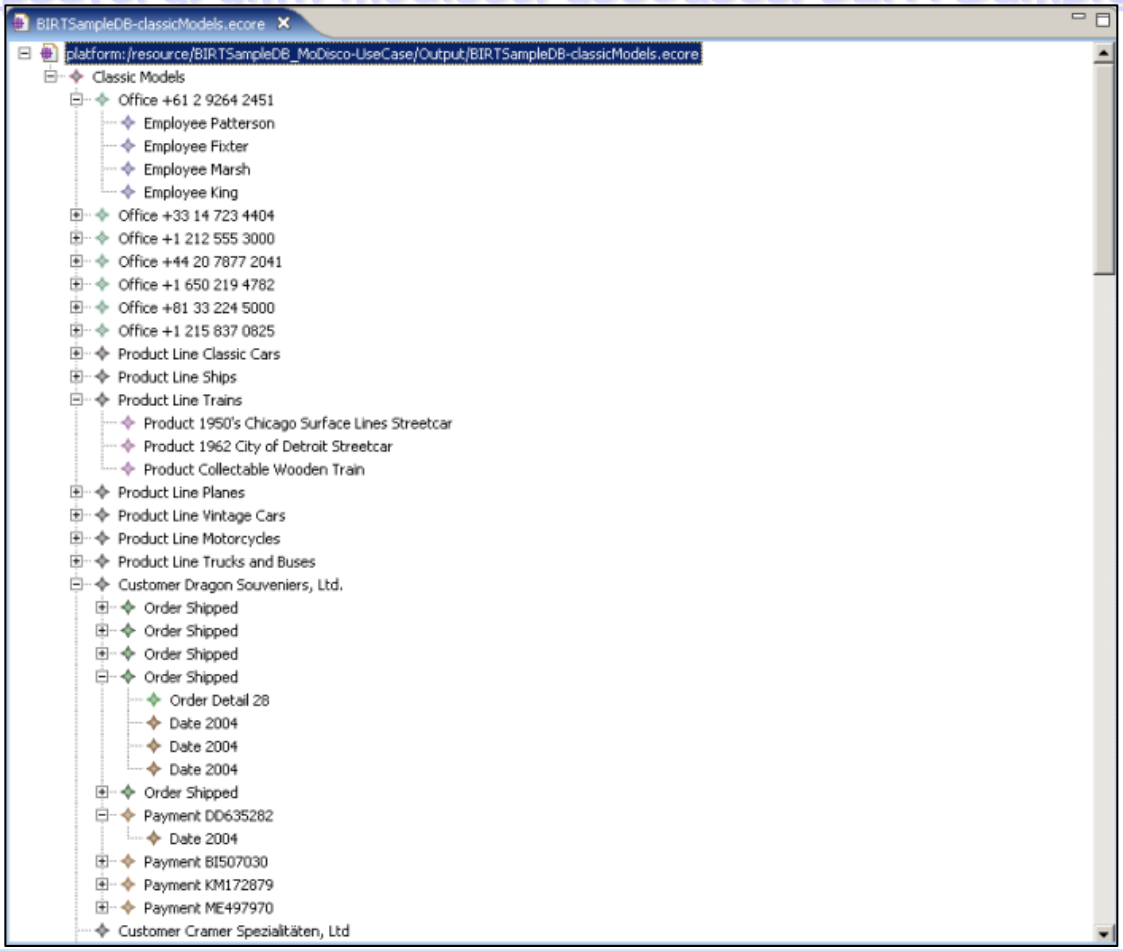
Possible Applications

- From a MySQL database (conversion to a specific metamodel from the discovered "content" model) http://www.eclipse.org/gmt/modisco/useCases/BIRTSampleDB/



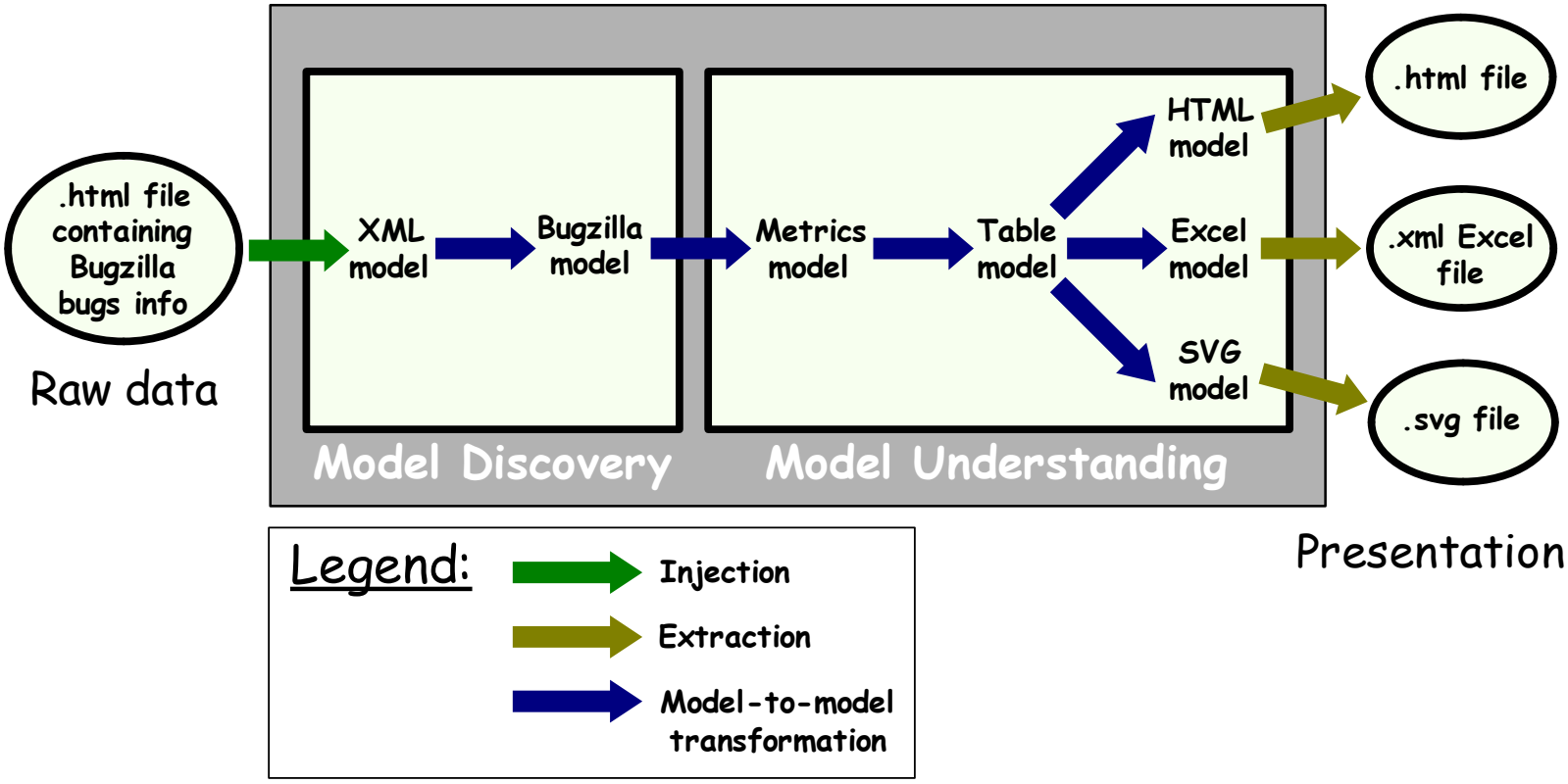
Possible Applications

- From a MySQL database (conversion to a specific metamodel from the discovered "content" model) http://www.eclipse.org/amt/modisco/useCases/BIRTSampleDB/



Possible Applications

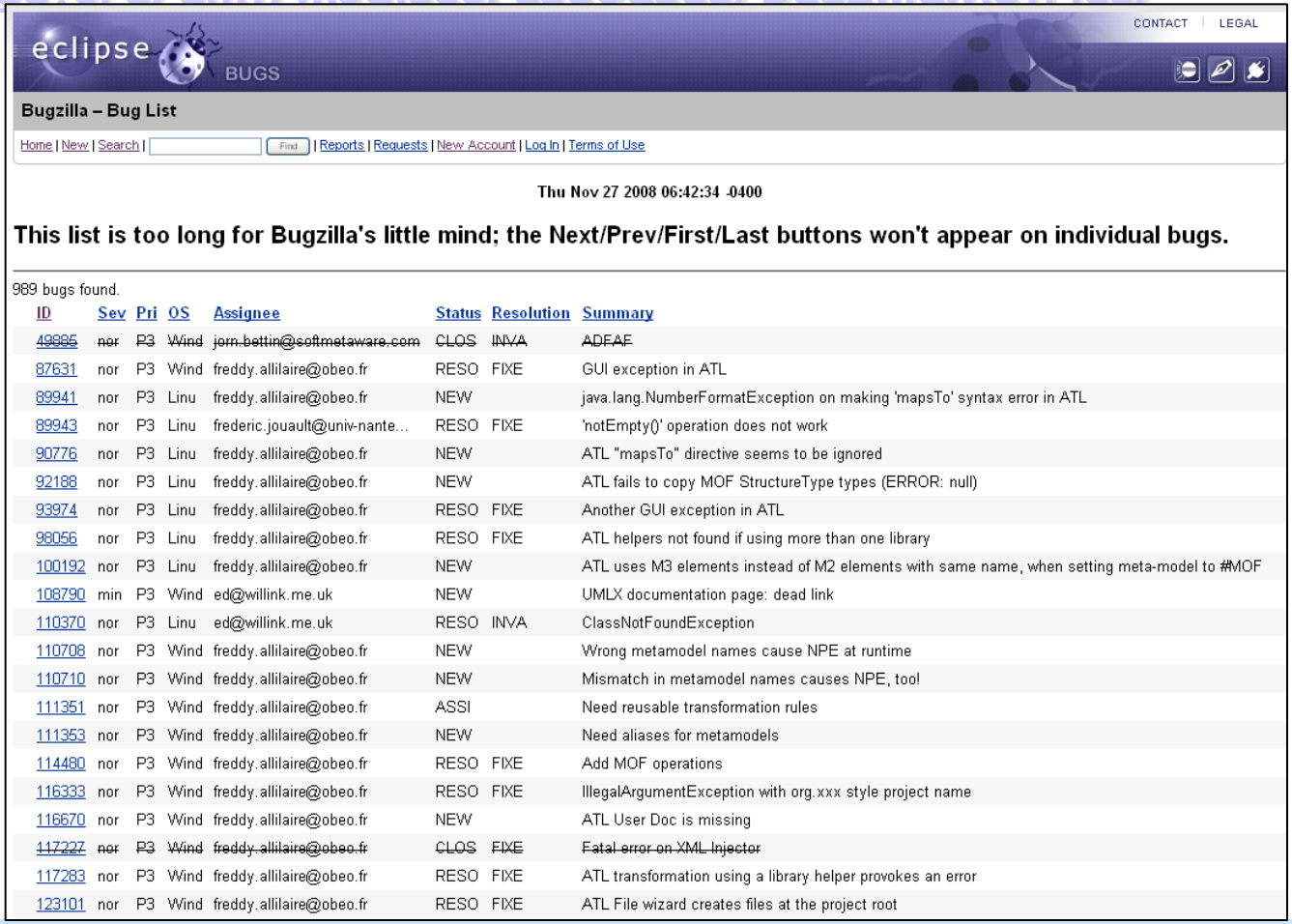
- From Bugzilla data (information discovery + metrics computation + visualization generation) http://www.eclipse.org/gmt/modisco/useCases/BugzillaMetrics/



Possible Applications

- From Bugzilla data (information discovery + metrics computation + visualization generation) http://www.eclipse.org/amt/modisco/useCases/BugzillaMetrics/
- Input

(Exported in XML)



eclipse BUGS

CONTACT | LEGAL

Bugzilla – Bug List

Home | New | Search | Find | Reports | Requests | New Account | Log In | Terms of Use

Thu Nov 27 2008 06:42:34 -0400

This list is too long for Bugzilla's little mind; the Next/Prev/First/Last buttons won't appear on individual bugs.

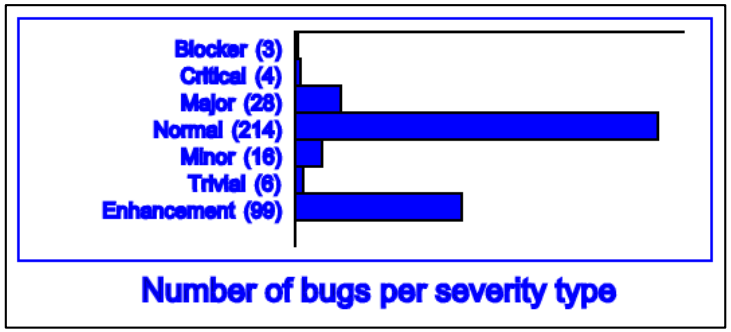
989 bugs found.

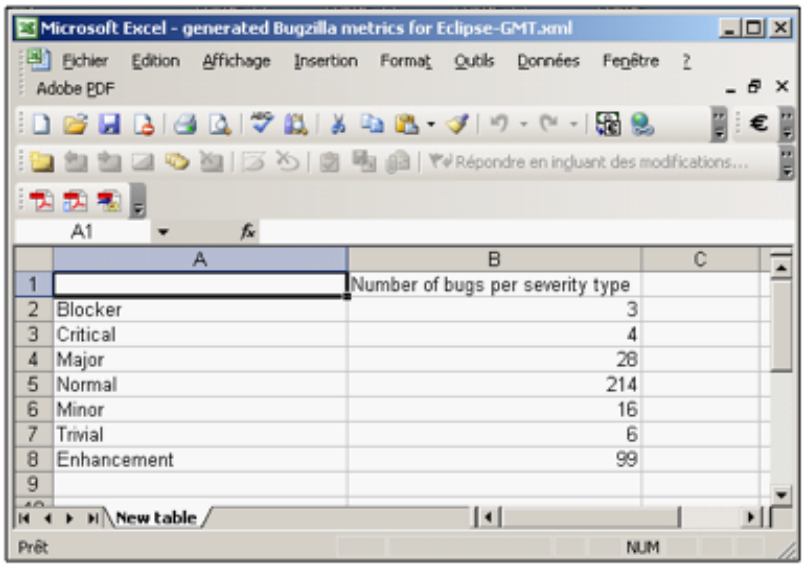
ID	Sev	Pri	OS	Assignee	Status	Resolution	Summary
~~49885~~	~~nor~~	~~P3~~	~~Wind~~	~~jorn.bettin@softmetaware.com~~	~~CLOS~~	~~INVA~~	~~ADFAF~~
87631	nor	P3	Wind	freddy.allilaire@obeo.fr	RESO	FIXE	GUI exception in ATL
89941	nor	P3	Linu	freddy.allilaire@obeo.fr	NEW		java.lang.NumberFormatException on making 'mapsTo' syntax error in ATL
89943	nor	P3	Linu	frederic.jouault@univ-nante...	RESO	FIXE	'notEmpty()' operation does not work
90776	nor	P3	Linu	freddy.allilaire@obeo.fr	NEW		ATL "mapsTo" directive seems to be ignored
92188	nor	P3	Linu	freddy.allilaire@obeo.fr	NEW		ATL fails to copy MOF StructureType types (ERROR: null)
93974	nor	P3	Linu	freddy.allilaire@obeo.fr	RESO	FIXE	Another GUI exception in ATL
98056	nor	P3	Linu	freddy.allilaire@obeo.fr	RESO	FIXE	ATL helpers not found if using more than one library
100192	nor	P3	Linu	freddy.allilaire@obeo.fr	NEW		ATL uses M3 elements instead of M2 elements with same name, when setting meta-model to #MOF
108790	min	P3	Wind	ed@willink.me.uk	NEW		UMLX documentation page: dead link
110370	nor	P3	Linu	ed@willink.me.uk	RESO	INVA	ClassNotFoundException
110708	nor	P3	Wind	freddy.allilaire@obeo.fr	NEW		Wrong metamodel names cause NPE at runtime
110710	nor	P3	Wind	freddy.allilaire@obeo.fr	NEW		Mismatch in metamodel names causes NPE, too!
111351	nor	P3	Wind	freddy.allilaire@obeo.fr	ASSI		Need reusable transformation rules
111353	nor	P3	Wind	freddy.allilaire@obeo.fr	NEW		Need aliases for metamodels
114480	nor	P3	Wind	freddy.allilaire@obeo.fr	RESO	FIXE	Add MOF operations
116333	nor	P3	Wind	freddy.allilaire@obeo.fr	RESO	FIXE	IllegalArgumentException with org.xxx style project name
116670	nor	P3	Wind	freddy.allilaire@obeo.fr	NEW		ATL User Doc is missing
~~117227~~	~~nor~~	~~P3~~	~~Wind~~	~~freddy.allilaire@obeo.fr~~	~~CLOS~~	~~FIXE~~	~~Fatal error on XML Injector~~
117283	nor	P3	Wind	freddy.allilaire@obeo.fr	RESO	FIXE	ATL transformation using a library helper provokes an error
123101	nor	P3	Wind	freddy.allilaire@obeo.fr	RESO	FIXE	ATL File wizard creates files at the project root

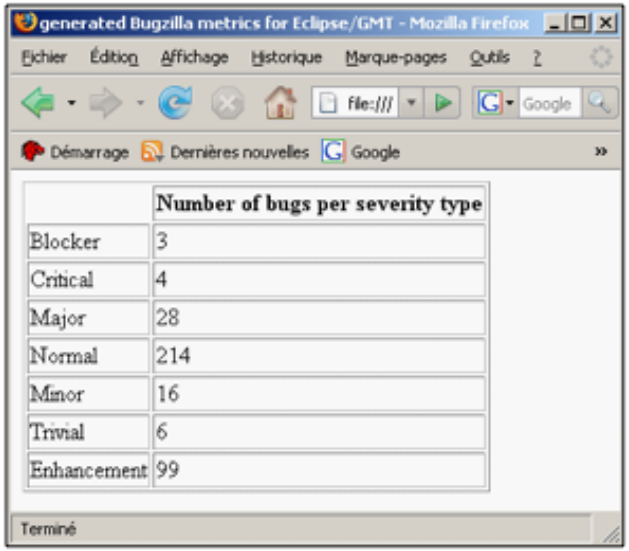
Possible Applications

- From Bugzilla data (information discovery + metrics computation + visualization generation) http://www.eclipse.org/gmt/modisco/useCases/BugzillaMetrics/
- Outputs

  (HTML, SVG, Excel…)

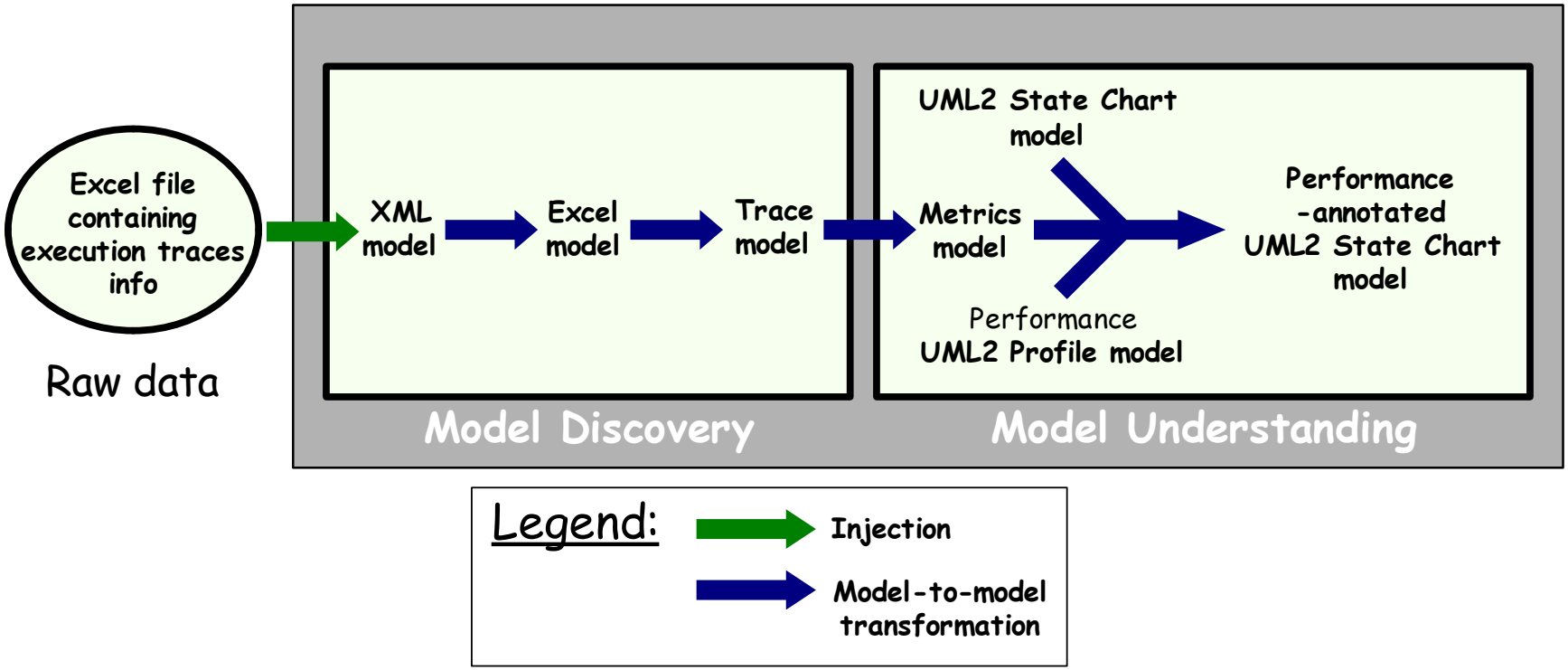






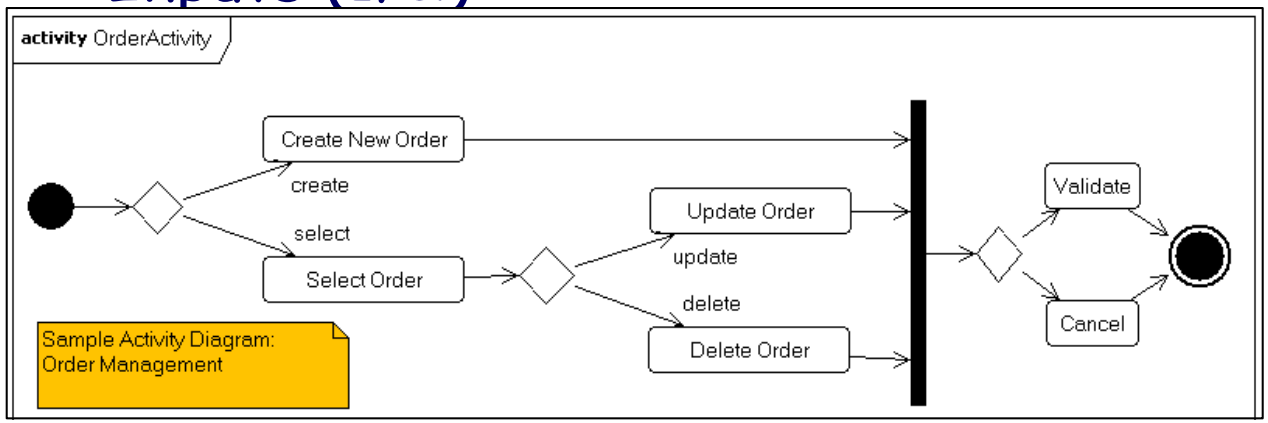
Possible Applications

- From an Excel file providing execution traces (traces discovery + metrics computation + profile application)
  http://www.eclipse.org/gmt/modisco/useCases/PerformanceAnnotatedUmlStateCharts/

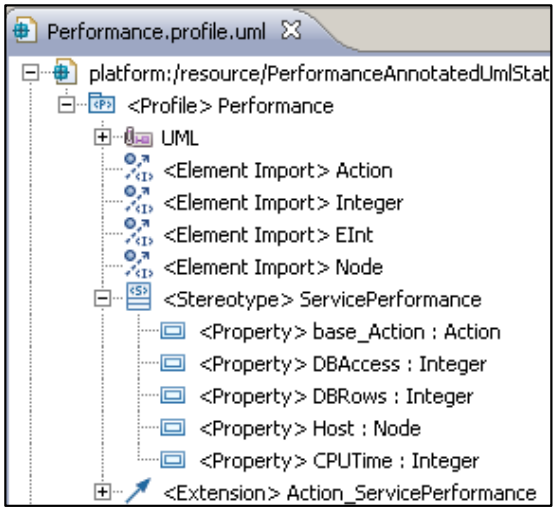


Possible Applications

- From an Excel file providing execution traces (traces discovery + metrics computation + profile application)
  http://www.eclipse.org/gmt/modisco/useCases/PerformanceAnnotatedUmlStateCharts/
- Inputs (1/2)



- UML2 model (state chart)

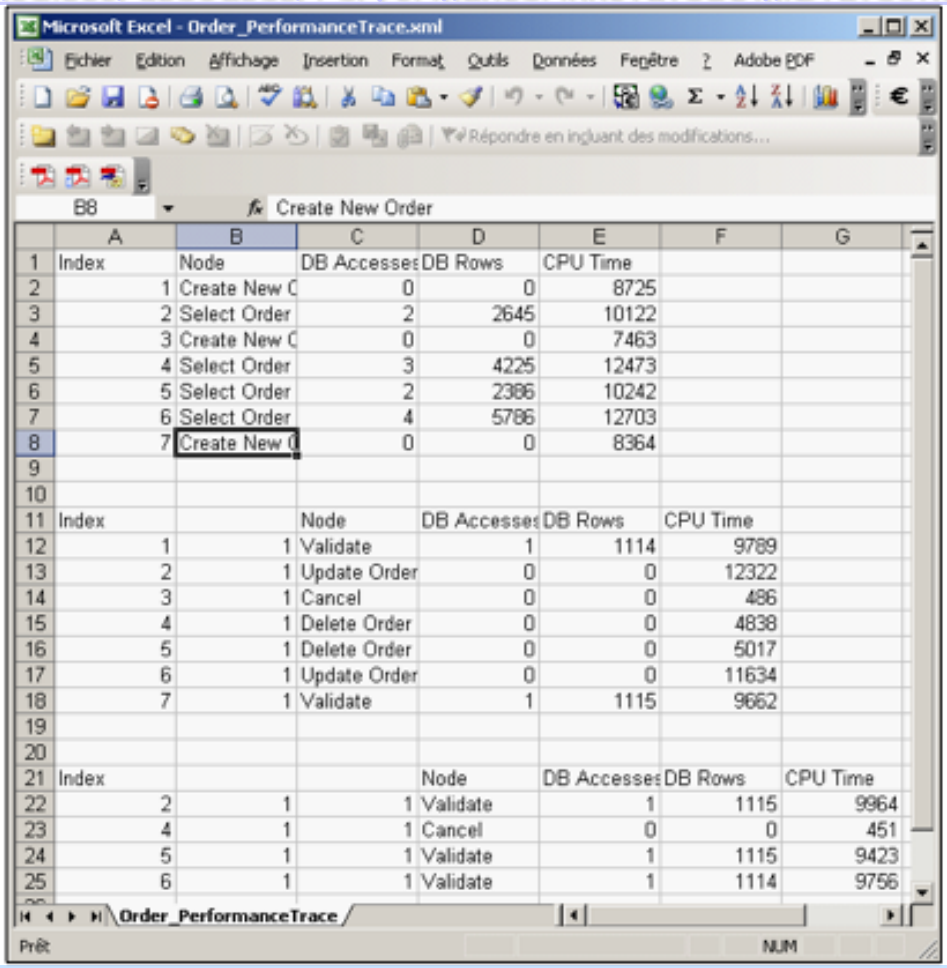


- UML2 model
("Performance" profile)

Possible Applications

- From an Excel file providing execution traces (traces discovery + metrics computation + profile application)
  http://www.eclipse.org/gmt/modisco/useCases/PerformanceAnnotatedUmlStateCharts/
- Inputs (2/2)
  - Excel file

(execution traces)



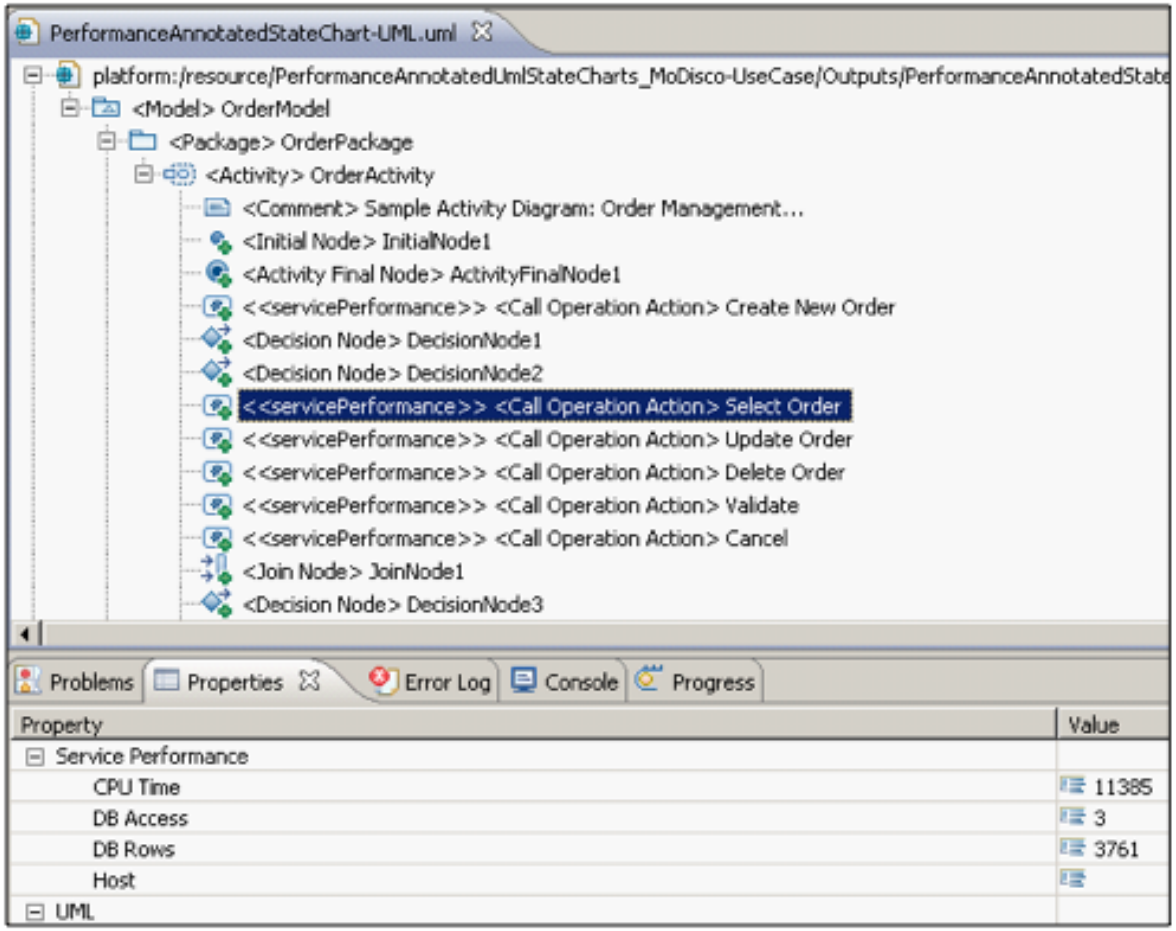
	A	B	C	D	E	F	G
1	Index	Node	DB Accesses	DB Rows	CPU Time		
2	1	Create New C	0	0	8725		
3	2	Select Order	2	2645	10122		
4	3	Create New C	0	0	7463		
5	4	Select Order	3	4225	12473		
6	5	Select Order	2	2386	10242		
7	6	Select Order	4	5786	12703		
8	7	Create New C	0	0	8364		
9							
10							
11	Index		Node	DB Accesses	DB Rows	CPU Time	
12	1	1	Validate	1	1114	9789	
13	2	1	Update Order	0	0	12322	
14	3	1	Cancel	0	0	486	
15	4	1	Delete Order	0	0	4838	
16	5	1	Delete Order	0	0	5017	
17	6	1	Update Order	0	0	11634	
18	7	1	Validate	1	1115	9662	
19							
20							
21	Index			Node	DB Accesses	DB Rows	CPU Time
22	2	1	1	Validate	1	1115	9964
23	4	1	1	Cancel	0	0	451
24	5	1	1	Validate	1	1115	9423
25	6	1	1	Validate	1	1114	9756

Possible Applications

- From an Excel file providing execution traces (traces discovery + metrics computation + profile application)
  http://www.eclipse.org/gmt/modisco/useCases/PerformanceAnnotatedUmlStateCharts/

- Output

- UML2 model (state chart with "Performance" profile applied)



A Concrete Application:

Legacy System Interoperability Discovery

- To be completed…