Collaborative Modeling

Papyrus and Modeling Technologies
Developer Summit

Mikaël Barbero
Maximilian Koegel
Philip Langer

October 28th, 2014
Collaboration on EMF-based models
Why support for collaborative modeling is **Important**

Industrial application of model-based engineering (MBE)

- Industrial projects are large
- Teams of developers need to collaborate
Why support for collaborative modeling is Important

Efficient collaboration in MBE

- Concurrent changes on models
- Multiple versions of models
- Compare, review, and merge model versions
Why support for collaborative modeling is Important

No proper collaboration support: no efficient work

- Inefficiency hurts the benefits of MBE
- Success of model-based projects is at stake

"It's so buggy. Frustrating."
Gerrit process for Collaborative Development
Why support for collaborative modeling is Challenging

Graph structure instead of flat text
- Graph matching and differencing
- Combinatorial complexity (change types, languages)

Diagrams, logical models, files
- Graphical differencing
- Multiple views (diagrams) on one or more models
- Model may spread multiple files

Models serve diverse purposes for different stakeholders
- Sketch, design, implementation
- Different requirements
  e.g., for visualizing changes, validation, reviewing
Why support for collaborative modeling is Challenging

Critical non-functional requirements

- **Reliability**
  Error destroys work and time
  Risk of losing the users’ trust
  Mitigates potential benefits of MBE

- **Understandability**
  Models are the most valuable assets of the project
  Users have to understand the models’ evolution
  Communication (e.g., comments) must be on model level

- **Tool integration and efficient workflows**
  Collaboration tools need to be tightly integrated
  Easy for plain text, hard for diagrams and models
  Modeling editors, diffing tools, reviewing platforms, ...
Why support for collaborative modeling is Challenging

Reliability of merge
- Must be safe and complete
- No unexpected changes

Understandability of changes
- Changes and their impact
- Model and diagram changes

Adaptability to specific needs
- Different perspectives
- Dedicated change reports (differential vs. complete, document vs. model)

Understandability of review
Review/validation results on model/diagram level
Why support for collaborative modeling is Challenging

Integration of many tools?
Integration of many (model)? (collaboration)?

Continuous Integration

Review System

emf

git

eclipse

Impedance Mismatch (line vs model)

OCL

R4E

Clone

Merge

Push for review

Feedback

Rate & comment

Diff

Result

Fetch patch

Fetch patch
What we cover today in

This Session

● Comparing models with EMF Compare
  ○ Model and diagram comparison
  ○ Comparing models split across resources
  ○ Reliability of differencing and merging

● Versioning models with Git
  ○ Integration of EGit and EMF Compare
  ○ Logical model resolution and merging

● Model review and workflow support
  ○ Plans on integration with Gerrit connector
  ○ Plans on integration with Papyrus
Comparing Models with EMF Compare
So far

Comparing Models
Now
EMF Compare
Now

EMF Compare
EMF Compare

Demo

- Preview of merge consequences
- Filters and Groups
- Diagrams comparison
What about the Consistency?
EMF Models are

Large Objects Graph
EMF Models are

Large Objects Graph
EMF Models are

Large Objects Graph
EMF handles cross resources references as Proxies
Very few EMF based tools gracefully support Resolution Failure
Solution: always do modifications with all

Resources Loaded

"Do not use a cannon to kill a mosquito"
— Confucius
What about the Performance?
How to create the Graph of Resources?
Search for model files in the Workspace
and...

Resolve All
Strategies to Optimize the Resolution of the Graph of Resources
First Solution

Limit the Scope of Lookup
First Solution

Limit the Scope of Lookup

● Workspace
  ○ Load every files in the **workspace**
● Project
  ○ Load every files in the containing **project** of the file from which the resolution is started
● Container
  ○ Load every files in the containing **folder** of the file from which the resolution is started
● Outgoing
  ○ Load all **reachable** (via cross-references & containment references) files starting with the file from which the resolution started
Second Solution

Multithreaded Resolution

[Image of a software preferences window with a checkbox for disabling multi-threading during model resolution]
Second Solution

Multithreaded Resolution

A

B

C

G

E

D

F

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A ➔ B ➔ C

G ➔ E ➔ D

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

Loading Thread

Unloading Thread

To be Loaded

To be Unloaded

Already Browsed
Second Solution

Multithreaded Resolution

- Loading Thread
- Unloading Thread

To be Loaded:

To be Unloaded:

Already Browsed:
Second Solution

Multithreaded Resolution

Diagram:

- A → B → C
- E → F → D

- B, G: To be Loaded
- D: To be Unloaded
- A: Already Browsed

Thread Directions:
- Blue: Loading Thread
- Red: Unloading Thread
Multithreaded Resolution

Second Solution

Loading Thread

Unloading Thread

To be Loaded

To be Unloaded

Already Browsed
Second Solution

Multithreaded Resolution

- **To be Loaded**
  - A
  - B
  - C
  - G

- **To be Unloaded**
  - A
  - B

- **Already Browsed**
  - A
  - B

- **Loading Thread**
  - Blue arrow

- **Unloading Thread**
  - Red arrow
Second Solution
Multithreaded Resolution

A

G

B

C

E

D

F

G

To be Loaded

A

To be Unloaded

A
B

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

Loading Thread

Unloading Thread

To be Loaded

To be Unloaded

Already Browsed
Second Solution

Multithreaded Resolution

A → B → C

G → E → D

<table>
<thead>
<tr>
<th>G</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>To be Loaded</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>To be Unloaded</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Already Browsed</td>
<td></td>
</tr>
</tbody>
</table>

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A ➔ B ➔ C

G ➔ E ➔ F ➔ D

To be Loaded

G C

To be Unloaded

B

Already Browsed

A B

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A → B → C

E, D

To be Loaded

B

To be Unloaded

A, B, C, G

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A ➔ B ➔ C

E ➔ D

To be Loaded

B C G

To be Unloaded

A B C G

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

- To be Loaded:
  - D
  - G

- To be Unloaded:
  - A
  - B
  - C
  - G
  - E

- Already Browsed:
  - F

- Loading Thread:
  - D
- Unloading Thread:
  - A
Second Solution

Multithreaded Resolution

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A → B → C

F

G

E → D

Already Browsed

To be Loaded

To be Unloaded

A  B  C  G  E  D

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

Diagram:

- **A** → **B** → **C**
- **E** → **F** → **G**
- **D**

Lists:

- **To be Loaded**
  - F
  - [ ] [ ] [ ] [ ] [ ]

- **To be Unloaded**
  - [ ] [ ] [ ] [ ] [ ] [ ]

- **Already Browsed**
  - A  B  C  G  E  D

Threads:

- **Loading Thread**
  - Blue arrows

- **Unloading Thread**
  - Red arrows
Second Solution

Multithreaded Resolution

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A → B → C
G → E → D

F
To be Loaded

D
To be Unloaded

A  B  C  G  E  D
Already Browsed

Loading Thread
Unloading Thread
Second Solution

Multithreaded Resolution

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A ➔ B ➔ C

G ➔ E ➔ D

F

To be Loaded

To be Unloaded

A B C G E D F

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A → B → C

G → E → D

F

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A

G

B

C

E

D

F

To be Loaded

F

To be Unloaded

A B C G E D F

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

A

B

C

D

E

G

F

To be Loaded

To be Unloaded

Already Browsed

Loading Thread

Unloading Thread
Second Solution

Multithreaded Resolution

Diagram showing a multithreaded resolution with nodes labeled A, B, C, E, F, and G, and arrows indicating the loading and unloading paths.
Third Solution

Minimize the Scope

Ancestor

Left (local)  Right (remote)
Third Solution

Minimize the Scope

Ancestor

Left (local)  Right (remote)
Third Solution

Minimize the Scope
Fourth Solution

Do not do any Model Resolution
Increasing

Reliability of EMF Compare

- Increase reliability of EMF Compare
  - Increase test coverage
  - Reduce number of faults

- Challenges
  - Model matching, differencing, and merging is inherently complex
  - Increased reliability requires comprehensive testing
  - Comprehensive testing is difficult for model merging: combinatorially sized search space

- Steps
  - Bottom-up fuzz testing
  - Isolation and fixing of bugs
Increasing reliability of EMF Compare with Fuzz Testing

- **Goal**
  - Find inputs that cause unexpected behavior of a software
  - Good cost effectiveness ratio
  - Suitable for combinatorially sized and practically infinite input space
  - Enable reproducibility of unexpected behavior for bug isolation

- **Basic idea**
  - Run software under test with *pseudo random but metamodel-conform inputs*
  - Monitor software under test during execution
  - Check for crashes and failures

- **Key ingredients**
  - Fuzz generator creates *pseudo random inputs*
  - Test coverage of the fuzzed input
  - Input-independent postconditions
Increasing reliability of EMF Compare with

Fuzz Testing: Two-way Merge

● **Goal**
  ○ Find combination of model changes
  ○ Applied to an EMF model
  ○ Causing erroneous matching, diffing, and/or merging

● **Fuzz Testing Framework for EMF tools**
  ○ Part of EMFStore
  ○ Repeatedly run JUnit tests with fuzzed input
  ○ Pseudo-randomly mutate EMF models

● **Basic Idea**
  ○ Generate an EMF model conforming to a metamodel: \( m_1 \)
  ○ Copy and mutate this model: \( m_2 \)
  ○ Compare and merge in both directions: \( m_1 \rightarrow 2 \) and \( m_2 \rightarrow 1 \)
  ○ Compare \( m_1 \) with \( m_2 \rightarrow 1 \) and \( m_2 \) with \( m_1 \rightarrow 2 \)
  ○ Assert that there are no differences
    ■ Between \( m_1 \) and \( m_2 \rightarrow 1 \)
    ■ Between \( m_2 \) and \( m_1 \rightarrow 2 \)
Fuzz Testing: Two-way Merge

- **Statement coverage of fuzz test: 40.5%**
  - 100 model elements
  - 500 seeds
  - Right-to-left and left-to-right merge: 1,000 tests

- **Results of 1,000 tests**
  - Initial error rate: 4.7% (47 errors)
  - Initial failure rate: 51.1% (511 failures)

- **Does not mean that there are 511 + 47 bugs**
  - $x$ bugs that occurred $(511 + 47)/x$ times
Increasing reliability of EMF Compare

Fuzz Testing: Two- & Three-way Merge

- **Overall statement coverage:** 53%
  - 100 model elements
  - 500 seeds
  - 2,000 two-way merge tests
  - 1,000 three-way merge tests

- **Overall results of 3,000 tests**
  - Initial error rate: 7.56 % (227 errors)
  - Initial failure rate: 61.4 % (1,842 failures)

- **Isolation and fixing**
  - Extract minimal unit test reproducing wrong behavior
  - Find, report, and fix underlying bug
Fuzz Testing: Two- & Three-way Merge

- Overall statement coverage: 53%
  - 100 model elements
  - 500 seeds
  - 2,000 two-way merge tests
  - 1,000 three-way merge tests

- Overall results of 3,000 tests
  - Initial error rate: 7.56% (227 errors)
  - Initial failure rate: 61.4% (1,842 failures)

- Isolation and fixing (10 bugs)
  - Current error rate: 1.86% (56 errors)
  - Current failure rate: 2.26% (68 failures)
  - Increased number of tests by 12 test cases

Increasing reliability of EMF Compare
Increasing reliability of EMF Compare

Fuzz Testing: Two- & Three-way Merge

● **Goal statement coverage: **80+ %
  ○ 100 model elements
  ○ 50,000 seeds
  ○ 100,000 two-way merge tests
  ○ 50,000 three-way merge tests

● **Goal for 150,000 tests**
  ○ Error rate: → 0 %
  ○ Failure rate: → 0 %
Versioning Models with Git
Created by

Linus Torvalds
As a replacement for BitKeeper
Git is Distributed
Primary SCM of the
Eclipse Foundation
(since 2011)
Popularized by

Famous Hosting Service

GitHub
EGit
What if EGit could prevent you from...

Creating dangling edges?
What if EGit could prevent you from...

Creating dangling edges?
What if EGit could prevent you from...

Creating dangling edges?

![Diagram showing a model and package with Class2 circled](image-url)
What if EGit could prevent you from...

Creating dangling edges?
What if EGit could prevent you from...

Creating dangling edges?
What if EGit could detect...

Semantic conflicts?
What if EGit could detect...

Semantic conflicts?
What if EGit could detect...

Semantic conflicts?
What if EGit could detect...

Semantic conflicts?
What if EGit could detect...

Semantic conflicts?
The Mediator is

Eclipse Team

EGit  Subversive  Eclipse CVS
The Mediator is

Eclipse Team
The Mediator is

Eclipse Team

![Diagram showing Eclipse Team, EGIt, and logical model ext. points]

?
The Mediator is

Eclipse Team
The Mediator is

Eclipse Team

Logical model ext. point

EGit

Logical model merger ext. point

Eclipse Team

EMF Compare
Deactivating Logical Model Support

Synchronize

- Always launch fetch before synchronization
- Automatically enable commit grouping in Git synchronizations
- Allow models (e.g., Java, EMF) to participate in synchronizations
The Mediator is

Eclipse Team
Deactivating

Logical Model Support

Comming Soon
A brief note about

Current support

Eclipse Team

Logical model ext. point
EGit
Logical model merger ext. point

Ongoing Work
A brief note about

Current support
A brief note about

Current support

http://download.eclipse.org/modeling/emf/compare/updates/egit-logical/nightly/
Logical Model Support on the
Command Line

http://wiki.eclipse.org
/EMF_Compare/Specifications/LogicalMergeCommandLine

Commands that trigger merge operations

- merge
- cherry-pick
- pull
- rebase
- revert
- stash apply
- submodule update

EGit
JGit
Eclipse Workbench (incl. Team)
Logical Model Support on the
Command Line

User

EGit

merge

get logical providers

return

return

JGit

get strategies

get logical providers

logical strategy

merge

return

return

Eclipse Team

logical model provider

merge

return

EMF Compare
Logical Model Support on the
Command Line

User

Git

merge

EGit / JGit
Eclipse Team
EMF Compare

return

return
Logical Model Support on the

Command Line

User → Git → Native Git bridge → Eclipse workbench & workspace provisioner

merge

? ? return return

return

EGit / JGit
Eclipse Team
EMF Compare
The provisionner will be

Oomph

- **Provisioning correct set of plugins in the Eclipse IDE.**
- Binding Git repos (incl. personal Gerrit push URL).
- **Checking out projects.**
- Setting workspace preferences.
- Configuring dynamic working sets.
- Keeping project preferences files in sync.
- The **configuration is model driven**, with the possibility to customize a lot for each project, each branch, each user...
The provisionner will be

Oomph
Git bridge as

Additional Git Commands

```

git logicalmerge <setup> <commit> [--help (-h)] [-m message]

  <setup>    : Path to the setup file. The setup file is an Oomph model
  <commit>   : Commit ID or branch name to merge
  --help (-h) : Displays this help message
  -m message : Sets the commit message to be used for the merge commit (in case
               one is created).
```

```

git logicalmergetool <setup> [--help (-h)]

  <setup>    : Path to the setup file. The setup file is an Oomph model
  --help (-h) : Displays this help message
```
EMF Compare — Git commands involving models
Version 3.1.0.201410261549

Table of Contents
- Add new Git commands in you system
- Create a setup file
  - Installation
  - Basic usage
- Git diff command with models : git logicaldiff
  - Others options available
- Git merge command with models : git logicalmerge
  - Others options available
- Git mergetool command with models : git logicalmergetool

When a user wants to compare or merge EMF models from the git command line interface, the operation is doing in a textual way. If he wants to compare or merge EMF models in a logical way, he needs to do that in an Eclipse environment similar to the one he used to create these models. As such, the environment requires some plugins to be installed but it may also requires some preferences to be set, some perspective to be activated etc.. Among these plugins, there are the mandatory ones that will be use to do the compare/merge operation: EMF Compare and EGit.

EMF Compare provides additional git commands in order to compare and merge models on the command line. These commands will use an Eclipse as a headless application (no Graphical User Interface) including EMF Compare and Egit to compare models in a logical way. To provisioned such Eclipse environment, the new git commands will call a program using Oomph.

Oomph Oomph is a technology that provisions a set of plugins in an Eclipse IDE, clones Git repositories, binds Git repositories to this IDE, checks projects, sets workspace preferences... The configuration is model driven, with files called Oomph setup model files.

Add new Git commands in you system
The basics git commands don't allow to compare or merge EMF models in a logical way. Additional git commands must be added to your system. Each git command is a shell script describing its behaviour. So, to add a new git command, a new script has to be developed. The new scripts corresponding to the git commands are:
- git logicalmerge: the "models compatible" version of the git merge command
- git logicaldiff: the "models compatible" version of the git diff command
- git logicalmergetool: the "models compatible" version of the git mergetool command
Easily get and test the latest

Model Collaboration Setup

Oomph-based setup & ready to use packages

- Nightly build of EMF Compare
- Nightly custom build of EGit
- Latest or nightly build of Papyrus
Easily get and test the latest

Model Collaboration Setup

**Ready-to-use Eclipse with collaborative modeling support**

Testing
We provide a ready-to-use and easy-to-update Eclipse version that contains the latest nightly builds of Papyrus, EMF Compare, and EGit (with logical model support). Please note that this version is intended for testing only. These ready to use Eclipse is available by two means:

- using **Oomph**. To download and install it, please follow the [installation guide](http://goo.gl/bfl20x). Once installed on your machine, you can keep it up to date with the latest nightly builds by selecting Help → Check for Updates in your installation.
- ready to use packages. **Download it, unzip it, launch it. Done.**

<table>
<thead>
<tr>
<th>Windows</th>
<th>Collaborative-Modeling-win32.zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows (x86_64)</td>
<td>Collaborative-Modeling-win32-x86_64.zip</td>
</tr>
<tr>
<td>Linux (GTK)</td>
<td>Collaborative-Modeling-linux-gtk.tar.gz</td>
</tr>
<tr>
<td>Linux (GTK / x86_64)</td>
<td>Collaborative-Modeling-linux-gtk-x86_64.tar.gz</td>
</tr>
<tr>
<td>MacOS X (Cocoa / X86_64)</td>
<td>Collaborative-Modeling-macosx-cocoa-x86_64.tar.gz</td>
</tr>
</tbody>
</table>

If you experience any issues with the installation or if you face bugs during testing this Eclipse version or if you have any other feedback, please let us know in the [discussion forums](http://goo.gl/tzPKT3).

Ready to use packages [http://goo.gl/tzPKT3](http://goo.gl/tzPKT3)
Oomph based setup [http://goo.gl/bfl20x](http://goo.gl/bfl20x)
Model Review
Status of developing of support for Model Review

- **Currently**
  - Requirement elicitation
  - Investigating feasibility
  - Growing developer & future user community

- **Beginning of 2015**
  - Initial prototype for UML model review
  - For collecting feedback of users
Minimum Requirements on Model Review

- Gerrit workflow for models
  Lift gerrit’s line level support to model level

**Review System**

- clone
- push for review
- feedback
- fetch patch
- rate & comment
- diff
- view changes of a review on model level
- add and view comments on model level
Model Review

- Support for
  - Papyrus UML models
  - Generic support for any GMF diagrams
  - Generic support for any EMF models

- Based on
  - EMF Compare & EGit
  - Gerrit & Reviews 4 Eclipse (Gerrit connector)
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
<td>Philip Langer</td>
</tr>
<tr>
<td>Branch</td>
<td>refs/heads/master</td>
</tr>
<tr>
<td>Change Id</td>
<td>7f8e6a17db0f3e324e1c08b05f32d1953aa9f4e8a2</td>
</tr>
<tr>
<td>Description</td>
<td>Changes name of class from Developer to Contributor</td>
</tr>
<tr>
<td>Reviewers</td>
<td>Philip Langer</td>
</tr>
</tbody>
</table>

**Patch Set 1**
- **Author**: Philip Langer
- **Committer**: Philip Langer

- Commit: 86b5c445713e91168c57db53d3f30b9a4b91c71d38f1
- End: refs/changes/02/2/1

- `/CORE/IT1_MSG`
- `/TestPythagorasProject/mathUtil`
Java - Changes name of class from Developer to Contributor [7be6a871e] - Eclipse

2 Changes name of class from Developer to Contributor [7be6a871e]
   Change 1

   Status: NEW
   Created: 29.08.2014 16:20
   Last commented: 2 mins 32 secs ago

   Attributes
   Owner: Philip Langer
   Branch: refs/heads/master
   Change-Id: 7be6a871e10be66f7c1d4b54e9b1e2a2

   Description
   Changes name of class from Developer to Contributor

   Change-Id: 7be6a871e10be66f7c1d4b54e9b1e2a2

   Review
   Reviewers:
   Philip Langer

   Add Reviewers...

   Patch Sets
   Patch Set 1
   Author: Philip Langer
   Commit: 866a5645213a214fd7d813d390f5af871c77158f1
   End: refs/changes/02/2/1

   COMMIT_MSG
   TestPapyrusProject/medalUrl

   Publish Comments... Compare With Base Restore Submit Abandon...
Right-click: Add comment

Re-use UML comments in Papyrus
Right-click: Add comment

Add Comment

build-74d714c.xmlant # Target test.dist

This one should be changed too!
Open issues for enabling

Model Review

● View changes of a model review
  ○ Integration of R4E and EMF Compare

● Review comments on model level
  ○ Show comments on model level
    ■ UML comments in Papyrus in own resource
    ■ How to provide generic EMF support?
  ○ Store comments on model elements in Gerrit
    ■ Gerrit supports line-based comments only
    ■ Own gerrit plugin seems to be the best solution?
Follow us, discuss with us, and contribute to
Collaborative Modeling Support
Ideas? Comments?

Contact us

Maximilian Koegel
mkoegel@eclipsesource.com

Philip Langer
planger@eclipsesource.com

Mikaël Barbero
mikael.barbero@obeo.fr

http://eclipsesource.com/

http://www.obeo.fr