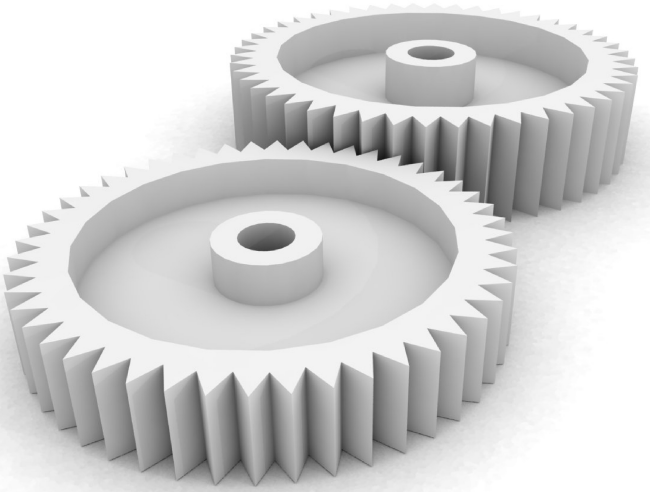
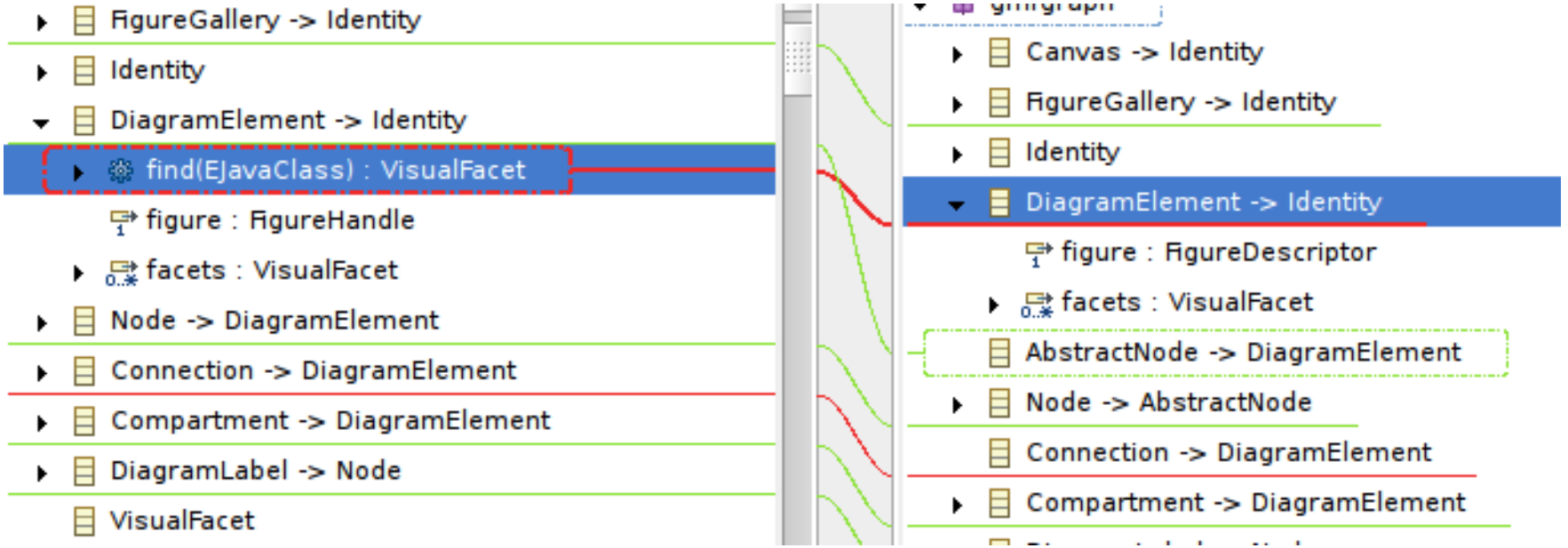
What **every developer** should know about EMF **Compare** ?



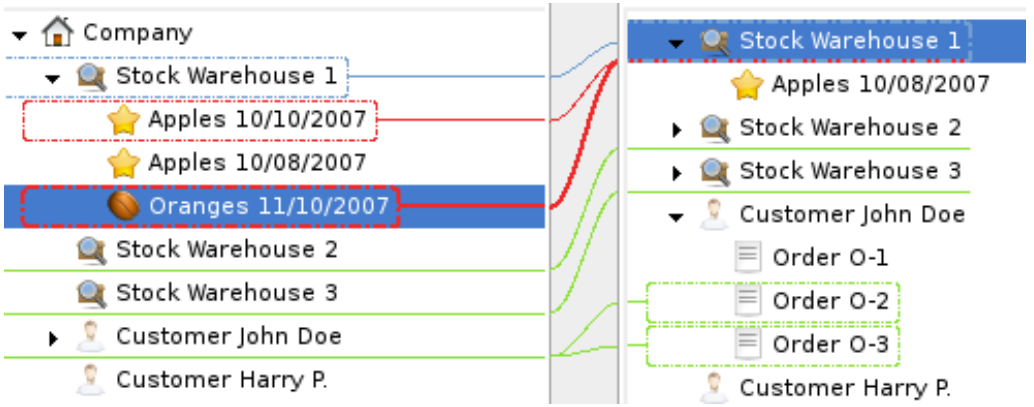
Cédric Brun

Obeo, France

in **action!**



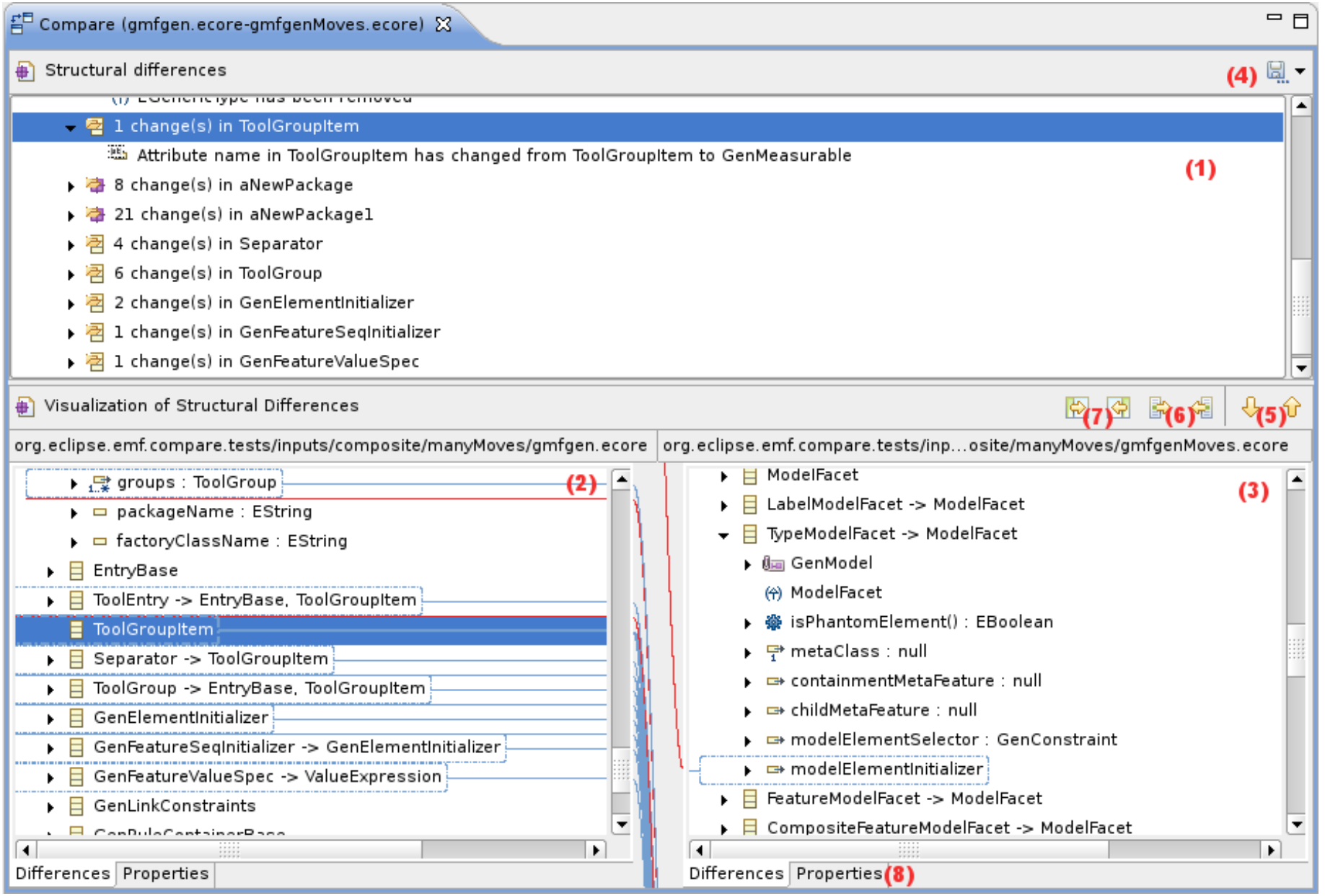
- SVN, CVS, GIT
- UML, EMF, DSL's
- History, Compare, Merge



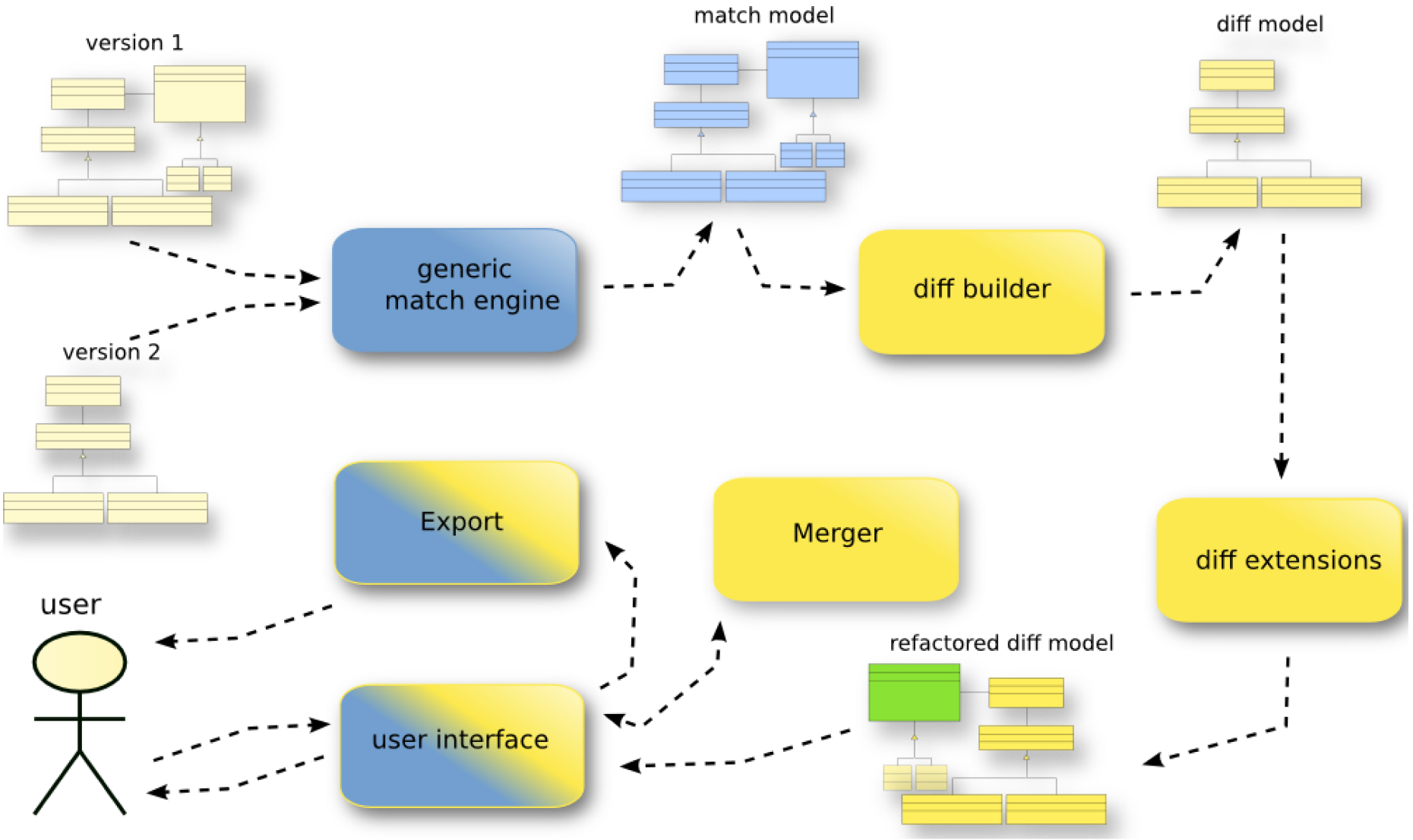
Component of **EMF**

- Graduated
- Stable API
- pure Java JAR
- Extensible

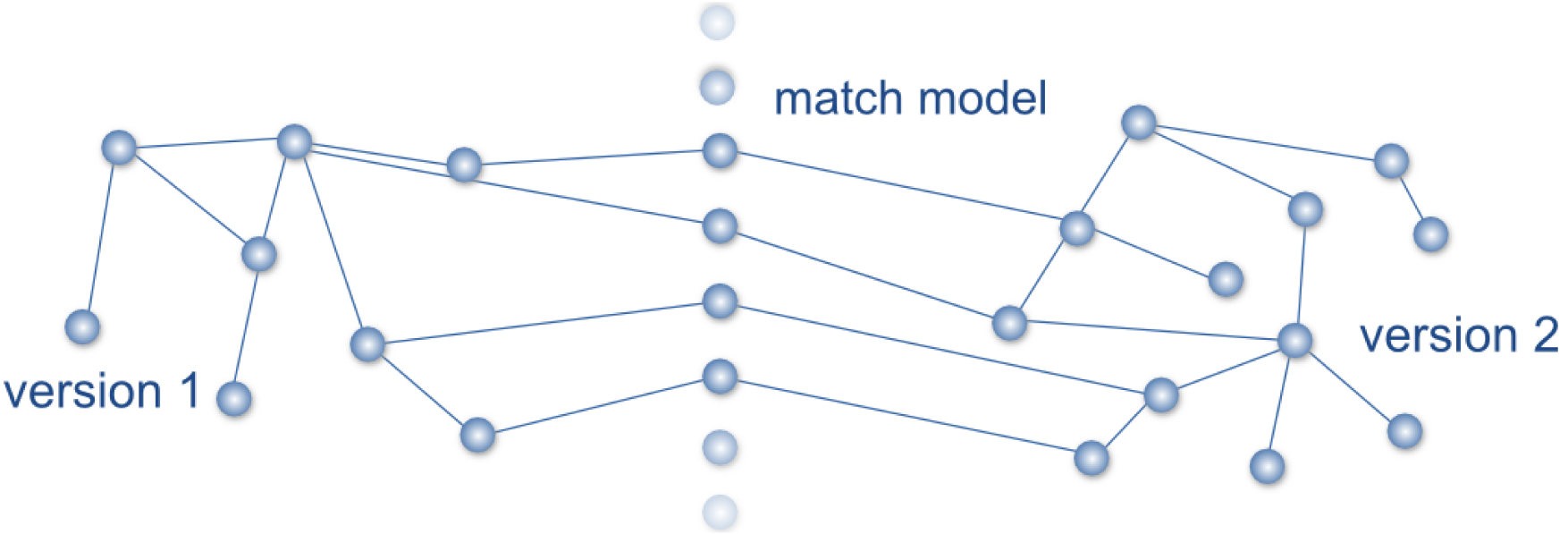
The Tool


Compare (gmfgen.ecore-gmfgenMoves.ecore)
Structural differences
(4)
1 change(s) in ToolGroupItem
Attribute name in ToolGroupItem has changed from ToolGroupItem to GenMeasurable
(1)
8 change(s) in aNewPackage
21 change(s) in aNewPackage1
4 change(s) in Separator
6 change(s) in ToolGroup
2 change(s) in GenElementInitializer
1 change(s) in GenFeatureSeqInitializer
1 change(s) in GenFeatureValueSpec
Visualization of Structural Differences
(7)
(6)
(5)
org.eclipse.emf.compare.tests/inputs/composite/manyMoves/gmfgen.ecore
org.eclipse.emf.compare.tests/inp...osite/manyMoves/gmfgenMoves.ecore
groups : ToolGroup
(2)
packageName : EString
factoryClassName : EString
EntryBase
ToolEntry -> EntryBase, ToolGroupItem
ToolGroupItem
Separator -> ToolGroupItem
ToolGroup -> EntryBase, ToolGroupItem
GenElementInitializer
GenFeatureSeqInitializer -> GenElementInitializer
GenFeatureValueSpec -> ValueExpression
GenLinkConstraints
ModelFacet
LabelModelFacet -> ModelFacet
TypeModelFacet -> ModelFacet
(3)
GenModel
ModelFacet
isPhantomElement() : EBoolean
metaClass : null
containmentMetaFeature : null
childMetaFeature : null
modelElementSelector : GenConstraint
modelElementInitializer
FeatureModelFacet -> ModelFacet
CompositeFeatureModelFacet -> ModelFacet
Differences
Properties
Differences
Properties
(8)

How is it **Working ?**



Who is **Who ?**



- Match Model /resource/org.eclipse.emf.compare.test
  - Match2 Elements 0.9981769904490151
    - Match2 Elements 1.0
    - Match2 Elements 1.0
    - Match2 Elements 1.0

Matching **Strategies**

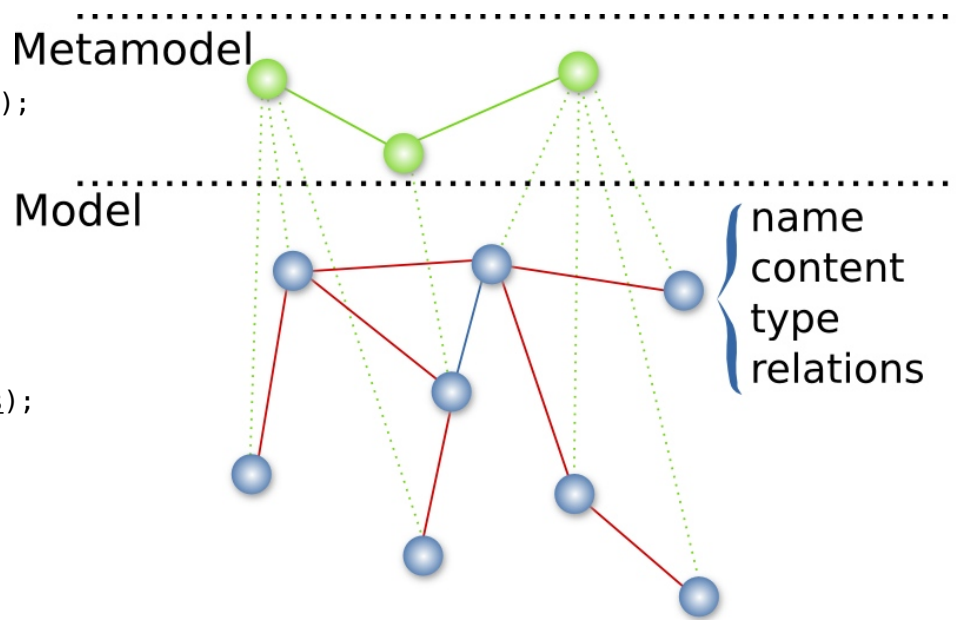
XMI-ID

```
options.put(MatchOptions.OPTION_IGNORE_XMI_ID, Boolean.FALSE);
MatchModel match = MatchService.doContentMatch(left,right options);
```

ID-Attribute

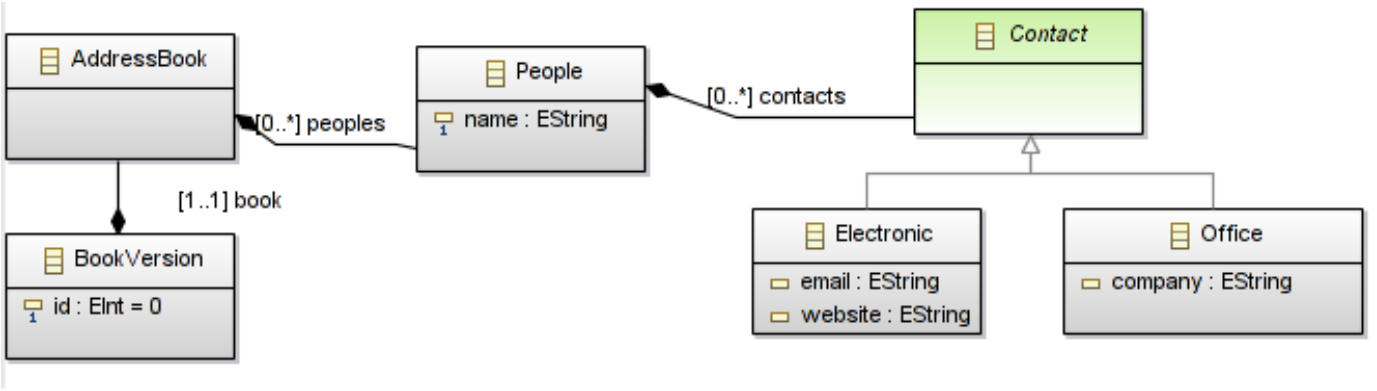
```
options.put(MatchOptions.OPTION_IGNORE_ID, Boolean.FALSE);
MatchModel match = MatchService.doContentMatch(left,right options);
```

GenericMatchEngine



defaults are XMI-ID, then ID-attributes and then reflective

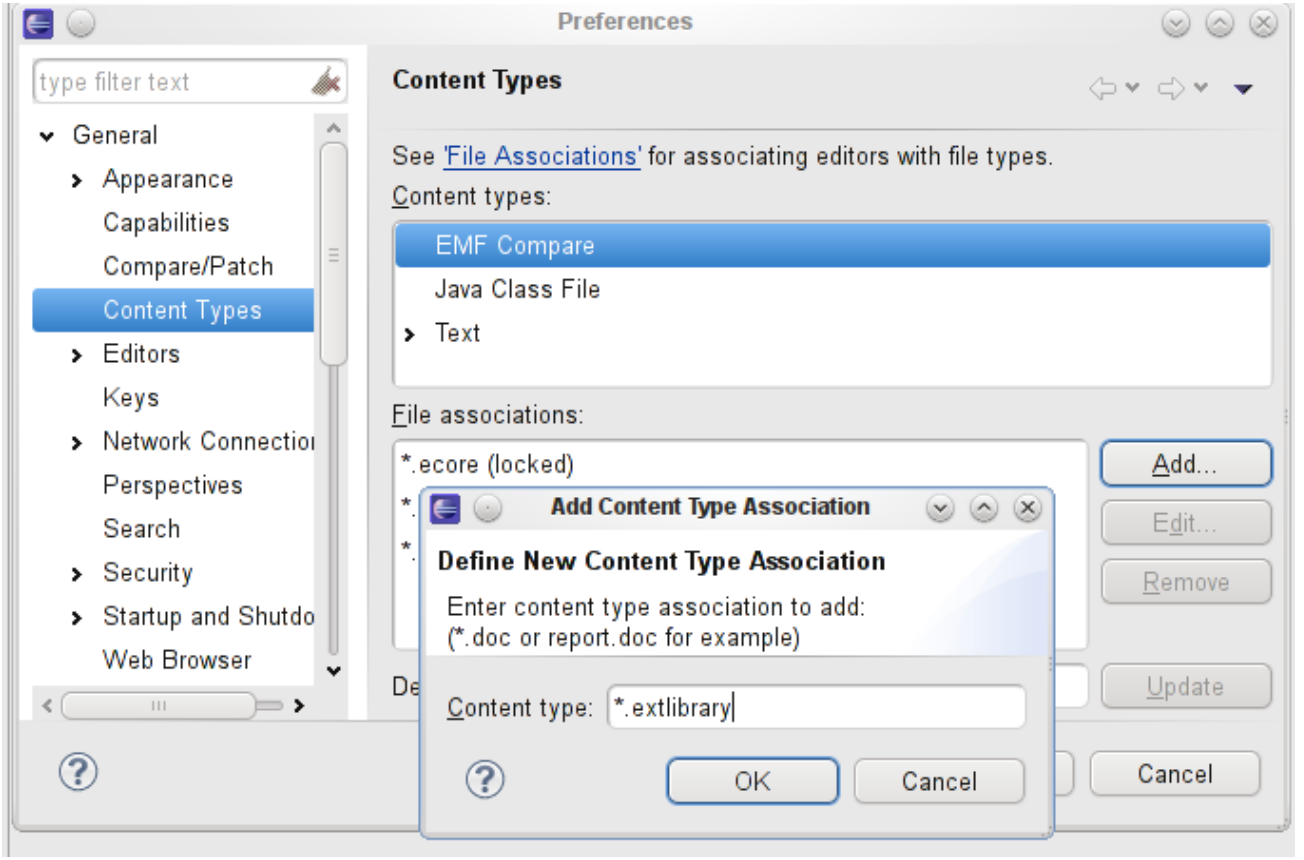
Your **own** matching strategy




```
public class AddressBookMatcher extends GenericMatchEngine {

/**
 * {@inheritDoc}
 */
@Override
protected boolean isSimilar(EObject obj1, EObject obj2) throws FactoryException {
/*
 * If we've got a People, only check the name similarity.
 */
if (obj1 instanceof People || obj2 instanceof People)
return nameDistance(obj1, obj2) > 0.8;
/*
 * Contacts are similar if : the associated people is similar + their content is quite the same.
 */
if (obj1 instanceof Contact && obj2 instanceof Contact) {
EObject obj1Parent = obj1.eContainer();
EObject obj2Parent = obj2.eContainer();
if (obj1Parent instanceof People && obj2Parent instanceof People)
return isSimilar(obj1Parent, obj2Parent) && contentDistance(obj1, obj2) > 0.5;
}

/*
 * If it's something we don't know about, then use the generic behavior.
 */
return super.isSimilar(obj1, obj2);
}
```

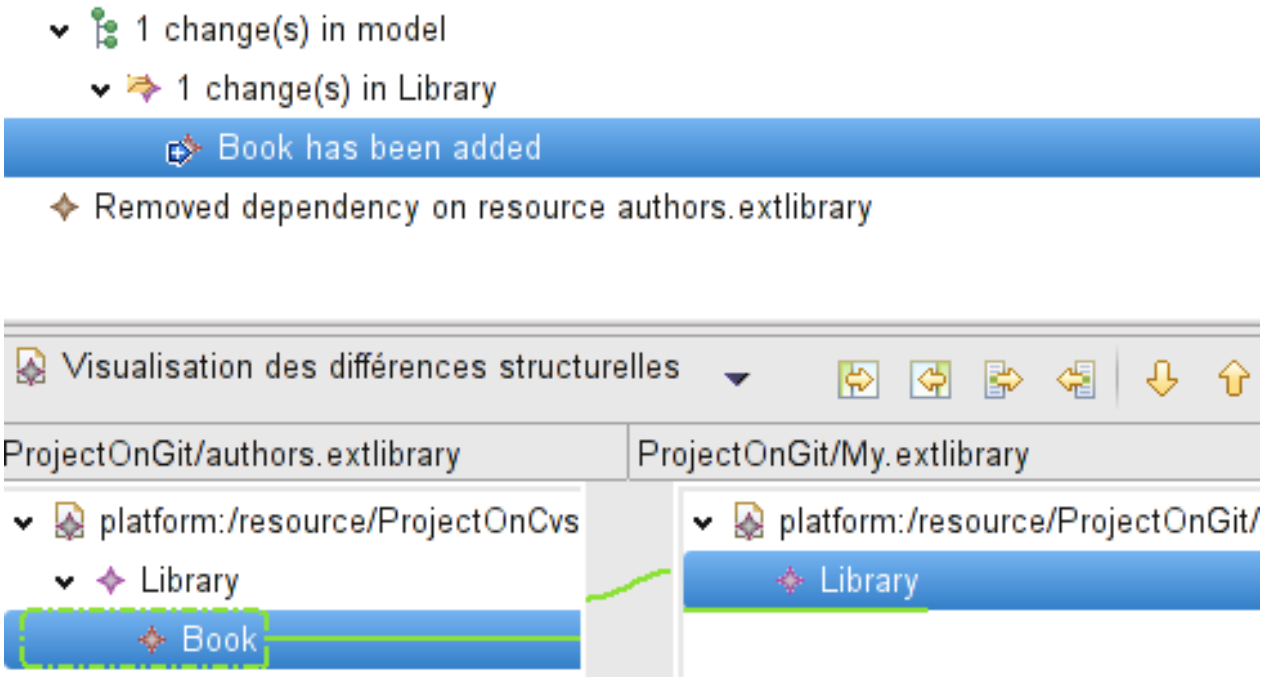
Comparing

```
People alice = createAlice();
v1Book.getPeoples().add(alice);

v2Book = EcoreUtil.copy(v1Book);

People bob = createBob();
v2Book.getPeoples().add(bob);
```



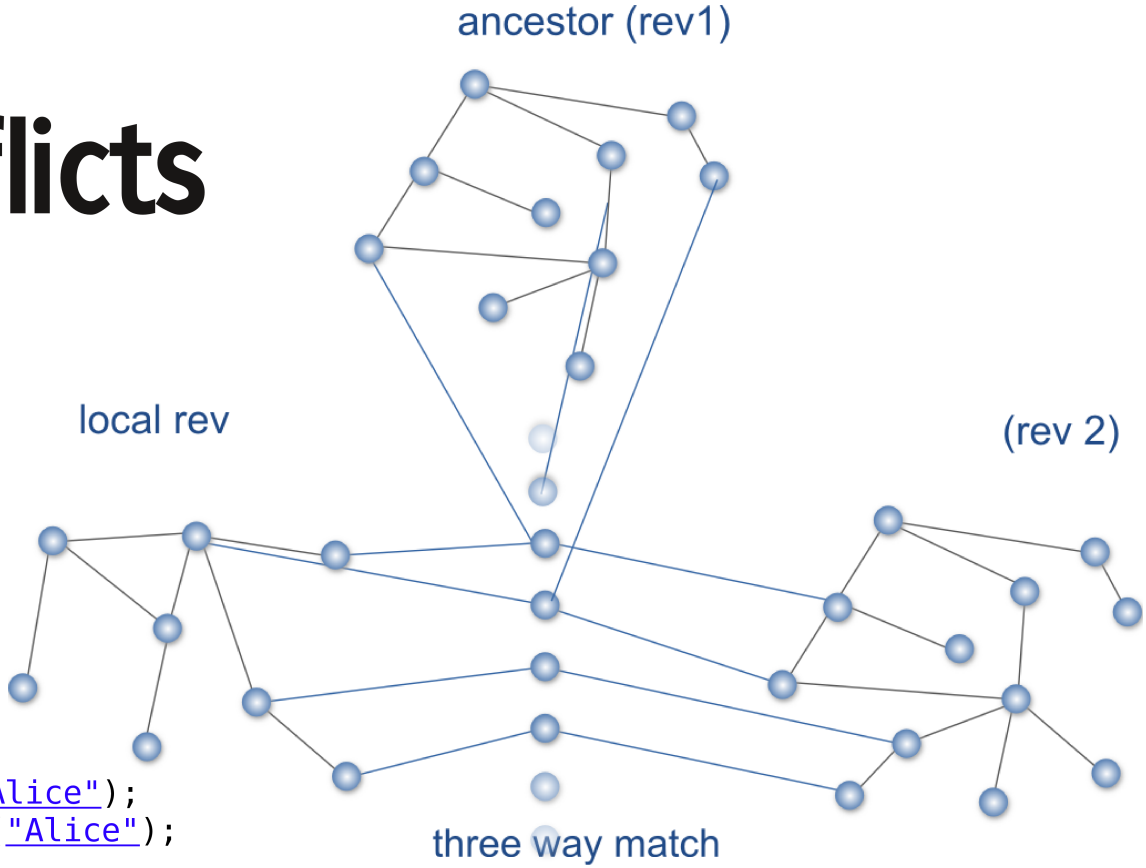

```
MatchModel match = new AddressBookMatcher().contentMatch(v1Book, v2Book, noOptions());

DiffModel delta = DiffService.doDiff(match);

DiffElement addbob = DiffHelper.isAdded(bob, delta);
assertNotNull("the addition of Bob has not been detected", addbob);
```

Merging




```
boolean mergeLeftToRight = true;
for (DiffElement diff : delta.getDifferences()) {
MergeService.merge(diff, mergeLeftToRight);
}

match = new AddressBookMatcher().contentMatch(v1Book, v2Book, noOptions());

delta = DiffService.doDiff(match);

assertEquals("We still have a difference whereas we merged all the diff elements", 0,
delta //$NON-NLS-1$
.getDifferences().size());
```

Detecting **conflicts**

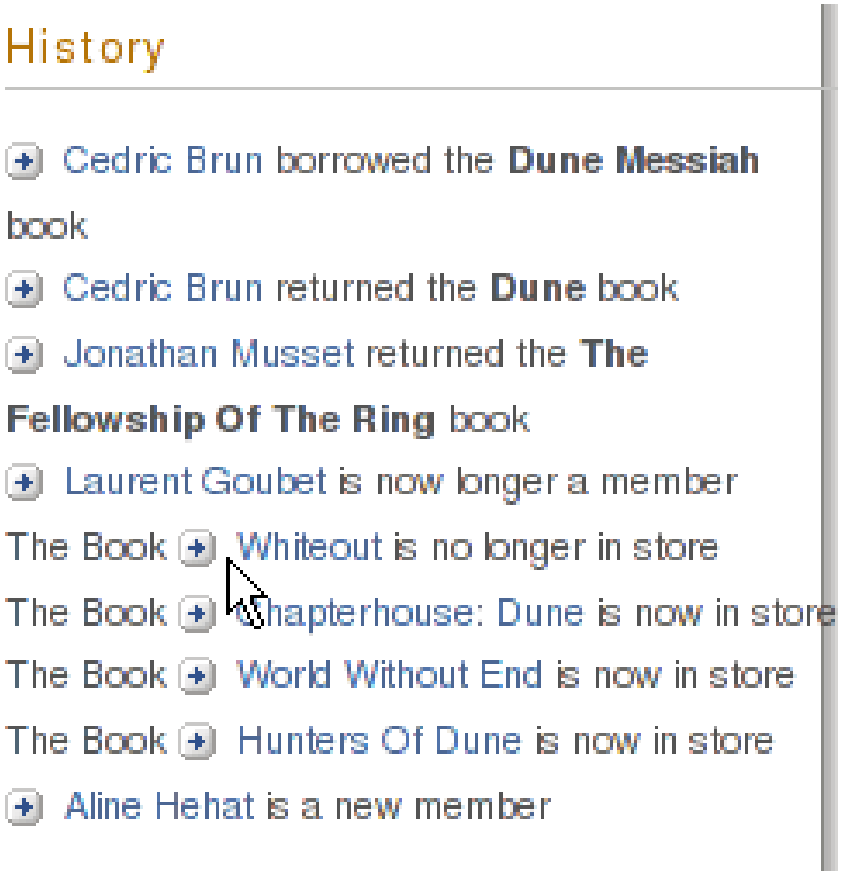


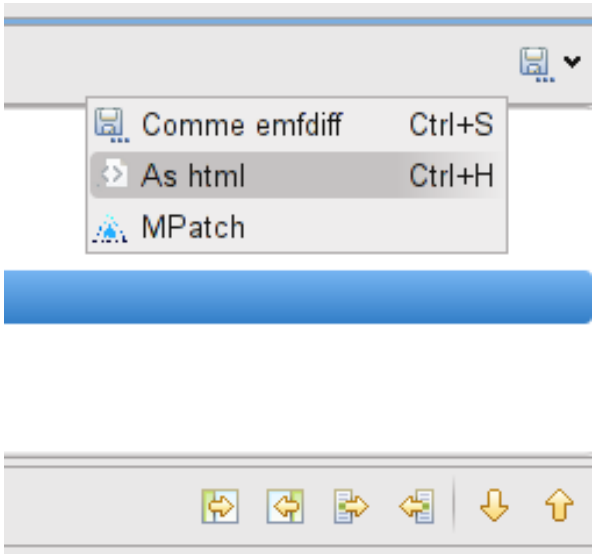

```
final People alice = createAlice();
v1Book.getPeoples().add(alice);

v2LocalBook = EcoreUtil.copy(v1Book);
v2RemoteBook = EcoreUtil.copy(v1Book);

People aliceLocal = v2LocalBook.getPeople("Alice");
People aliceRemote = v2RemoteBook.getPeople("Alice");

aliceLocal.setBirthday("07/05/84");
aliceRemote.setBirthday("07/07/84");

MatchModel match = new AddressBookMatcher().contentMatch(v2LocalBook, v2RemoteBook, v1Book,
noOptions());

DiffModel delta = DiffService.doDiff(match, true);
List<DiffElement> conflicts = new ArrayList<DiffElement>();

for (DiffElement diff : delta.getDifferences()) {
if (diff.isConflicting()) {
conflicts.add(diff);
}
}

assertEquals("We should have a conflict on Alice's birthday", 1, conflicts.size());
}
```

Exporting the delta.

Diff is a model => You can use Acceleo to generate code based on a diff

e.g : SQL migration scripts based on a entity models comparison





Other **usages** we've heard of ?

- Non regression Testing
  - Selective assertEquals() on models
- Transforming a destructive process in an incremental one
- Using the diff + original version as an input for a process
- Interactive Model Transformation
  - Leveraging the UI + custom Match and Merge

What's **cooking** ?

- UML dedicated support
- Enhanced generic editor
- GMF and Papyrus Integration

References

- Website :
  - http://www.eclipse.org/modeling/emf?project=compare
  - http://wiki.eclipse.org/index.php/EMF_Compare
- Support : EMF newsgroup and mailling list
- Downloads :
  - Indigo Update Site
  - Eclipse Modeling Package
- **Source Code**
  - git://git.eclipse.org/gitroot/emfcompare/org.eclipse.emf.compare.git