EMF Comparison Framework

One year later ...

Cédric Brun

Obeo
France
Working with models in Eclipse?
EMF / GMF : modeling support

- Modeling, Meta-modeling
- Editors and modelers
M2T / Acceleo : generate text from models

- Map Eclipse Summit Europe 2007
  - Topic Working with models in Eclipse
  - Topic EMF/GMF : modeling support
    - Thread Modeling, Meta-modeling
    - Thread Editors and modelers
  - Topic M2T/Acceleo : generate text from model
  - Topic M2M : transform models in models
  - Topic Have a try ?
  - Topic No team-work support
  - Topic Back in 2006
  - Topic Eclipse Summit Europe 2006
- Topic Result ?
- Topic History
- Topic Why should you bother ?
M2M : transform models to models

- UML Tools, Query/Transaction/Validation, ...

```xml
<Model> mindmap
  <Class> DocumentRoot
  <Class> Map
    <Property> title
    <Property> author
  <Class> Topic
    <Property> title
  <Class> Thread
  <Class> ThreadItem
  <Class> Relationship
  <Association> map
    <Association> ownedTopics
    <Association> ownedRelationships
    <Association> ownedThreads
    <Association> parentTopic
    <Association> subTopics
```
Had a try?
No team-work support
Back in 2006...
Modeling symposium lighten the fact that this component is essential and missing.

Obeo and Intalio contribute their initial implementations
Result?

- differencing
- merging
- team API integration
- diff export
- extensibility
History

- October 2006 : Eclipse Summit Europe
- November 2006 : EMF Compare Proposal
- March 2007 : IP approval
- June 2007 : First builds on Eclipse.org
- September 2007 : Stable builds for 0.7.0
Why should you bother?
As an Eclipse user

- develop using models!
As an RCP developer

- compare your own business objects
- do not depend on Eclipse IDE
- « in memory » comparison
- produce history reports
As a tool developer

- incremental processing
- « mymodel » support
- XML based file comparison

```xml
<extension point="org.eclipse.core.contenttype.contentTypes">
  <content-type
    id="org.eclipse.emf.compare.ui.contenttype.ModelContentType"
    name="Model File"
    priority="normal"
    file-extensions="mymodel">
  </content-type>
</extension>
```
As a researcher

- testbed for models/graphs comparison algorithms
How does it works?
Principles: models everywhere
Phase 1 : matching elements
Match engine

- are « a » & « b » similars ?
- produce a match model

version 1  match model  version 2
Generic matching engine

- heuristic based
- metamodel agnostic
- 4 metrics
  - name
  - content
  - type
  - relations
Phase 2 : differencing
Differencing engine

- produces the diff model from the match model

- 70 changes in model
  - 70 changes in gmfgen
    - 3 changes in LinkLabelAlignment
      - 1 changes in MIDDLE
      - 1 changes in TARGET
      - 1 changes in SOURCE
    - 3 changes in TypeModelFacet
      - 3 changes in modelElementInitializer
        - GenElementInitializer has been removed from reference eType in modelElementInitializer
        - typeModelFacet has been removed from reference eOpposite in modelElementInitializer
        - EGenericType has been removed
    - 4 changes in ViewmapLayoutType
  - 2 changes in Palette
    - 2 changes in groups
Phase 3: extending the diff model
Diff model export

version 1

version 2

user

generic match engine

match model

diff model

diff builder

Export

Merger

user interface

refactored diff model

diff extensions
Diff model export

- export a model containing « diff » and « match » model: emfdiff file
Extensibility
Performances : figures

- **UML model**
  - UML2.0, 130+ elements : 0.3 second
  - UML2.0, 4600+ elements : 1 second
  - UML1.3, 65 000+ elements : 4 minutes

- **Ecore model**
  - gmfgraph.ecore, 450+ elements : 1 second
  - gmfgen.ecore, 1200+ elements : 3 seconds
  - 6000+ elements, very flat model : 20 seconds
  - 45 000+ elements : 4 minutes
Performances parameters

- Quantity of information
- Unstructured model
- Number of differences

Performances parameters:
- Quicker: higher performance
- Slower: lower performance

Value:
- Low: low performance
- High: high performance
Current status

- 0.7.0 stable release
  - merge/differencing
  - reference documentation
  - « 2 way » comparison
  - Ecore/XMI Id handling
  - metamodel agnostic
  - match engine extensibility
  - export API

- 0.8.0 interim builds
  - « 3 way » comparison
  - diff engine extensibility
  - diff model extension
  - performances enhancements

- Community
  - « up-to-date » doc is in the wiki
  - great feedbacks
  - bugs and enhancements requests
  - patches
  - coming soon: release review

- Eclipse.org
  - automated builds
What's next?
Roadmap

- 0.8.0 release (end of the year)
  - 3 way comparison
  - diff model extension
  - diff engine extensibility
  - community tests
  - tutorial documentation

- Ganymede simultaneous release
  - stability/efficiency
  - other tools integrations
  - API validation
Alternatives engines and settings

- specific properties (big models...)
- performances
- weight-based engine
Proxy resolving on version control systems
Integration in modelers
Thanks for your attention, questions?

- Further information:
  - [http://www.eclipse.org/modeling/emft/?project=compare#compare](http://www.eclipse.org/modeling/emft/?project=compare#compare)