Model Versioning
and Enterprise Architect
My Background

- OOM & Model Engineering, TU Vienna
- 2011 PhD: *Model Versioning*, TU Vienna
- Sparx Trainer & Consultant
- LieberLieber Product Manager
Who Needs Modeling?

**Information Management**
- Allianz
- AXA
- Swisscom
- Commerzbank

Companies collecting, controlling and managing information
- Insurance companies
- Telecommunication companies
- Banks
- etc.

**Cyber Physical Systems**
- Daimler
- VW
- BMW
- Siemens
- Airbus
- etc.

Manufacturers of embedded devices
- Automotive
- Healthcare
- Defense
- Railway
- Customer Electronics
- etc.

**Application Development**
- Software AG
- SAP
- etc.

Companies developing application software
- Web Applications
- Mobile Applications
- Desktop Applications
- etc.
Modeling Infrastructure is Challenge for Everyone

Consultancy for Methodologies, etc.

Tools Supporting Modeling Methodologies

Modeling Infrastructure

Hosting of Modeling Infrastructure

Information Management

Cyber Physical Systems

Application Development

- Configuration Management (Versioning)
- Connected Models, Distributed Modeling
- Variant Management
- Team Collaboration
- Tools Integration
- Web Access

LieberLieber Software
Agenda

• Motivation
• Built-In Feature
• Third Party Market
• Conclusion
Enterprise Architect Challenge

Configuration Management

Team Collaboration
Collaboration in General

Same Time
synchronous

Face to Face Interactions
decision rooms, single display
groupware, shared table, wall displays,
roomware,…

Continuous Task
team rooms, large public display, shift
work groupware, project
management,…

Different Time
asynchronous

Remote Interaction
video conferencing, instance messaging,
chats, virtual worlds, shared screens,
multi-user editors,…

Communication & Coordination
e-mail, bulletin boards, blogs,
asynchronous conferencing, group
calendars, workflow, wikis,
version control,…

Time/Space
Groupware Matrix

Same Place
colocated

Different Place
remote

Configuration Management, Change Management and Collaborative Modeling

- Working collaboratively on a model is hard
- Versioning for EA Models is hard and error-prone
- Tracking Changes in Models is very complex

RESULT → Modeling with EA is often used without Configuration Management
EA Built-in Features
EA Built-in Features

• Manage multiple file copies
• Package Baselines
• XMI Export/Import with Merge Sets
• Reusable Asset Service
• VCS Integration
• (Time-Aware Modeling)
Typical Versioning Concepts
Manage Multiple File Copies

 ✓ Easy to use
 ✓ Works for one person well
 ✓ Good Performance

- Change Management not possible
- Configuration Management not possible
- Teamwork not possible
- Comparison of EAP-Files not possible
- Transfer content only via XMI
Typical Versioning Concepts

Package Baseline

- Easy to use
- Works for teams as package backup well
- Not bad for review of one changed diagram

- Change Management only with high effort
- Configuration Management only with high effort
- Comparison between the baselines not possible
- Slow for big packages
- No overview of available baselines
- For review of complex changes not usable
- Selective restore (merge) not usable
- Many elements properties not visible
- Blows up the EAP-files
- 3-way Diff and Merge not supported
- Cross package dependencies causes problems
Typical Versioning Concepts
XMI Export/Import

• Merge Sets
• Reusable Asset Service
• XMI in VCS (EA Integration)
XMI Merge Using Baseline Merge Sets

• Used when users are editing separate parts (as in separate elements, diagrams or sub-packages) within the same containing package

✓ Allows integrating work from multiple people
✓ DBMS and files-based
✓ Used when users are editing separate parts (as in separate elements, diagrams or sub-packages) within the same containing package
✓ Better than simple import
✓ Kind of 3-way merge
  – Only union (Vereinigungsmenge)
  – Complex handling of multiple files
  – No Conflict Resolution
  – Not intended to change the same element
  – Configuration Management only with high effort
  – Similar problem than baseling because of similar technique used
Reusable Asset Service (RAS)

- XMI not in VCS, but in a shared repository, accessible via Cloud Service

For each Package, the RAS automatically identifies the:
- Version of the Package held in the registry
- Diagrams and elements (including child Package elements) contained by the Asset Package
- Dependencies on parent Packages of any external elements that the Asset Package references
- Dependencies on MDG Technologies
Typical Versioning Concepts

**EA VCS Integration**

- Out of the box available
- Team work possible
- Model under Configuration Control
- No conflicts (good & bad)

- Working independently of each other not possible
- Requires high discipline and effort
- Model structure according to versioning paradigm
- Change Management only with high effort
- Bad performance for big models
- For review of complex changes not usable
- Selective restore (merge) not usable
- Many elements properties not visible
- Cross package dependencies causes problems
- Many CVS solutions not supported (e.g. Git)
Summary Built-In features

1. Multiple File Copies
2. Baselines
3. XMI Export/Import incl. Merge Files
4. SVN Integration ("Pessimistic Versioning")
5. (Time-aware Modeling)

Independent of DBMS/file-based repository!
VCS – Version Control Systems

• Pessimistic (EA):
  • Model is locked → Sequential work

• Optimistic:
  • Parallel work, but conflicts possible
Why not using traditional diff tools?

• Switch of notation – mental map?
• Hard to understand changes
• Impossible to understand the intention behind changes
Why is Three-Way Diff and Merge so Important

Two-Way Diff compares two Models to each other

✓ Same attribute has different names EnergyHighLevel and EnergyLowLevel
✓ Bob or Verena has changed?
  – Has Bob renamed?
  – Or has Verena renamed?

✓ TempLevel Attribute is missing in Verena’s version
  – Bob or Verena has changed?
    – Has Bob added or renamed?
    – Or has Verena deleted or renamed?
Why is Three-Way Diff and Merge so Important

Three-Way Diff compares each Model to their both predecessor

Bob

Verena

 ✓ TempLevel Attribute is missing in Verena’s version
 ✓ Verena has deleted!

 ✓ Verena and Bob they both have differently renamed the attribute EnergyLevel, Bob to EnergyHighLevel and Verena to EnergyLowLevel!
Why 3-way Diff is useful?

2-way diff: Common base version is not considered

→ Changes are hard to understand
→ Merge cannot cover all intentions
3-Wege-Diff and Merge

V0 → V1.1 → V1.2
V1' → V2.1 → V2.2
V2.1' → V3.1 → V3.2

Copy

Diff Model

Diff Model

Diff Model

Copy

Research Project 2011:
K. Wieland: "Conflict-tolerant Model Versioning";
Why is Three-Way Diff and Merge so Important

- Allows me to see during merging if it’s my changes or theirs
- Enables branching of models
- Same approach as for source code (since 60ies)
LemonTree ©
3rd Party EA Tool

- Standalone Application
- Compares & Merges EA Models
- EAP-files and MBSE (e.g. MS SQL) are supported
- 3-way diff and merge is supported
- Integrates into VCS (e.g. SVN, Git, PTC, etc.)
- Versioning of complete EAP-Files (similar to text files)
Especially for Safety Projects

“In general, standards such as IEC 61508 demand the application of configuration management.

This refers to all artifacts, including UML models.

The solution from LieberLieber is our key to revealing the changes that have been made to a revision.”

STEFAN MÜLLER
HIMA PAUL HILDEBRANDT GMBH
SAFETY-RELATED AUTOMATION SOLUTIONS
New challenges can be tackled now
Collaborative Modeling under Version Control

Frank

Model
Version V1

Version V1
Change V1'

Check Out
08:32
Check Out
09:05
Check In
11:34
Check In
12:56

Merge Changes

Verena

Version V1

+ Change V1''

Version V2 = V1+V1''

Version V3 = V1+V1''+V1'

Time
Review of Model Changes

Model Version V0.4

Model Version V0.4''

Model Version V0.4'

Are the reviews considered?

Integrate review comments

Version V0.5

Version V0.6 = 0.4 + 0.5 + 0.4''

Version V1.0

10.09.15 Released for review

13.09.15 Review with adaptations

13.09.15 Adaptations

15.09.15 2nd Review with adaptations

16.09.15 Adaptations accepted

17.07.13 new version released
Model Branches

**Feature Team**

- V B1.1`
- V B1.3`
- V B1.4`

**Branch Version**

- Branch V B1.2
- Branch V B1.3
- Branch V B1.4
- Branch V B1.5
- Branch V B1.6

**Main Version**

- V A1.1`
- V A1.3`
- V A1.4`

**Basis Team**

- V A1.2`

**VCS**

- Merge only some features

**Time**

- 01.03.13 Branch created
- 29.07.13 merged
Conclusion

- Team Collaboration vs. Configuration Management
- Change tracking is essential
- Many techniques, but how do I want to work?
- Features of VCS are necessary for today’s challenges
Thank you!

konrad.wieland@lieberlieber.com