



**Where there is modeling,  
there is merging**

ECLIPSECON FRANCE, 24/06/2015



**intel**  
Software

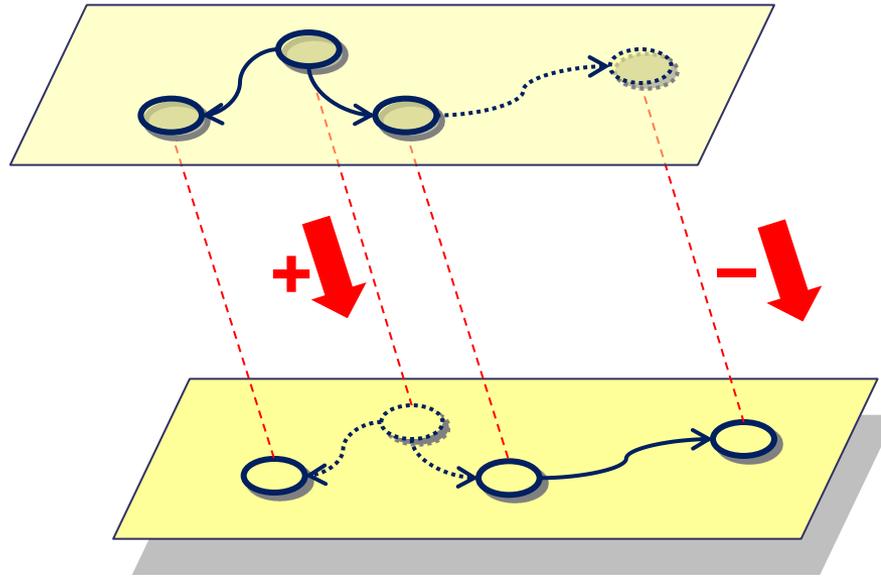


**Atos**



**THALES**

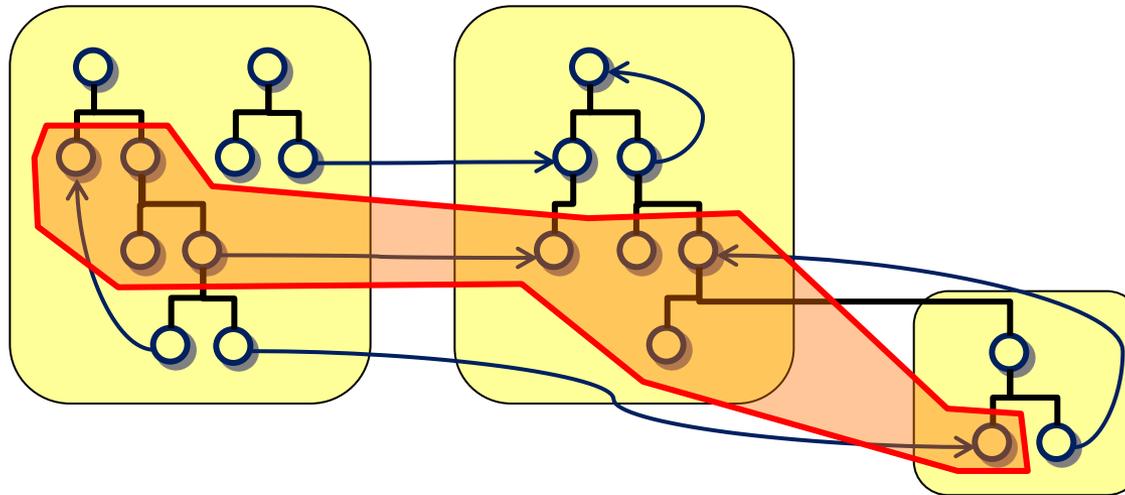
# Model merging?



- Transfer data
- Align (totally/partially) model structures
- Report changes
- ...

## Vision

- Merging = **primitive, consistency-preserving operation** for model manipulation, transformation, evolution
- Operates on arbitrary **model scopes** whose behavior can be customized



[http://wiki.eclipse.org/EMF\\_DiffMerge](http://wiki.eclipse.org/EMF_DiffMerge)

## Case 1: Version Control

INTEL

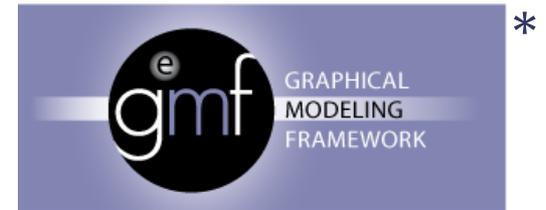
Version control of models  
with multiple SCMs  
based on EMF Diff/Merge\*  
in Intel® CoFluent™ Studio



# Once upon a time... a product

A modeling and simulation framework  
for predicting system behavior and performance  
before starting hardware and software design

■ <http://cofluent.intel.com>

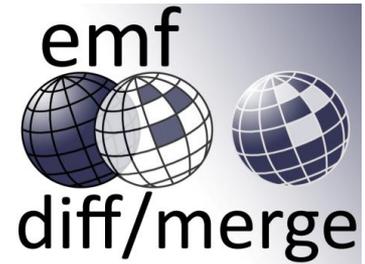


## How to compare models?

- Locally
- Shared using various SCMs

## Needs to

- Be sexy!
- No framework labels like Edge, Node
- Ensure graphic/semantic cohesion



- Model consistency preservation
- Provides extendible API
- Easy integration
- Very responsive team on forum/bugs

# What we've done

## GUI integration with Eclipse Compare

- Compare with each other
- Compare with local history

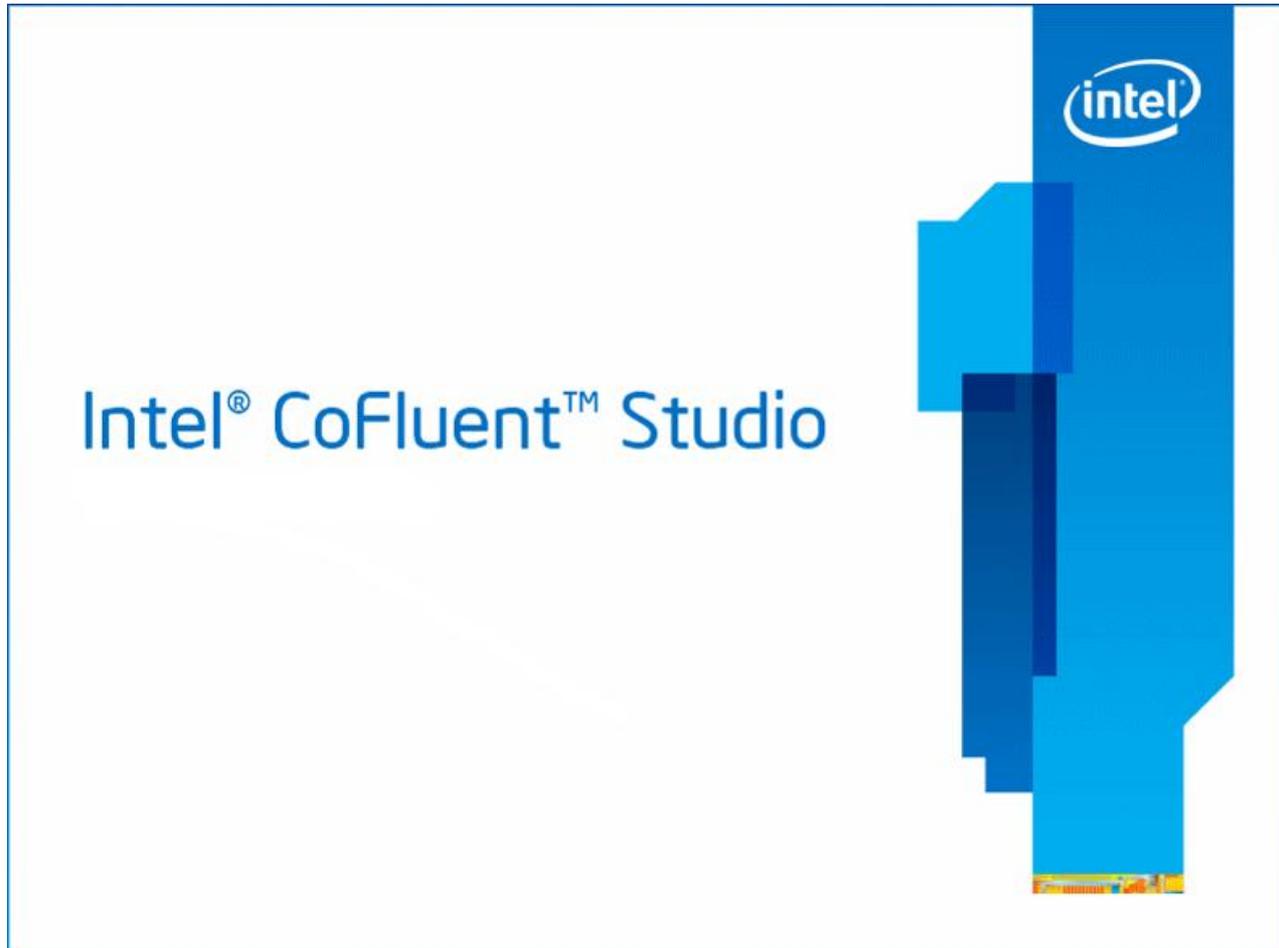
## GUI integration

- With Git\*
- With Perforce\*
- With Subversion\*

## Logical models integration

## Customized UX

- LabelProvider/ContentProvider
- Menus
- Resource load/unload



# Wait... happy ending!

## Let's contribute

- Logical model support
- Local history support
- Git\* support
- Perforce\* support
- Subversion\* support

## Represent 60 classes & ~4000 LOC

<https://www.eclipse.org/forums/index.php/t/1066288/>

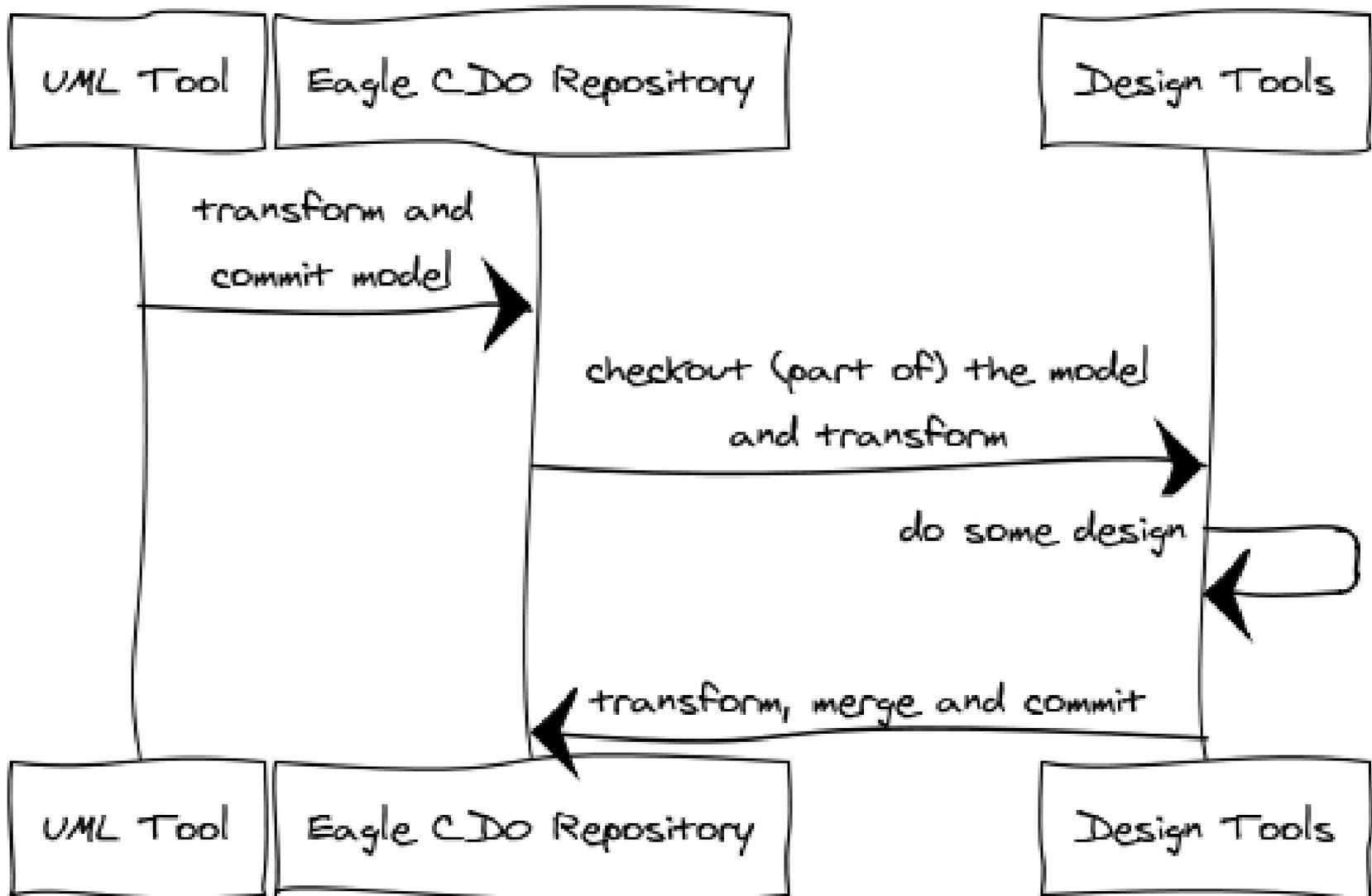
## Case 2: Incremental model transformations

ATOS

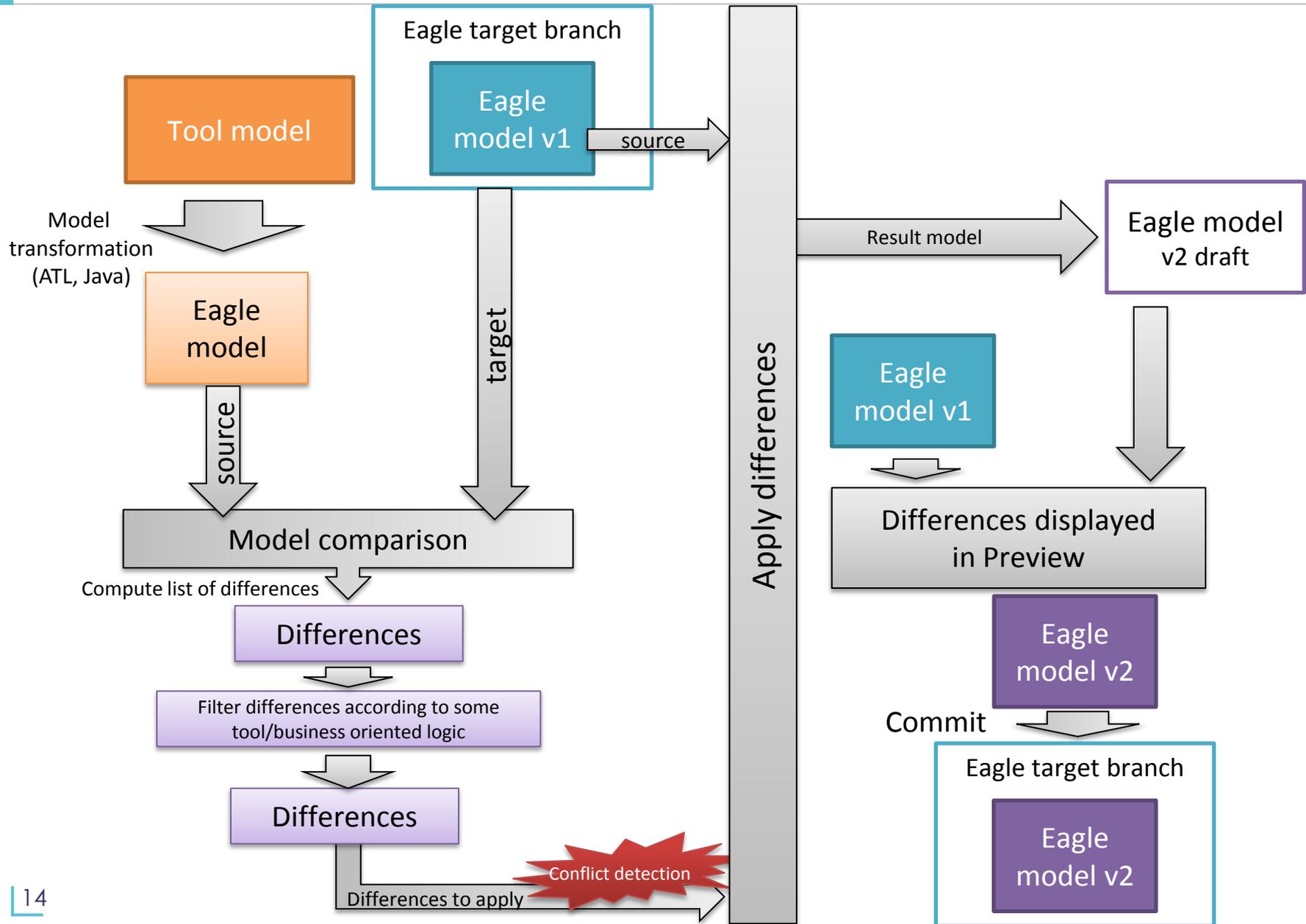
CDO-based model infrastructure with incremental import/export



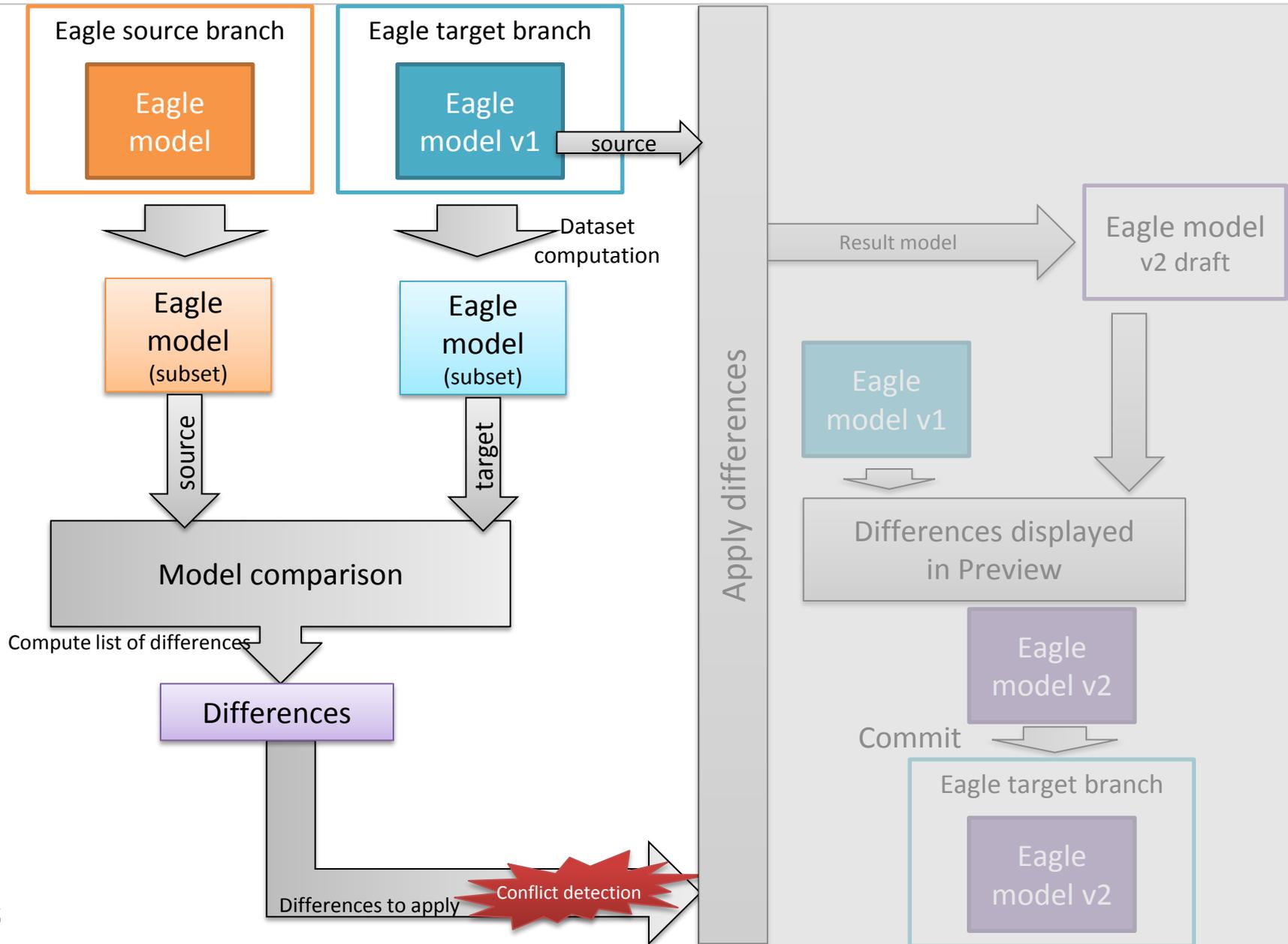
# Solution overview



# Tool transformation and import



# Merge between branches



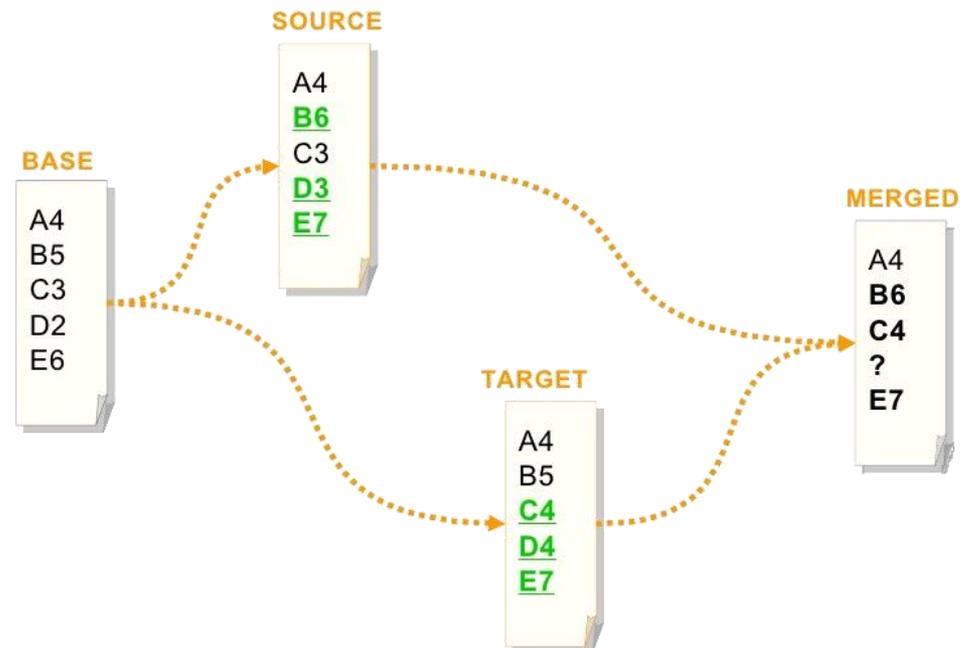
# CDO & Diff/Merge integration

## Diff/Merge customization

- Match policy (ID-based)
- Merge policy

## 3-way merge using the ancestor

pulled from the CDO history



## Let's contribute!

- Tracking of model scope for merge actions
- Higher-level 3-way conflict detection

## Case 3: Modeling Patterns

THALES

Modeling Pattern tool  
for Sirius-based environments

The Thales logo is displayed in a white rectangular box. It features the word "THALES" in a bold, blue, sans-serif font. A small blue dot is positioned above the letter "A".

THALES

■ **Pattern [engineering]: a solution to a recurring problem**



■ **"Modeling pattern": a modeling principle that reflects a pattern**

- Data
- Constraints
- Graphical representations

# A dedicated tool?

## Usage of the tool

- Extract a pattern from a model
- Store it in a catalog
- Reuse it in other models [with roles]
- Evolve patterns and models, check conformance/synchronize when needed

## It is rather fun because

- All is done through dialogs, no programming
- Diagramming concerns are handled via Sirius



**[http://wiki.eclipse.org/EMF\\_DiffMerge/Patterns](http://wiki.eclipse.org/EMF_DiffMerge/Patterns)**

thanks to **MERGE**  
SAFETY & SECURITY



# UML Designer

graphical tooling to edit and visualize UML models

The screenshot shows the 'New Pattern' dialog box in Capella. The 'Create new pattern' section includes a warning: 'Included elements have dependencies to other elements in the model (-> marks in Content tab)'. The 'Properties' tab is active, showing fields for Catalog (Lib1), Name (Measurement and Data Acquisition), Version (1.0), Environments (Capella 0.8.0), and Authors (OCO). A description is provided: 'This pattern defines a straightforward integration of an optical sensor into a system whose behavior is based on an analysis of the optical data.' The 'Image' section contains a diagram showing data flow between 'Launch data acquisition' and 'Launch image acquisition' components. The 'Include layout and style' checkbox is checked, and the 'Use template...' button is visible. The background shows a project overview with components like 'Measurement Engineer', 'Forecaster', and 'Webmaster'.

The screenshot shows the 'New Pattern' dialog box in UML Designer. The 'Create new pattern' section includes the instruction: 'Specify the properties and contents of the new pattern, and how this pattern can be applied in models.' The 'Properties' tab is active, showing a tree of 'Pattern elements': '<Association> subjectsToObservers', '<Class> ConcreteObserver (role: ConcreteObserver Role)', '<Class> Observer (role: Observer Role)', and '<Class> Subject (role: Subject Role)'. The 'Roles' section on the right lists 'Observer Role', 'ConcreteObserver Role', and 'Subject Role' with 'Add', 'Delete', 'Rename', 'Up', and 'Down' buttons. The 'Include all dependencies...' checkbox is checked. The background shows a UML class diagram with 'Observer' and 'Subject' classes and their association.



**« EMF Convergence » ?**



**The end \o/**

**TIME FOR QUESTIONS  
(UNLESS WE REALLY TALKED TOO MUCH)**