

Decoupling Operation-Based Merging from Model Change Recording

Stephen Barrett, Patrice Chalin, Greg Butler

Department of Computer Science and Software Engineering
Concordia University, Montréal, Québec

International Workshop on
Models and Evolution

Oslo, Norway – 3 October, 2010

Opening Remarks

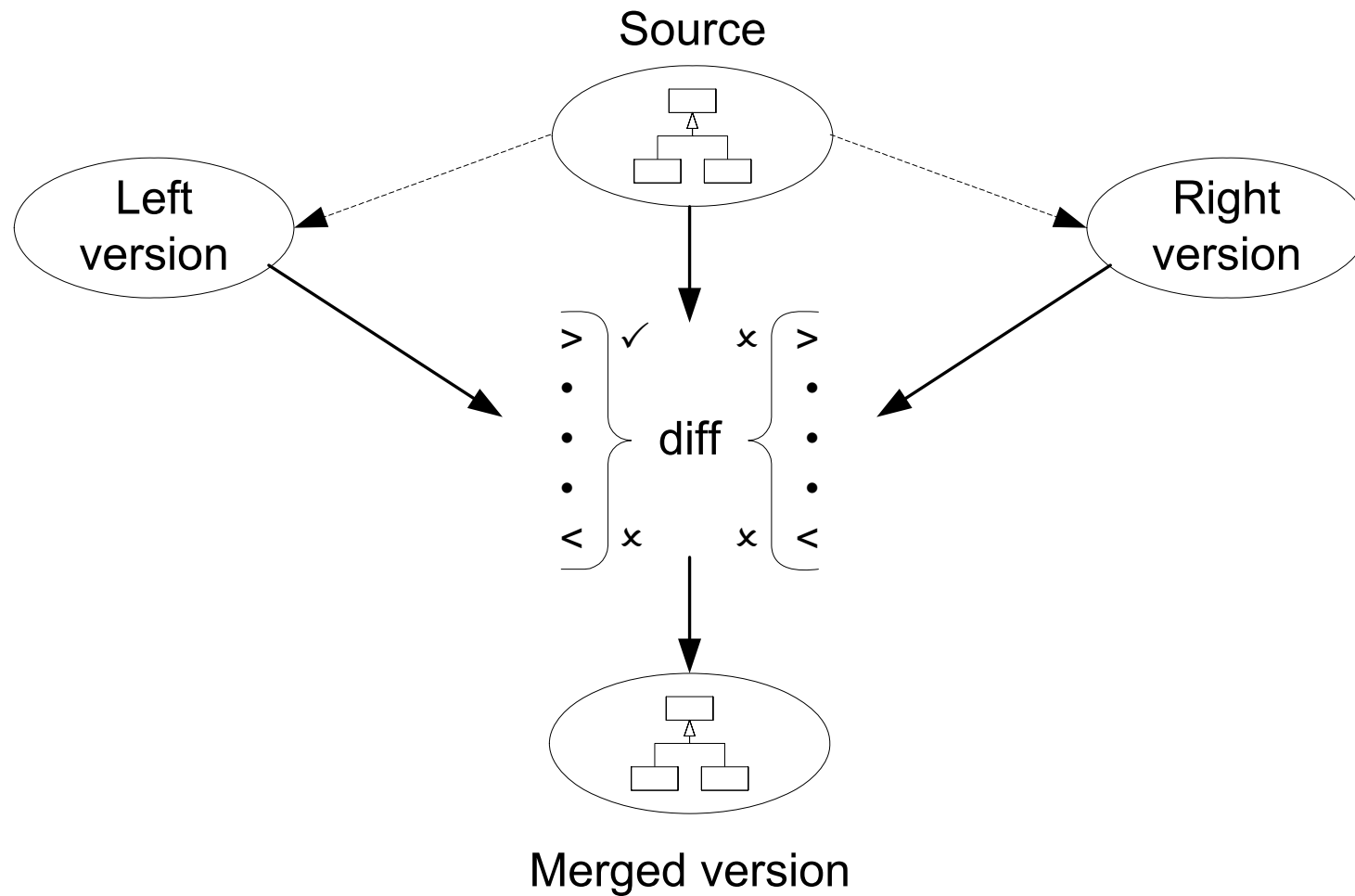
Models Considered

- Decoupling — applicable to models that can be expressed using Ecore
- Mirador — subset of UML class diagrams

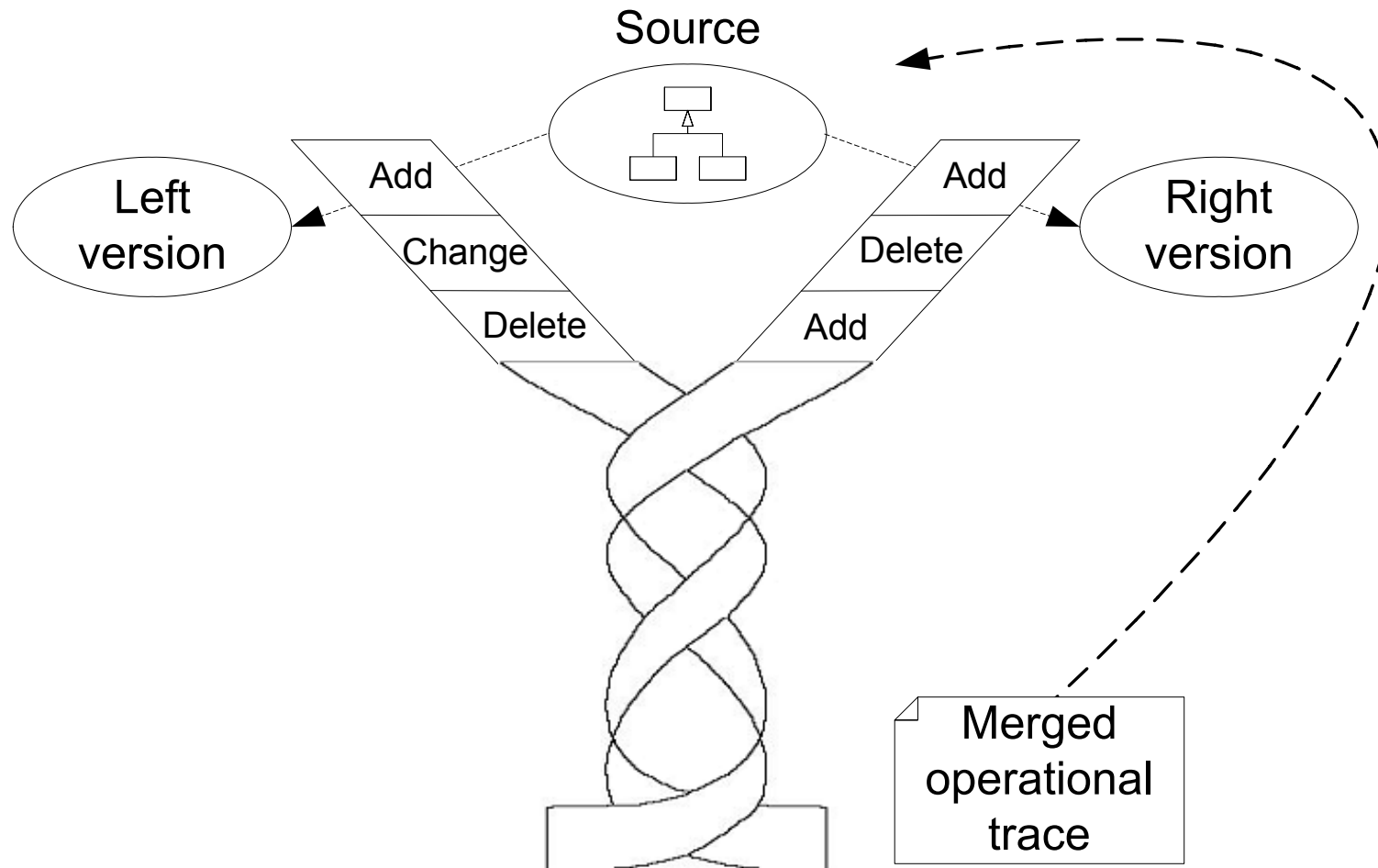
Evolution Challenges Addressed

- Operation-based merging independent of change recording mechanism
- Tracking cumulative change within model
- Merging across different modeling tools

State-Based Merging



Operation-Based Merging



Merge Approaches: State vs. Operation

	State-Based	Operation-Based
Implementation	easier	more difficult
Merge choice granularity	coarse	fine
Solve add-delete problem	3-way merge	nonexistent
Change recorder coupling	none	tight
Can use techniques of other approach	no	yes

Phases of Decoupling Solution

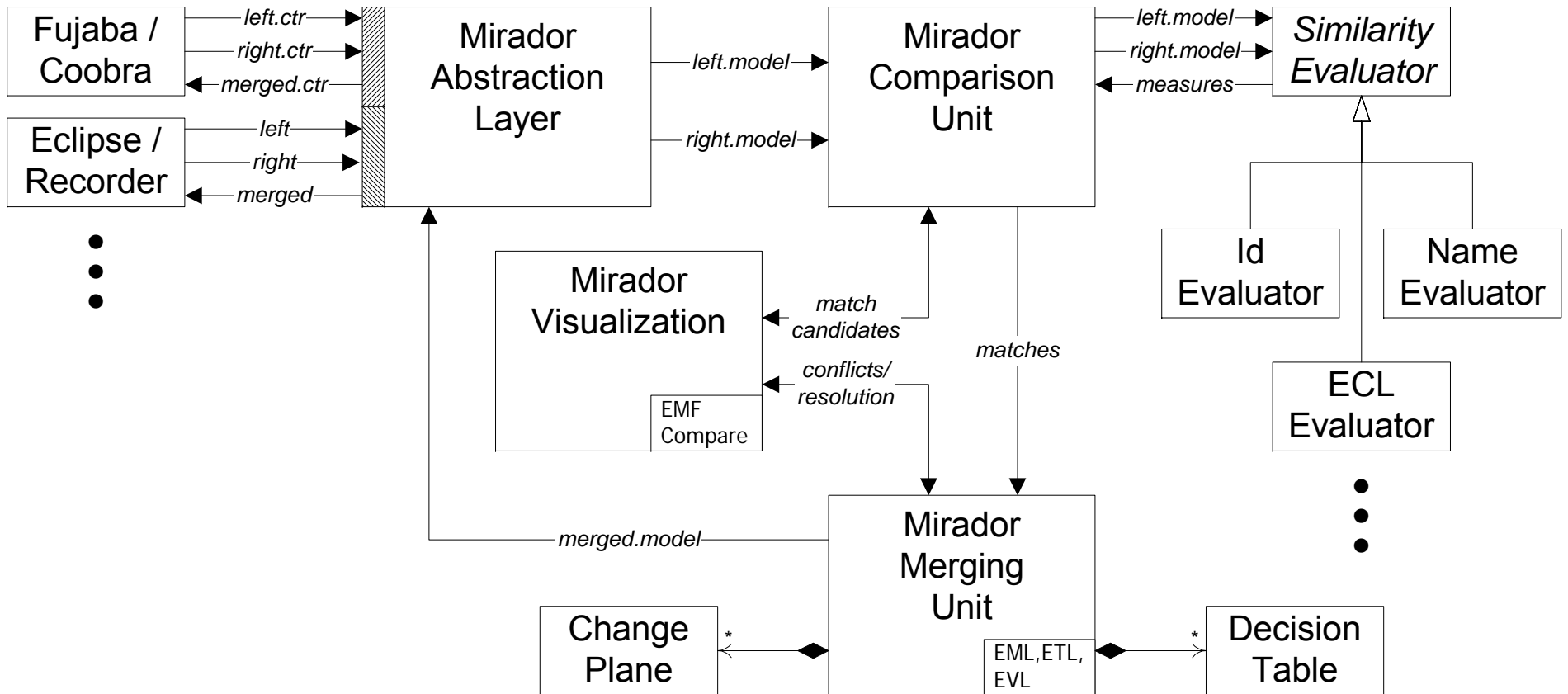
1. Change Record Isolation

- Abstract away change recorder formatting differences
- Employ common structure to represent change operations

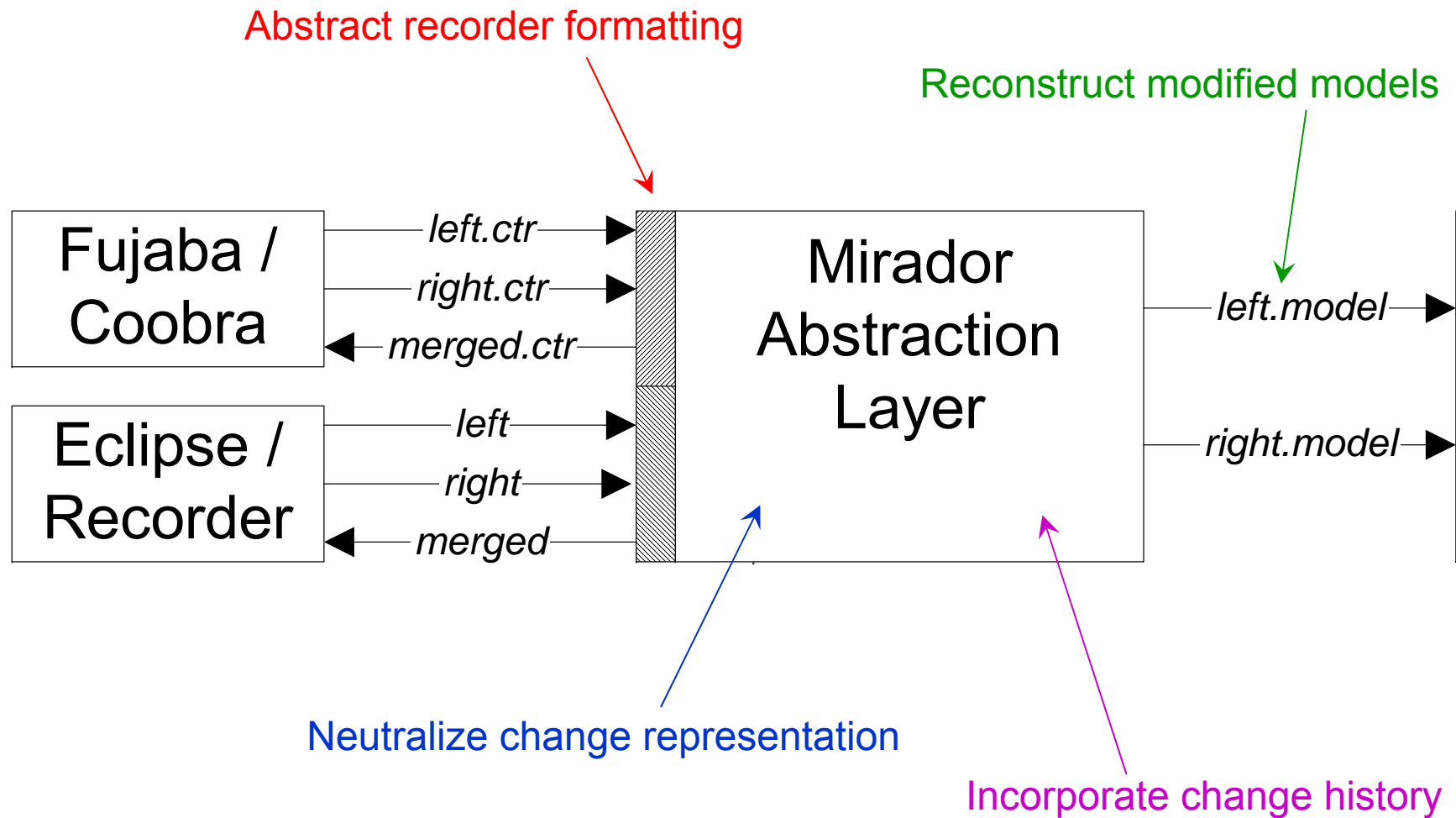
2. Model Evolution

- Incorporate change history into model as incremental differences
- Reconstruct modified models from change history

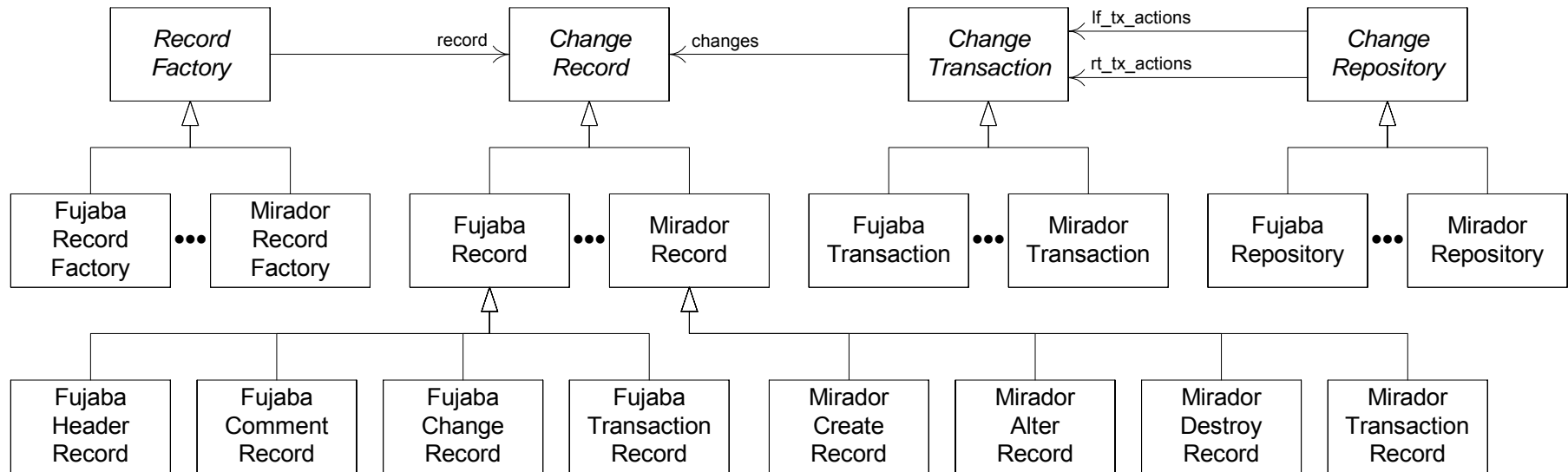
Mirador Architecture



Mirador Change Log Input



Isolation: Common Change Record Hierarchy

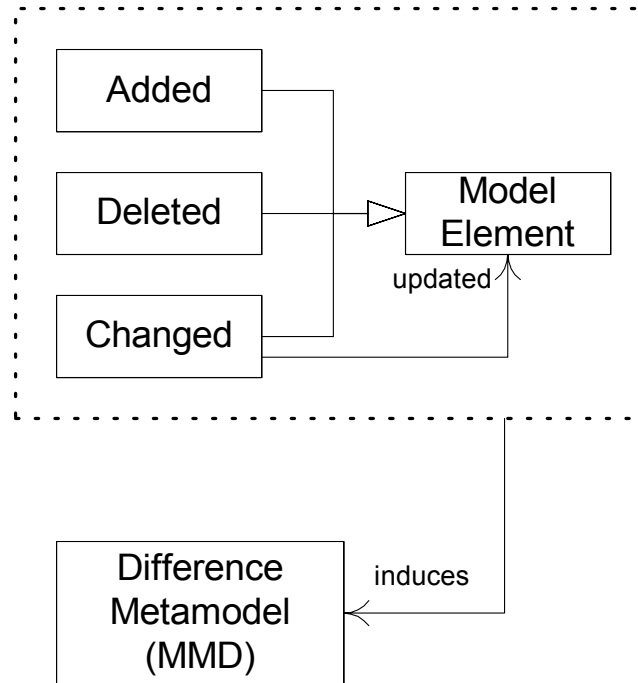


Sub-hierarchy for each recorder type.

Mirador classes form a LCD of merge features.

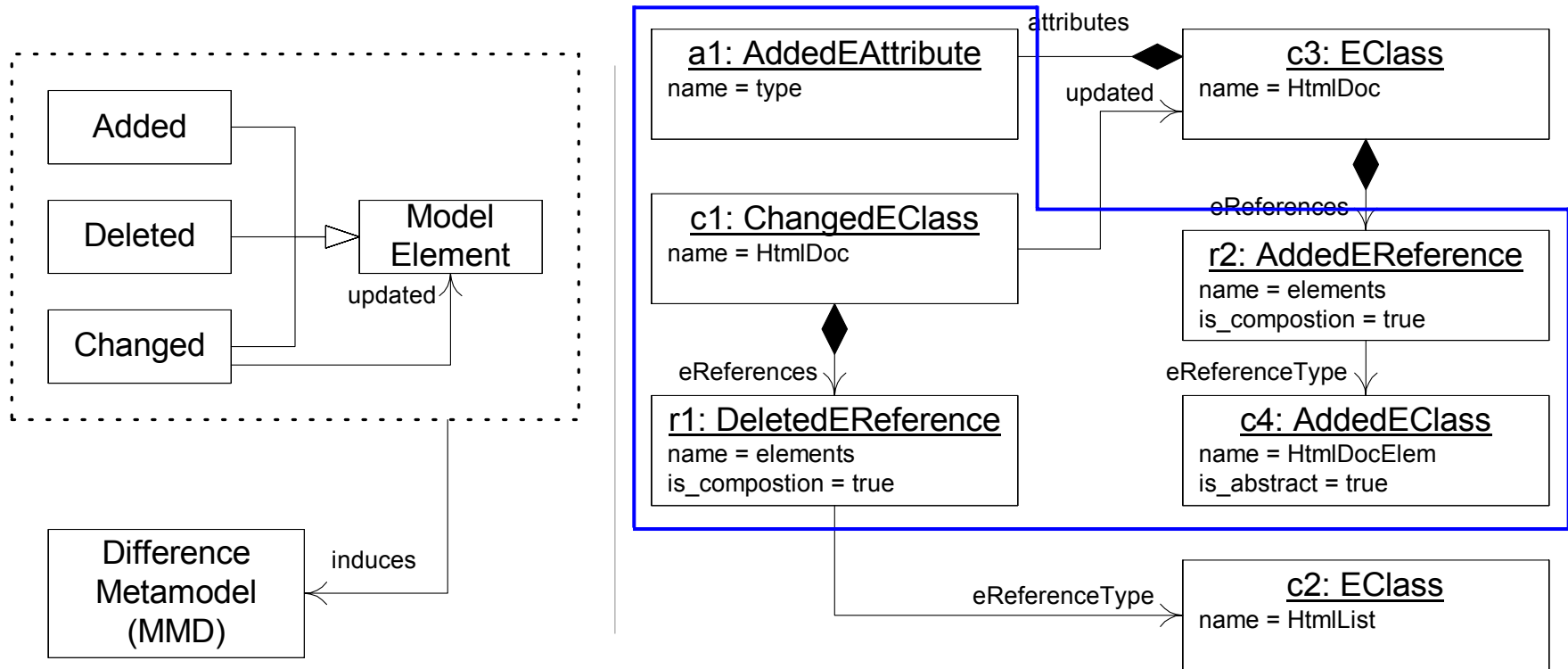
Downstream process need only know LCD format.

Evolution: Aggregate Difference Representation



Cicchetti, Di Ruscio, Pierantonio - 2007

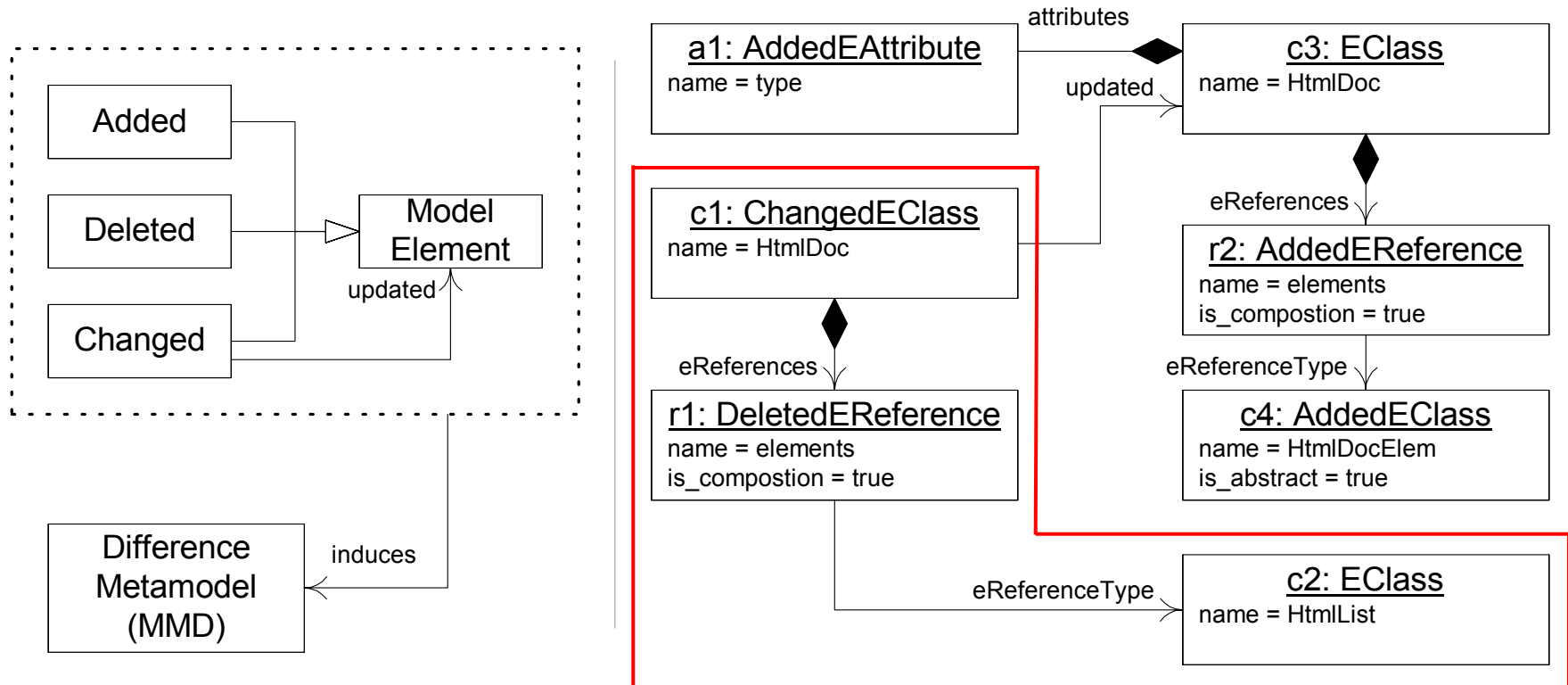
Evolution: Aggregate Difference Representation



Cicchetti, Di Ruscio, Pierantonio - 2007

MMD elements = Added + Changed + Deleted

Evolution: Aggregate Difference Representation

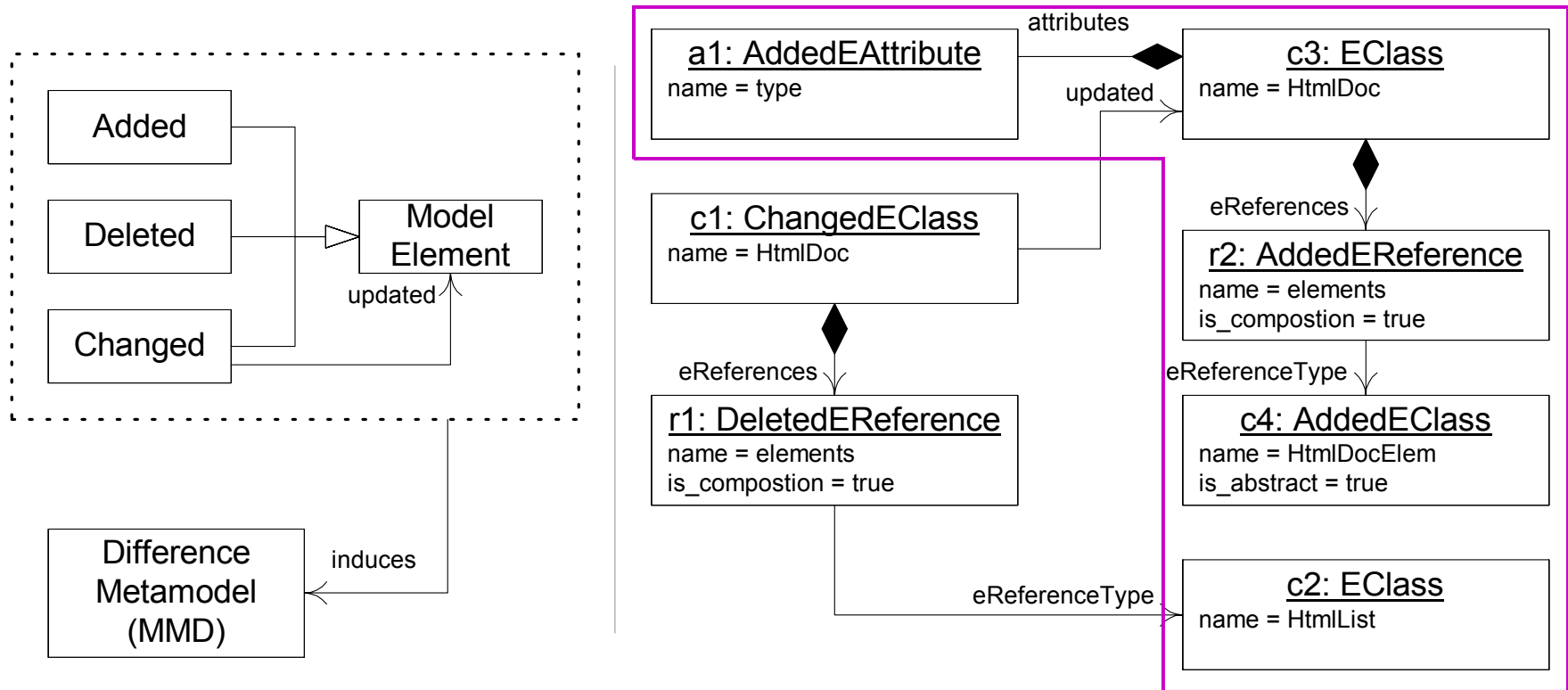


Cicchetti, Di Ruscio, Pierantonio - 2007

MMD elements = Added + Changed + Deleted

Initial model = Unchanged MM + Changed + Deleted

Evolution: Aggregate Difference Representation



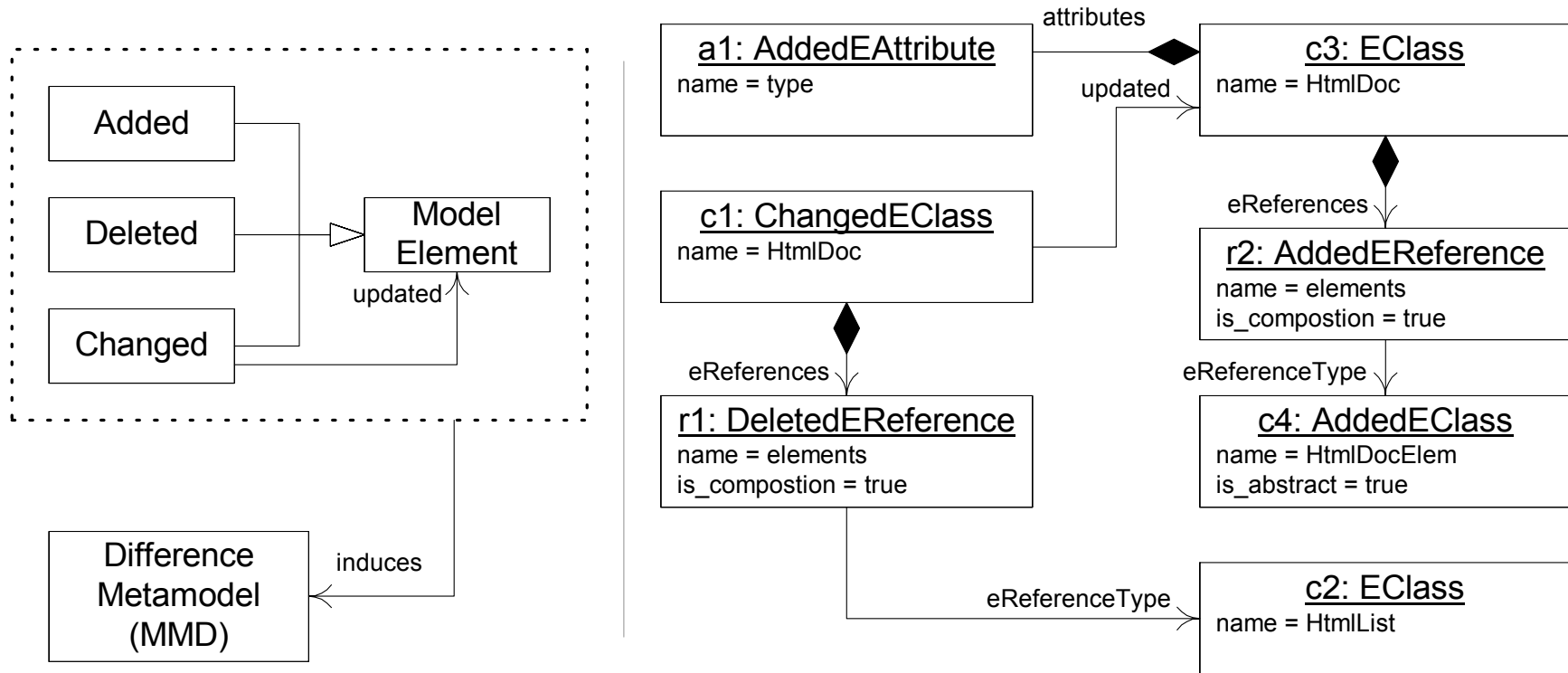
Cicchetti, Di Ruscio, Pierantonio - 2007

MMD elements = Added + Changed + Deleted

Initial model = Unchanged MM + Changed + Deleted

Final model = MM elements + Added

Evolution: Aggregate Difference Representation



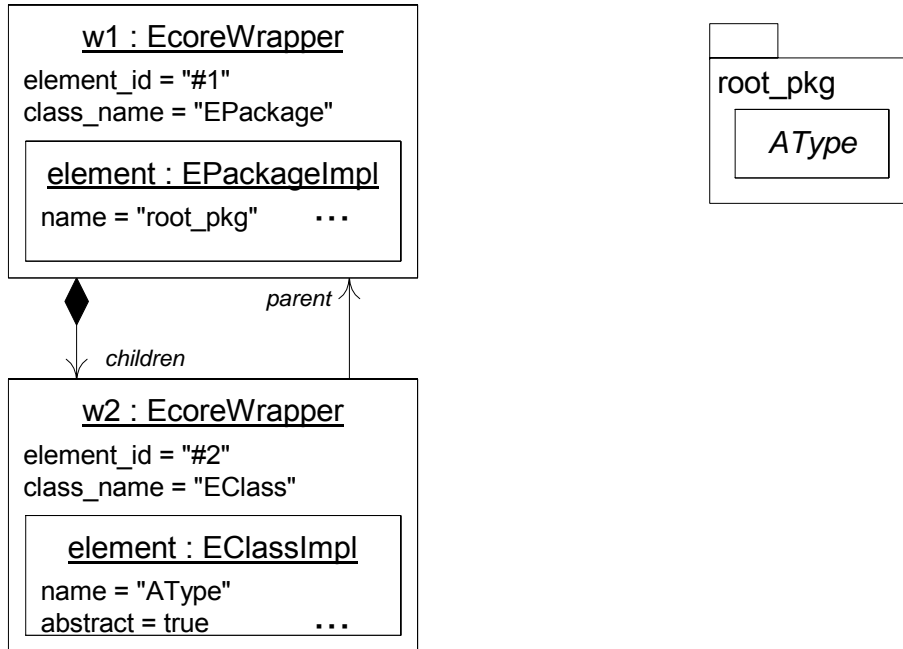
Cicchetti, Di Ruscio, Pierantonio - 2007

Model holds initial and final states.

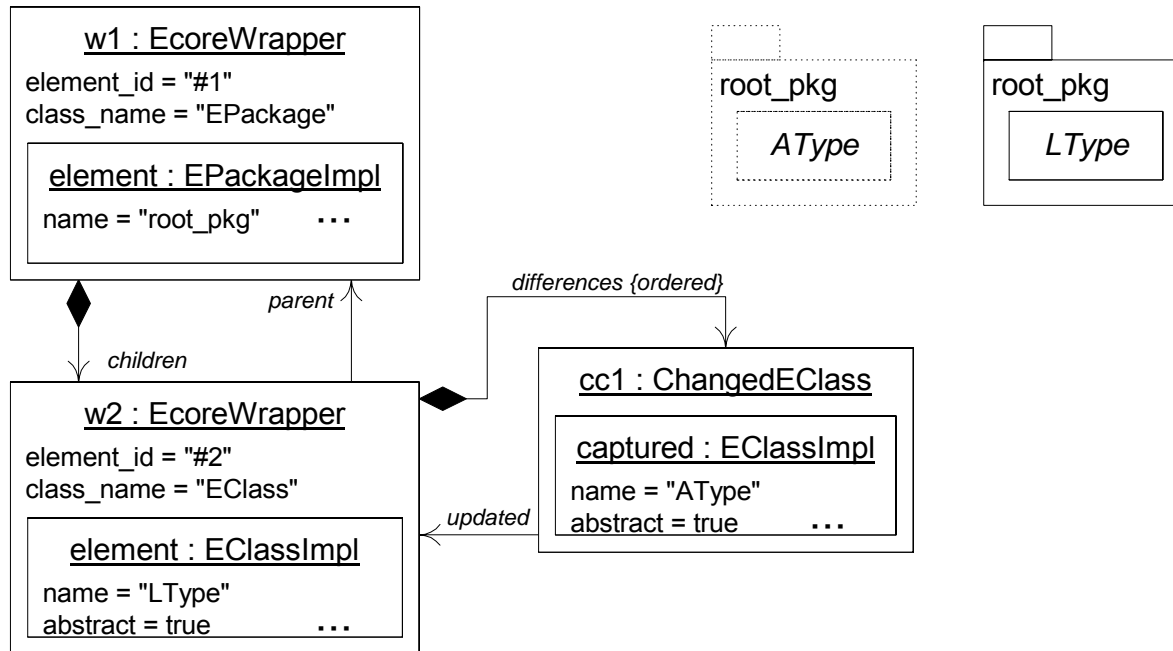
Diff across "updated" link pair to get aggregated change op.

No linkage to Changed and Deleted MMD objects.

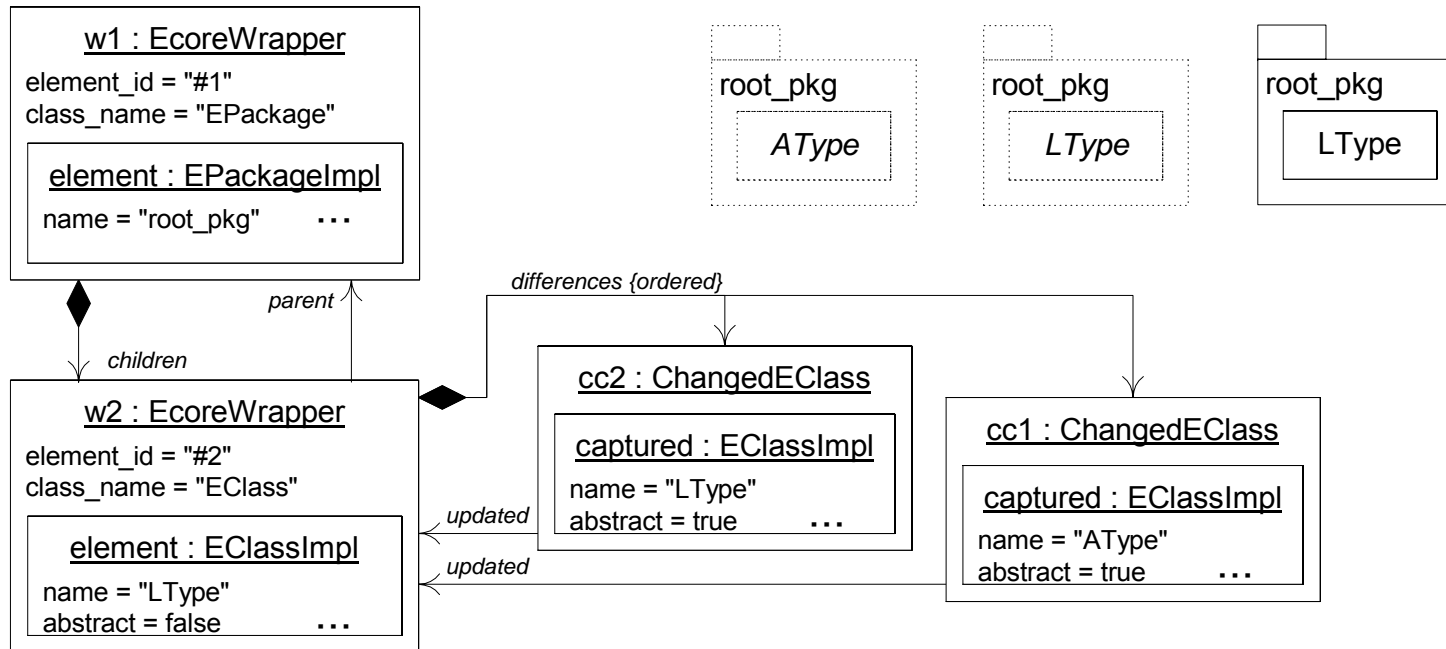
Evolution: Incremental Difference Representation



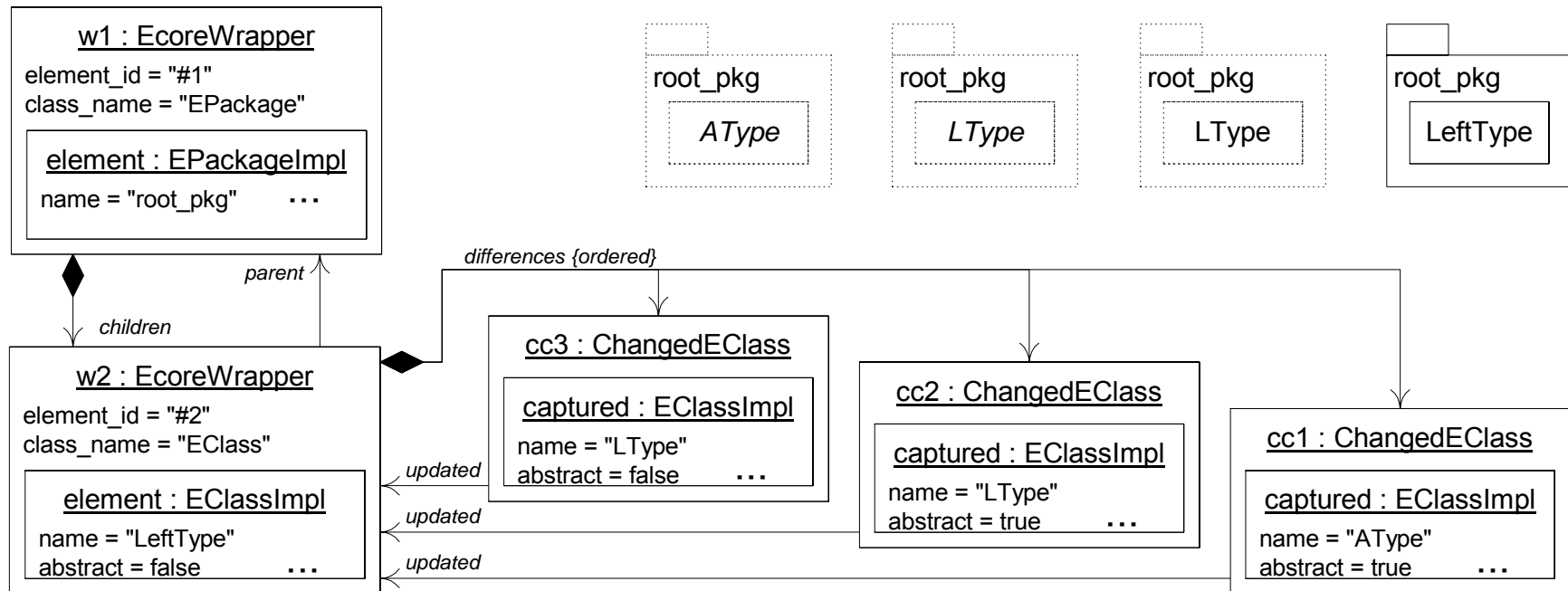
Evolution: Incremental Difference Representation



Evolution: Incremental Difference Representation



Evolution: Incremental Difference Representation



Model holds initial, final, and every state in between.

Diff across *adjacent* "updated" link to get inc. change op.

Wrappers link to Changed and Deleted MMD objects.

Wrappers are Mirador specific.

Decoupling Summary

Input Abstraction	Isolation	Evolution	Output /Export
Recorder → Recorder-specific log (textual) change ops (OO)	Recorder-neutral change ops	Cumulative differences held in model	MMDx model in Ecore

Abstraction

- Abstract away recorder formatting
- Dedicated interface for each supported type of change recorder

Decoupling Summary

Input Abstraction	Isolation	Evolution	Output /Export
Recorder → Recorder-specific log (textual) change ops (OO)	Recorder-neutral change ops	Cumulative differences held in model	MMDx model in Ecore

Isolation

- Removes dependency on change recorder
- Replaces with dependency on common change record classes
- Enables merging of models created with different modeling tools

Decoupling Summary

Input Abstraction	Isolation	Evolution	Output /Export
Recorder → Recorder-specific log (textual) change ops (OO)	Recorder-neutral change ops	Cumulative differences held in model	MMDx model in Ecore

Evolution

- Differences become first class artifacts
- Change operations must be extracted
- Model can be used by other tools
- Wrappers impair complete independence

Decoupling Summary

Input Abstraction	Isolation	Evolution	Output /Export
Recorder → Recorder-specific log (textual) change ops (OO)	Recorder-neutral change ops	Cumulative differences held in model	MMDx model in Ecore

Output / Export

- Exported Ecore model is wrapper free
- Any state may be exported; final is default
- Exported models may be used for any state-based analyses and manipulations

Open Questions

Is ensuring “everything is a model” worthwhile?

How can the effectiveness of a merge be measured?

Thank You

Questions?