



Changes to Network Model Data

CIM University Track II

Pat Brown, EPRI

pbrown@epri.com



Some ongoing work will be tagged

Warning:
Draft!

= not yet reached IEC WG consensus



Change is central...

- So far... snapshots
 - EQ, DL, case inputs / outputs
- But network models are always changing (EQ)
 - Completed construction projects, storm restoration, residential PV installation <- History
 - Generator interconnect requests, transmission line evaluations, developers proposing DER <- Prospective future
- And we want to study different states (SSH)
 - Outages
 - Contingencies
 - Seasonal/future loads
- Describing change (and change over time) is essential to supporting network analysis



CIM model support for change over time

- Approaches to history in other areas of CIM
 - **Mostly CIM is temporally 'flat'**
 - Have scattered support for various forms of history/change:
 - **Instances w/ timestamp** (e.g., maintenance/inspection reports using ProcedureDataSet child classes)
 - **Attribute w/ timestamp** (e.g., telemetered values using MeasurementValue)
 - **Schedules** (e.g., interchange schedule using Curve classes)
- In WG13
 - Needed comprehensive history of change over time
 - But 'obvious' strategies obviously not adequate:
 - periodic snapshots
 - last time updated
 - birth/death dates



What was needed

An information model that describes change

- A technology-independent way to describe a collection of changes to shared data that occurred (or might occur) as a result of some activity
- Needed **CRUD** (creates, updates, deletes)



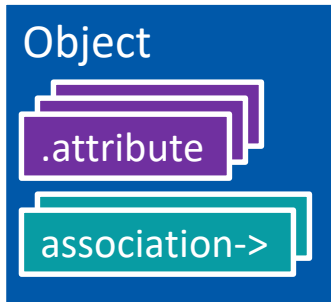
Changes in network model part of CIM

- Original solution: ‘incrementals’ or ‘difference models’
- 61970-552 describes exchange of network model data using RDFS syntax
 - Has ‘full models’ and ‘difference models’
 - Both models expressed in terms of the same CIM classes
 - Has limited amount of meta-data in header (used for exchanging both types of models)
 - Interpretation and application of model completely dependent on header information (**which is not part of the canonical CIM**)



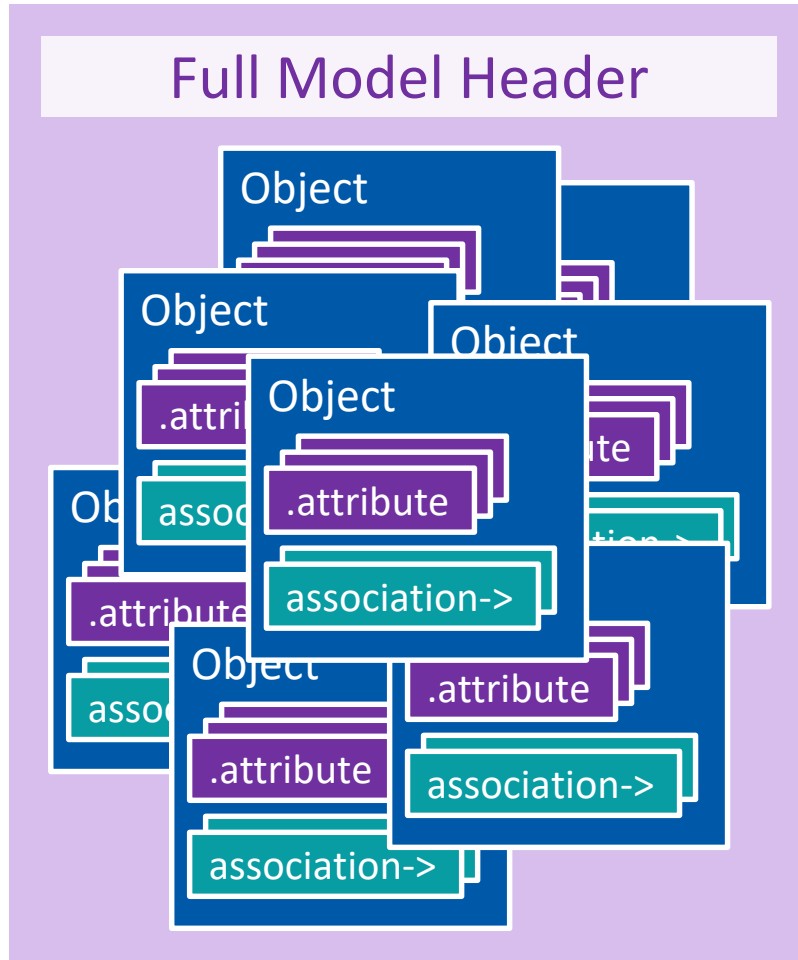
61970-552

- CIM object representation
 - Object instance
 - 0 to many attributes
 - 0 to many associations



61970-552 full model

■ CIM 'full model'

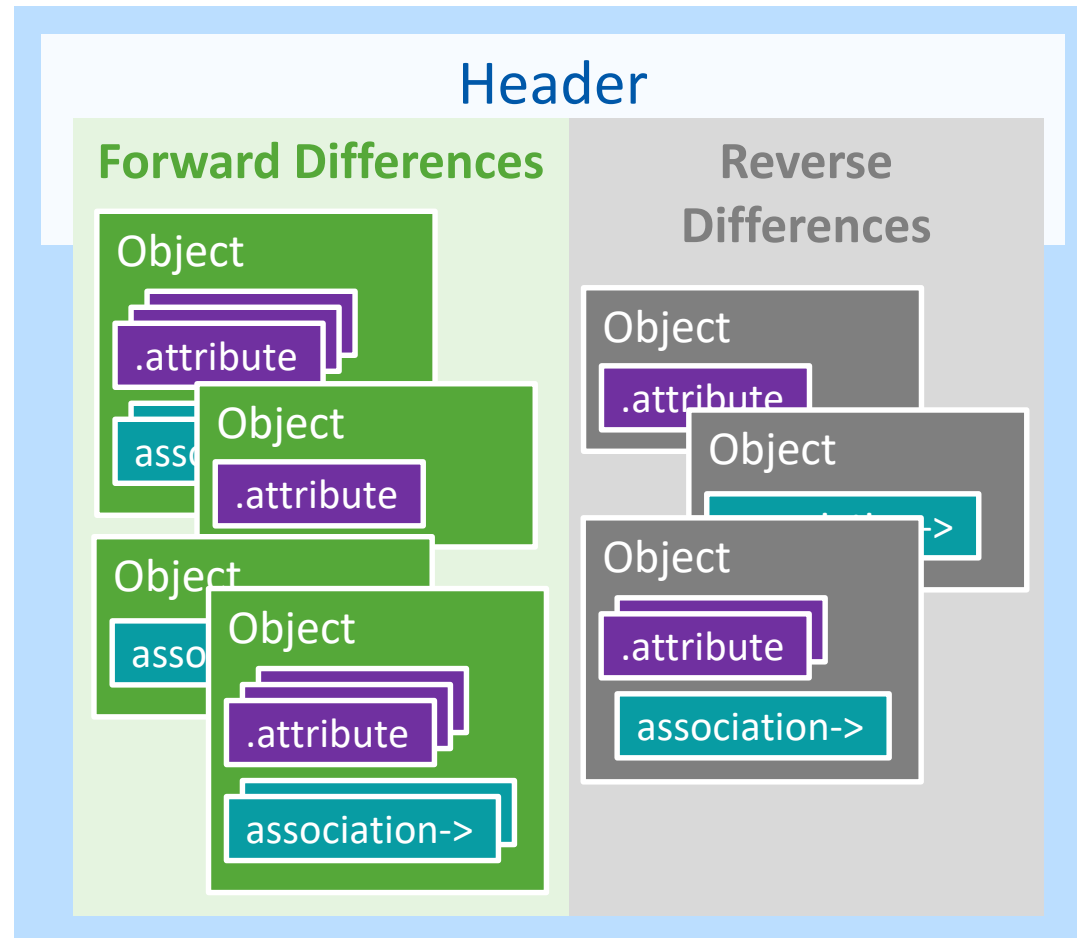


- Has a collection of objects
 - Of specific classes (defined by profiles)
 - Without dangling associations
 - That are electrically logical
- Header
 - Version
 - Identification of modeling authority, type of model and profile to which contained objects conform



61970-552 change model

- CIM 'difference model'
- Has 2 collections of objects
 - one 'forward', one 'reverse'
- Header
 - Version
 - Identification of modeling authority, type of model and profile to which contained objects conform
 - Meaning of collections of objects



61970-552 change model

- Getting **CRUD** from 'forward' and 'reverse' statements

Object, Object.attribute or Object association->	In Forward Differences	Not in Forward Differences
In Reverse Differences	UPDATE	DELETE
Not in Reverse Differences	CREATE	<i>unchanged</i>



61970-552 change model

- Difference model drawbacks
 - Header is not part of canonical UML model
 - > Neither full nor change models are identifiable, persistent 'things'
 - Changes are described by combination of objects (statements) and header
 - > Changes aren't fully modelled in canonical UML
 - > Receiver must create own storage strategy
 - Changes have no business context information
 - > No ability to describe purpose, nature, timing, or intended usage of changes



New CIM change model

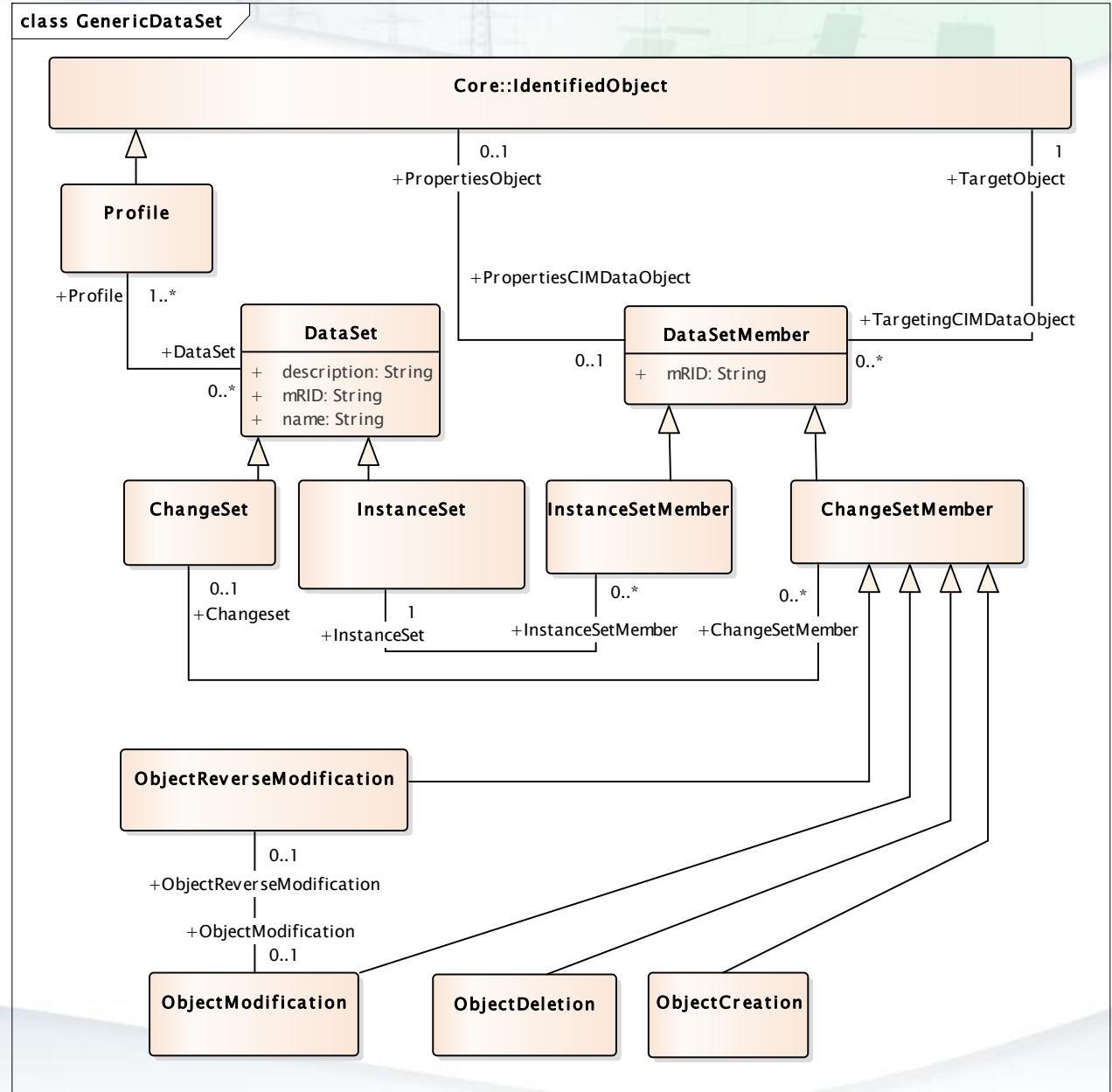
■ Requirements

- Sets of changes have identity, persistence and business context
 - > Explicitly model collections of changes (ChangeSets) in UML
 - > Allow modelling of real-world 'things' to leverage ChangeSets
 - Historic changes to physical network model (result of field work)
 - Future projects
 - Outages
 - Contingencies
 - Study scenarios
- Changes themselves are clearly described, not inferred, and are not dependant on serialization
 - > Explicitly model changes in UML
 - Object creation
 - Object deletion
 - Property modification (attributes and associations)



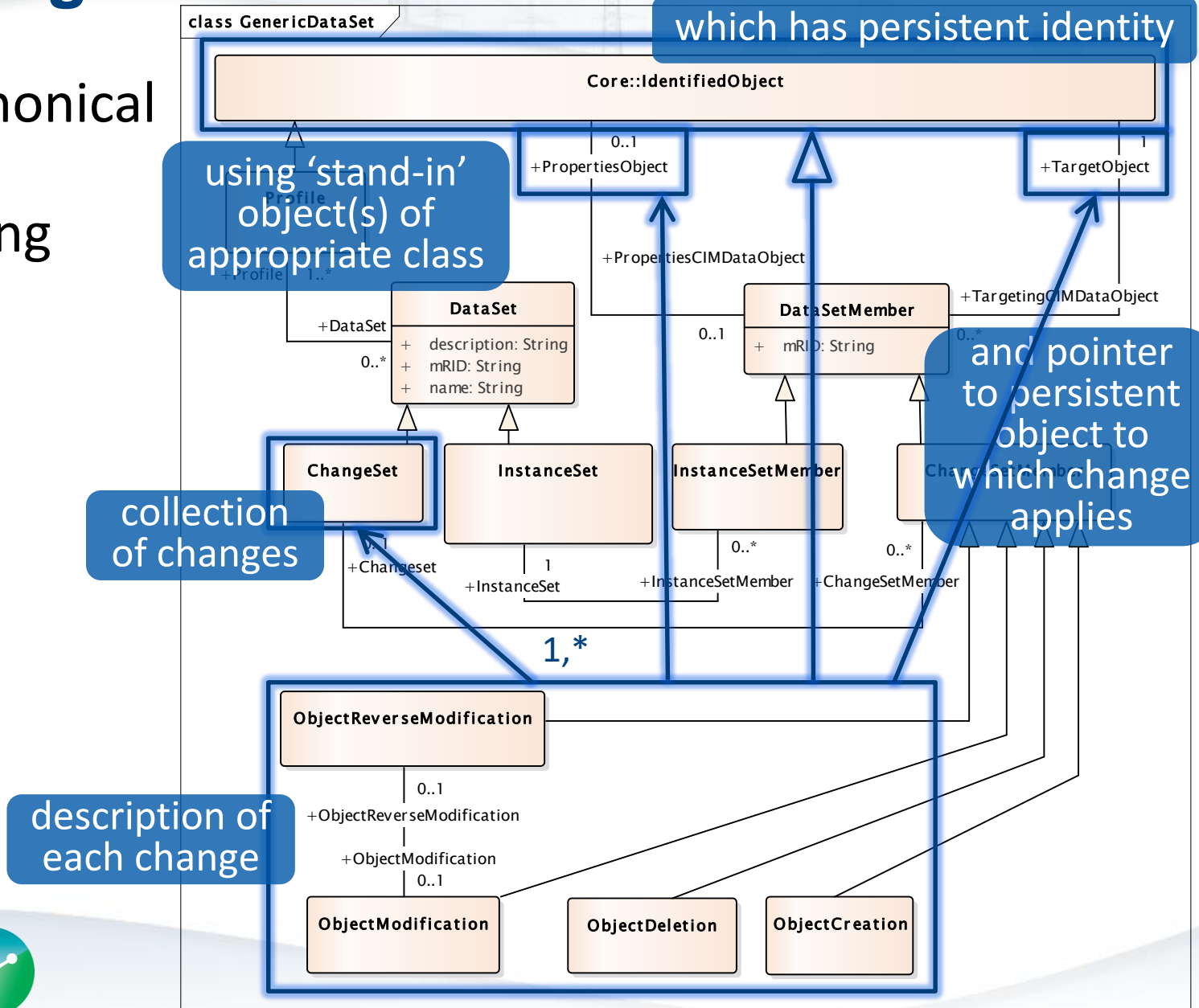
CIM change model - UML

- New canonical model describing changes



CIM change model - UML

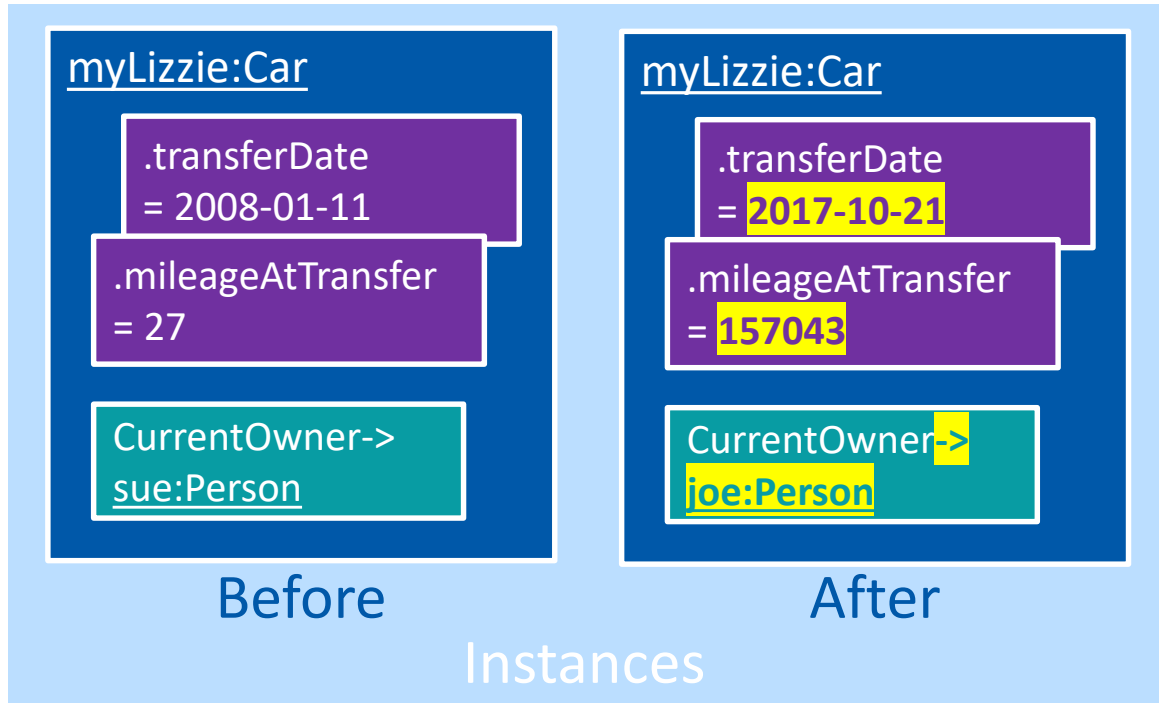
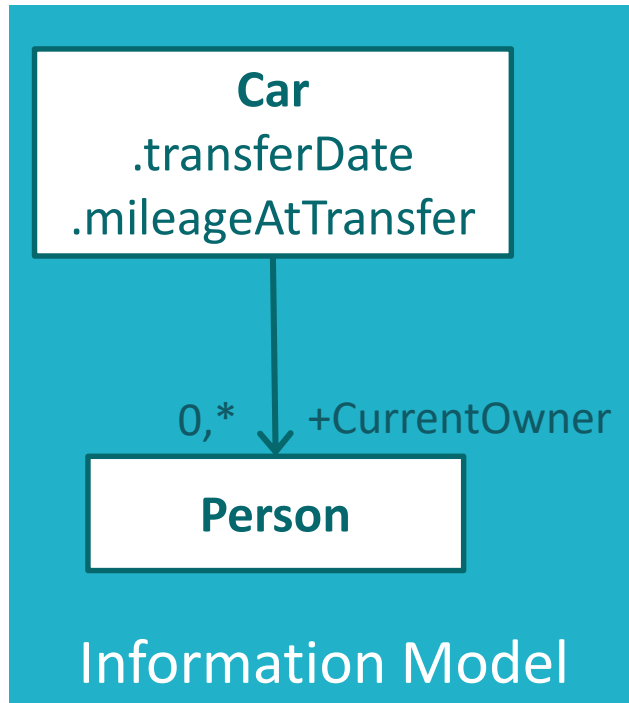
- New canonical model describing changes



CIM change model – simple example

- Example of model describing changes

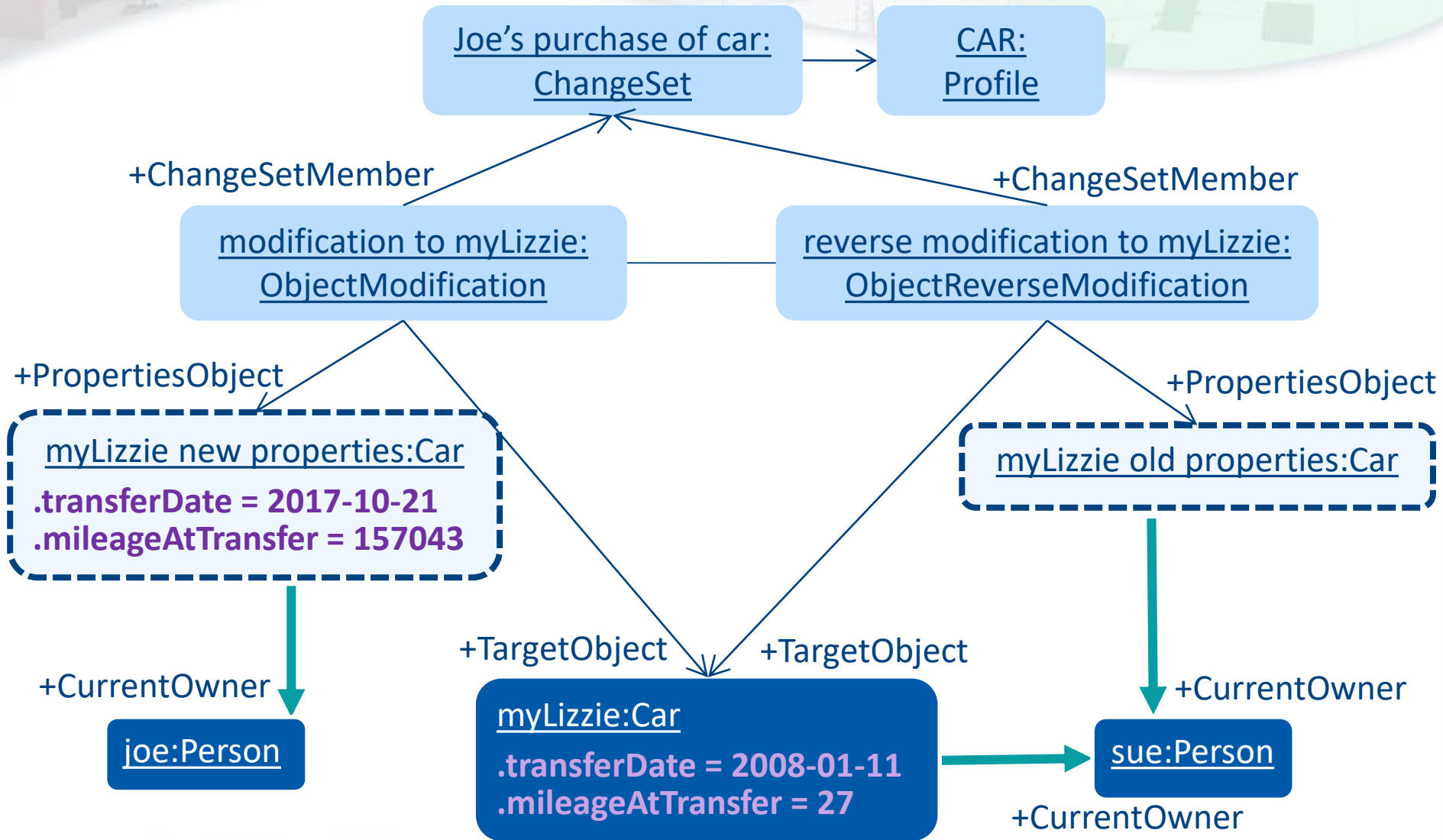
Sue sells car (named ‘My Lizzie’) to Joe,
date and mileage at sale are tracked



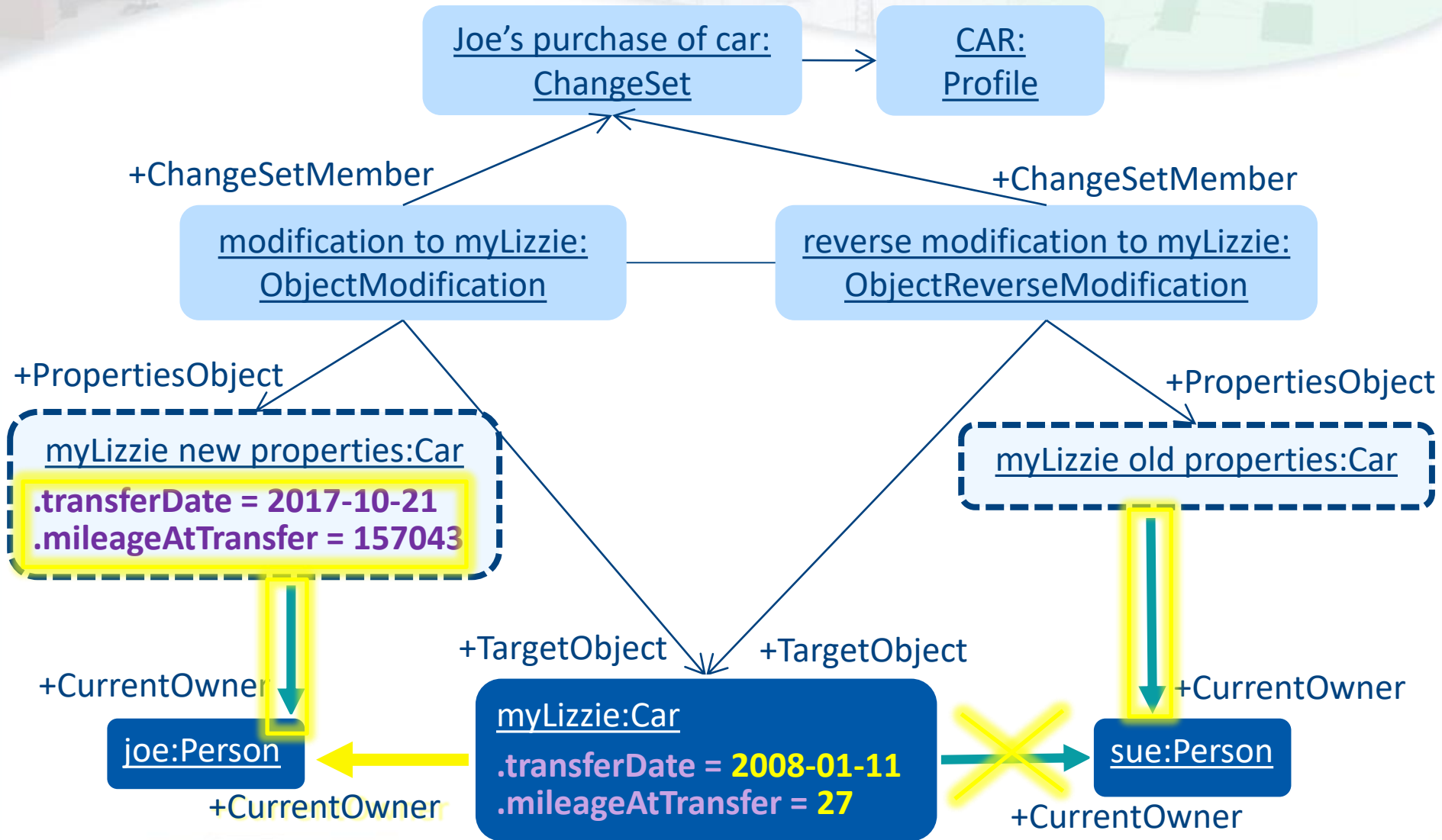
- On existing **myLizzie:Car** object
 - Update 2 attribute values
 - Change 1 association on existing object



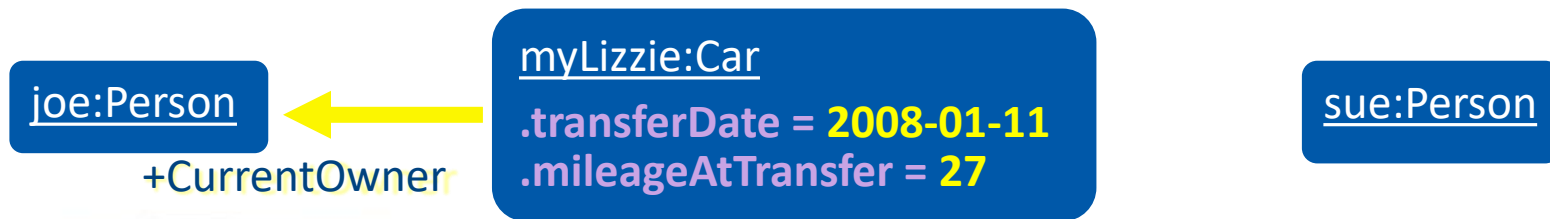
CIM change model – change class instances



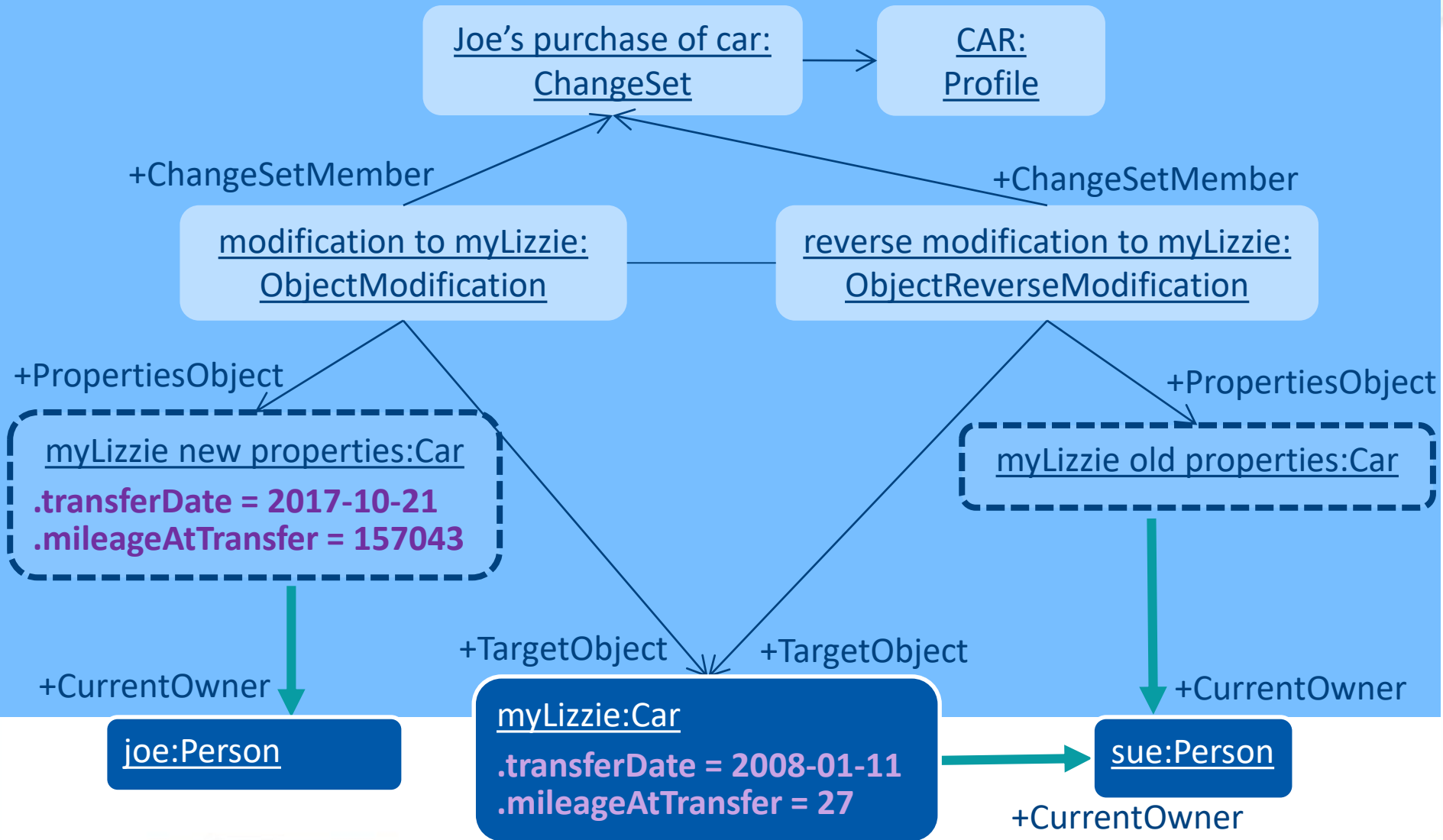
CIM change model – in action



CIM change model – application result

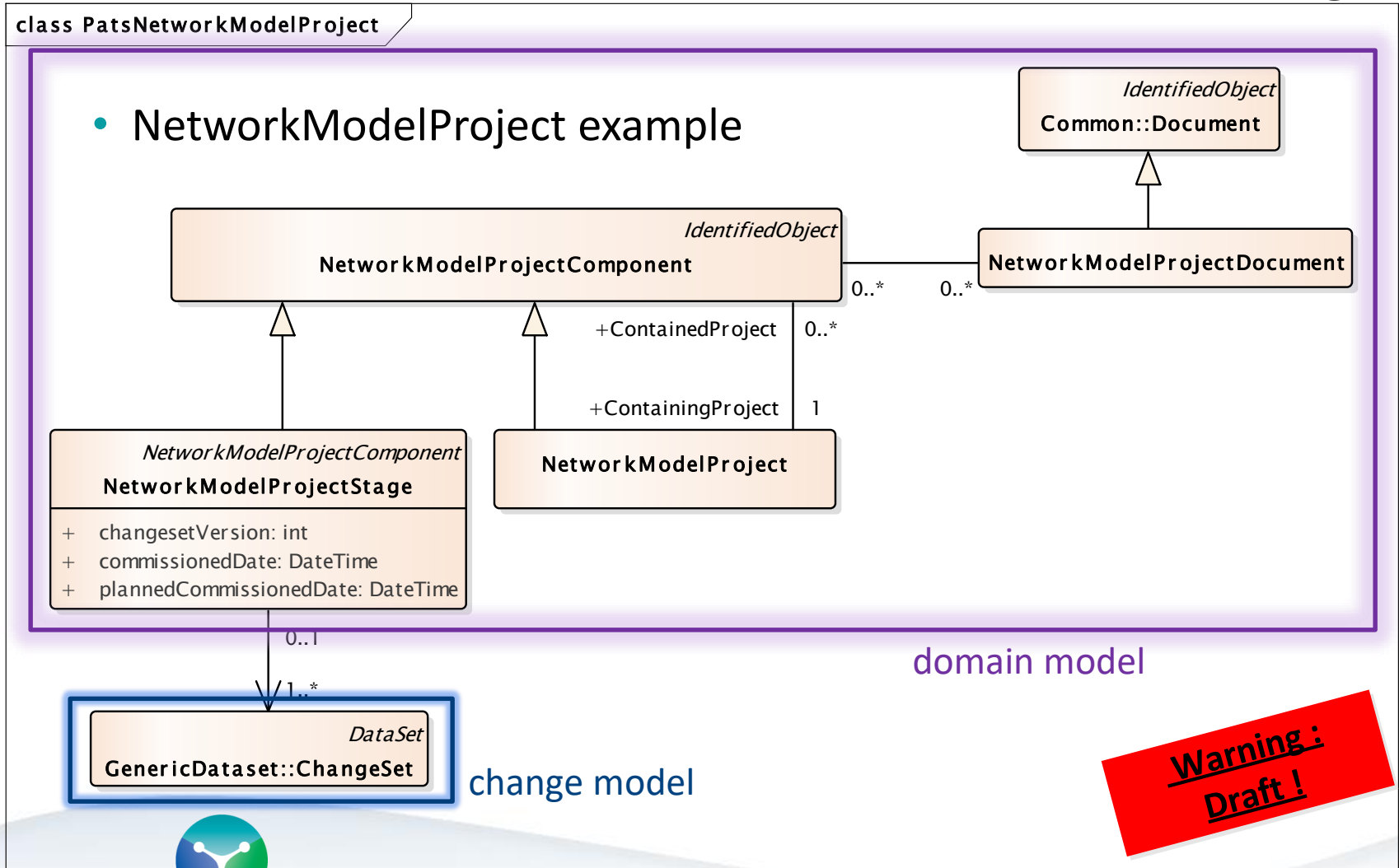


CIM change model – change itself described



CIM change model – potential uses

- New canonical models for entities that include changes

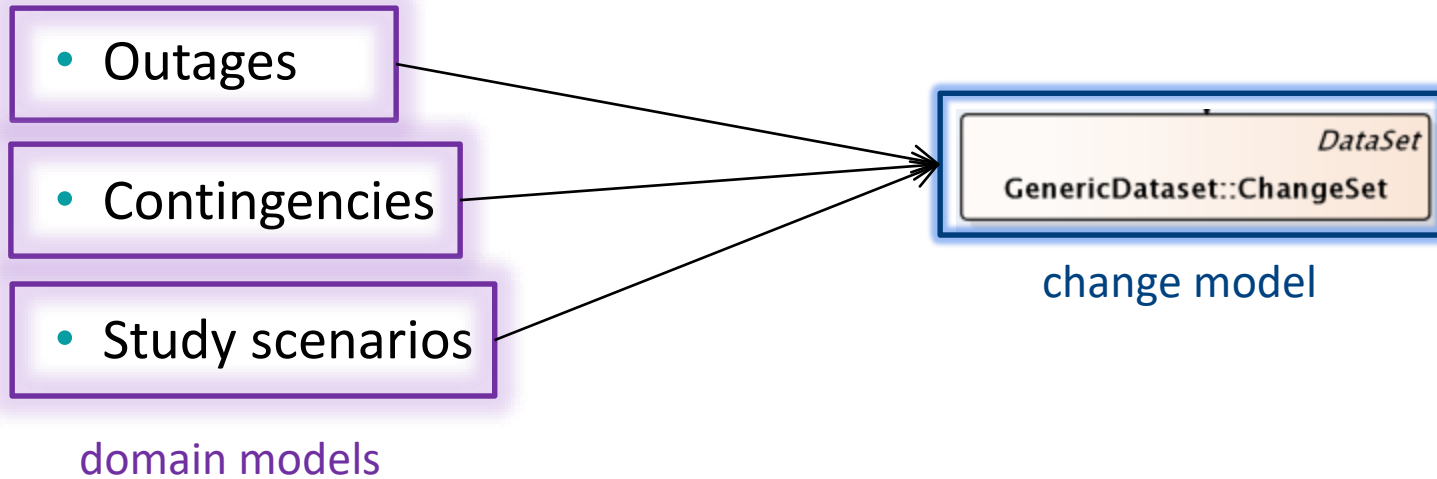


**Warning:
Draft!**



CIM change model – potential uses

- New canonical models for entities that include changes



Warning :
Draft !



CIM change model summary

- In the network analysis domain, change matters
 - past completed (for version history)
 - future prospective (for case creation)
- CIM UML change model allows:
 - Changes
 - Sets of changes
 - 'Things' which refer to sets of changesto be treated as just one more kind of shared data
- Change models have value to other parts of the CIM where history matters...

