Multi Model Storage using EMF and GMF

Helge Sören Klimek

"Rien ne se perd, rien ne se crée, tout se transforme."
"Nothing is lost, nothing is created, all is transformed."

-- Antoine-Laurent de Lavoisier (1743-1794)

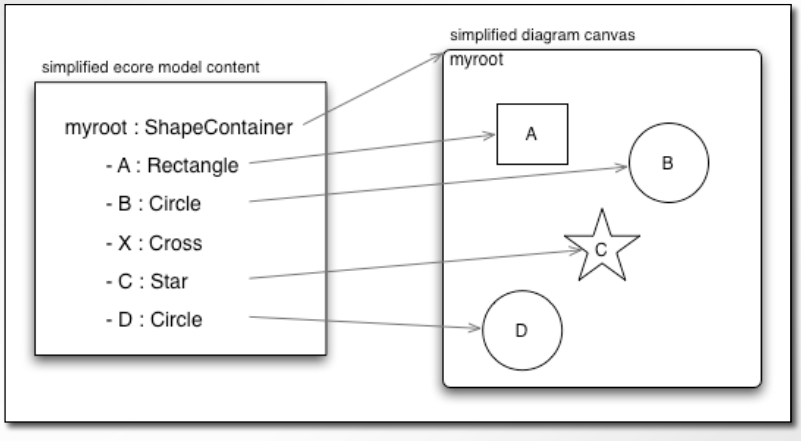
Introduction

- Crowded diagrams in graphical model editors are difficult to read
  - Limitation on the number of elements in a diagram
  - Large models have to be split up in smaller portions
- "*Big models*"
  - Different diagrams based on the same model
  - Single diagrams do not contain all the model elements
  - Model elements are (re-)used in different diagrams (e.g. for displaying different semantics)
- Models are not started from scratch, but based on existing models

Graphical Modelling Framework (GMF)

- Generates editors for EMF based meta models
- *Default* GMF behaviour:
  - GMF maps 1 meta-model element on 0..1 diagram element
  - *"Type based"* mapping
  - Diagram root element contains diagram elements
    - All elements in the root will be displayed in the diagram (iff their types are mapped)

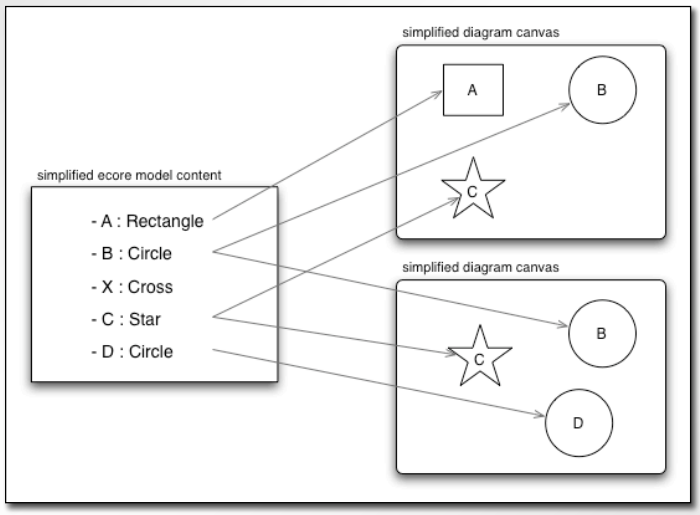
Graphical Modelling Framework (GMF)



GMF Requirements

- One diagram is insufficient for big models
- Mapping of 1 model element to 0..n diagram elements is required
  - No common diagram root
  - Direct mapping not possible
- Reuse as much as possible from generated GMF code, but
  - Display a subset of elements in one diagram
  - Different diagrams based on same main model
  - Instance based diagrams

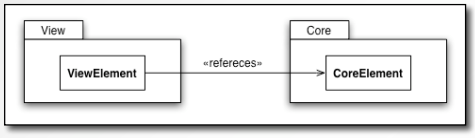
GMF Requirements

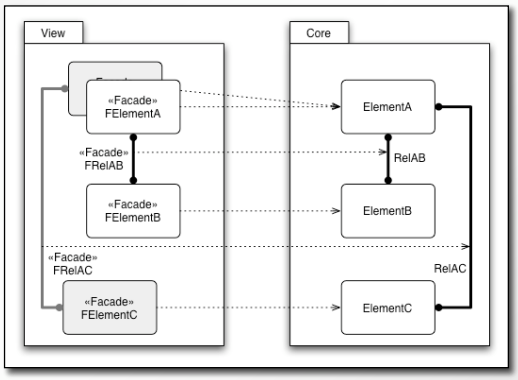


Facades

- The meta-model is divided into two parts: Core and View
- Core and View elements are uniquely identifiable
- The GMF diagram (root) element is a Core element
- Every diagram element is a View element
- Every View element is a *facade* for a Core element and references that Core element

Facades





Modifications (1/3)

- Facades need to resolve their core elements
  - Unique identification of model elements
  - Default Identification of elements depends on resource (e.g. XMI default: indices)
  - EMF allows to define ID fields which were used for identification
- Element creation
  - Facades reference core elements
  - Every facade has to be stored in the model, too
  - GMF advices as hook for manipulating elements

Modifications (2/3)

- Removing / Deleting elements
  - *Deleting*: physically erases an element from the model
  - *Removing*: *deletes* the facade element only and keeps the core element, thereby detaching an element from the view.
  - How to handle the removal of not needed elements
    - Deletion of core elements may harm existing diagrams
    - Keeping all core elements leads to pollution of the main model
  - Concept of reference counting may help
    - Similar to garbage collector
    - Running it manually allows to keep elements for a certain time period.
    - Garbage collection is meta-model specific
  - Always *remove* and manually start garbage collector

Modifications (3/3)

- Resolving edges
  - View elements keep references to their core counterparts
  - References are unidirectional: An edge in the core part knows its referenced core elements but not the view elements in a diagram (A core edge references core elements only!)
  - Additional tooling for resolving edges
    - *Assumption:* Diagrams are valid model instances
    - Iterate over view elements of the diagram
      - Look up referenced core IDs
      - Compare with core IDs from edges “to” and “from” side until match is found

Thank you.