Data Model Workshop

Annotation Session

Mark Cresitello-Dittmar

May. 28, 2021

Annotation

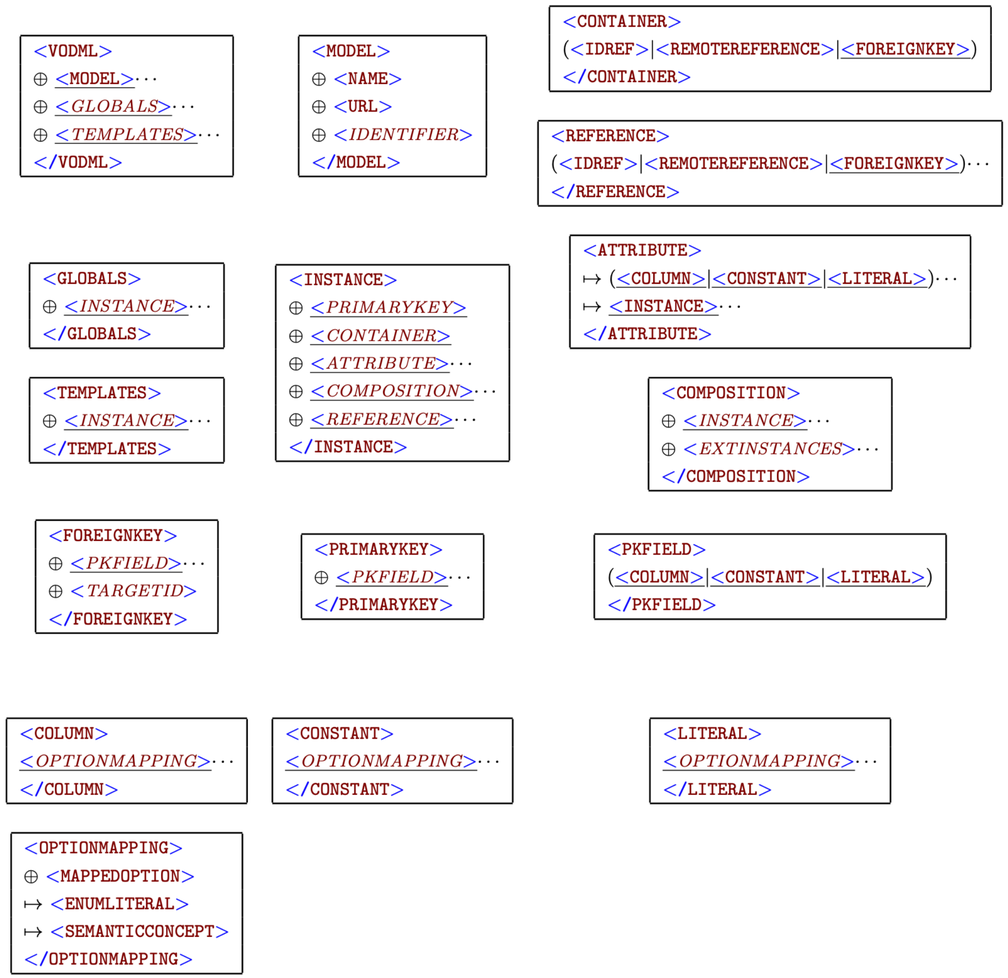
Basis for comparison

- VO-DML Mapping Syntax V1.0 [WD - 20170323]
  - Played a small role in the generation of the syntax
  - Used the syntax in workshop case implementations
- ModelInstanceInVOT - V1.0 [WD - 20200915]
  - Thorough read of the syntax
  - Generated element hierarchy diagram to aid comparison
- **Both shown to support the workshop use cases**
  - It is possible other cases would expose limitations in either syntax

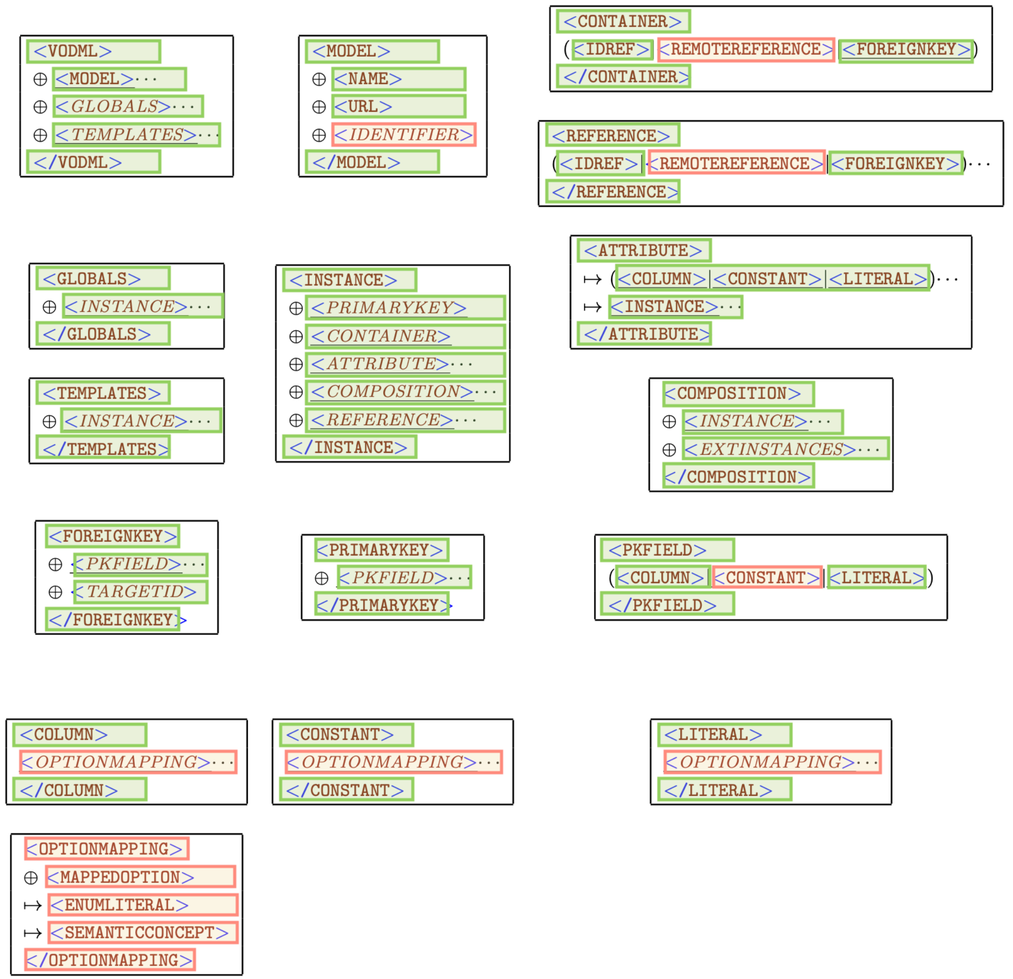
Coverage

Usage within workshop cases

7.3 VODML Element Hierarchy

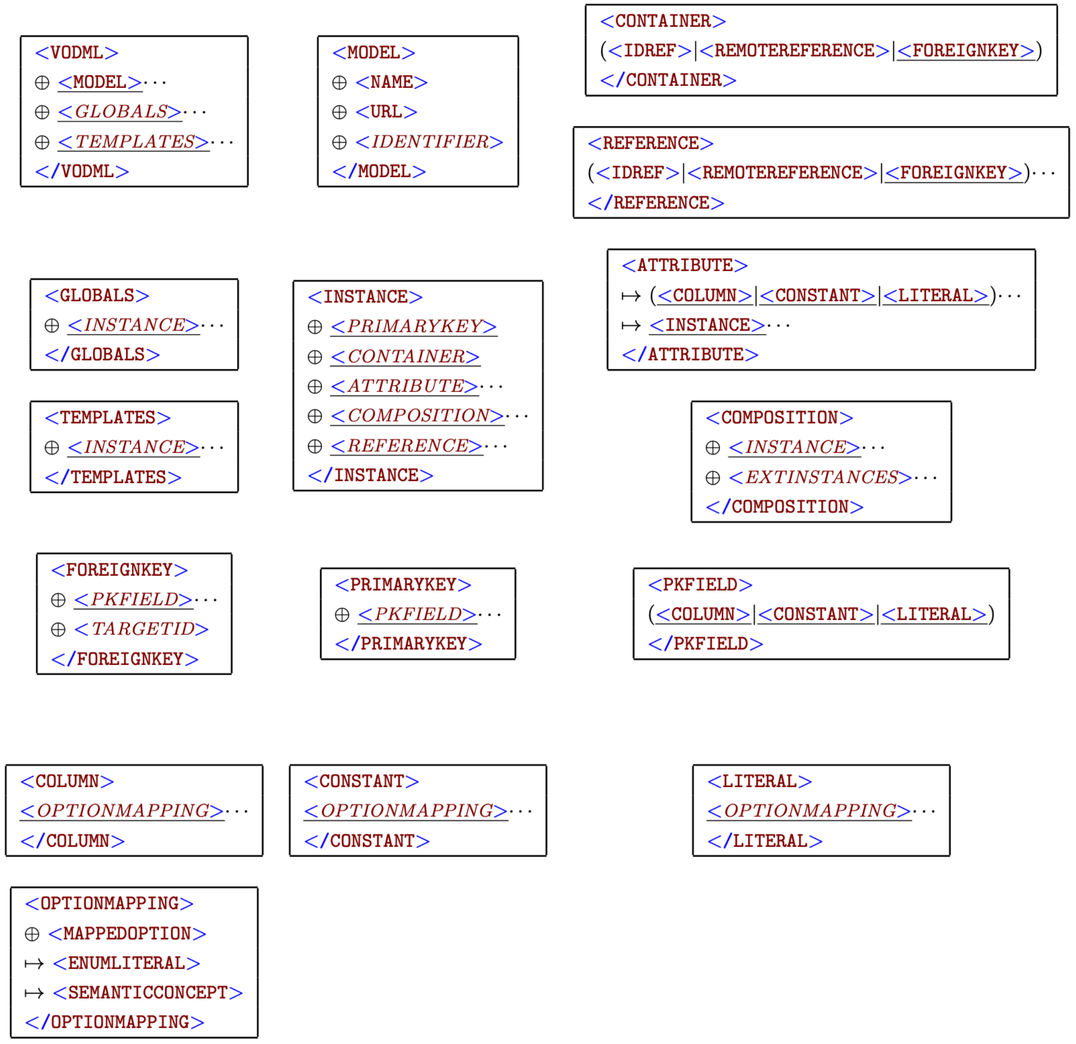


7.3 VODML Element Hierarchy



VO-DML Mapping Syntax

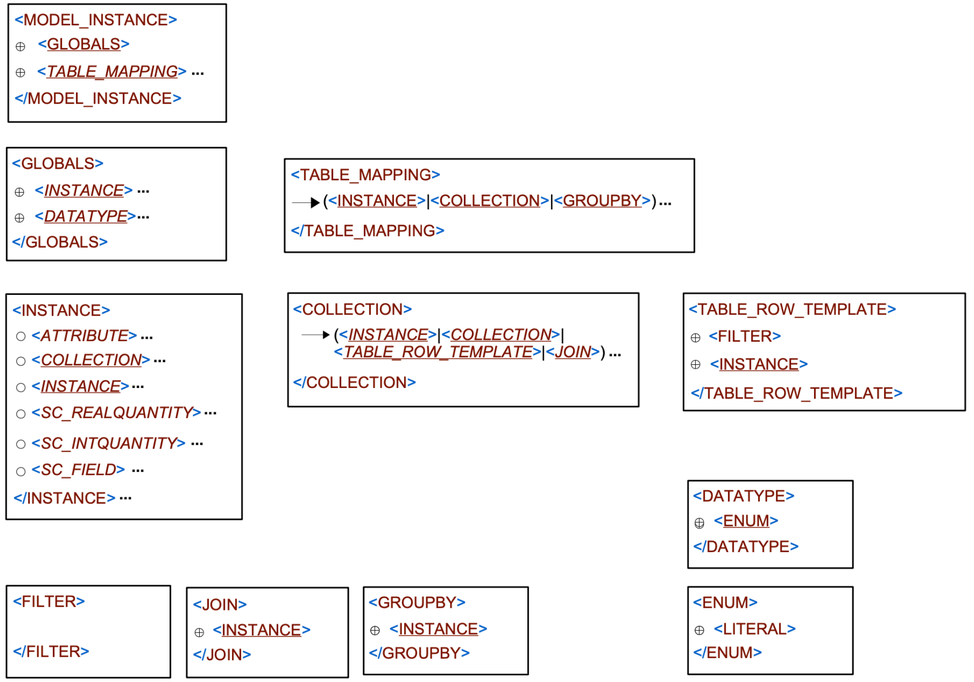
7.3 VODML Element Hierarchy



ModelInstanceInVOT Syntax

Element Hierarchy

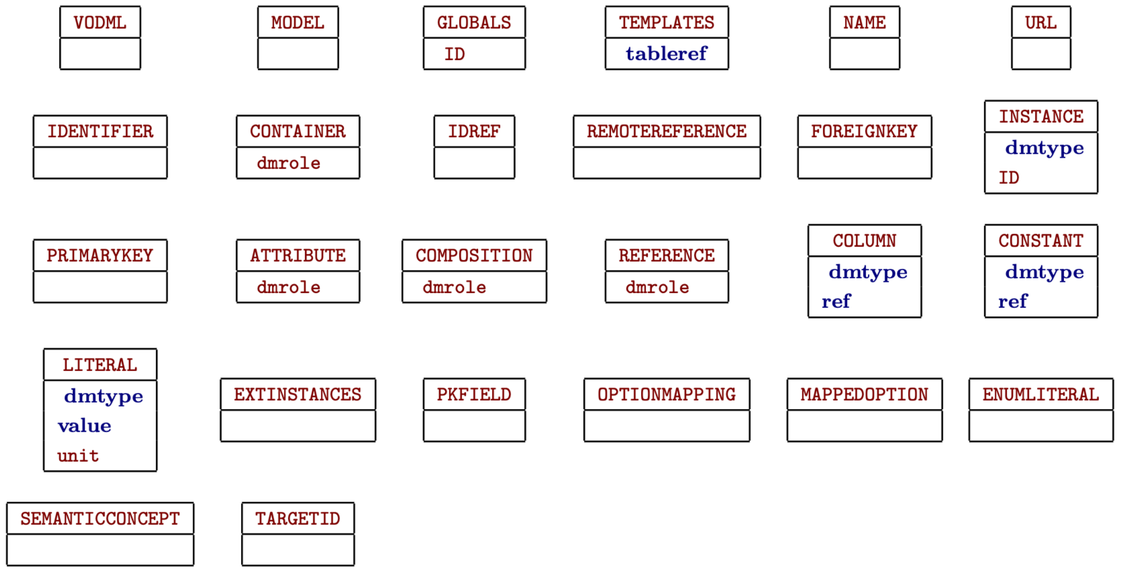
- *italic* – optional
- underlined – has subelements
- ⊕ - order is mandatory
- ○ - in any order
- ⟶ - choice between alternatives
- ··· - may be repeated



Includes elements with sub-elements only

VO-DML Mapping Syntax

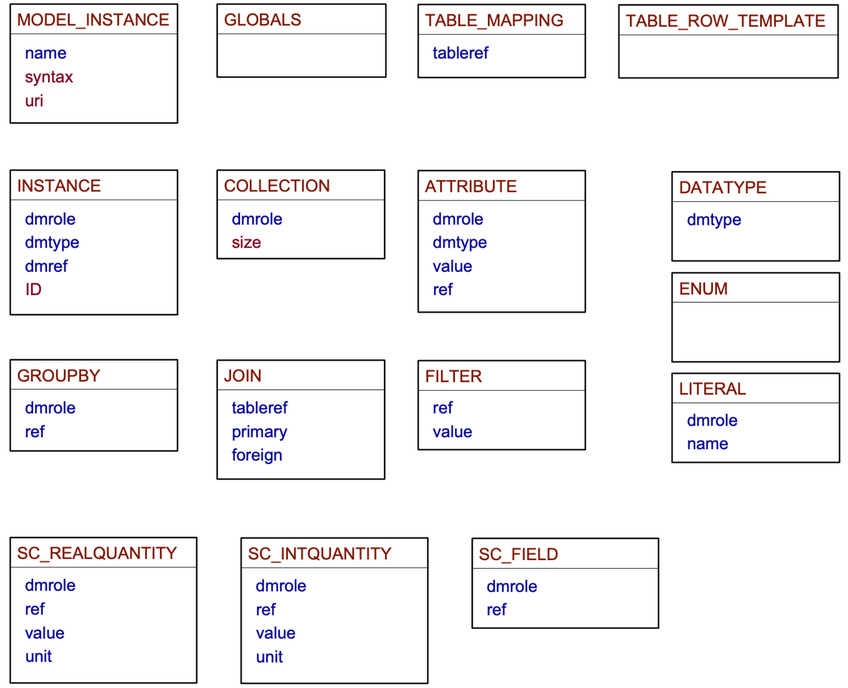
7.4 VODML Attribute Summary



26 elements + 6 unique attributes = 32 total

ModelInstanceInVOT Syntax

Attribute Summary

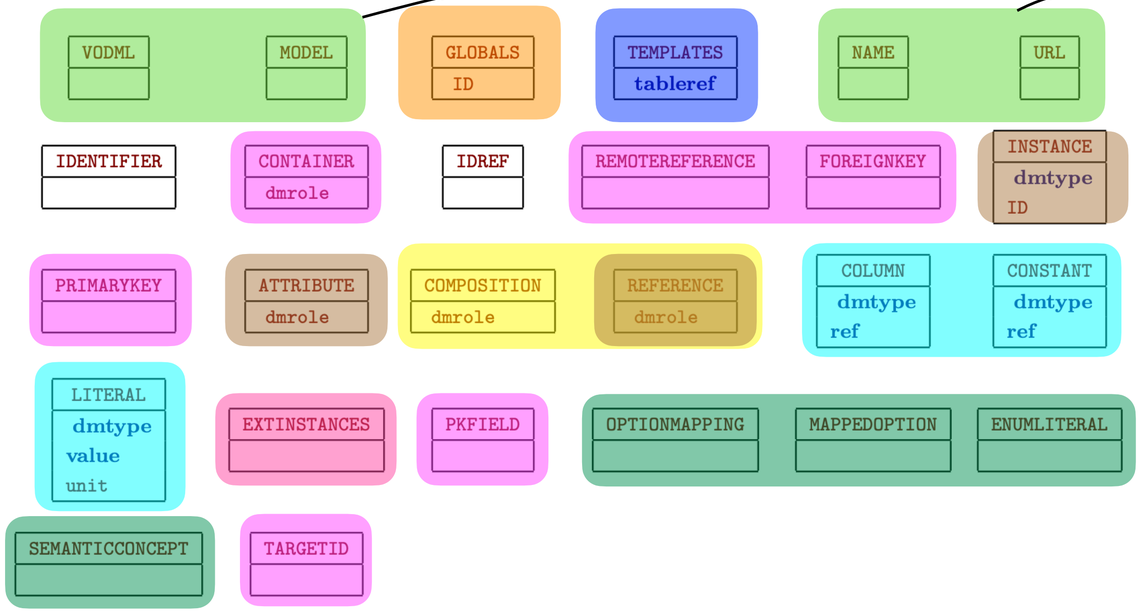


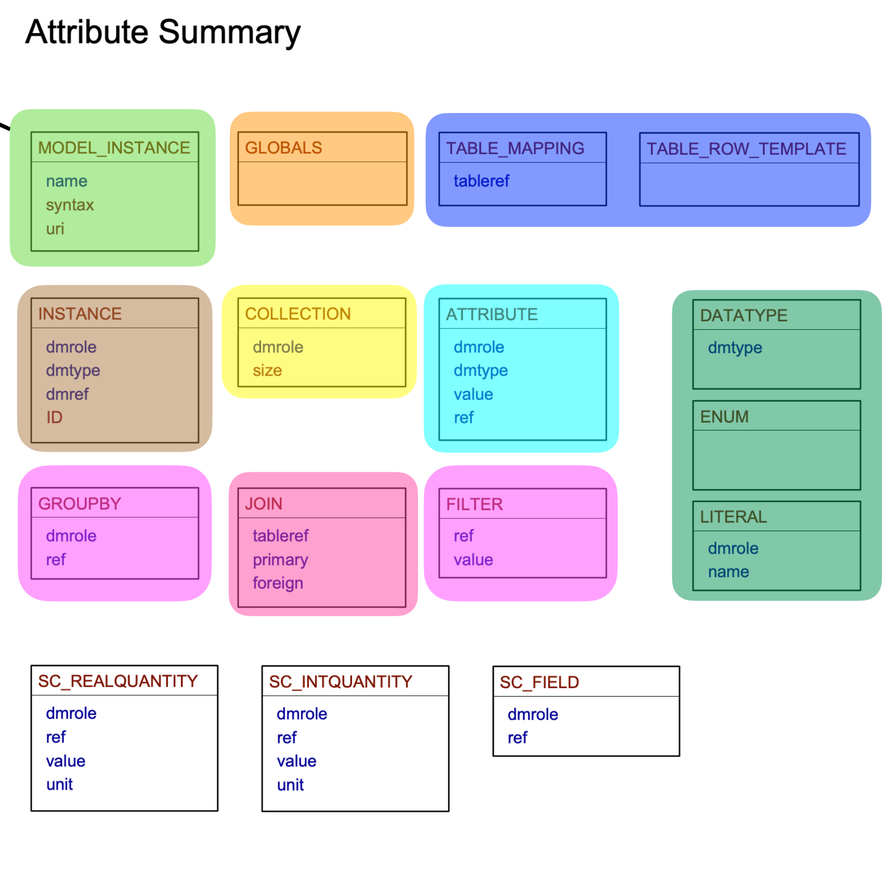
16 elements + 14 unique attributes = 30 total

VO-DML Mapping Syntax

ModelInstanceInVOT Syntax

7.4 VODML Attribute Summary





* Significant overlap in concepts (no surprise)
* Some consolidation and shuffling of elements to attributes
* ORM elements most different

Annotation

Comments on syntax

- ModelInstanceInVOT
  - Reduced to 1 Model_Instance: cannot annotate ‘root’ instances from multiple models. Example: mango:Source and cube:SparseCube
  - Combines role with type: annotation changes when instance becomes a child of another instance (slightly).
  - ORM elements [FILTER|GROUPBY] allow only 1 key: GAIA multi-band case potentially has srcid + filter
  - TABLE_MAPPING: ties annotation too directly to VOTable structure
  - SC_*Quantity: elements unnecessary

Annotation

Comments on syntax

- VO-DML Mapping
  - Usage of ORM elements was a challenge: syntax provides the components which can be combined/used in different ways.
    - I leaned heavily on test examples to figure out what to do
  - Distinction of [COLUMN|CONSTANT|LITERAL] not strictly necessary
  - [REFERENCE|COMPOSITION|ATTRIBUTE] can be useful at I/O level, but distinction possibly more relevant to power users than general usage.

Annotation

Comments on usage

- Had to add IDs to VOTable TABLE/FIELD elements in many cases. They were not contained in the native serialization.
  - Used LITERAL instead of CONSTANT for PARAMs, which duplicates the value, but does not modify native serialization.
- Added TABLE to hold primary key values in order to make compact annotation
- Annotating complex PARAMs: time-series case, ssa_location element
  - <PARAM name="position" datatype="double" arraysize="2" ref="_icrs" value="123.222 -10.000" />
  - Neither syntax allows for annotating into array elements (tag which is longitude and which latitude), nor (I think) permit annotating the PARAM to represent the complex type coords:SphericalPoint as a whole, leaving the client to interpret the content.
  - instead, this must be annotated with LITERALs, duplicating the individual values in the annotation.
- Data issues - not annotation related
  - Trouble handling (RA,DEC) in sexagesimal format: bug in QTable
  - pmRA units ’s/yr’ failed conversion to ‘deg’; ’s’ is Time Unit.: bug in data (s/b “arcsec/yr”)

Moving Forward

Hybrid Solution?

Completely UNTESTED and UNREVIEWED proposal
To facilitate discussion

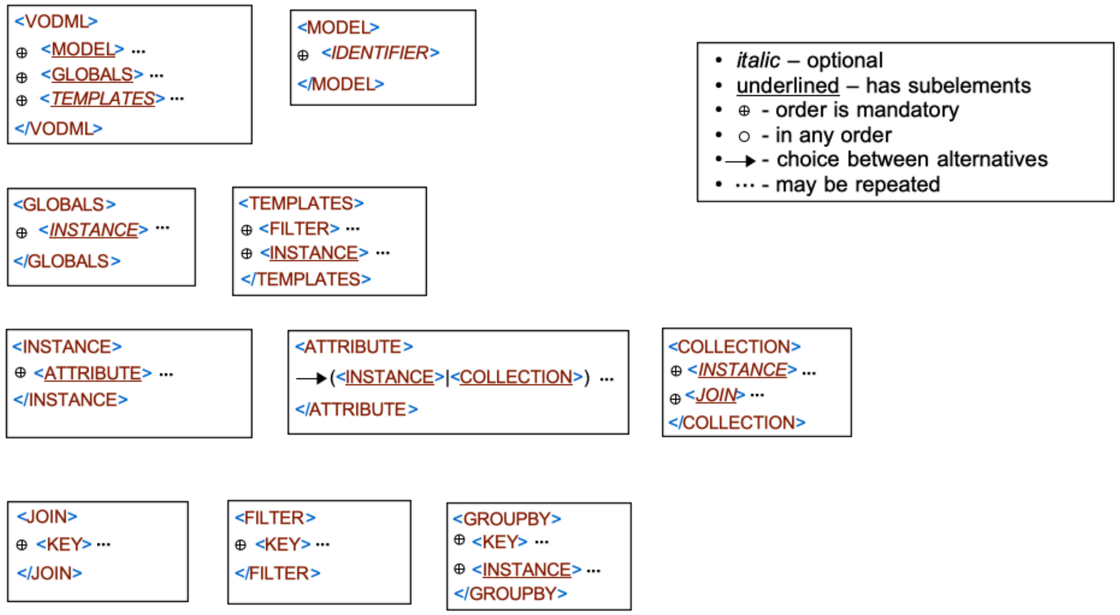
Annotation

Mapping syntax migration possibilities

- Retain multiple model declaration
- More usage of attributes vs elements
- ATTRIBUTE: provides dmrole, remains separate from INSTANCE
- Consolidate COLUMN, CONSTANT, LITERAL, REFERENCE into INSTANCE
- ORM as ‘actions’ FILTER, JOIN, GROUPBY; distributed where applied
- Consolidate KEY handling to KEY element; allow >1
- Postpone OPTIONMAPPING and subelements

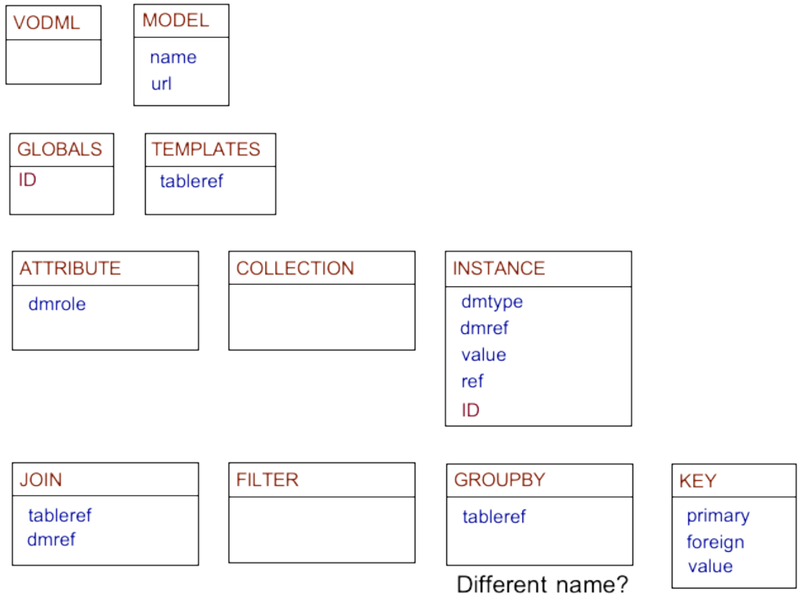
Merged Mapping Syntax

Element Hierarchy



- *italic* – optional
- underlined – has subelements
- ⊕ - order is mandatory
- ○ - in any order
- ⟶ - choice between alternatives
- ··· - may be repeated

Attribute Summary



11 elements + 11 unique attributes = 22 total