

The Data Point Model methodology and its evolution to the DPM Refit



Eurofiling Online Conferences 2022

High level overview on the new DPM Refit metamodel

Wednesday, 22 June 2022, 10:10 – 10:40

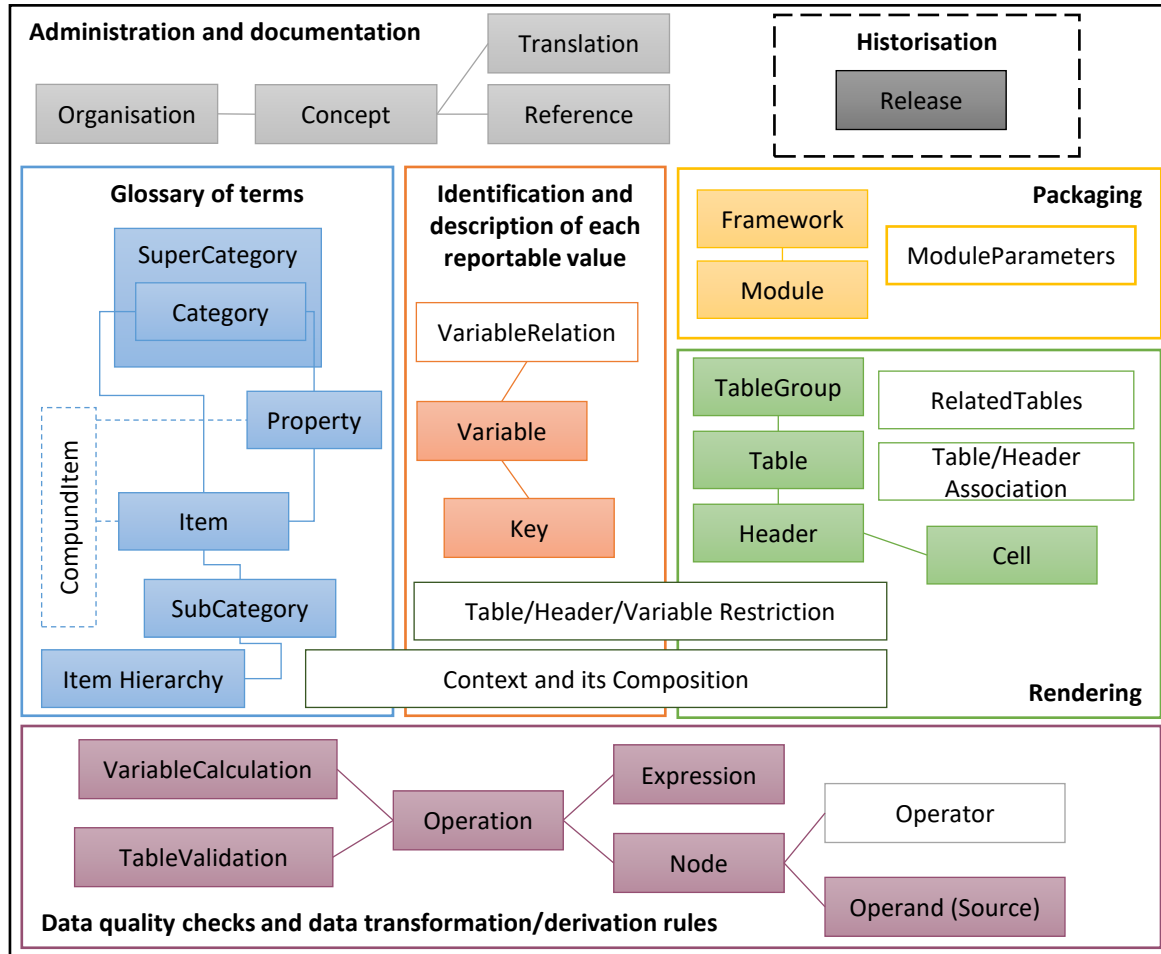
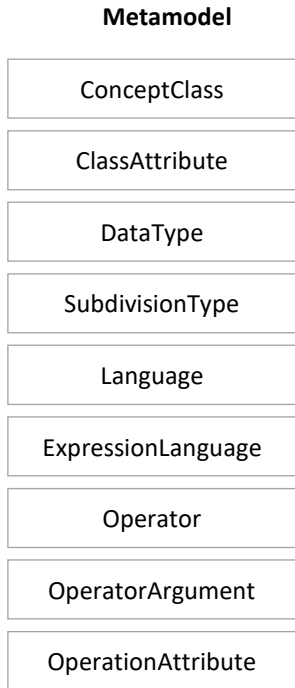
Bartosz Ochocki, consultant, EIOPA

Outline:

1. DPM meta-model overview
2. Meta-model information, model administration and documentation
3. Historization: releases, deactivations, application dates
4. Glossary: basic and advanced modelling concepts
5. Representation of information requirements
 - Role and definition of variables
 - Rendering: tables, headers, cells
 - Packaging: frameworks, modules, table groups
6. Operations: data quality checks and data derivation rules

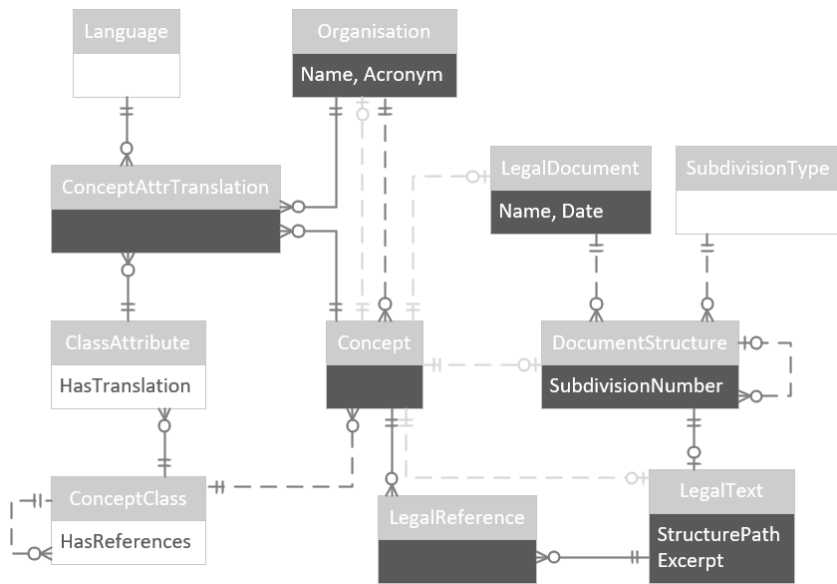


DPM Refit meta-model overview



Meta-model information, model administration and documentation

Class	Parent Class
Category	
Subcategory	
Property	
Item	
Framework	
Module	Framework
ModuleVersion	Module
TableGroup	Framework
Table	
TableVersion	Table
Header	Table
HeaderVersion	Header
Cell	Table
TableAssociation	Table
Variable	
VariableVersion	Variable
Key	
Context	
Restriction	
Operation	
OperationVersion	Operation
OperationNode	OperationVersion
LegalDocument	
DocumentStructure	LegalDocument
Organisation	
Release	

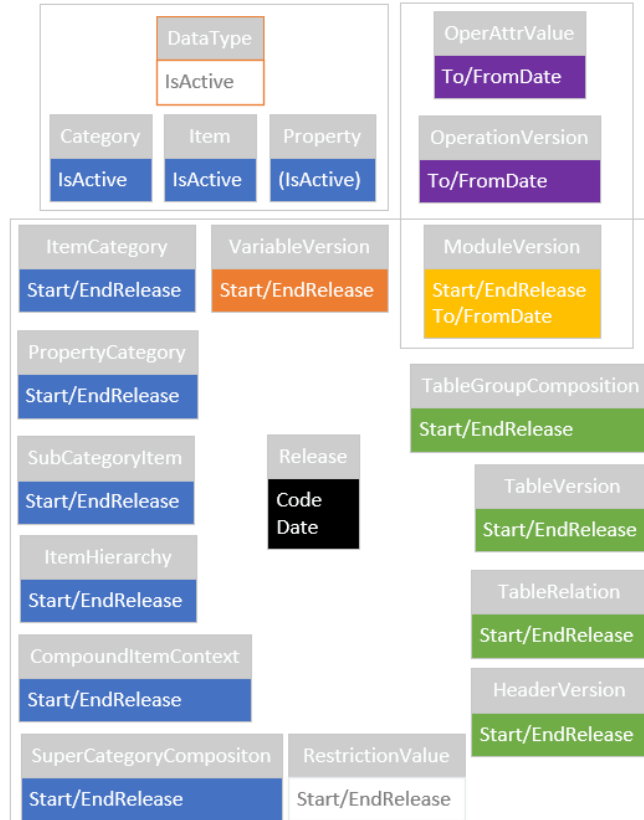
- Concept** enables indication of ownership (**Organisation**) and provides means for translations or provision of references to legal acts, regulations, standards, etc
 

The diagram illustrates a complex class hierarchy and relationships. Key classes include:

 - Language** and **Organisation** (with attributes Name, Acronym).
 - Concept** (with attributes Name, Date) is associated with **Language** and **Organisation**.
 - ConceptAttrTranslation** is associated with **Concept** and **Organisation**.
 - LegalDocument** (with attributes Name, Date) is associated with **Concept**.
 - SubdivisionType** is associated with **Concept**.
 - DocumentStructure** (with attribute SubdivisionNumber) is associated with **Concept** and **LegalDocument**.
 - LegalText** (with attributes StructurePath, Excerpt) is associated with **DocumentStructure**.
 - LegalReference** is associated with **Concept** and **LegalText**.
 - ClassAttribute** (with attribute HasTranslation) is associated with **Concept**.
 - ConceptClass** (with attribute HasReferences) is associated with **Concept**.
- ConceptClass**
 - identifies inheritance of ownership (e.g. headers and cells inherit owner from table they are part of, modules and table groups inherit owner from framework they belong to, etc)
 - indicates if certain class can have references
- ConceptAttribute** lists attributes of Classes (i.e. columns of entities) identifying which are translatable by means of **ConceptAttrTranslation** (note: reference to **Organisation** enables same language but different authors)
 - legal references indicate referred **LegalDocument**, their **Structure** and **Text** (all are concepts and can be ownerable and may be translatable)
- under consideration on physical level: global uniqueness of ID (GUID or derived from owner and class type and inheritance: may be helpful in merging different physical implementations of DPM models)

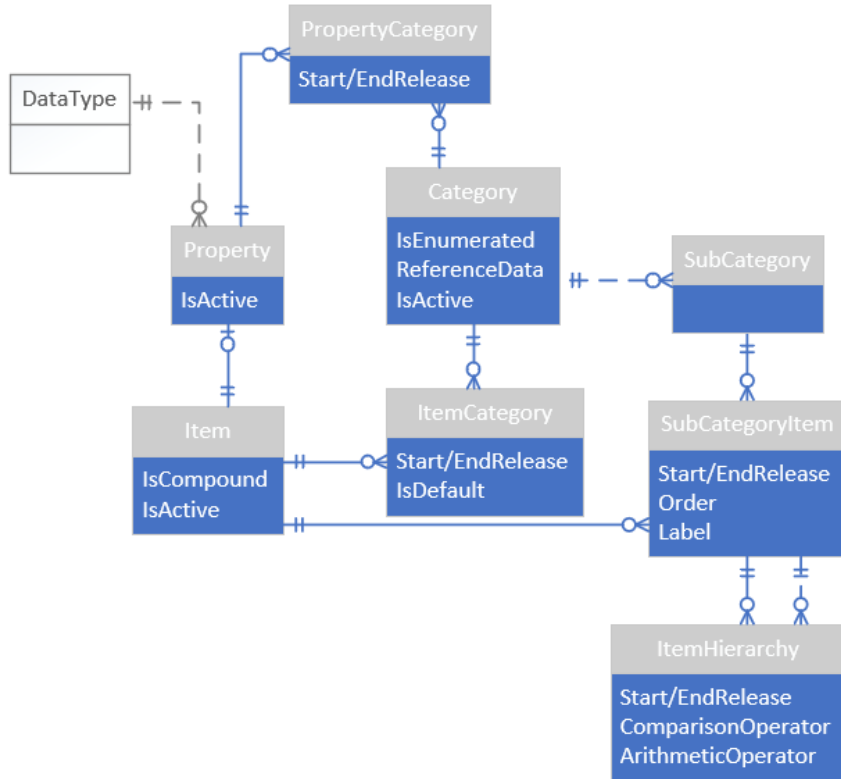
Historization: releases, deactivation and application dates

- Reference to a **Release** (i.e. publication)
 - Start Release** is mandatory, **End Release** is optional
 - on **objects** or **connections between objects** (to enable changing their composition)
 - indication of evolution of concepts** rather than defining constraints (e.g. decommissioning of use of concepts or values is imposed by data checks)
 - does not address model administration purposes** (e.g. creation/modification dates), workflow/development process stages (IWD, PWD, ...) neither it helps to handle indication of temporary metadata (work-in-progress for internal comments/feedback)



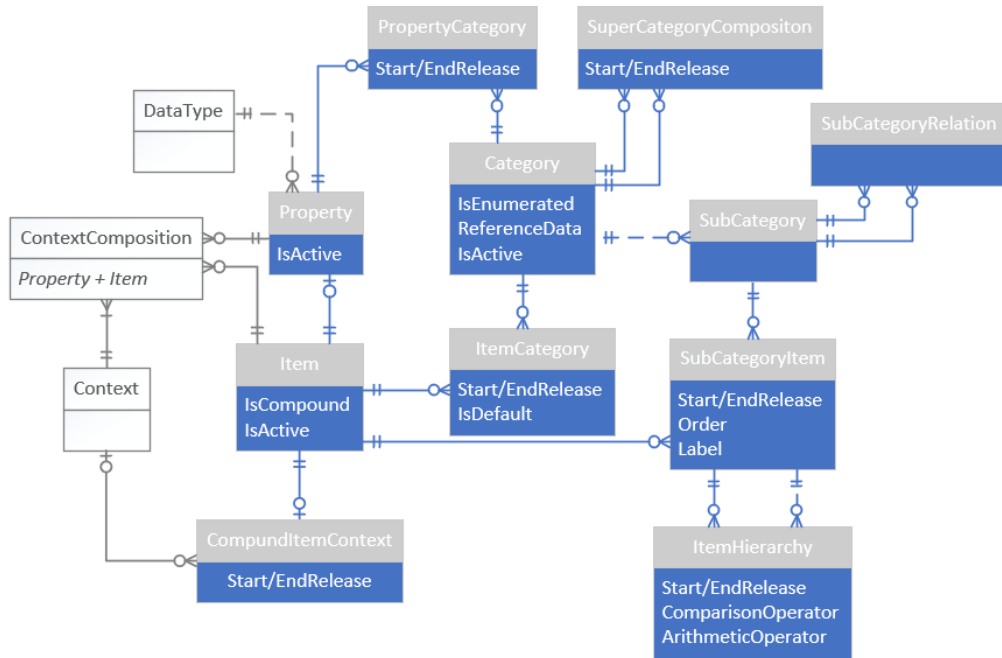
- Categories and Item (therefore also Properties that are counterparts of property Items) can be marked as **deactivated** (e.g. no longer to be used)
- Modules and Operations may have their own life cycle within a release start/end - **application dates** at some point in the future (e.g. module applicable starting from next year, enabling/disabling validations related to Brexit or planned changes in composition of currencies)

Glossary: basic modelling concepts



- Enumerated **Categories** group **Items** that share common nature/semantics
 - Item can change Category (helps fixing modelling flaws)
- **Subcategories** are lists (e.g. to be used as dropdowns), that can be arranged in **ItemHierarchies** (if needed, e.g. to reflect nesting or arithmetic relationships)
 - Subcategories (and ItemHierarchies) can change composition
- Not-enumerated **Categories** are no longer data type constraints – these data types are assigned to **Properties** instead
- **Properties:**
 - **quantitative** - identify what is measured (previously “metrics” – determine data type and time duration of observation (no longer “specific” set of Items from a selected Category))
 - **qualitative** - provide perspective/characteristic; in case of enumerated properties contextualise Category Items
 - Properties can change Category (helps fixing modelling flaws)
- Each property has its counterpart Item: enables arranging them in Subcategories to be used as dropdowns

Glossary: more complex modelling concepts

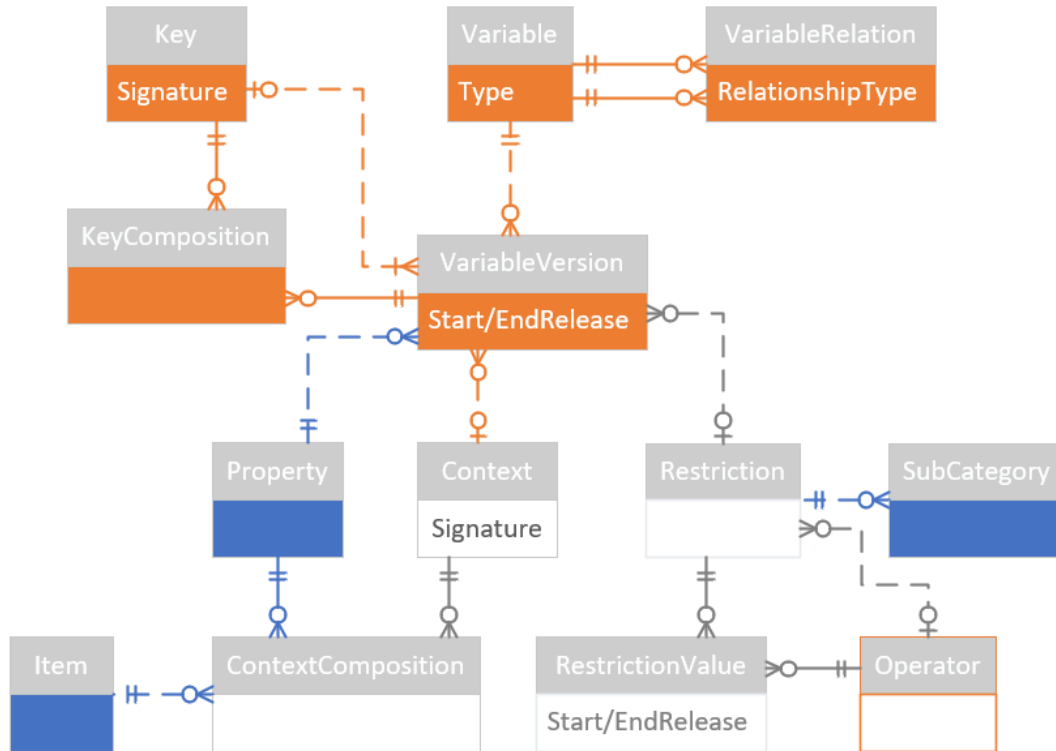


- **SuperCategory** is a union of **Categories**, e.g. frequently requested mix of breakdowns by “Countries” and “International organizations” (i.e. IMF, BIS, ECB, ... along countries for exposures/risk classification)
- **SubCategories** may be related to one another, e.g. a “full list of countries” can be indicated as a “master” for regional/topic oriented subsets (which may need to be updated once the “master” list is updated)
- **Compound Items** are composed of Property-Item pairs – simplified representation of complex terms

Instrument type	Treasury bills
	=
Instrument type	Debt securities
Issuer sector	General Government
Original maturity	< 1 year

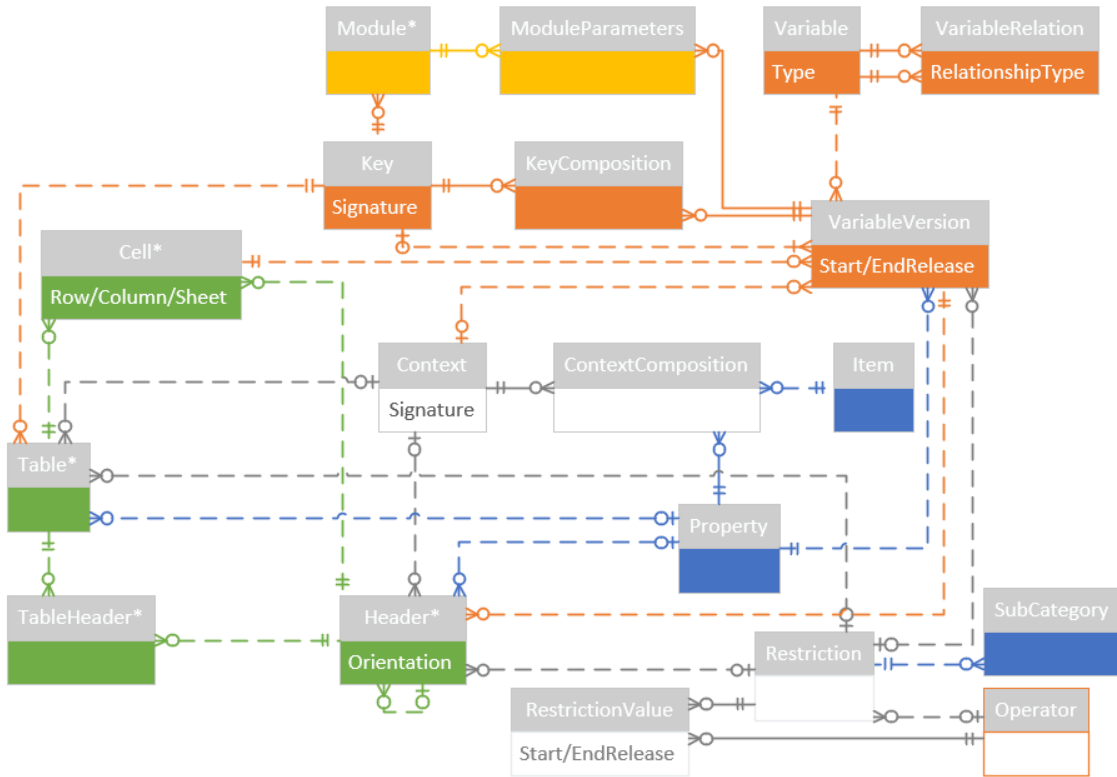
- Compound item composition can be versioned

Information requirements: role of variables



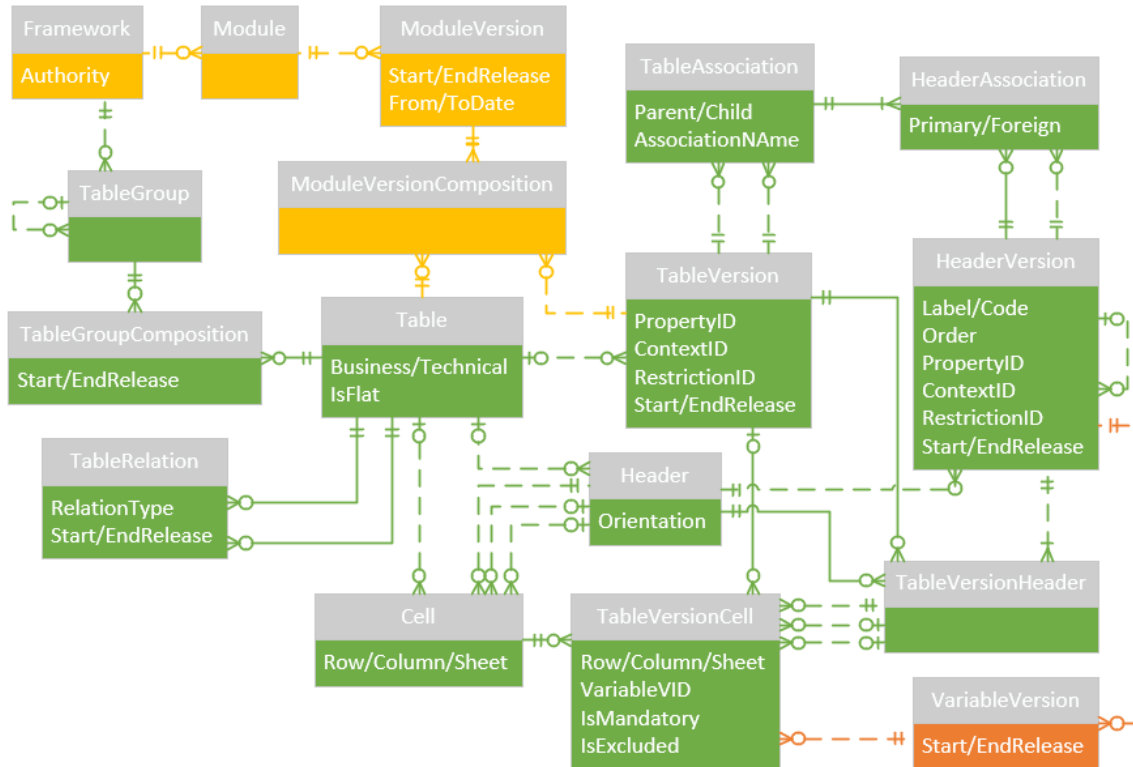
- Variable** represents each distinct reportable value (regardless of its occurrence in rendering) – quantitative or qualitative
 - types: **Fact** (monetary amount, text, enumeration, ...), **Key** (e.g. column identifying each a row in an open table) and **Attribute** (unit of measure, value precision/accuracy, a comment of a Fact or a Key)
 - relationships**: *Key-Fact*, *Fact-Attribute*, *Equivalent-Facts* (same meaning but modelled differently), ...
- VariableVersion**:
 - must refer to Property
 - may refer to Restriction or Context (composed of one or many Property-Item pairs)
 - composition may change between Releases (changes in modelling, fixing of bugs, etc)

Information requirements: definition (derivation) of variables



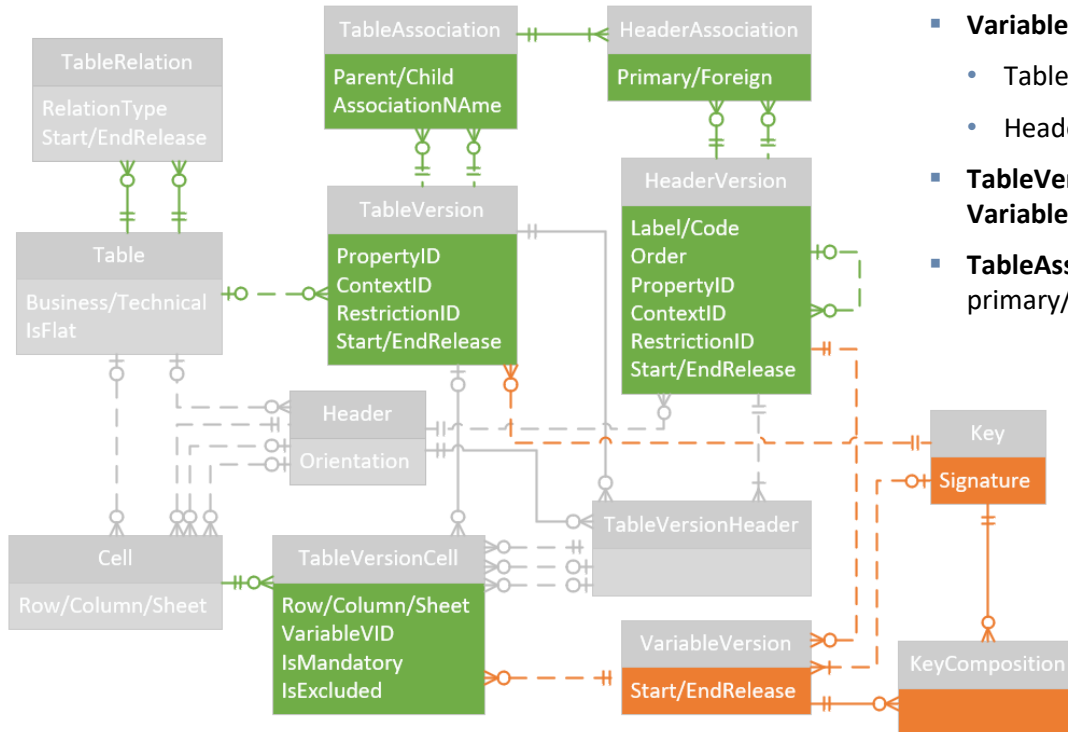
- Fact variables** can be derived from Table Cells: Table level glossary terms contribute to definition of Table Headers (where information can be inherited from upper levels) and is further propagated to cells on intersection of these headers
- Key variables** are linked from Headers which have no cells attached (this is to ensure open/semi-open sheets and columns are modelled in the same way as it is merely a rendering decision if these are displayed as sheets or columns)
 - KeyComposition** arranges Keys in sets to support rendering-less models
- ModuleParameters** apply globally for Module, for example Attribute Variables such as report currency and period/date

Rendering: overview



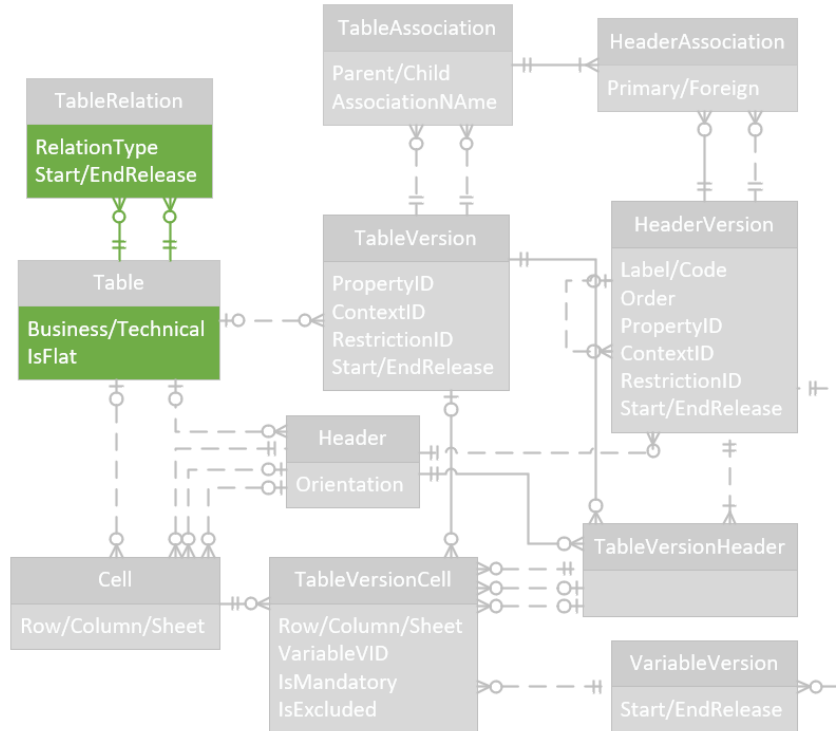
- **Tables** have **Headers** and **Cells**
 - **Cells** are **Fact variables**
 - **Headers** can be **Key variables**
- **Tables** can be **grouped**
- **Tables** can **relate** to one another (variants, normalisation – PK/FK)
- **Tables** are referred from **Modules**

Rendering: relations to variables and tables association



- **VariableVersion** refers to and inherits glossary description from:
 - TableVersionCell (fact variable) or
 - HeaderVersion (key variable)
- **TableVersion** refers to **Key** that though **KeyComposition** identifies **Variables** that are keys in a given table
- **TableAssociation** and **HeaderAssociation** identifies primary/foreign key dependencies between tables

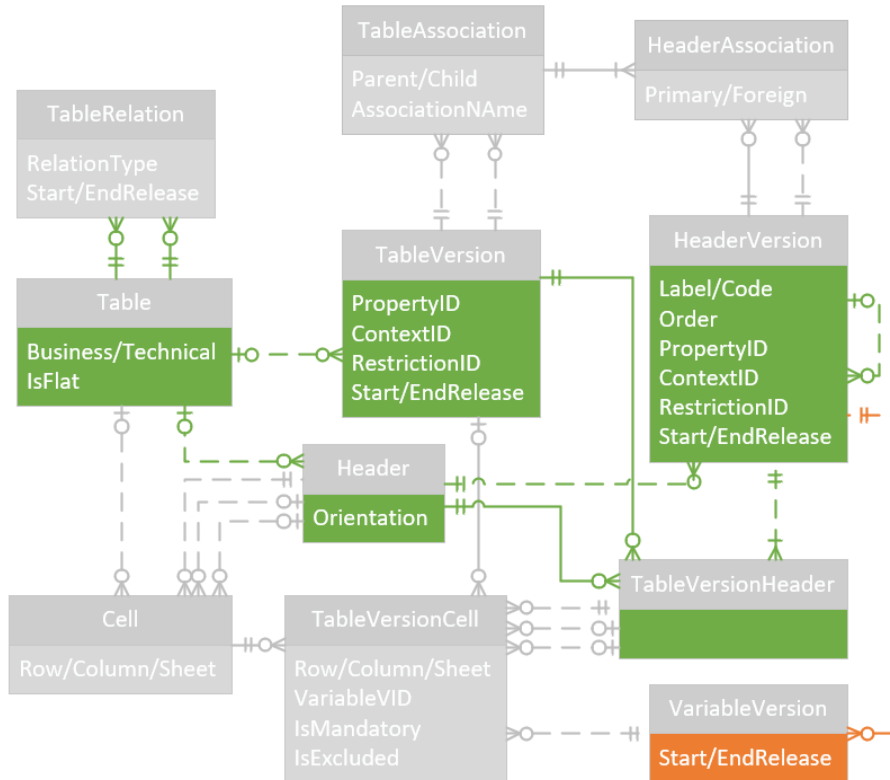
Rendering: relations between tables



- ### TableRelation:

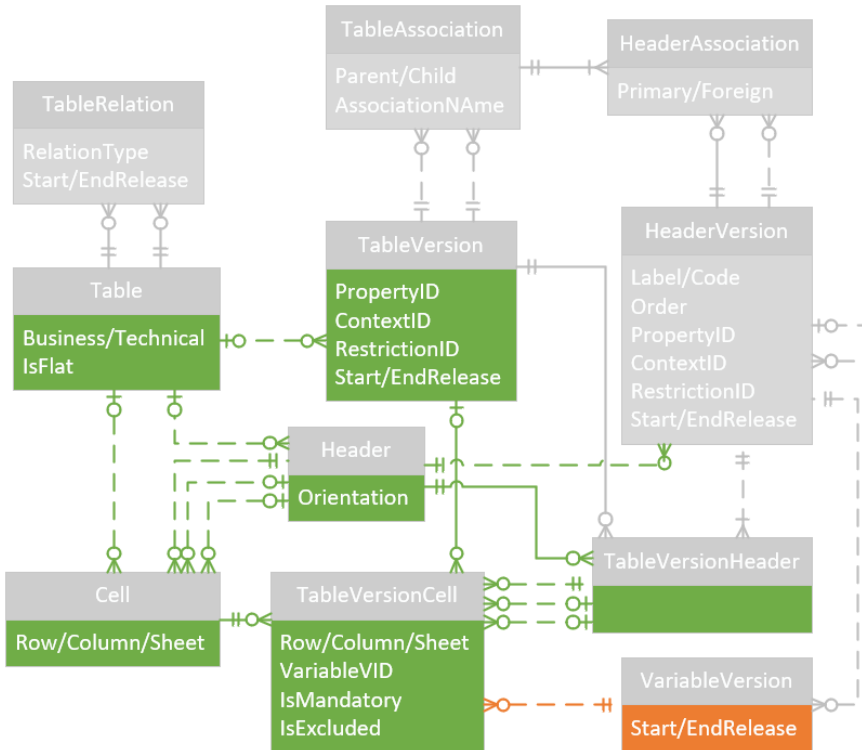
- “Business” Table (as originally defined by policy/in regulation) can result in many “Technical” Tables or vice-versa (normalization/denormalization for modelling purposes)
 - Variants of one table can be related (e.g. solo/group/branches balance sheets)

Rendering: headers



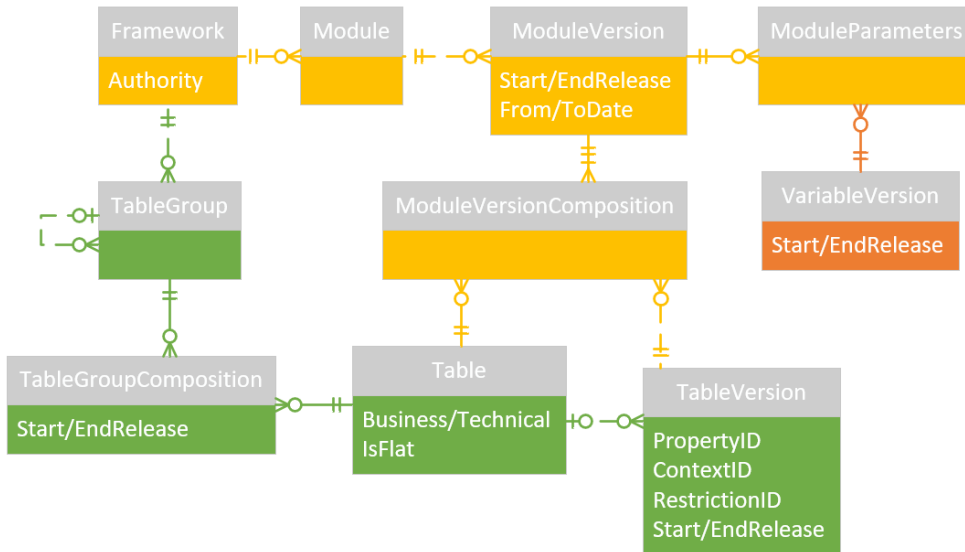
- **TableVersion** may link to Property, Context (Property-Item pairs) or Restriction applicable on table level (common to/shared by all headers/cells)
- **Headers orientation** can be Rows/Columns/Sheets - one set of related headers per disposition (note: typical open tables require only columns)
- separation of **TableVersionHeader** enables modification of headers structure without a need for duplicating objects for the entire table (all headers, cells, etc)
- **HeaderVersion:**
 - contribution to definition of a fact variable: column/row/sheet headers may point to Property and/or Context
 - definition of a key variable
 - points to a Property and may point to a Restriction
 - does not link/result in a Cell
 - links to VariableVersion

Rendering: cells



- **Cell** points to at least one leaf-level Header (e.g. a column in open rows table) but may also point to two or maximum three (one for each: Column, Row and Sheet)
- on **TableVersionCell** it can be indicated if a cell is mandatory or excluded (this information may change for a table between table versions/releases)
- **TableVersionCell** definition results from intersection of headers for this cell (including inheritance) and points to a **Fact Variable** unless combination of characteristics from this intersection is illogical (e.g. *“Equity instruments”* issued by *“Government”* with certain *“Maturity period”*)

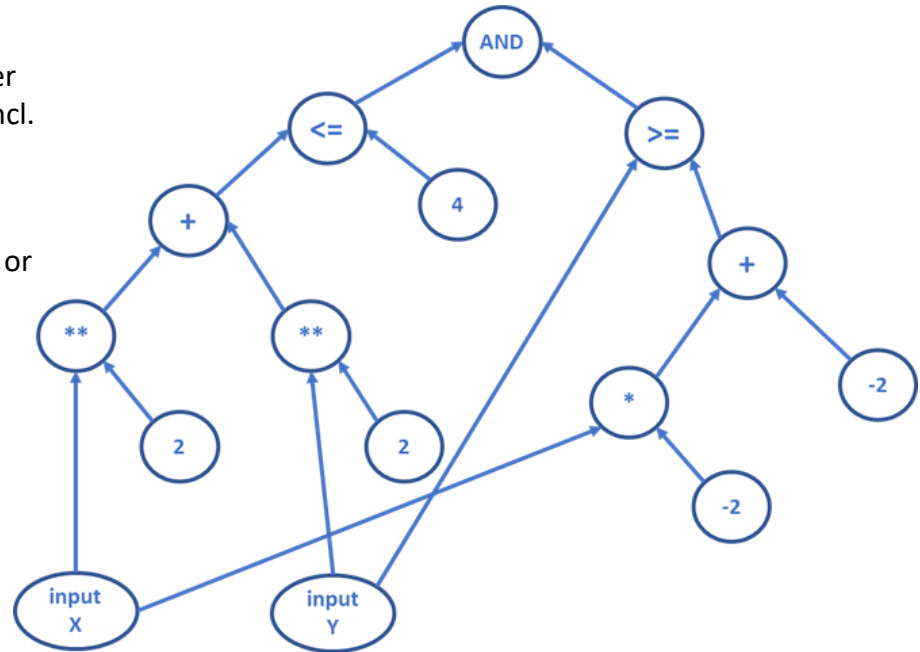
Packaging: frameworks, modules, table groups



- **Framework** – regulatory reporting requirements organized by subject/thematical area
- **Module** – set of information requirements to be reported together (in one report)
- **ModuleVersion**
 - **FromDate** and **ToDate** identify its application dates (may differ from model publication dates i.e. releases)
 - Gathers **Tables** (their Versions) that are required to be reported together
- **ModuleParameters** are Variables (e.g. Attribute) applied globally
- **TableGroups** can gather other **TableGroups** (can be nested with multiple levels) and via **TableGroupComposition** link to **Tables** to arrange them e.g. in families, by subject area, purpose, etc
- Composition of **TableGroups** may change between releases

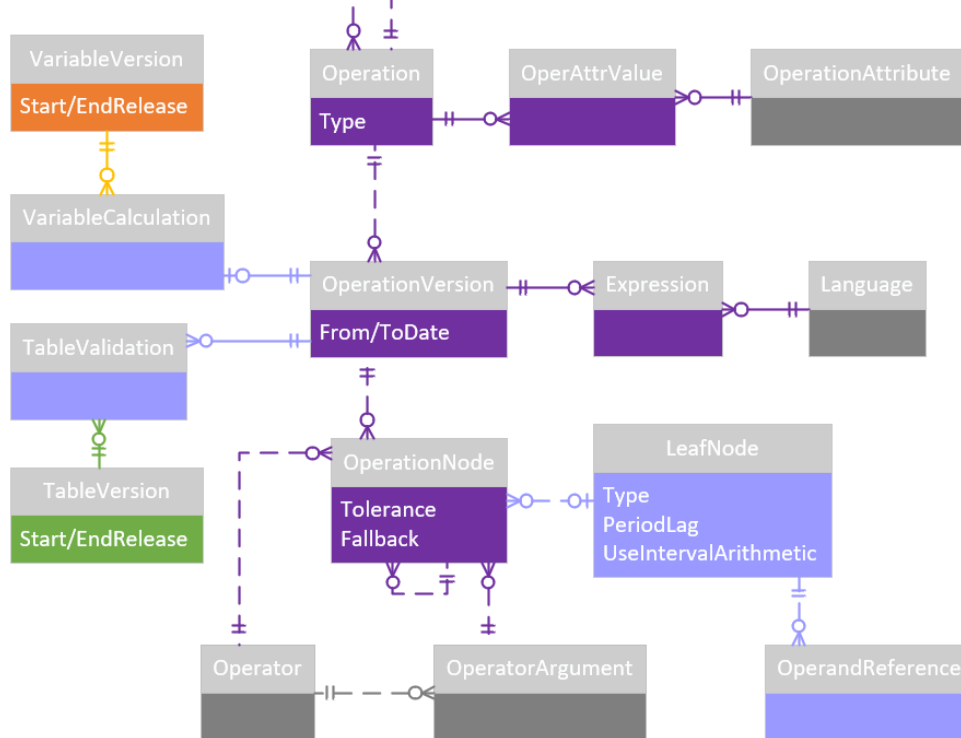
Operations: principles and assumptions

- reflected in metamodel using tree structure (AST) comprising of nodes - operators and operands, the latter referring to terms from glossary, rendering, variables (incl. external sources)
- syntax agnostic
- automatically populated: parsed from any/some syntax or defined and reviewed in graphical interface by users



Operations: representation in metamodel

Sequenced/nested operations, preconditions, severities



- **Operations** enable definition of **data quality checks** as well as **data derivation rules** (transformations and computation of Variables)
- Operations can be sequenced/nested and have dedicated **types**, e.g. precondition, severity, etc
- **OperationVersions** have application dates (are independent from Release)
- **Expression languages** can be natural language, programming language or various syntaxes
- **OperationNode**: one of defined **Operators** or an **Operand** (referring any artefact of a model, e.g. Variable, Table Header, Cell, Item, Property, Data Type, ...)
 - Note: Variables may indicate external data sources (e.g. master/reference data)

Time for your questions!

THANK YOU