

Neue OMG-Standards: DMN 1.6, B-FEEL & SDMN

Industriestandards zur Automatisierung
von Prozessen und Entscheidungen



About me



Falko Menge

Senior Principal Solution Architect

Open Standards Ambassador

- 18+ years experience in Process Orchestration
- 15+ years experience in Open Standards
 - Chair of DMN, SCE, SDMN, ~~KPMN~~
 - Voting Member of BPMN, ~~EMMN~~ & DD
 - Member of BPMN Model Interchange Working Group (MIWG)
 - Chair of BPMN MIWG Test Automation Subgroup
 - Member of DMN Technology Compatibility Kit (TCK) at WfMC
 - Member of Serverless Workflow Working Group at CNCF
 - OCEB 2.0 Certification Exam Author



github.com/falko



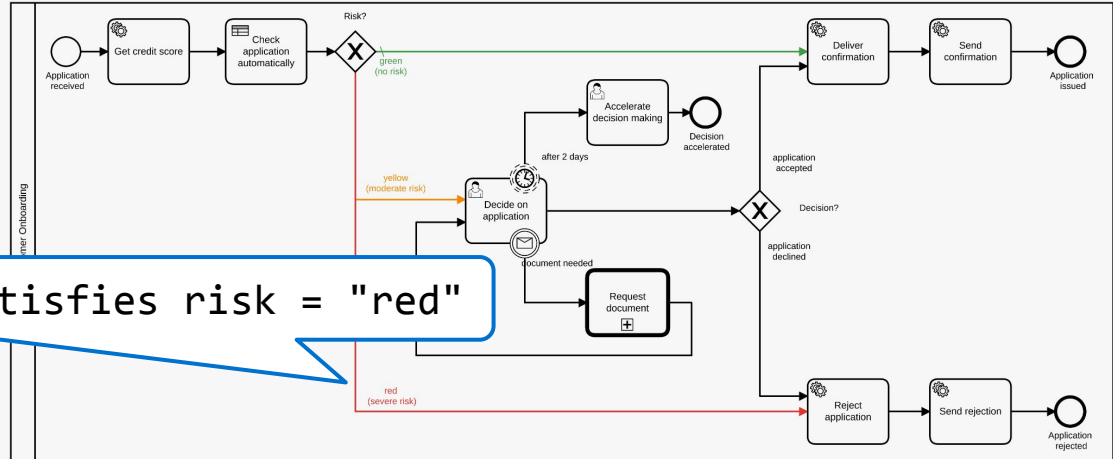
x.com/falko_menge



linkedin.com/in/falko-menge

BPMN 3.0?

- We are already working on it! Just not in the way you may expect
- Core standard deliberately kept stable
- Extensions: [colors](#), [internationalization](#) (i18n), [icons](#)
- New specifications grow ecosystem
 - **DMN** to implement Decision Tasks (Business Rule Tasks)
 - **FEEL** as default expression & type language
 - **SDMN** data models



= some risk in riskLevels satisfies risk = "red"

Design Goals of DMN & FEEL



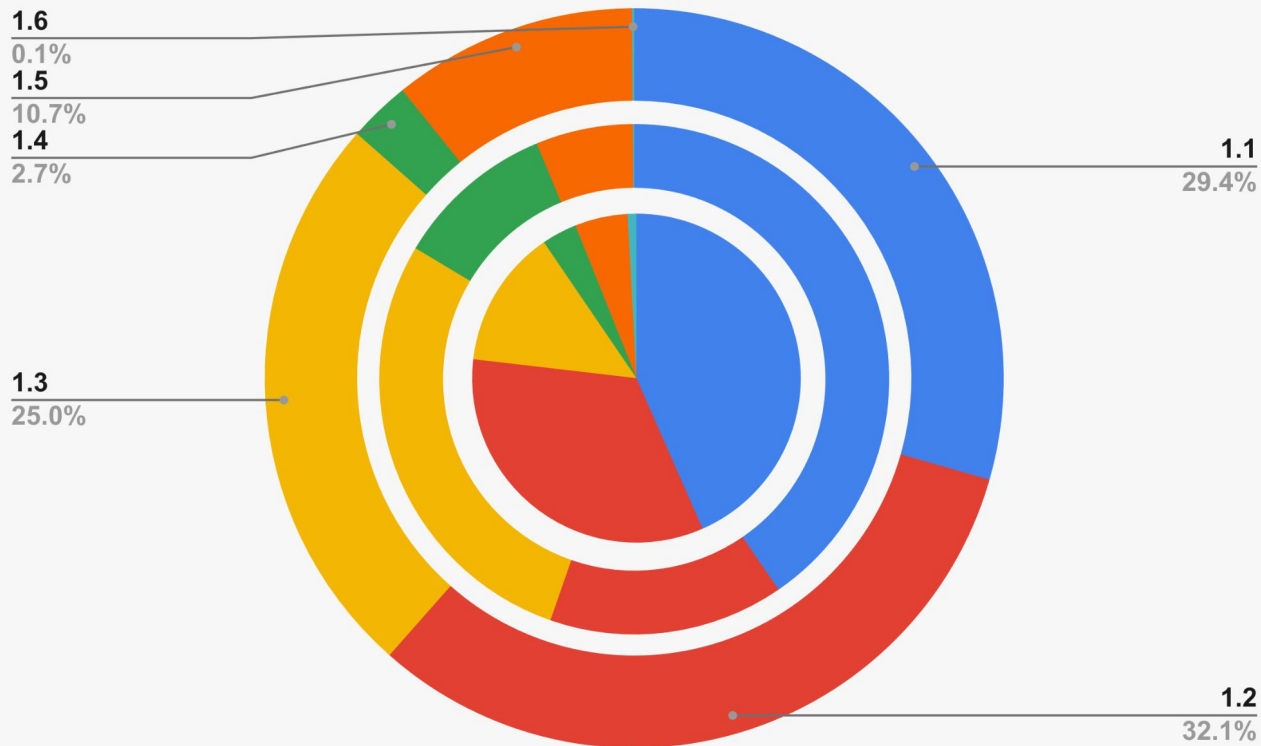
DMN: Decision Model and Notation

- Modeling language for business decisions from requirements to automation
- Common notation to **bridge gap between business analysts and developers**
- Member of the BPMN ecosystem of OMG standards
 - **most successful** standard after BPMN itself

FEEL: Friendly Enough Expression Language

- Fundamental for executable & interoperable decision logic
- Like JSONPath or jq but **standardized** incl. **function library**
- **Business-friendly**: names with spaces, counting from 1, singleton list/null coercion
 - Easier than Excel formulas
- **Used by all languages** of BPMN ecosystem, e.g. BPMN, DMN, and SDMN

Adoption of DMN Versions



Data as of 2024-09-11

Outer Ring: **GitHub** Search by Namespace URL

Middle Ring: **GitHub** Search by XML Schema file name

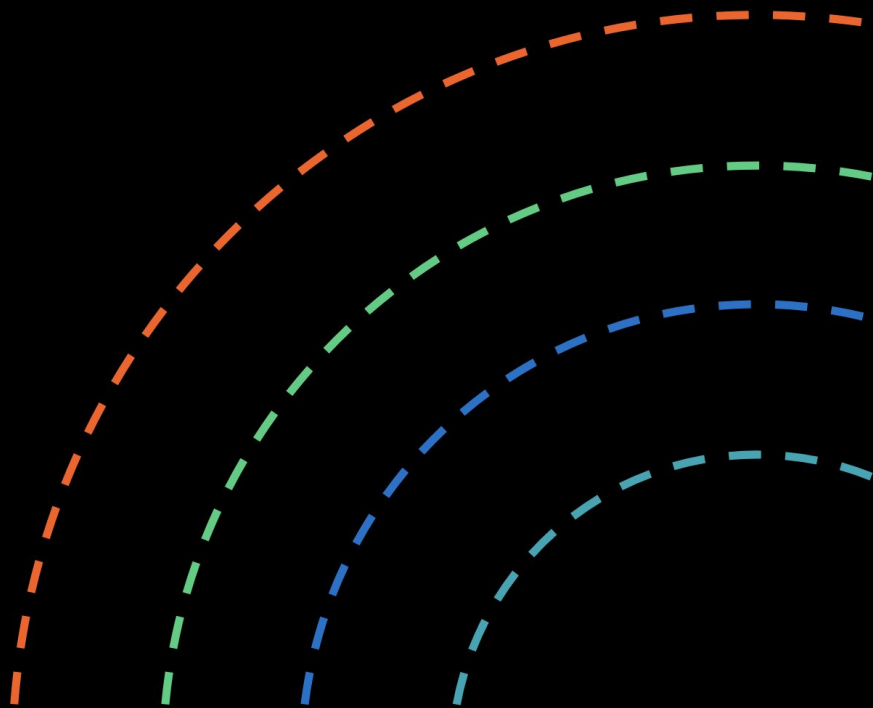
Inner Circle: **Google** Search by Namespace URL



FEEL 1.5

DMN 1.5

AB approval: 2023-03-20
Publication: August 2024



New Features in FEEL 1.5



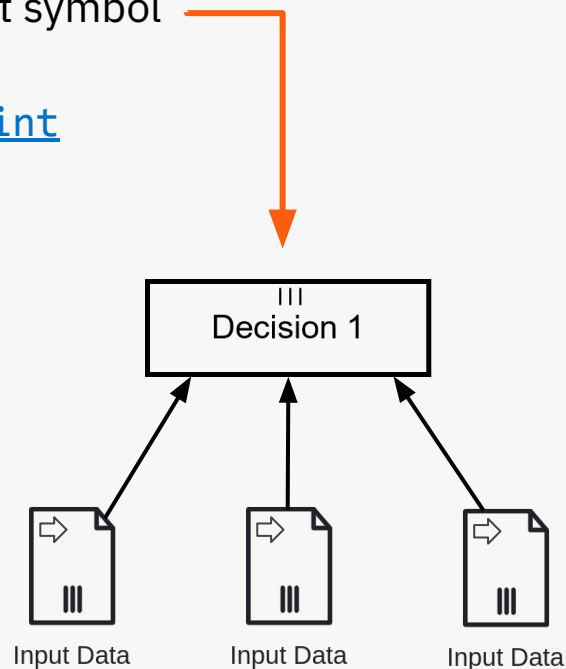
- Short-paced maintenance release
 - Addresses feedback from users, implementers, and the [DMN TCK](#)
- New FEEL functions
 - Numbers in scientific notation: `1.5E-23`
 - For-loop expression with date ranges: `@"2024-09-18"..@"2024-09-20"`
 - `list replace([2, 4, 7, 8], 3, 6) = [2, 4, 6, 8]`
 - `list replace ([2, 4, 7, 8],
function(item, newItem) item < newItem, 5) = [5, 5, 7, 8]`
 - String-to-Range conversion: `range("[18..21)")`
 - UnaryTests with equality operators: `"=" | "!="`

New Features in DMN 1.5

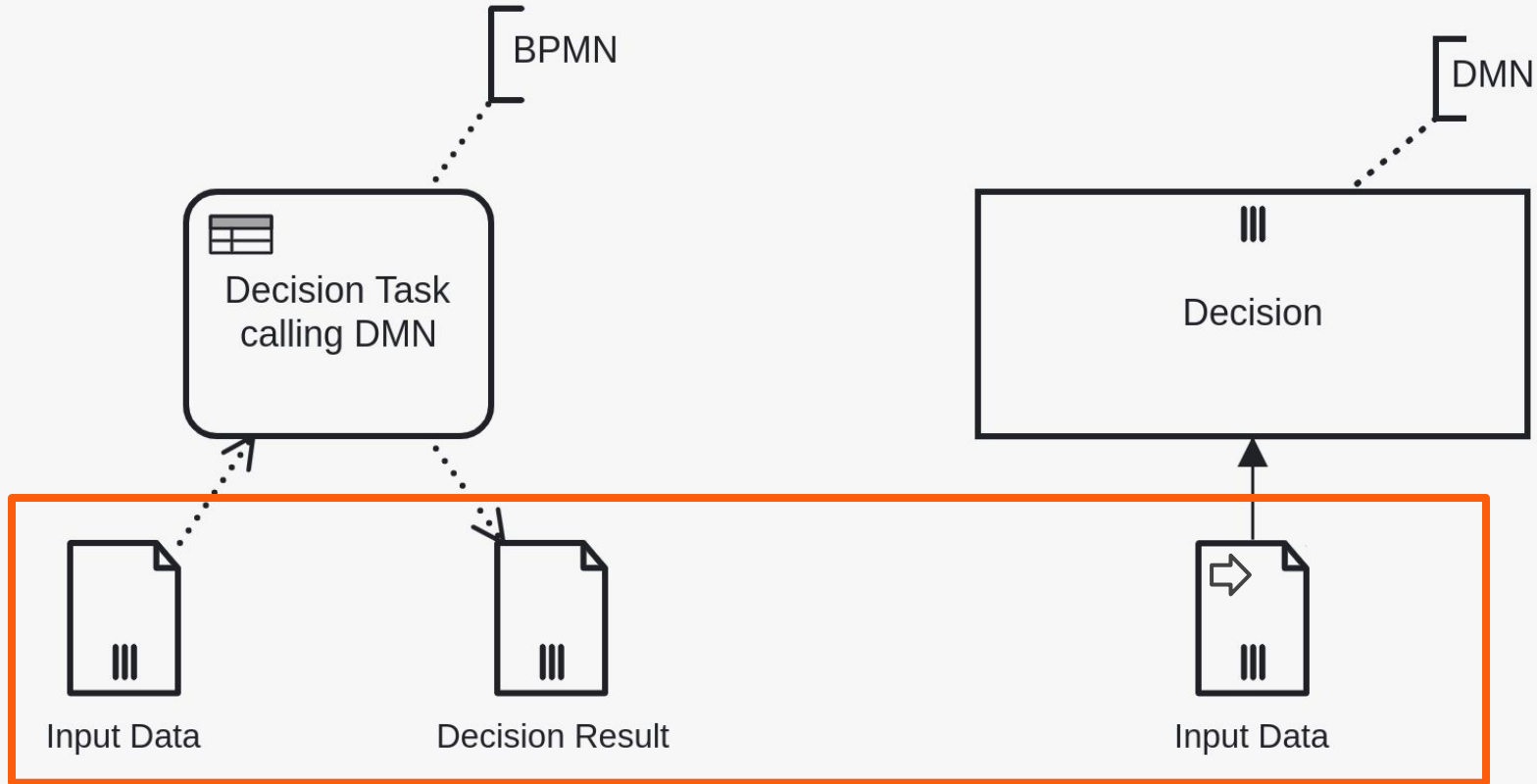
- Alternative notation for Input Data using BPMN Data Object symbol
- Replace `allowedValues` with more powerful [typeConstraint](#) based on `UnaryTests`, e.g. `age > 0`
- Import(s) into default namespace
- Improved examples
=> all fully backwards compatible

omg.org/spec/DMN

[List of all changes in Jira](#) (for OMG members)



Harmonization of Data Notation



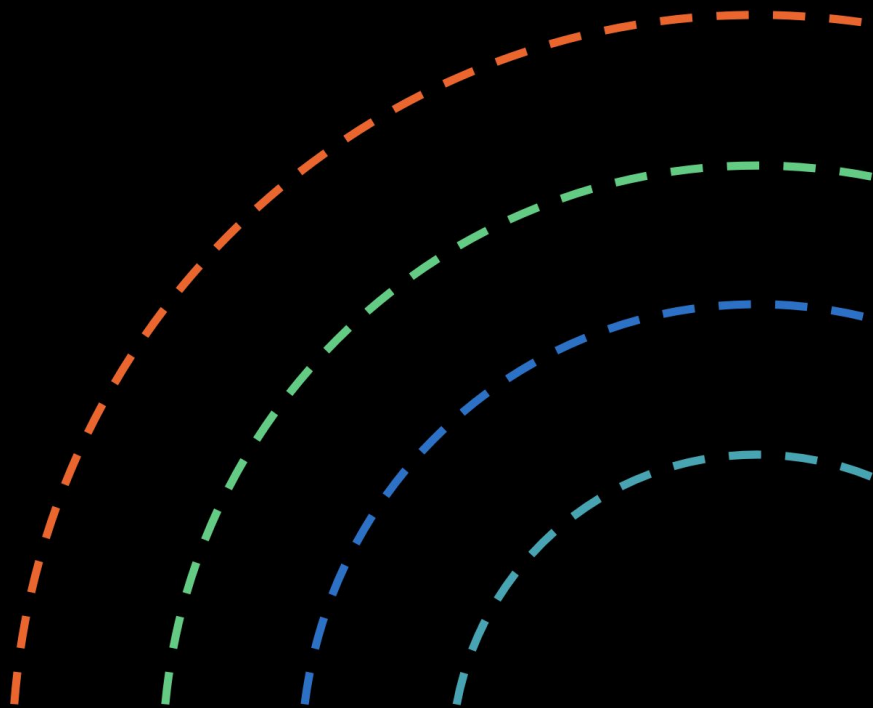
CAMUNDA

(B-)FEEL 1.6

DMN 1.6

AB vote: 2024-06-13

Publication: 2026



New Features in FEEL 1.6



- Short-paced maintenance release
 - Addresses feedback from users, implementers, and the [DMN TCK](#)
- FEEL to JSON mapping
- New FEEL functions
 - Single-parameter versions of all rounding functions & number function
 - `number(from: number)`
- Descendent operator ... (similar to .. in JSONPath or jq)
- Type conversion from decimal to integer
- Deprecated: IANA time zone use in time strings

What is the result of these expressions?

- "Today is " + today()
- "a" = 1
- sum(1, null, 3)

=> **null**

- In FEEL, **null** represents missing data **or** an **execution error**
- This **Design Flaw** caused frequent discussions over the years

B-FEEL (alternative FEEL dialect)



- Business-Friendly Enough Expression Language (B-FEEL)
- Alters the **semantics** to be friendlier and more intuitive toward **non-IT users**
 - **Same grammar** as “traditional” FEEL => short addition chapter in the spec
 - **null** represents missing data **only**
 - Not backward-compatible
 - => additional dialect that does not require a DMN 2.0
- When an error occurs, B-FEEL operators and built-in functions return **non-null values of the expected return type**
 - Errors MUST still be reported on a separate channel
- Vision: B-FEEL **de-facto standard** & “traditional” FEEL **deprecated**
 - Design future models for graceful behavior and use Strict Mode where needed

B-FEEL Examples

Expression	FEEL	B-FEEL
"a" = 1	null	false
"a" < 1	null	false
"a" <= null	null	false
null >= 1	null	false
not("a")	null	false
true and "x"	null	false
false or "x"	null	false
"a" in [1..100]	null	false
null between 1 and 100	null	false
"a" != 1	null	true

Null checks are not business friendly



Expression that don't produce **null** any more:

- `"Today is " + today() => "Today is 2024-09-18"`
- `"a" = 1 => false`
- `sum(1, null, 3) => 4`

Workarounds that are no longer needed:

- `if totalIncome = null then false else totalIncome > 0
=> totalIncome > 0`
- `if incomeItems.value[item!=null] = [] then 0
else sum(incomeItems.value[item!=null])
=> sum(incomeItems.value)`

New Features in DMN 1.6



- **Strict Mode** and clarified **error handling & reporting**
- Multiple imports clarified
- Integration with **Open Neural Network Exchange (ONNX)** as entry point for AI & ML
- Makes Java, PMML & ONNX optional
- New property '**value**' on 'date', 'date and time', 'time', 'years and months duration', 'days and time duration'

- Improved examples
 - all fully backwards compatible
 - Correct the typeRef for BKM's

[List of all changes in Jira](#) (for OMG members)

[...] while its basic functionality is little changed, DMN is a “living language,” continually made more powerful and easier to use as new use cases arise. - Bruce Silver

Plans for DMN 1.7



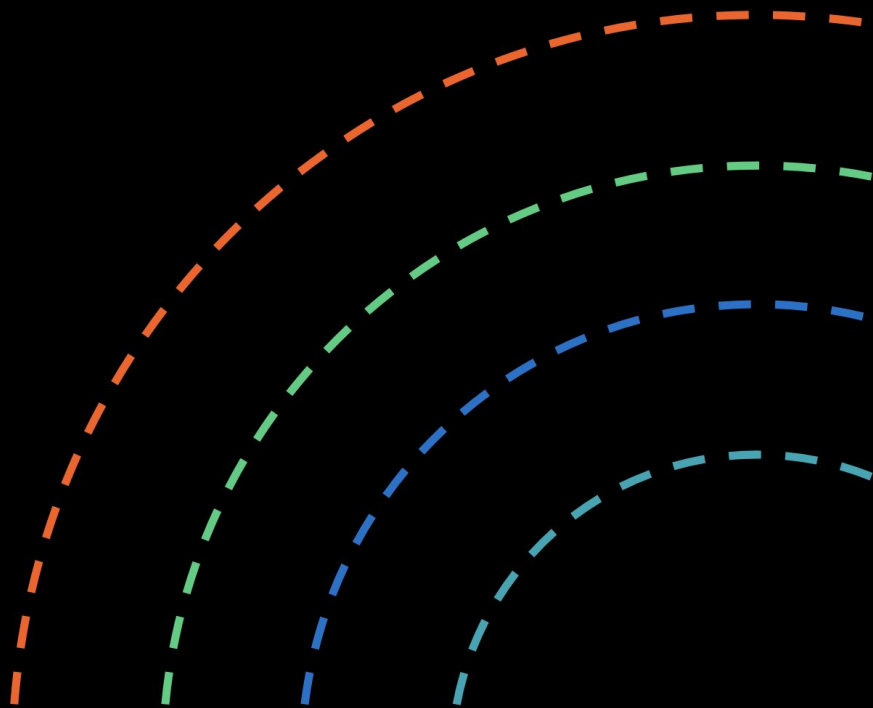
- Timeline: ~~March~~ December 2025 approval by OMG Architecture Board (AB)
- Deprecate traditional FEEL in favor of B-FEEL
- Retrieve Input Data from external sources, e.g. via REST call
- Import of interchangeable FEEL functions (incl. into current namespace)
- New XML versioning scheme based on xsi:schemaLocation
 - => allows permanent XML namespace URI
 - => less market fragmentation
 - => easier referencing among specifications
- Adopt [Specification Common Elements \(SCE\) 1.0](#)

CAMUNDA

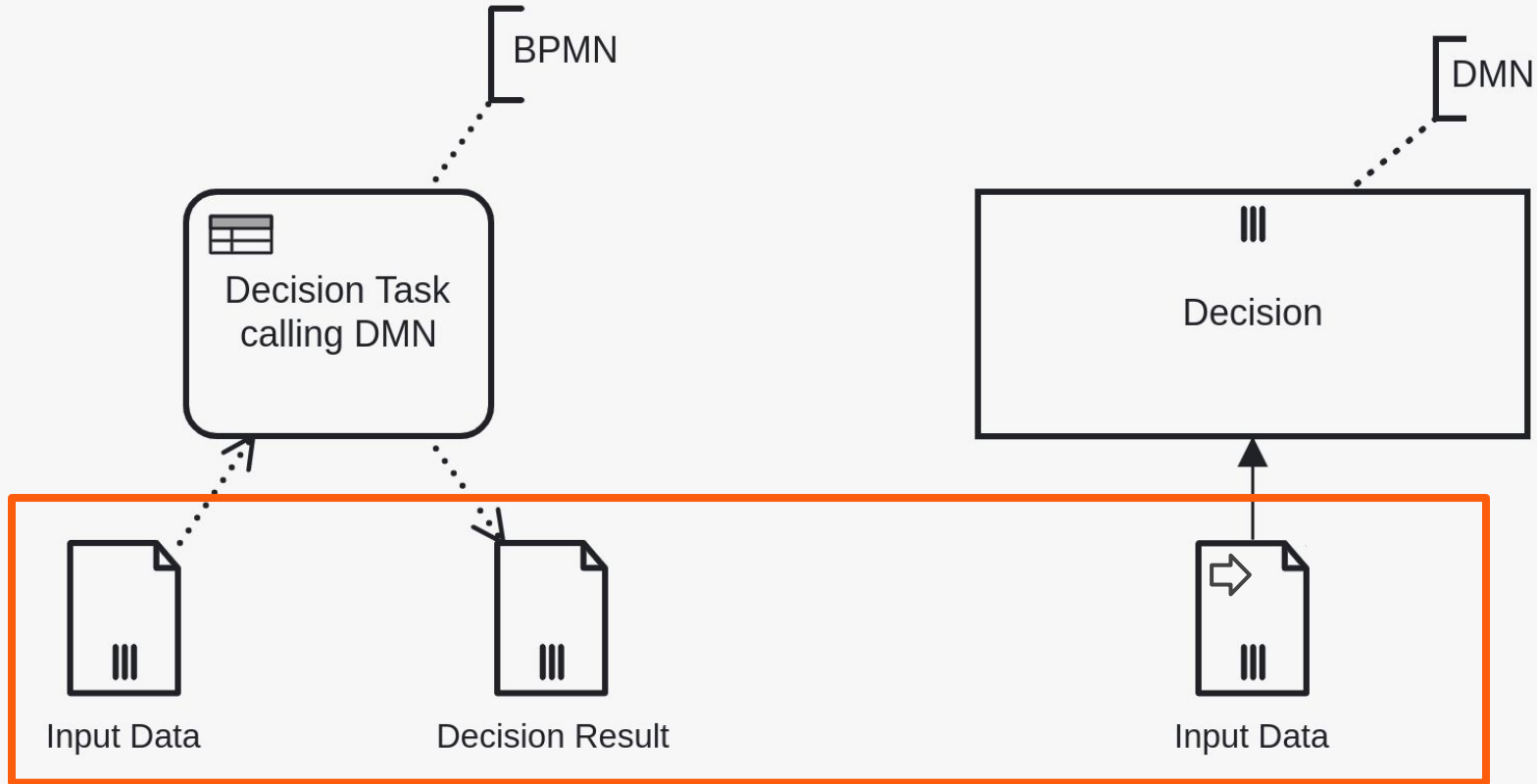
Shared Data Model and Notation (SDMN)

AB vote: 2024-03-17

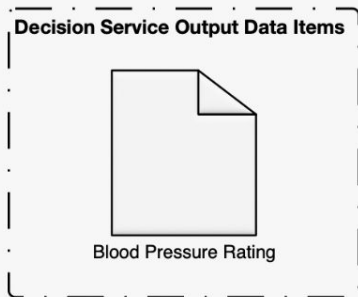
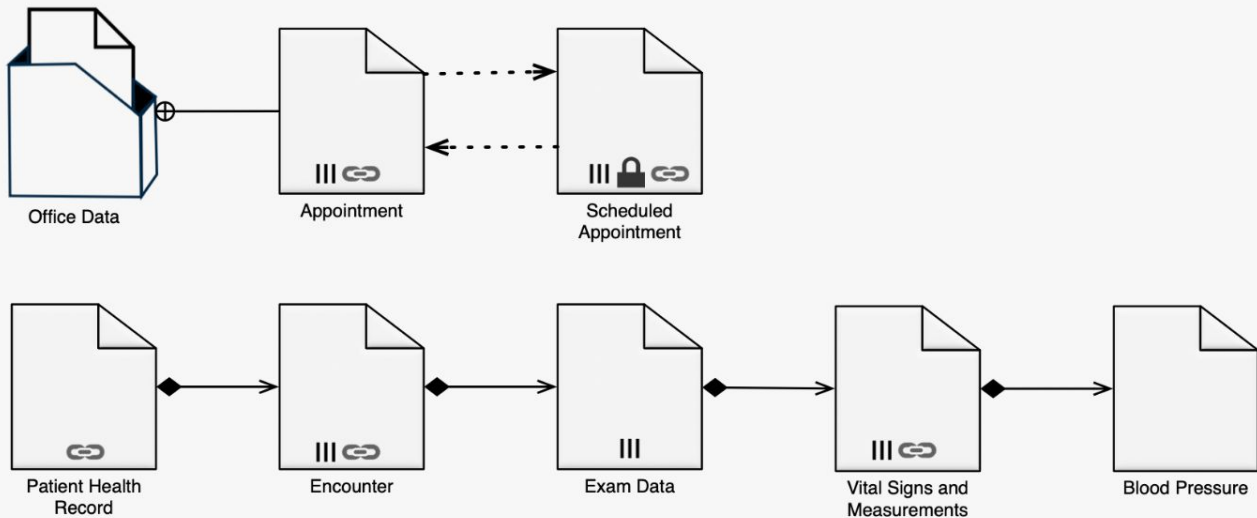
Publication: 2026?



SDMN: Shared Data Model and Notation

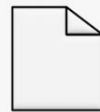


SDMN: Data Item Notation

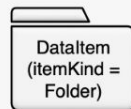


Legend

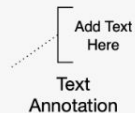
Shapes



Data Item



Group



Markers



Collection



Semantic Reference

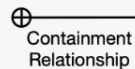


Locked



Hidden Relationships

Lines



Containment Relationship



Composition Relationship



Reference Relationship

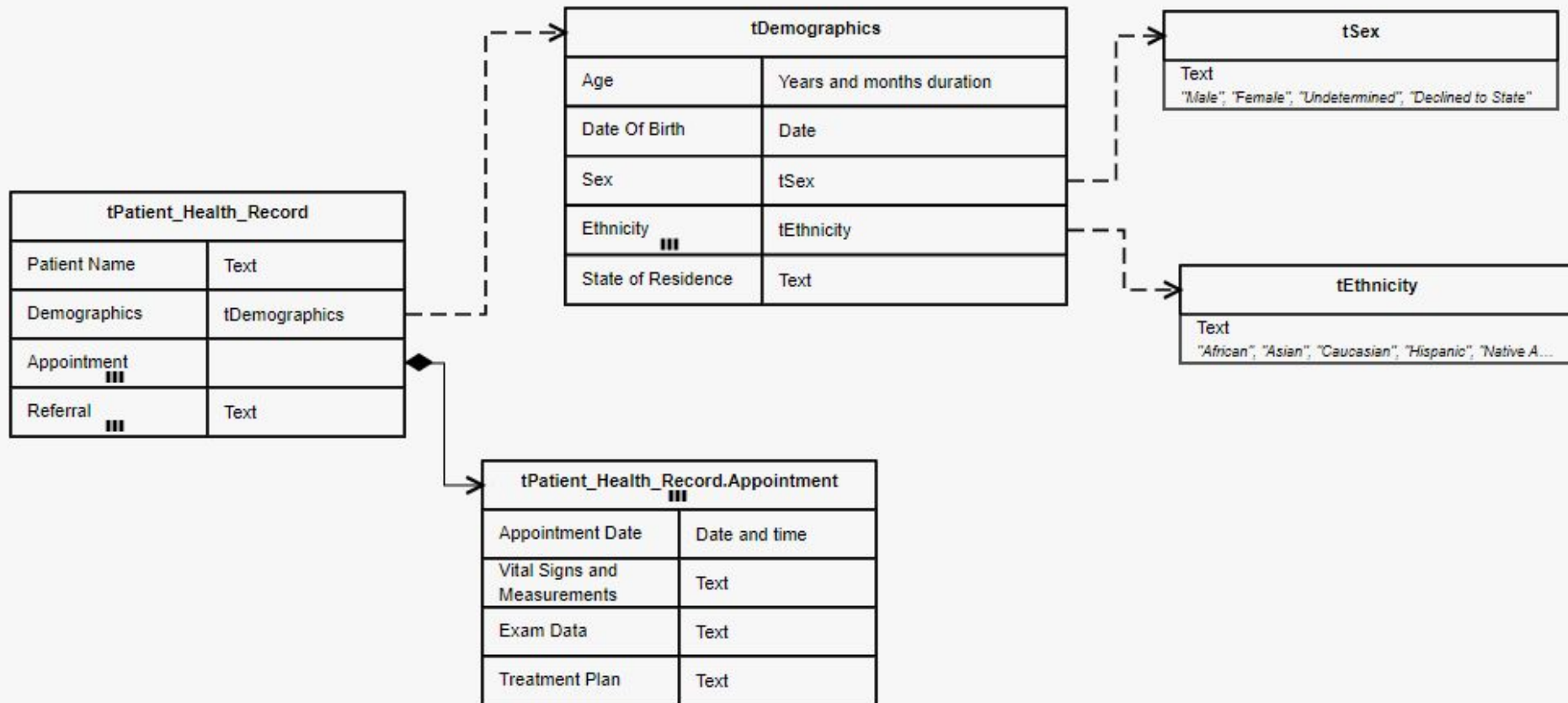


Data Association



Association

SDMN: Item Definition Diagram



Use cases that SDMN addresses



- **Design data model upfront** using SDMN Item Definition Notation, i.e. before any BPMN, DMN, forms, or integrations are designed
=> then use it for:
 - Code completion for FEEL expressions
 - Validation
 - “Templating”, i.e. generate Inputs and Outputs for Call Activities and Tasks
 - Form generation

Import SDMN into BPMN or DMN

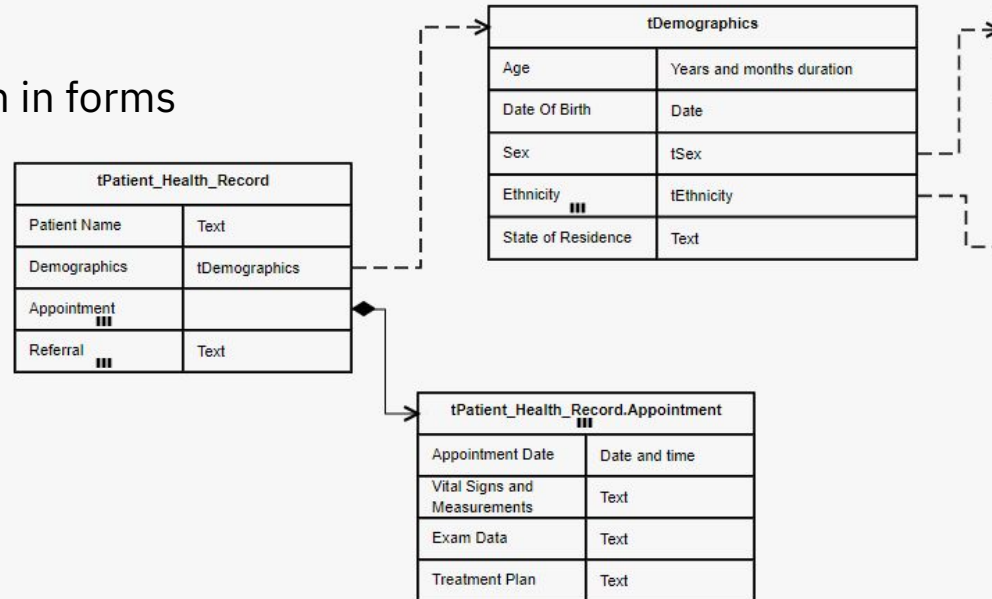
- **Visualization** of the Data Outline extracted from BPMN, DMN, forms, and connectors
 - Generate SDMN diagram out of extracted implicit schema

Import DMN with item definitions into SDMN

Advantages of the SDMN Notation



- Business-friendly, i.e. not for computer scientists
- Designed to be compatible with BPMN & DMN through reuse
- Visual graph (not just table or tree list)
- Similar to DMN's Boxed Expression notation
- A bit like UML but simpler
- Abstracts away from UI representation in forms



omg.org/spec/SDMN

Key takeaways



DMN 1.6

“living language”
more powerful
and easier to use



B-FEEL

Ends abuse of **null**
Errors **MUST** still be reported



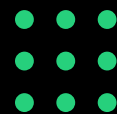
SDMN

Data models shared among DMN
models and with BPMN
processes

How to get involved



- Report issues and propose solutions:
omg.org/report_issue.htm
- Create an account and join Revision Task Force (RTF):
omg.org/login/request
- Become a trial OMG member:
omg.org/memberservices
- Join DMN TCK
github.com/dmn-tck



Thank You



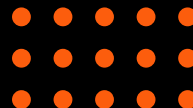
falko.menge@camunda.com



linkedin.com/in/falko-menge



x.com/falko_menge



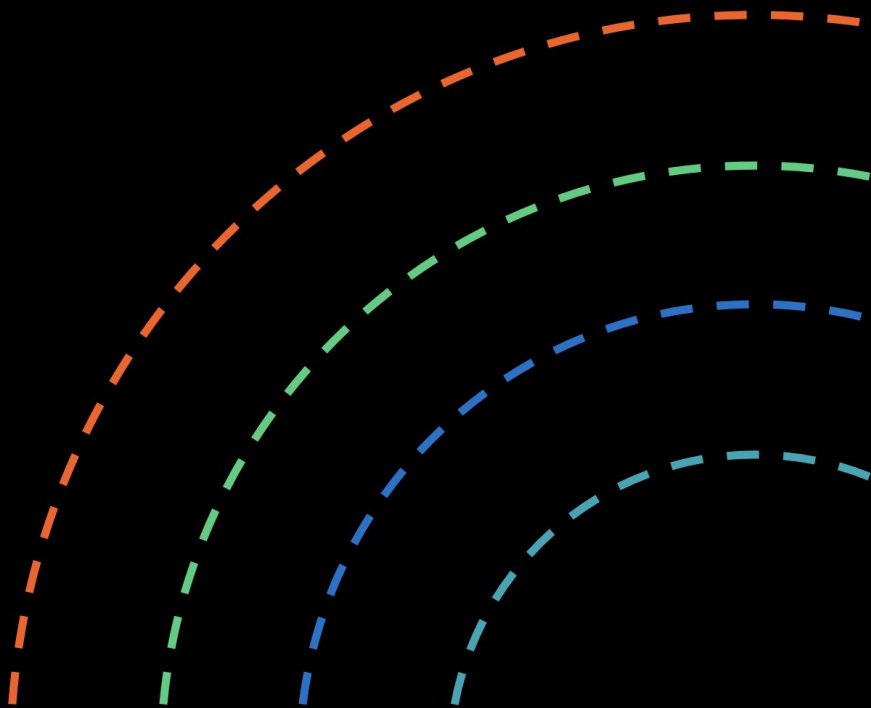
Bonus Material

CAMUNDA

Specification Common Elements (SCE)

AB vote: 2024-06-10

Publication: 2026?



Specification Common Elements (SCE)



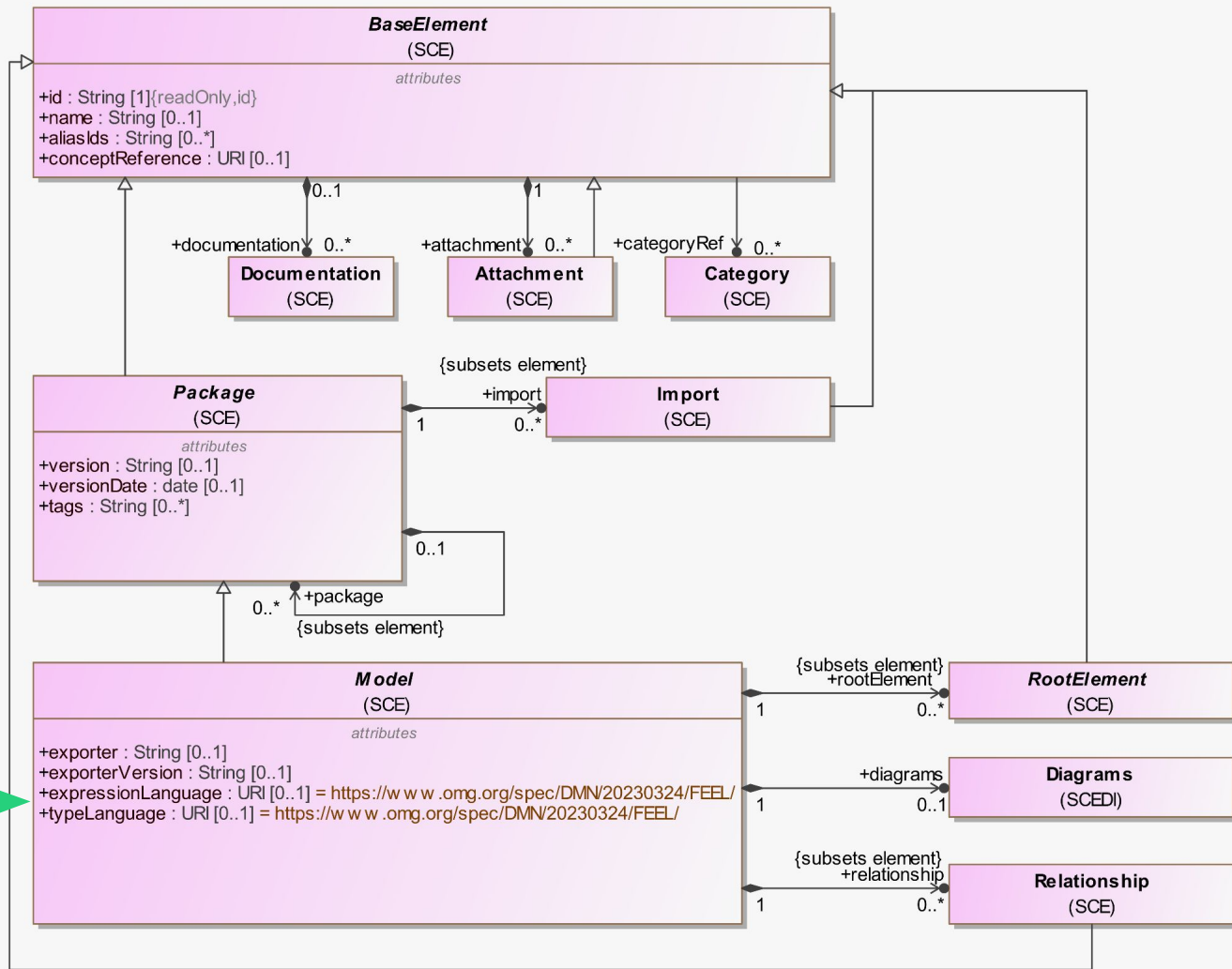
- Generalizes **structural elements and patterns** introduced by BPMN 2.0
- Adopted by DMN 1.x and CMMN 1.x but by **copy & paste**
- SCE emerged during development of Shared Data Model and Notation (SDMN)
- SDMN 1.0 first specification directly based on SCE
 - => good example of how to apply SCE
- **Backwards compatible** with BPMN and DMN
 - => may adopt SCE to eliminate redundancy

Other specifications may leverage SCE components, especially if they want to be **compatible with BPMN's ecosystem of standards**.

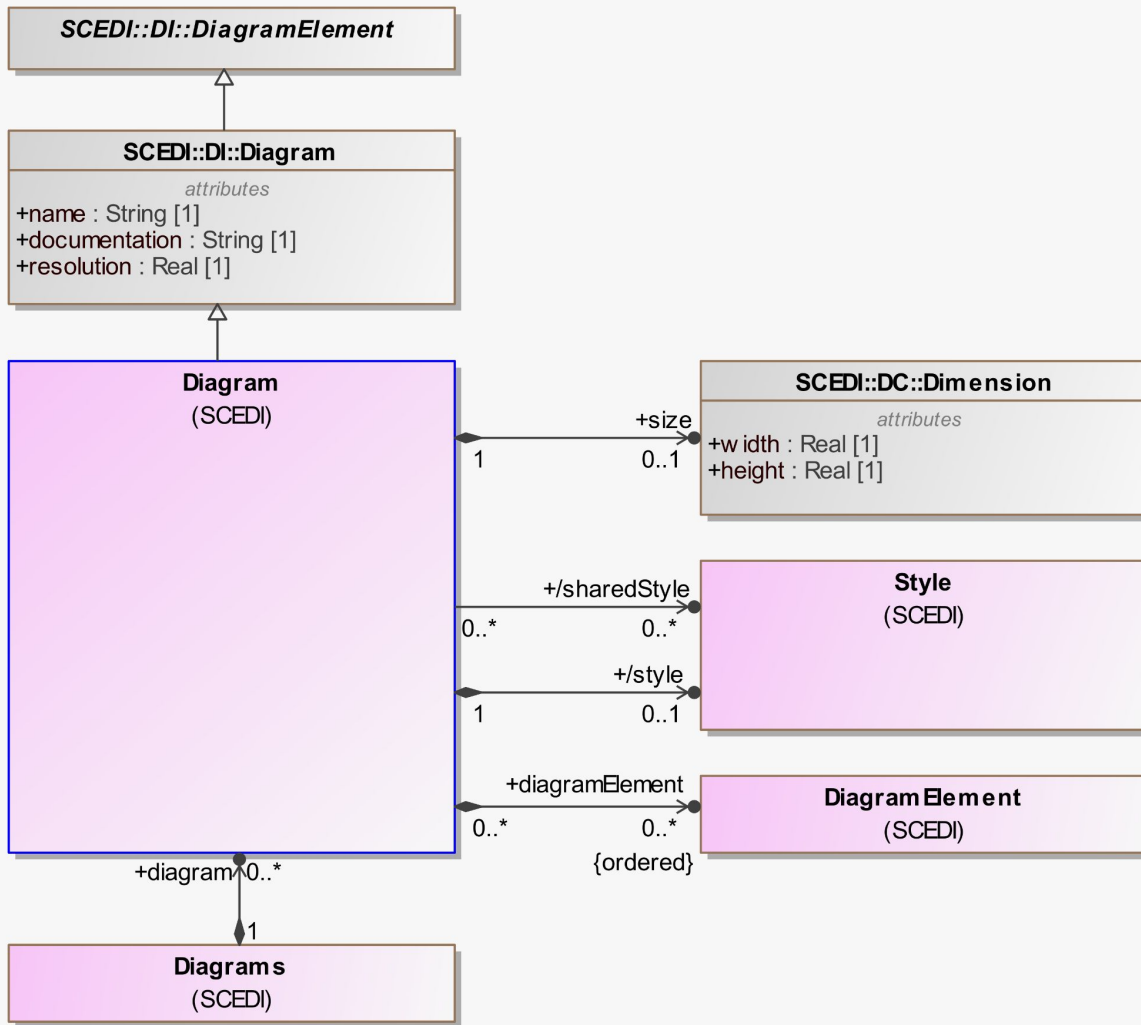
[omg.org/spec/SCE](https://www.omg.org/spec/SCE)

SCE Meta Model

FEEL as default



SCEDI Meta Model



Reusable without
language-specific
DI Meta Model!

=> SDMN 1.0 has
less than 50 pages