

---

# Managing Complex Interoperability Solutions using Model-Driven Architecture

---

Nico Bau, Michael Gerz – Fraunhofer FKIE

Francisco Loaiza, Steven Wartik – Institute for Defense Analyses

16th ICCRTS

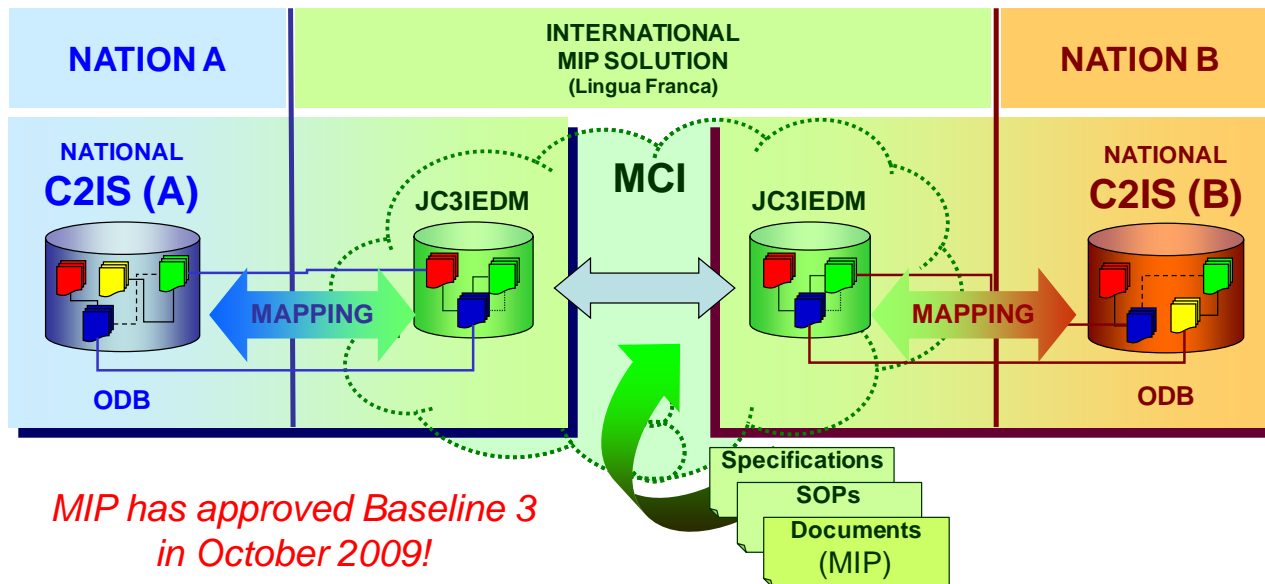
June 21-23, Québec City, Kanada

# Outline

- Multilateral Interoperability Programme (MIP)
- JC3IEDM
- Model-Driven Architecture (MDA)
- Query-View-Transformation (QVT)
- Summary

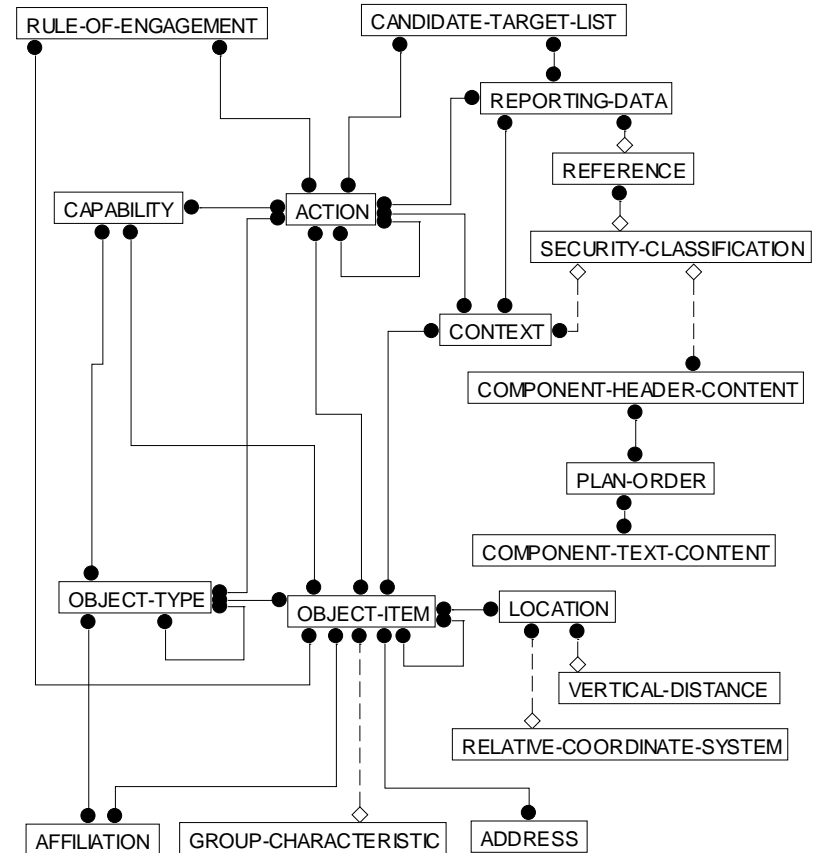
# Multilateral Interoperability Programme

“The aim of the Multilateral Interoperability Programme (MIP) is to achieve **international interoperability** of Command and Control Information Systems (C2IS) **at all levels** from corps to battalion, or lowest appropriate level, in order to **support multinational (including NATO), combined and joint operations** and the advancement of digitization in the international arena.“



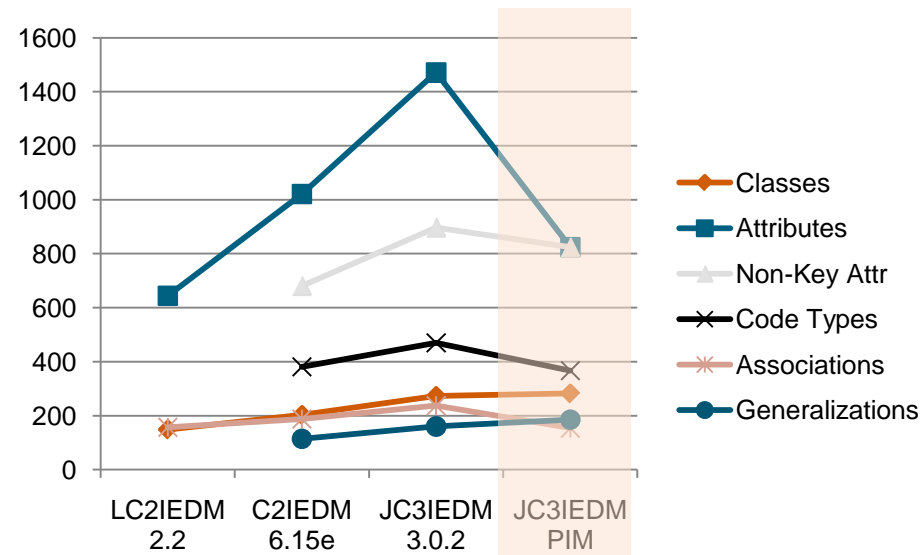
# JC3IEDM

- **J**oint **C**onsultation, **C**ommand, and **C**ontrol **I**nformation **E**xchange **D**ata **M**odel
- NATO ratification as **STANAG 5525**
- Latest version: **JC3IEDM 3.0.2**
  - Plans & Orders, ATO, MMW, CBRN, ...
- Complex data model based on generic core concepts
- Entity relationship model
- Semantic definitions
  - Business Rules
  - Free-text documentation



# JC3IEDM Restructuring (1)

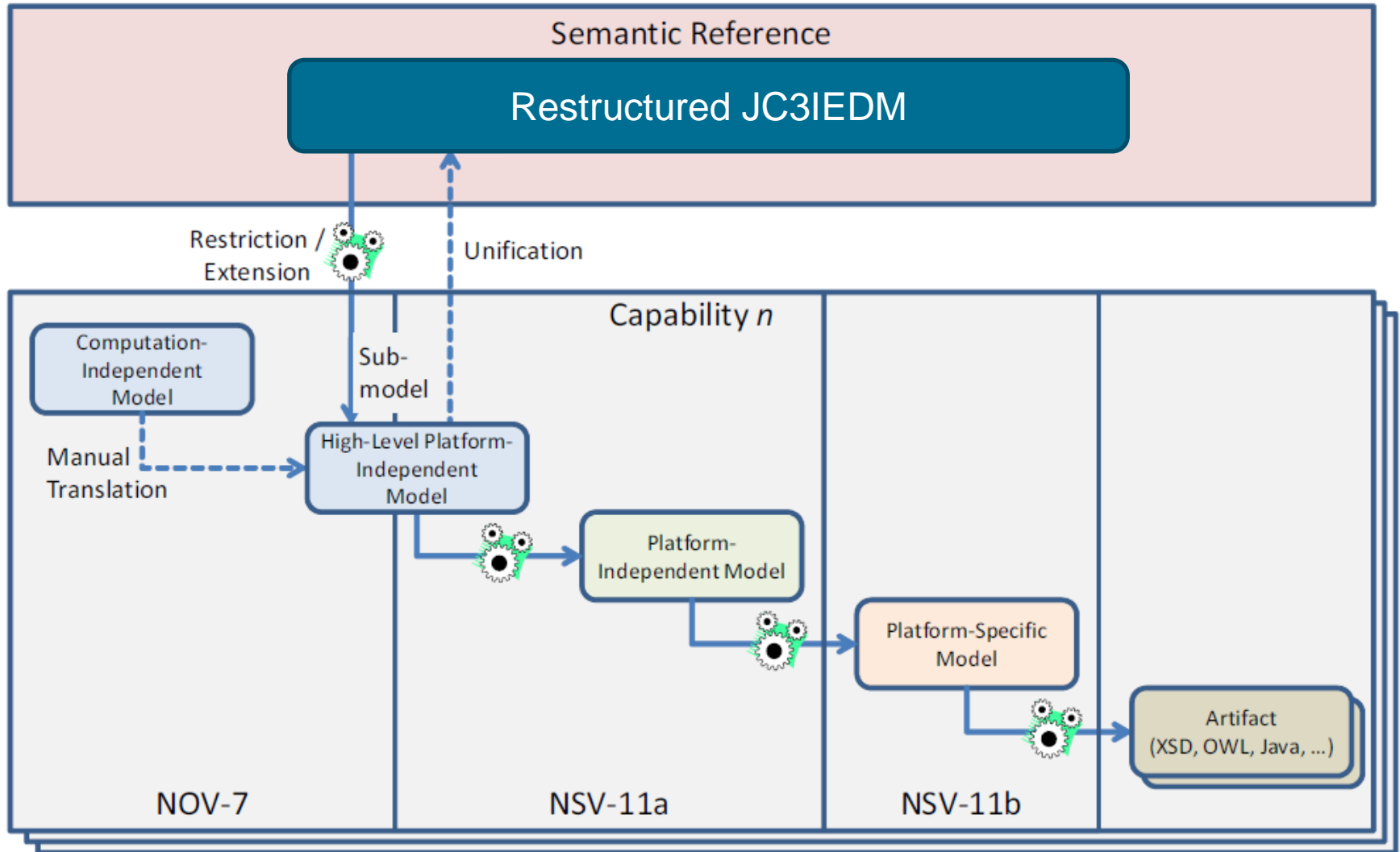
- JC3IEDM has been transformed from a small, generic hub into a comprehensive data model
- Configuration Management
  - Growing size and complexity results in configuration management challenges
  - Tracking and applying changes is laborious
- Faster response to user requirements
  - From operational requirement to the field
- Keep existing information exchange services stable
- ➔ Incremental delivery of independent capabilities
- ➔ Modular interoperability solution



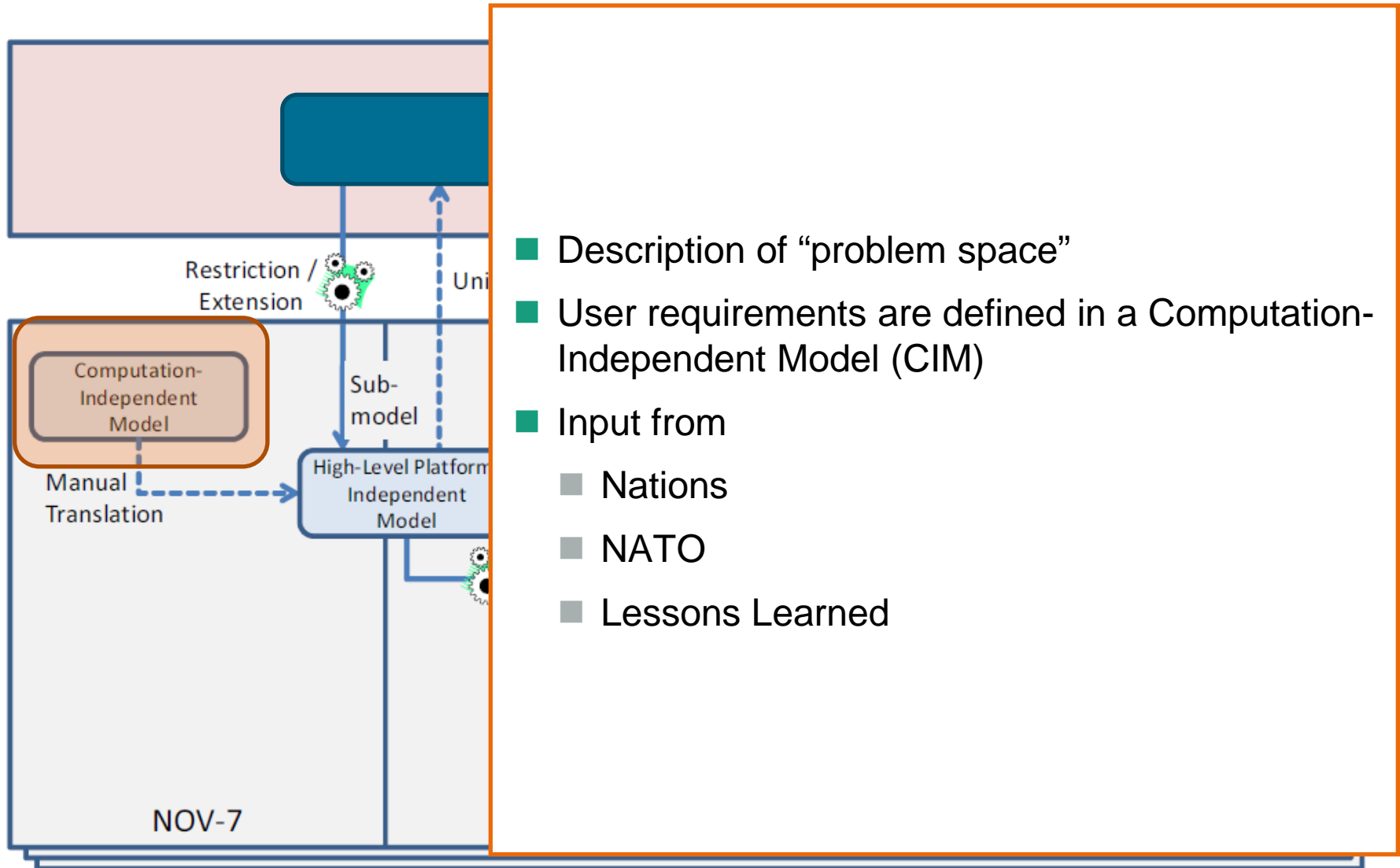
# JC3IEDM Restructuring (2)

- Entity-Relationship model is platform-specific
  - Database concepts, e.g., key attributes, discriminator codes for sub-typing
  - Not perfectly suited to other application areas
- Resolve well-known problems/workarounds, e.g.,
  - Deletion/update of information
  - Grouping of information
  - Archiving
- Make the model independent from a specific exchange mechanism
- Generalize existing concepts
- Provide a sound basis for the definition of capability- and COI-specific sub-models

# Model-Driven Architecture (MDA) in MIP

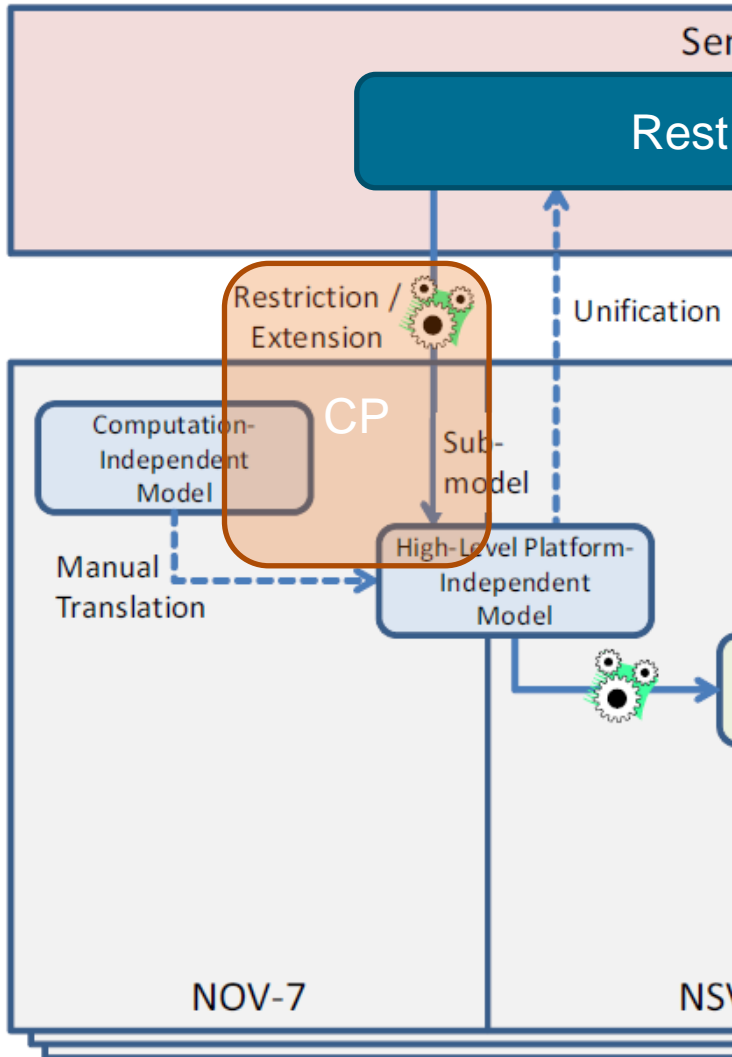


# Model-Driven Architecture (MDA) in MIP



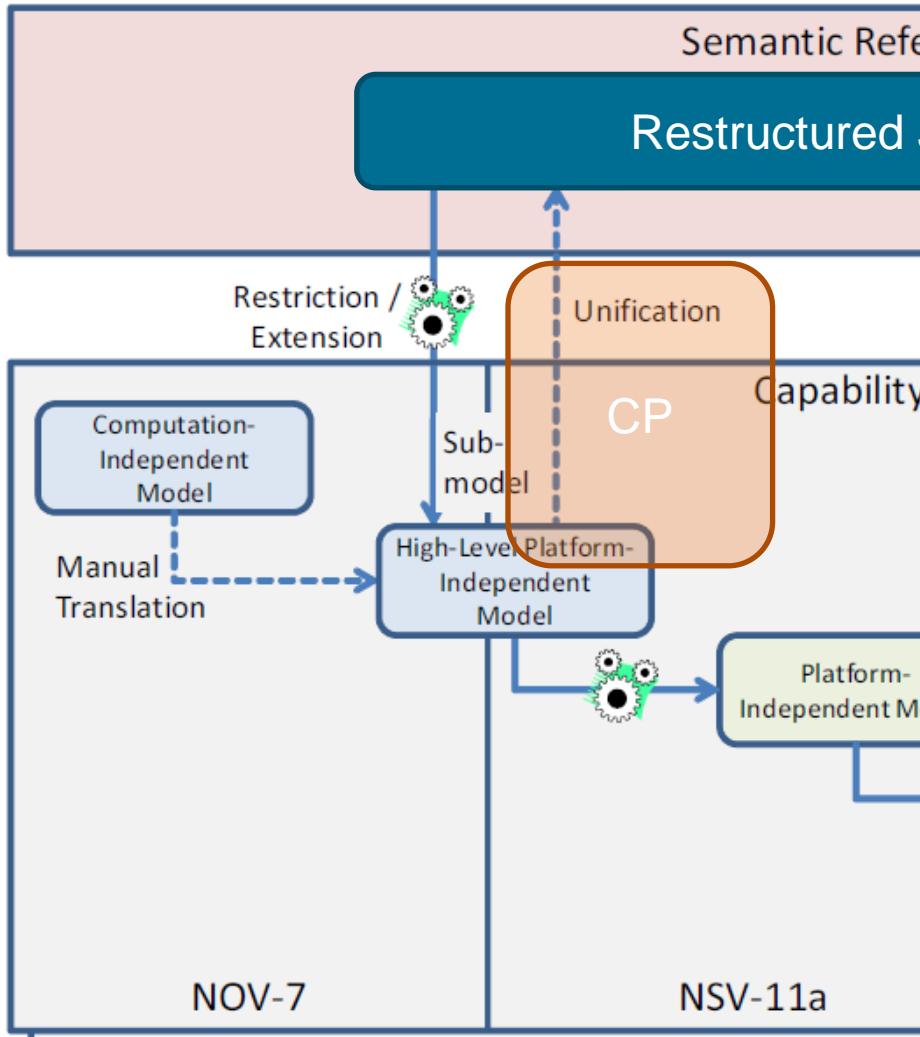


# Model-Driven Architecture (MDA) in MIP



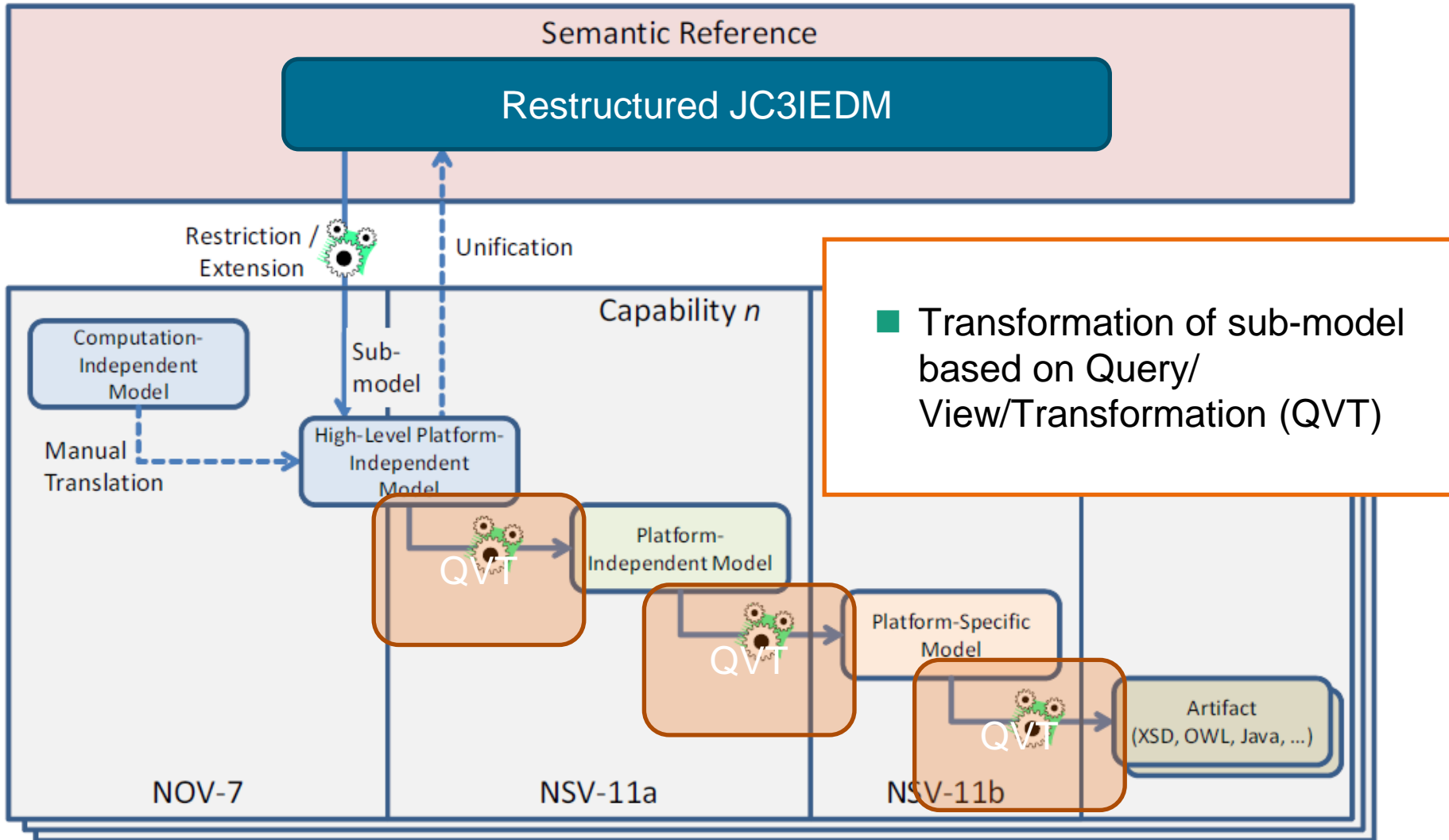
- Mapping to “solution space”
- Formal change proposal in XML format
- Describes a subview of the JC3IEDM
  - Classes
  - Associations
  - Attributes
  - Domain values
- Extends/modifies this subview
  - New classes, associations, ...

# Model-Driven Architecture (MDA) in MIP

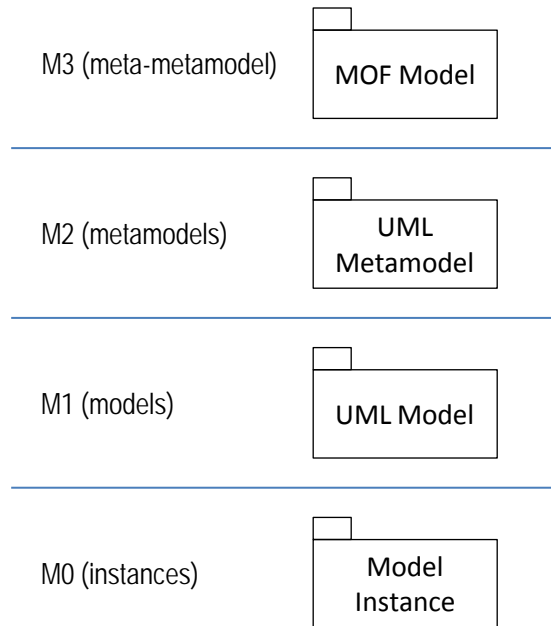


- Extensions may be fed back into Semantic Reference
  - New classes
  - New associations
  - New attributes
  - New domain values
- Impact analysis for other sub-models
- XML change proposal

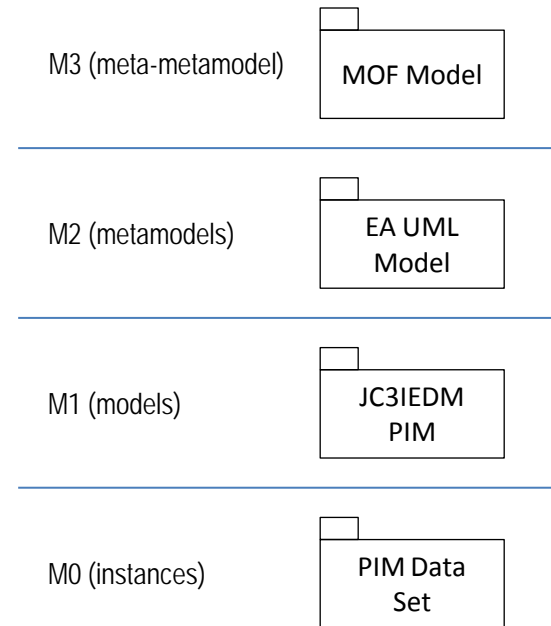
# Model-Driven Architecture (MDA) in MIP



# Model-to-Model Transformations using QVT

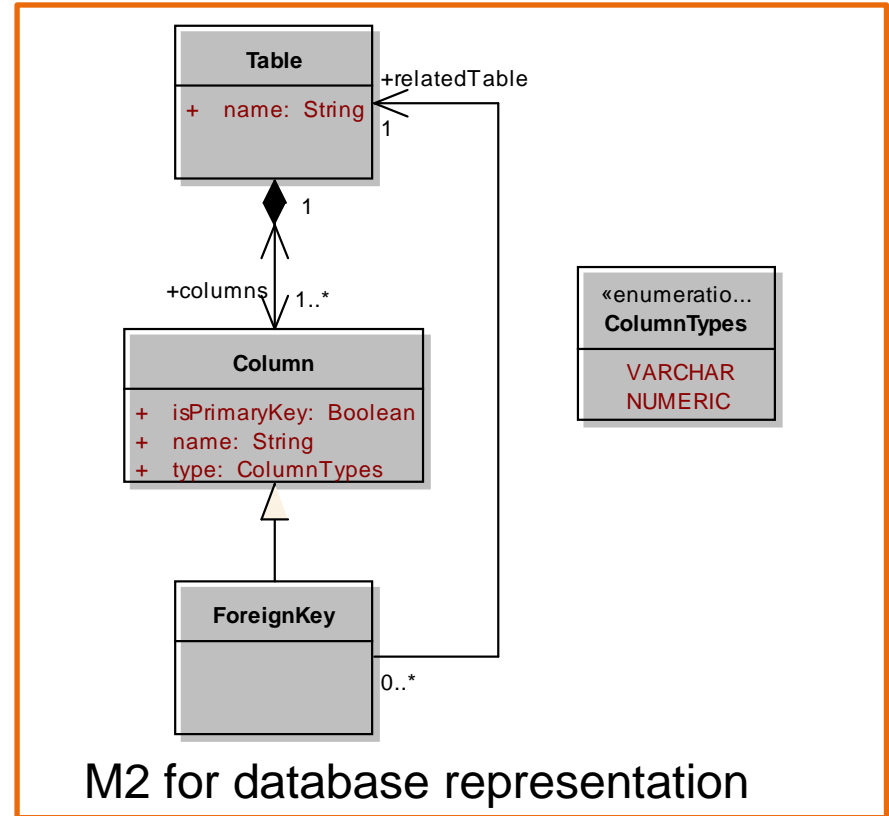
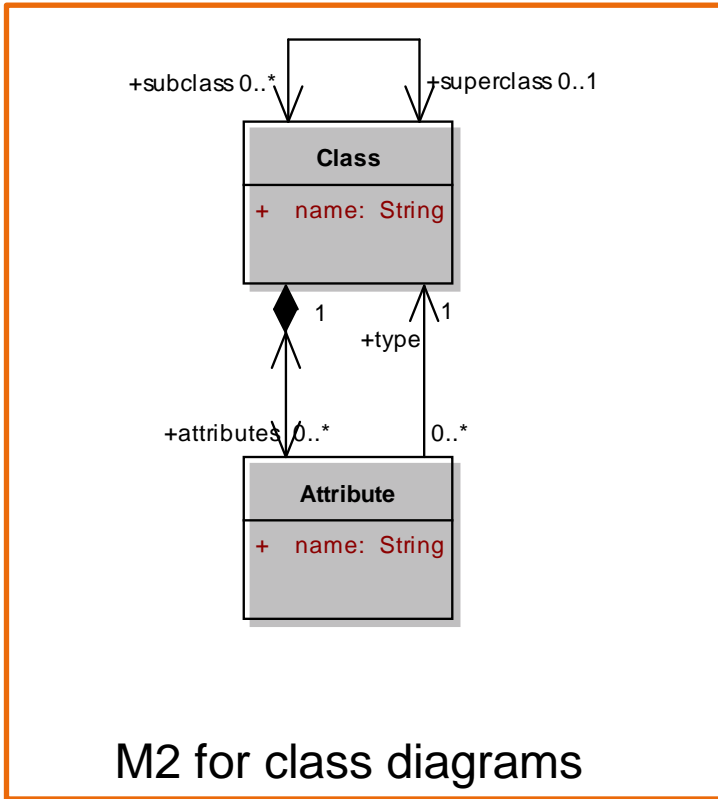


The Standard MOF Layers

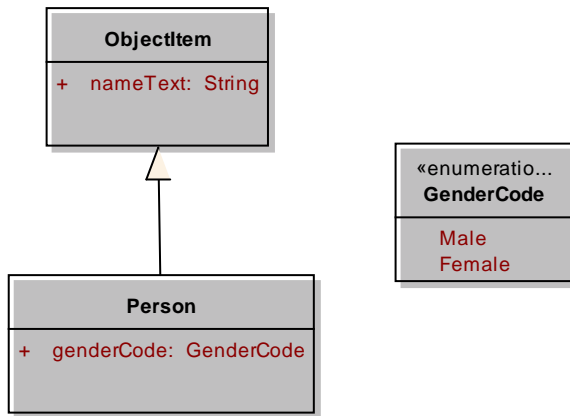


Mapping of JC3IEDM to MOF

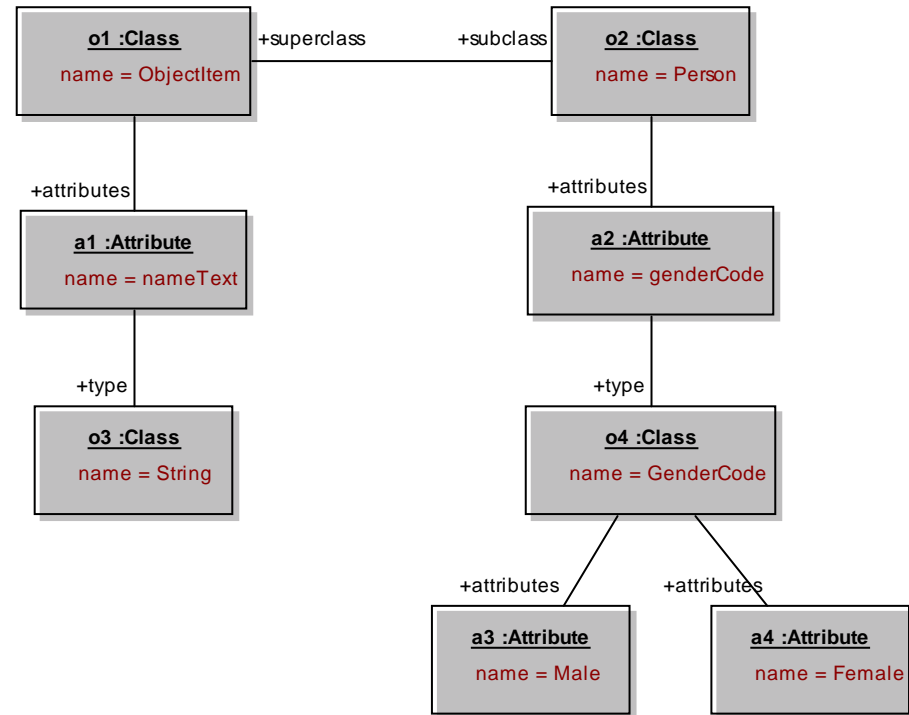
# MOF Level 2 Metamodels



# JC3IEDM Instantiation



Example class diagram



Corresponding object diagram

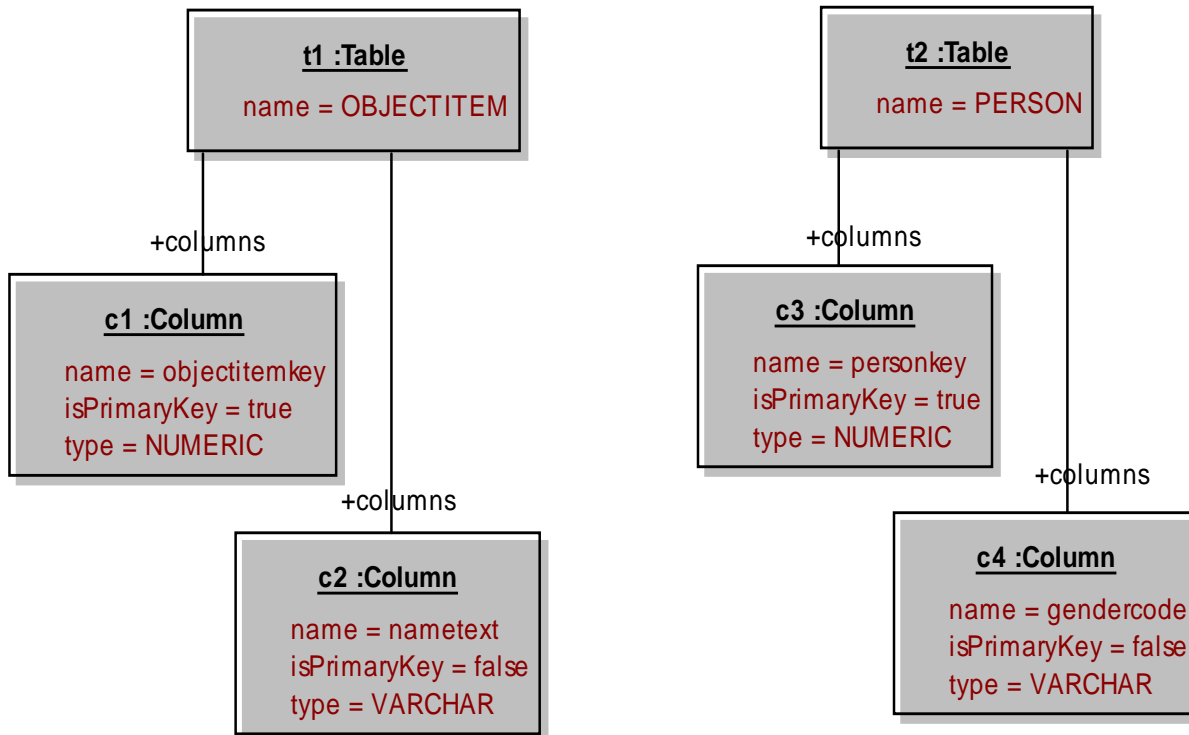
# QVT Script Example

```
transformation Class_DBMS(classModel: Classes, dbmsModel: Tables) {
  top relation Class_Table {
    domain classModel class: Class {} { name <> 'String' };
    enforce domain dbmsModel table: Table {
      name = class.name.toUpper(),
      columns = pk: Column {
        name = class.name.toLowerCase().concat('key'),
        type = ColumnTypes::NUMERIC,
        isPrimaryKey = true
      }
    };
    where { Attribute_Column(class, table); }
  }

  relation Attribute_Column {
    domain classModel class: Class {
      attributes = attr: Attribute {}
    };
    enforce domain dbmsModel table: Table {
      columns = col: Column {
        name = attr.name.toLowerCase(),
        type = if attr.type.name = 'String' or isEnumeration(attr.type.name)
              then ColumnTypes::VARCHAR
              else ColumnTypes::NUMERIC
            endif,
        isPrimaryKey = false
      }
    };
  }

  query isEnumeration(name: String): Boolean {
    let size: Integer = name.size() in name.substring(size-4,size) = 'Code'
  }
}
```

# QVT Output Example



Resulting PSM for an RDBMS Representation  
of the PIM Classes



# Summary (1)

- Restructured JC3IEDM PIM
  - Modular
  - Extensible
  - Simple, consistent
  - Work in progress, probably finished by the end of this year
  
- Change Management
  - Changes Proposals are formal XML documents
  - Changes can be applied to sub-model and/or the whole JC3IEDM PIM
  - Automated impact analysis
  - ➔ Different COIs can work in parallel on different sub-models
  - ➔ First step towards a domain-based approach

# Summary (2)

- Model Driven Architecture
  - Allows for a cleaner, easier to understand PIM and consistent PSMs
  - Improvement in traceability from conception down to the implemented solutions
  - Enhanced solution quality through largely automated conversion of the business constraints and conditions into executable code
  - QVT scripts describe transformations elegantly
  - Transformations and tools are/will be provided by MIP
  - Implementers are invited to extend/improve the transformations for use in their national systems
  
- Model, Tools, Examples are available at <http://mda.cloudexp.com>

Questions? Comments?

Thank you very much for your Attention!