Model-Driven Approach for Metadata Specifications

**UCMIS, a subset of UML class diagram items, is intended for data modeling**
- It focuses on core items that are familiar from object-oriented programming
- The subset focuses on items that describe classes, describe their relationships to each other, and their attributes
- The subset ensures structural interoperability between UML tools

**Canonical XMI**
- Canonical XMI (see Appendix B of the OMG XMI 2.5.1 specification) constitutes a specific constrained format of XMI that minimizes variability, provides more predictable identification and ordering, and ensures syntactic interoperability
- UCMIS class models as Canonical XMI can be imported into many UML tools (but no tool exports as Canonical XMI)

**Transformations**

**Transformation from UCMIS model as Canonical XMI to documentation and encodings**
- Software tool: UCMIS Model to Text (UCMIS.M2T)
  - UCMS.M2T is a tool for the generation of the classifier documentation (including UML diagrams) and syntax representations of a model confirming to UCMIS
  - It uses the Eclipse Acceleo implementation of the OMG standard MOF Model to Text Transformation Language (MOFM2T) with the Eclipse OCL2PQL plug-in

**Transformation from proprietary XMI to Canonical XMI**
- Software tool: to-canonical-xmi (set of XSLTs)
  - Intensively tested for Enterprise Architect XMI flavor
  - Basic tests for flavors other major UML editing tools
  - Output is Canonical XMI which can be imported into many UML tools

**Interoperability**
- UCMIS models as Canonical XMI ensure interoperability on the structural and syntactic level between UML tools.

**Model-Driven Products**
- Field-level documentation: one page per class and data type
- Syntax representations: XML Schema, RDF (ontology in Turtle, JSON-LD, in the works: SHACL and ShEx)
- Further model processing in UML tools

**Model-Driven Approach for Metadata Specifications**

**UML Model Creation**
- Model editing in a UML tool like Enterprise Architect
- Using only items of UCMIS
- Exporting to XMI (often proprietary flavour)

**Includes**
- the conceptual structure of a metadata specification,
- the documentation of all individual elements such as classes, data types, and class relationships.