T4U

EUROPEAN INSURANCE AND OCCUPATIONAL PENSIONS AUTHORITY

XBRL integration

11:00 – 11:30, Slot 2

26/11/2014, Brussels

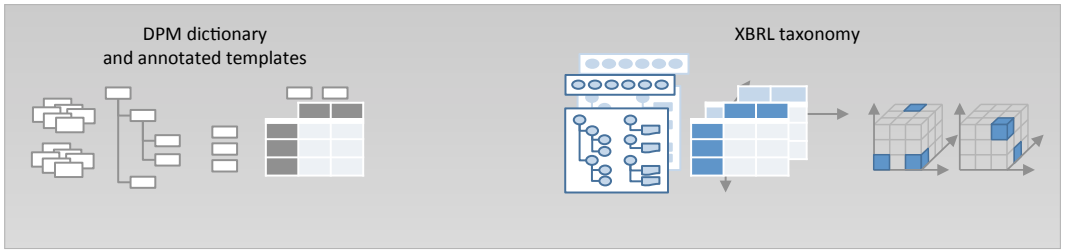
Eurofiling Session on EIOPA Tool for Undertakings

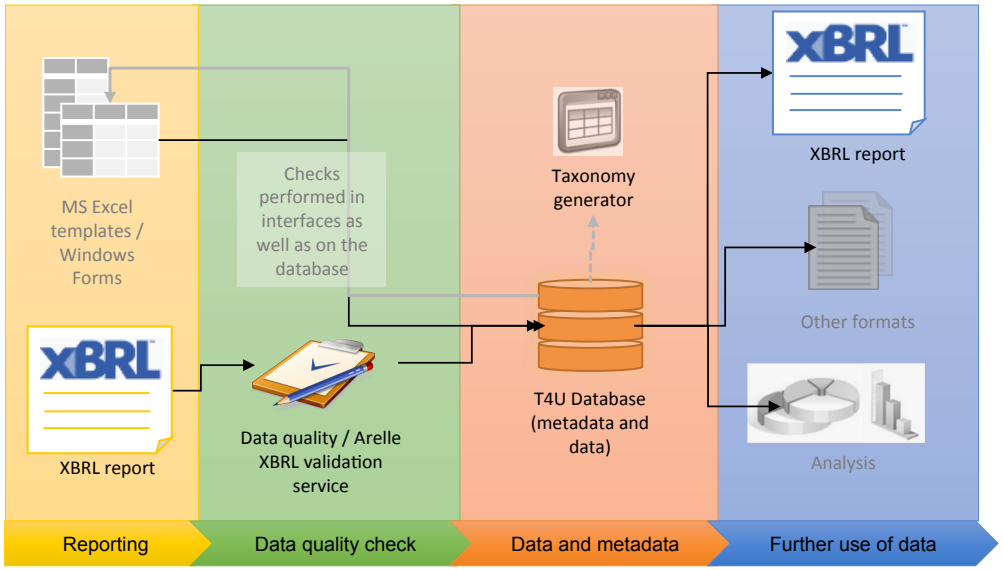
Agenda

- XBRL technical architecture
- XBRL report input and validation
- XBRL report output
- Arelle implementation strategy within $^{T}4_{U}$
- Benefits to NCA integration of Arelle

XBRL Technical Architecture

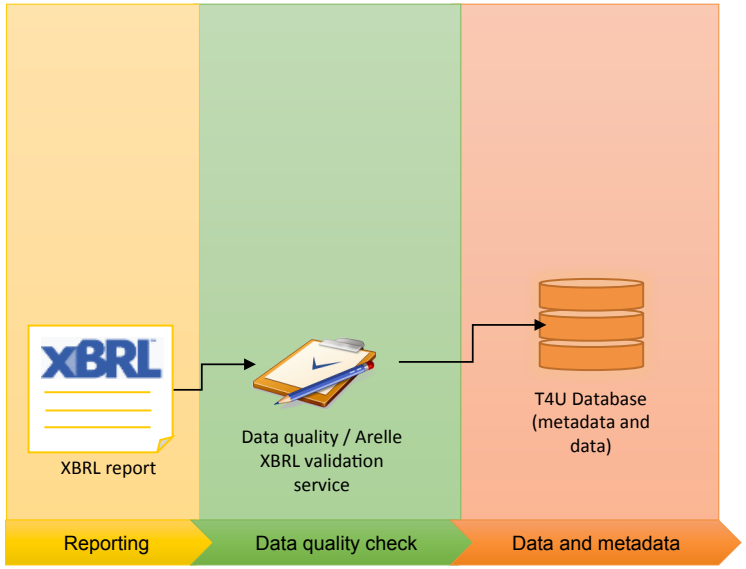
- XBRL as a syntax (angle brackets)
  - XBRL reports (xml files)
    - Interchange media
  - XBRL taxonomies (for validation)
    - Standard publication media
- XBRL as an abstract model
  - $^{T}4_{U}$ DPM database
    - Metrics, dimensions, members
    - Forms, axes, cells, rules
  - Classical relational table structure
- XBRL as a basis for validation
  - Data point validity
    - Metric value and precision
    - Metric dimensions and aspects
    - Enumerations





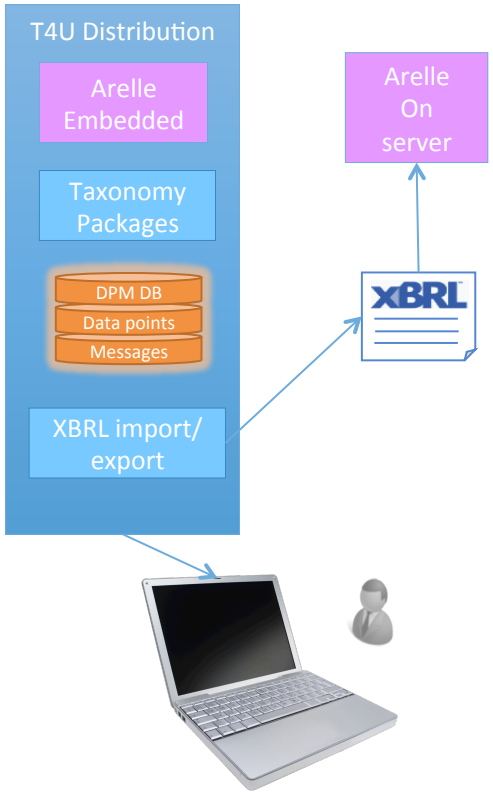
XBRL Report Import and Validation

- Native (C#, transliteration from Arelle)
  - Import validation against DPM DB
  - Data types, dimensions, enumerations
  - EBA/EIOPA Filer Manual validations
  - XBRL streaming support
- Arelle validation against DPM DB
  - Same as native
- Arelle validation against DTS
  - DTS in taxonomy packages
  - Full XBRL validations
  - EBA/EIOPA Filer Manual validations
  - Formula support options



Arelle implementation strategy

- Local implementation strategy
  - C# transliteration of Arelle features for "Native" XBRL report import and output
  - Embedded minimal Arelle as a zip with embedded taxonomy packages
  - Loads itself into application's executable directories
  - Self-updates because it's embedded in T4U distributions
  - T4U status line shows Arelle status line (integrated to T4U GUI)
- $^{T}4_{U}$ Remote Server strategy
  - production-quality CGI server
  - remote validation optional
- Validation results
  - Interface messages are captured in $^{T}4_{U}$ Database tables
  - GUI consistent for both local and remote implementation
- Validation performance
  - Profile each feature
    - time
    - memory usage
  - Evaluate DPM support where more efficient than DTS features
- EIOPA (& EBA) Filer Manuals



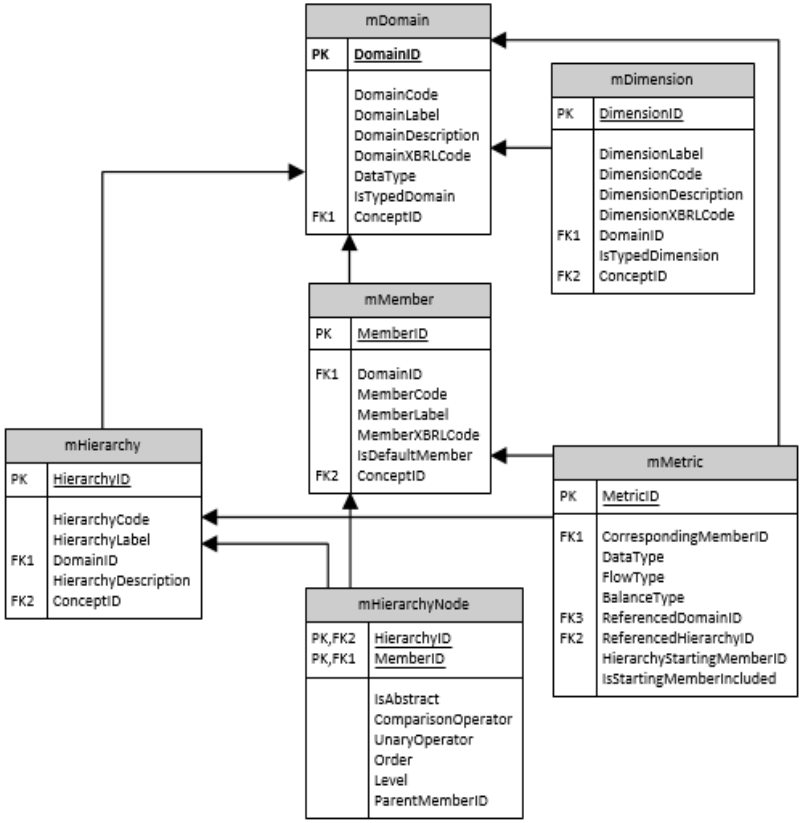
Native (C#) implementation

- Arelle C# Parser
  - Streaming architecture
  - Filer manual and DPM-based validations
- Arelle C# Saver
  - Non-validating
- Arelle Command Interface
  - Import interfaces
    - Native C# parser
    - Embedded Arelle (Python) DPM-DB interface (no DTS)
    - Embedded Arelle (Python) DTS interface
  - XBRL Report Saver
    - Native C# saver
    - Embedded Arelle (Python) DPM-DB interface (no DTS)
    - Embedded Arelle (Python) DTS interface
  - Arelle validator
    - Embedded Arelle
    - Server-hosted Arelle

Validations

- DPM-based (database XBRL approach)
  - Use metric code to determine schema type
    - Data type validity
  - Use database table queries to determine
    - Module of framework (schemaRef)
    - Filing indicators allowed by module
    - Metrics valid by filing indicators
    - Dimensional signature validity by filing indicators
- DTS-based (traditional XBRL approach)
  - Use XML schema to determine
    - Data type and XML particle structure validity
  - Use XBRL dimensions to determine
    - Dimensional validity (irrespective of filing indicators)
  - Use XBRL formula to validate structure
- Python and C# Filer Manual checks

DPM structure supports validation



- dictionary contains definitions of domains (mDomain)
- each domain:
  - consists of members (mMembers), and
  - is associated with dimensions (mDimensions) that further contextualize members in the information requirements section of the model (database)
- members are gathered in hierarchies (mHierarchy and mHierarchyNode)
  - for documentation purposes
  - in order to support management of the dictionary
  - to describe basic arithmetical relationships between members (following the nesting and values of mHierarchyNode.ComparisonOperator and mHierarchyNode.UnaryOperator)
- metrics (mMetric) are members of a selected domain that are further associated with attributes:
  - period type
  - data type (which could take form of a list of members of another domain by referencing a hierarchy of its members)

$^{T}4_{U}$ to Arelle Interface

- Local operation (Arelle Python implementation)
  - Arelle is installed/updated from a zip within $^{T}4_{U}$ distribution
  - Input sources
    - DPM database or XBRL instance files
    - Taxonomy packages included (Arelle operates “offline”)
  - Output messages serialized into XML
    - parsed into DPM DB (error code, severity, text and fact references)
  - Status line feedback from Arelle to $^{T}4_{U}$ Status Widget
- Remote operation (Arelle web service)
  - Input sources
    - XBRL instance files (posted to server, no access to DPM database)
  - Output messages as above

Validations Integrated to Database

Arelle Validation Status

Data Point Signature	Message Code	Value
MET(s2md_met:ei1025)|s2c_dim:AZ (<s2c_typ:IDG>33</s2c_typ:IDG>)	xmlSchema:valueError	Fact value enumeration error: s2md_met:ei1025 context c2 value s2c_CU:GDYNIA
MET(s2md_met:di1043)|s2c_dim:AZ (<s2c_typ:IDG>33</s2c_typ:IDG>)	sqlDB:factDimensionsError	Fact Qname s2md_met:di1043 dimensions not allowed for filing indicators; Extra dimensions of fact: none, missing dimensions of fact: none, different dimensions of fact: dim: s2c_dim:AZ fact: <s2c_typ:IDG>33</s2c_typ:IDG> DPMsig: *
MET(s2md_met:di1044)|s2c_dim:AZ (<s2c_typ:IDG>33</s2c_typ:IDG>)	sqlDB:factDimensionsError	Fact Qname s2md_met:di1044 dimensions not allowed for filing indicators; Extra dimensions of fact: none, missing dimensions of fact: none, different dimensions of fact: dim: s2c_dim:AZ fact: <s2c_typ:IDG>33</s2c_typ:IDG> DPMsig: *
MET(s2md_met:ei1004)|s2c_dim:AZ (<s2c_typ:IDG>33</s2c_typ:IDG>)	sqlDB:factDimensionsError	Fact Qname s2md_met:ei1004 dimensions not allowed for filing indicators; Extra dimensions of fact: none, missing dimensions of fact: none, different dimensions of fact: dim: s2c_dim:AZ fact: <s2c_typ:IDG>33</s2c_typ:IDG> DPMsig: *
MET(s2md_met:ei1017)|s2c_dim:AZ (<s2c_typ:IDG>33</s2c_typ:IDG>)	sqlDB:factDimensionsError	Fact Qname s2md_met:ei1017 dimensions not allowed for filing indicators; Extra dimensions of fact: none, missing dimensions of fact: none, different dimensions of fact: dim: s2c_dim:AZ fact: <s2c_typ:IDG>33</s2c_typ:IDG> DPMsig: *
MET(s2md_met:ei1025)|s2c_dim:AZ (<s2c_typ:IDG>33</s2c_typ:IDG>)	sqlDB:factDimensionsError	Fact Qname s2md_met:ei1025 dimensions not allowed for filing indicators; Extra dimensions of fact: none, missing dimensions of fact: none, different dimensions of fact: dim: s2c_dim:AZ fact: <s2c_typ:IDG>33</s2c_typ:IDG> DPMsig: *
MET(s2md_met:ei1318)|s2c_dim:AZ (<s2c_typ:IDG>33</s2c_typ:IDG>)	sqlDB:factDimensionsError	Fact Qname s2md_met:ei1318 dimensions not allowed for filing indicators; Extra dimensions of fact: none, missing dimensions of fact: none, different dimensions of fact: dim: s2c_dim:AZ fact: <s2c_typ:IDG>33</s2c_typ:IDG> DPMsig: *
MET(s2md_met:ei1516)|s2c_dim:AZ (<s2c_typ:IDG>33</s2c_typ:IDG>)	sqlDB:factDimensionsError	Fact Qname s2md_met:ei1516 dimensions not allowed for filing indicators; Extra dimensions of fact: none, missing dimensions of fact: none, different dimensions of fact: dim: s2c_dim:AZ fact: <s2c_typ:IDG>33</s2c_typ:IDG> DPMsig: *
MET(s2md_met:ei1521)|s2c_dim:AZ (<s2c_typ:IDG>33</s2c_typ:IDG>)	sqlDB:factDimensionsError	Fact Qname s2md_met:ei1521 dimensions not allowed for filing indicators; Extra dimensions of fact: none, missing dimensions of fact: none, different dimensions of fact: dim: s2c_dim:AZ fact: <s2c_typ:IDG>33</s2c_typ:IDG> DPMsig: *
MET(s2md_met:ei1633)|s2c_dim:CS (s2c_CS:x18888)	sqlDB:factDimensionsError	Fact Qname s2md_met:ei1633 dimensions not allowed for filing indicators; Extra dimensions of fact: none, missing dimensions of fact: none, different dimensions of fact: dim: s2c_dim:CS fact: s2c_CS:x18888 DPMsig: s2c_CS:x18

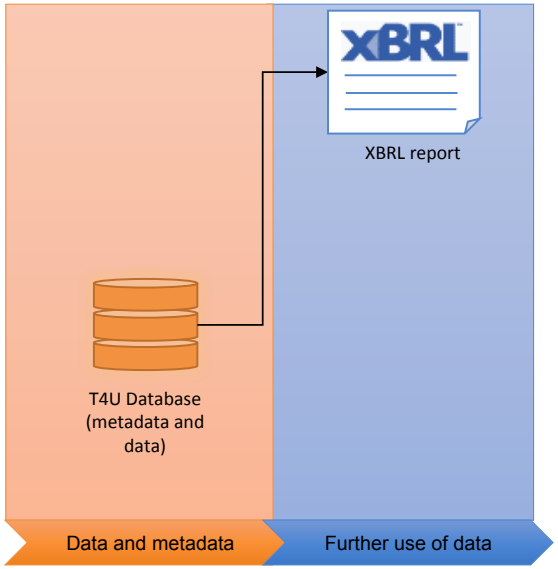
Filer Manual Validations

Category	Rule	Specification	C# Native Import	Arelle Import DPM DTS	Arelle Validation
Filing syntax rules	1.1	Filing naming	√	√	√
	1.4	Character encoding of XBRL instance documents	√	√	√
	1.5	Taxonomy entry point selection	√	√	√
	1.6	Missing Filing indicators	√	√	√
	1.6.1	Multiple filing indicators for the same reporting unit	√	√	√
	1.6.2	Filing indicators in several tuples	√	√	√
	1.7	Implication of no facts for an indicated template.			
	1.7.1	No facts for non-indicated templates	√	√	?
	1.09	Valid XML-XBRL	∂	∂	√
	1.10	Valid according to the defined business rules	tbd	tbd	?
	1.11	Taxonomy extensions by reporters			
	1.12	Completeness of the instance			
Instance syntax rules	2.1	@xml:base			√
	2.2	xbrli:xbrl/link:schemaRef content	√	√	√
	2.3	xbrli:xbrl/link:schemaRef	√	√	√
	2.4	xbrli:xbrl/link:linkbaseRef	√	√	√
	2.5	XML comment and documentation			
Context related rules	2.6	xbrli:xbrl/xbrli:context/@id			
	2.7	Unused xbrli:xbrl/xbrli:context	√	√	√
	2.8	Identification of the reporting entity			
	2.9	One reporter	√	√	√
	2.10	xbrli:xbrl/xbrli:context/xbrli:period/*	√	√	√
	2.11	xbrli:xbrl/xbrli:context/xbrli:period/xbrli:forever	√	√	√
	2.13	XBRL period consistency	√	√	√
	2.14	xbrli:xbrl/xbrli:context/xbrli:entity/xbrli:segment and xbrli:xbrl/xbrli:context/xbrli:scenario	√	√	√
	2.15	xbrli:xbrl/xbrli:context/xbrli:entity/xbrli:segment and xbrli:xbrl/xbrli:context/xbrli:scenario	√	√	√
Fact related rules	2.16	Duplicate facts	√	√	√
	2.17	@precision	√	√	√
	2.18	@decimals			not testable
	2.19	zero value, empty, nil value @xsi:nil	√	√	√
	2.2	@xml:lang	√	√	√
Unit Related Rules	2.21	Duplicates of xbrli:xbrl/xbrli:unit	√	√	√
	2.22	Unused xbrli:xbrl/xbrli:unit			
	2.23	xbrli:xbrl/xbrli:unit/* content			
	2.24	xbrli:xbrl/xbrli:unit/xbrli:measure			
Additional Guidance	3.1	One Currency	√	√	√
	3.2	Non-monetary numeric units	√	√	√
	3.3	Decimal representation			
	3.4	Unused namespace prefixes	√	√	√
	3.5	Re-use of canonical namespace prefixes	√	√	√

- C# native implementation
  - Checked when parsing to import XBRL instance to DPM database
- Arelle (Python) implementations
  - DTS-based XBRL instance validation
    - Plug-in implementation
    - Independent of $^{T}4_{U}$ tool
  - Non-DTS streaming import validation
    - Requires $^{T}4_{U}$ DPM database

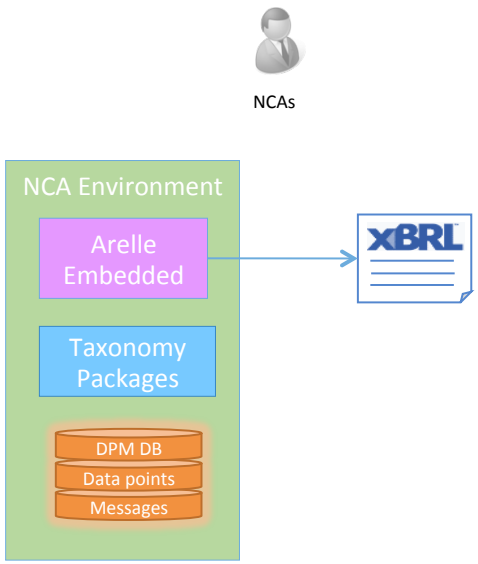
XBRL Report Output

- Native
  - Data types and precision adjustment
    - From Database to XBRL spec
  - XBRL streaming file organization
- Arelle using only DPM DB
  - Same as native
- Arelle with DTS
  - DTS and object model are loaded
  - XBRL elements validated



$^{T}4_{U}$ benefits to NCA Arelle Integrators

- Reuse DPM-based features for NCA's internal integration
  - DPM-based validations (vs. DTS-based validations)
    - Performance in production
  - DPM-based data checks (vs. formula checks)
- Pattern for NCA's integration (C# - Arelle) like $^{T}4_{U}$



T4U

EUROPEAN INSURANCE
AND OCCUPATIONAL PENSIONS AUTHORITY

Thank you!

Herm Fisher