Major technical changes
in the Solvency II 2.2.0PWD XBRL Taxonomy

EIOPA Meet the Market

6 June 2017, Frankfurt am Main

Major changes

- Validation of multiple values for a fact

- LEI and ISIN validation

- Currency related validations

- Filing indicator and content template checks (TV0, TV24)

- Standard formula and Article 112 issue (ghost data points)

- Different severity for regular and non-regular reporting

- Deactivating assertions by means of XBRL

- Technical table

Validation of multiple values for a fact

- topic addressed in EIOPA_XBRL_Taxonomy_Documentation_2.1.0, Annex 2. Multiple values for a fact

- some facts in Solvency II represent predefined lists of options, i.e. the LOGs identify the set of allowed values to be reported in a cell

> Representing the activities of the broker involved, as considered by the undertaking. In
> case the activities are combined all activities must be mentioned separated be a ",":.
> 1 - Intermediary for placement
> 2 - Underwriting on behalf of
> 3 - Financial services

- Filing Rules section „VI.3 Cases where multi value elements reporting is applicable" guides that an undertaking shall report a pattern based on the numbers assigned to each option in ascending order and separated by commas, for example *"1"* or *"1,2"* or *"1,3"*, *"2,3"*, *"1,2,3"*, …

- implementation:
  - ideal: Extensible Enumerations 1.1 (https://www.xbrl.org/Specification/extensible-enumerations-1.1/PR-2017-02-08/extensible-enumerations-1.1-PR-2017-02-08.html)
    - proposed solution: a list of QNames (similar to singe value enumeration) of members belonging to a dictionary hierarchy identified on the definition of a metric
    - concern: Proposed Recommendation status with ongoing discussion in the XBRL Specifications Working Group on the final shape
  - in 2.1.0: enumeration of all options, e.g. ^(1|1,2|1,2,3|2|2,3|3)$
    - potential problem for large enumerations (from 1 to 18 gives over 260 000 combinations)
    - incorrectly implemented with some values missing (e.g. 1,3 above) – hence technically deactivated
  - in 2.2.0: regex, e.g. ^(1$|1,){0,1}(2$|2,){0,1}(3$|3,){0,1}$

- concerns:
  - defined as a string not enumeration item (although applicable hierarchies exist in the dictionary)
  - still requires specific implementation in GUI rather than standard solution

LEI and ISIN validation (1)

- implemented in test expressions of the technical validations (TV1 to TV21) checking patterns for instrument (TV5, TV9) and entity codes (other)

- applies to metrics (si1495, si1552, si1553, si1558, si1559, si1899, si1900, si1901, si2132, si2179, si2205) and dimensions in contexts (IW, UI, CA, CE, CV, GO, IZ, OV, RF, ZS)

- In 2.1.0 defined as regex expressions: ^ISIN/[A-Z0-9]{12}$, ^LEI/[A-Z0-9]{20}$

- In 2.2.0PWD defined as regex + custom function:
  - LEI: https://taxonomies.xbrl.org/taxonomy/5909aa8cbc180d68739ac316

  ### Legal Entity Identifier Taxonomy 2017-05-03 (PWD)   Login

  Taxonomy Registry > Legal Entity Identifier Taxonomy (2017-05-03 (PWD))

  Description:

  This taxonomy provides a standard concept and datatype for including Legal Entity Identifiers in XBRL Reports. Formula validation ensures that the identifiers have a valid checksum. In additional, formula validation will ensure that any entity identifiers defined using a scheme of "http://standards.iso.org/iso/17442" are in the correct format, and pass checksum validation.

  Entry points:

  Legal Entity Identifier
  Element, item type and associated formula rules for enforcing format and checksum of Legal Entity Identifiers.

  Details:

Name:	Legal Entity Identifier Taxonomy
Version:	2017-05-03 (PWD)
Published:	3 May 2017
Publisher:	XBRL International Inc.
Download:	lei-taxonomy-PWD-2017-05-03.zip

  - ISIN: custom implementation of EIOPA

  EIOPA_SolvencyII_XBRL_Taxonomy_2.2.0_PWD > eiopa.europa.eu > eu > xbrl > func
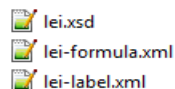
  - func.xsd
  - func-formula.xml
  - isin-check.xml

eiopa

LEI

- product of XII

- included in the 2.2.0PWD taxonomy package

EIOPA_SolvencyII_XBRL_Taxonomy_2.2.0_PWD › www.xbrl.org › taxonomy › int › lei › PWD › 2017-05-03

- lei.xsd
- lei-formula.xml
- lei-label.xml

- defines:
  - lei.xsd and lei-label.xml:
    - LegalEntityIdentifier concept
    - files not referenced from the Solvency II taxonomy files
  - lei-formula.xml
    - custom function checking LEI checksum lei-fn:validate-checksum + supportive functions
      - applied in Solvency 2 validations
      - referenced from eiopa.europa.eu\eu\xbrl\func\func.xsd (see next slide)
    - value assertions:
      - check for LegalEntityIdentifier concept and checks for entity identifiers in concepts
      - „turned off" in func-formula.xml by attaching a precondition evaluating always to false()

- concerns:
  - PWD status
  - undesired (and hence prohibited or not referenced) content of additional concept and three validations
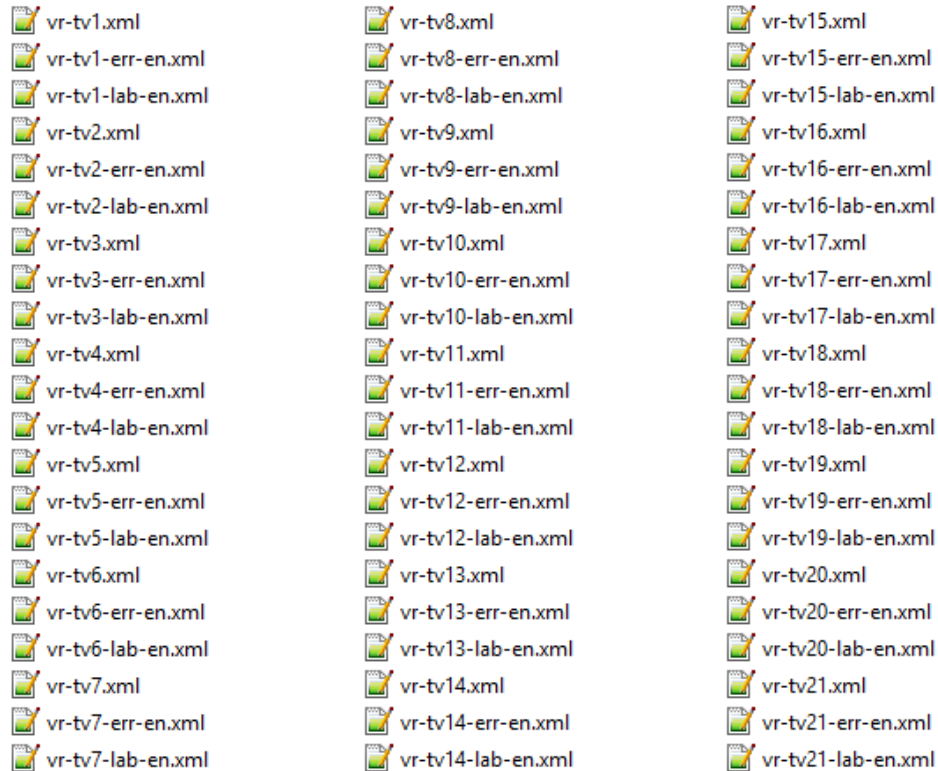
ISIN

- 12-character alpha-numerical code where two first alphabetic characters are the ISO 3166-1 alpha-2 code for the issuing country

- Luhn algorithm
  - convert any letters to numbers (A - 10, B - 11, C - 12, (…)): AU0000VXGZA -> 103000003133163510 (A = 10, G = 16, U = 30, V = 31, X = 33, Z = 35)
  - collect odd and even characters: 103000003133163510 => (1, 3, 0, 0, 3, 3, 1, 3, 1), (0, 0, 0, 0, 1, 3, 6, 5, 0)
  - multiply the group containing the rightmost character by 2: (0, 0, 0, 0, 1, 3, 6, 5, 0) -> (0, 0, 0, 0, 2, 6, 12, 10, 0)
  - add up the individual digits: $(1 + 3 + 0 + 0 + 3 + 3 + 1 + 3 + 1) + (0 + 0 + 0 + 0 + 2 + 6 + (1 + 2) + (1 + 0) + 0) = 27$
  - take the 10s modulus of the sum: 27 mod 10 = 7
  - subtract from 10: 10 - 7 = 3
  - take the 10s modulus of the result and compare with check digit: 3 mod 10 = 3
  - compare the result with the check digit: 3 = 3

- implemented in EIOPA_SolvencyII_XBRL_Taxonomy_2.2.0_PWD\eiopa.europa.eu\eu\xbrl\func\isin-check.xml

EIOPA_SolvencyII_XBRL_Taxonomy_2.2.0_PWD › eiopa.europa.eu › eu › xbrl › func

- func.xsd
- func-formula.xml
- isin-check.xml
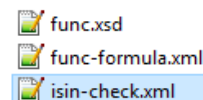
- defines isin_fn:isin-checksum-sig and supportive function

- concerns:
  - issues with some tools – requires experts' review and testing with various tools
  - solution to be preferably managed at XII level (similarly to LEI)

LEI and ISIN functions applications

- eiopa.europa.eu\eu\xbrl\func\func.xsd referenced from every module

- examples:

  - TV1: (lei-fn:validate-checksum(substring(xfi:fact-typed-dimension-value($a,QName("http://eiopa.europa.eu/xbrl/s2c/dict/dim"CA"))/s2c_typ:ID,5)) and matches(string(xfi:fact-typed-dimension-value($a,QName("http://eiopa.europa.eu/xbrl/s2c/dict/dim","CA"))/s2c_typ:ID), "^LEI/[A-Z0-9]{20}$")) or …

  - TV5: (isin_fn:isin-checksum-sig(substring(xfi:fact-typed-dimension-value($a,QName("http://eiopa.europa.eu/xbrl/s2c/dict/dim","IW"))/s2c_typ:ID,6)) and matches(string(xfi:fact-typed-dimension-value($a,QName("http://eiopa.europa.eu/xbrl/s2c/dict/dim","IW"))/s2c_typ:ID), "^ISIN/[A-Z0-9]{12}$")) or …

  - TV23: (lei-fn:validate-checksum(substring($a,5)) and matches($a, "^LEI/[A-Z0-9]{20}$")) or …

vr-tv1.xml	vr-tv8.xml	vr-tv15.xml
vr-tv1-err-en.xml	vr-tv8-err-en.xml	vr-tv15-err-en.xml
vr-tv1-lab-en.xml	vr-tv8-lab-en.xml	vr-tv15-lab-en.xml
vr-tv2.xml	vr-tv9.xml	vr-tv16.xml
vr-tv2-err-en.xml	vr-tv9-err-en.xml	vr-tv16-err-en.xml
vr-tv2-lab-en.xml	vr-tv9-lab-en.xml	vr-tv16-lab-en.xml
vr-tv3.xml	vr-tv10.xml	vr-tv17.xml
vr-tv3-err-en.xml	vr-tv10-err-en.xml	vr-tv17-err-en.xml
vr-tv3-lab-en.xml	vr-tv10-lab-en.xml	vr-tv17-lab-en.xml
vr-tv4.xml	vr-tv11.xml	vr-tv18.xml
vr-tv4-err-en.xml	vr-tv11-err-en.xml	vr-tv18-err-en.xml
vr-tv4-lab-en.xml	vr-tv11-lab-en.xml	vr-tv18-lab-en.xml
vr-tv5.xml	vr-tv12.xml	vr-tv19.xml
vr-tv5-err-en.xml	vr-tv12-err-en.xml	vr-tv19-err-en.xml
vr-tv5-lab-en.xml	vr-tv12-lab-en.xml	vr-tv19-lab-en.xml
vr-tv6.xml	vr-tv13.xml	vr-tv20.xml
vr-tv6-err-en.xml	vr-tv13-err-en.xml	vr-tv20-err-en.xml
vr-tv6-lab-en.xml	vr-tv13-lab-en.xml	vr-tv20-lab-en.xml
vr-tv7.xml	vr-tv14.xml	vr-tv21.xml
vr-tv7-err-en.xml	vr-tv14-err-en.xml	vr-tv21-err-en.xml
vr-tv7-lab-en.xml	vr-tv14-lab-en.xml	vr-tv21-lab-en.xml

Currency of facts

- Reporting currency declared on a fact for s2md_met:ei1930 item (http://eiopa.europa.eu/xbrl/s2md/dict/met)

```
<xbrli:context id="c">
<xbrli:entity>
  <xbrli:identifier scheme="http://standards.iso.org/iso/17442">…</xbrli:identifier>
</xbrli:entity>
<xbrli:period>
  <xbrli:instant>2015-12-31</xbrli:instant>
</xbrli:period>
</xbrli:context>
<s2md_met:ei1930 contextRef="c">s2c_CU:EUR</s2md_met:ei1930>
```

- in general all monetary facts in an instance document must refer to this currency xbrli:measure of xbrli:unit (here with iso4217: prefix):

```
<xbrli:unit id="uEUR">
  <xbrli:measure>iso4217:EUR</xbrli:measure>
</xbrli:unit>
```

- if a fact contains a s2c_dim:AF (http://eiopa.europa.eu/xbrl/s2c/dict/dim) dimension with s2c_CA:x1 (http://eiopa.europa.eu/xbrl/s2c/dict/dom/CA) member in context and additionally s2c_dim:OC dimension then a currency in xbrli:unit must match the currency declared for OC dimension

```
<xbrli:unit id="uPLN">
  <xbrli:measure>iso4217:PLN</xbrli:measure>
</xbrli:unit>

<xbrli:context id="c">
<xbrli:entity>
  <xbrli:identifier scheme="http://standards.iso.org/iso/17442">…</xbrli:identifier>
</xbrli:entity>
<xbrli:period>
  <xbrli:instant>2015-12-31</xbrli:instant>
</xbrli:period>
  <xbrli:scenario>
    <xbrldi:explicitMember dimension="s2c_dim:AF">s2c_CA:x1</xbrldi:explicitMember>
    <xbrldi:explicitMember dimension="s2c_dim:OC">s2c_CU:PLN</xbrldi:explicitMember>
(…)
  </xbrli:scenario>
</xbrli:context>

<s2md_met:mi84 contextRef="c" decimals="0" unitRef="uPLN">400</s2md_met:mi84>
```
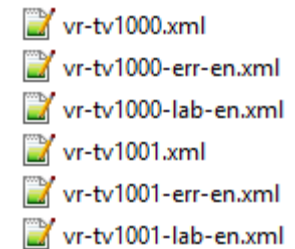
- s2c_dim:AF with s2c_CA:x1 in context may also appear with no s2c_dim:OC dimension, in which case the general rule applies (i.e. currency on xbrli:unit must match the reporting currency)

Currency related validations in 2.2.0 PWD

- implemented by means of two technical validations
  - TV1000:
    - $a: all monetary facts that have x0 (default) value for AF dimension
    - $b: s2md_met:ei1930 (Reporting currency enumerated metric)
    - test: local-name-from-QName(xfi:measure-name(xfi:unit-numerator(xfi:unit($a)))) eq local-name-from-QName($b)
    - error: There is at least one monetary fact reported in reporting currency not matching basic information data
  - TV1001:
    - $a: all monetary facts that have x1 value for AF dimension
    - test: local-name-from-QName(xfi:fact-explicit-dimension-value($a ,QName('http://eiopa.europa.eu/xbrl/s2c/dict/dim','OC'))) eq local-name-from-QName(xfi:measure-name(xfi:unit-numerator(xfi:unit($a))))

- concerns:
  - simple test but checked for every single monetary fact results in large number of evaluations
  - TV1000 may be profiled for special treatment of s2md_met:ei1930

vr-tv1000.xml
vr-tv1000-err-en.xml
vr-tv1000-lab-en.xml
vr-tv1001.xml
vr-tv1001-err-en.xml
vr-tv1001-lab-en.xml

SR.01.01.01.01
Appendix I: Quantitative reporting templates

| Ring-fenced fund/matching portfolio/remaining part | Z0010 | s2c_PU:x60 (Ring fenced fund or Matching Adjustment Portfolio) |
| Fund/Portfolio Number | Z0020 | RFF0001 |

Content of the submission

		C0010
Template Code - Template name		
SR.02.01.01 - Balance sheet	R0790	s2c_CN:x1 (Repoted)
SR.12.01.01 - Life and Health SLT Technical Provisions	R0800	s2c_CN:x1 (Repoted)
SR.17.01.01 - Non-Life Technical Provisions	R0810	s2c_CN:x1 (Repoted)

OK

SR.01.01.01.01
Appendix I: Quantitative reporting templates

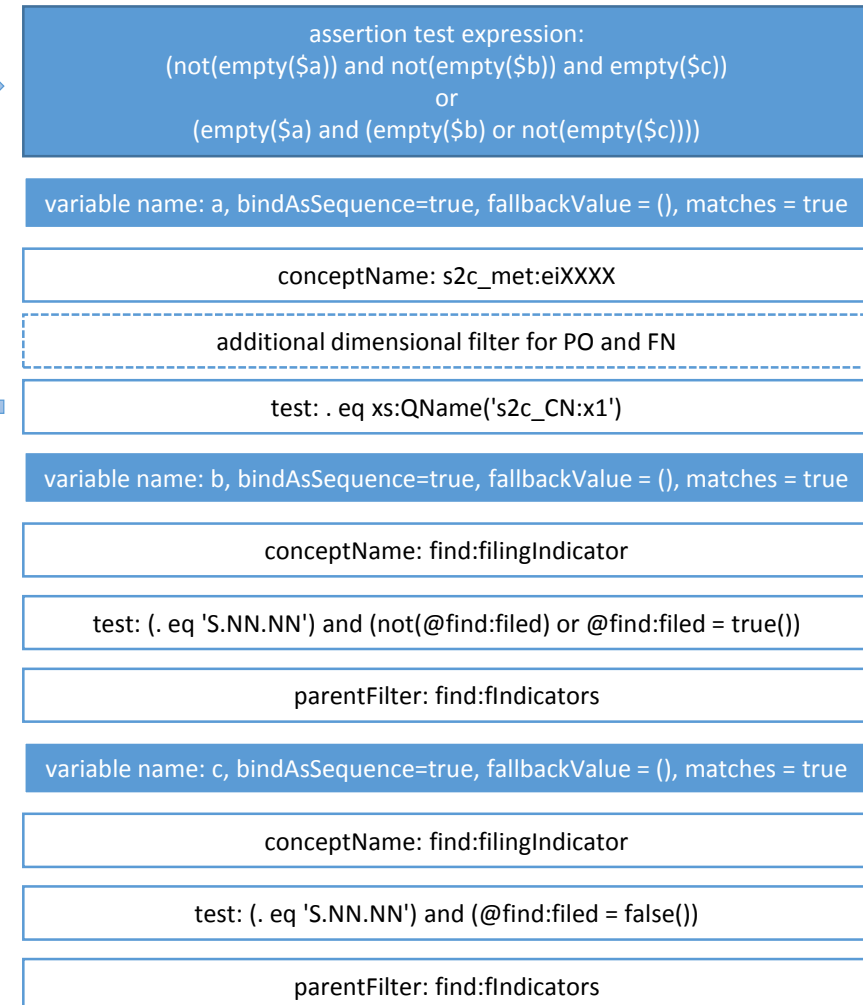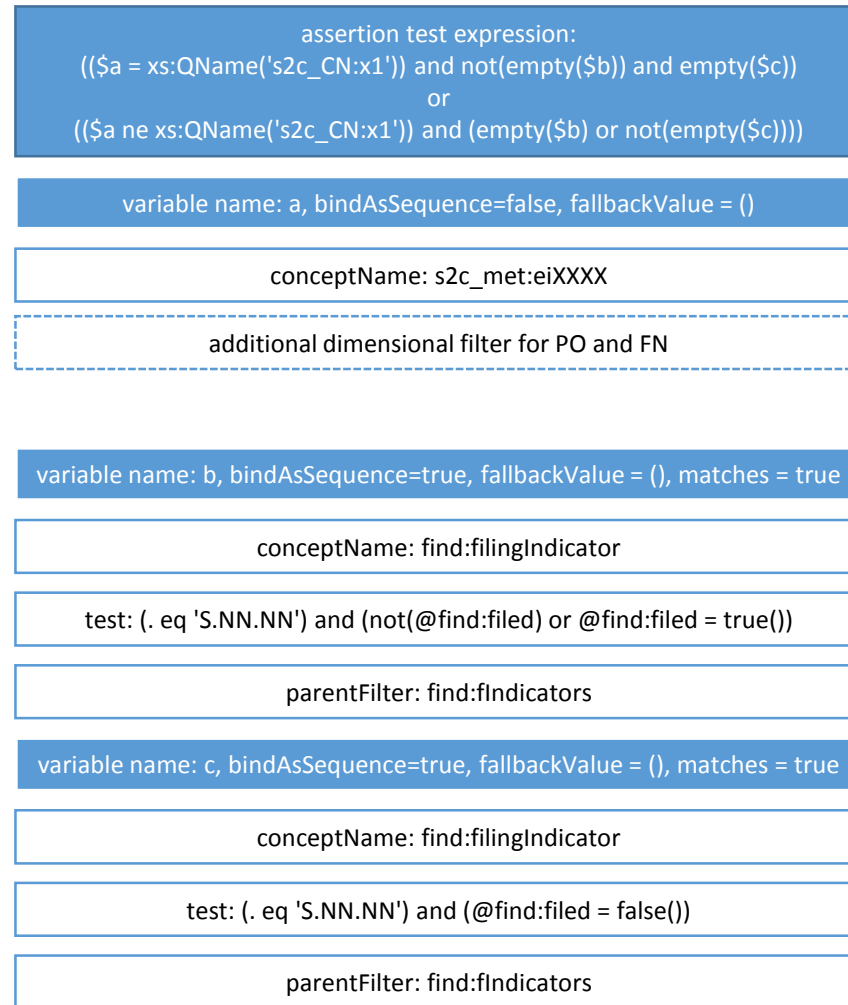| Ring-fenced fund/matching portfolio/remaining part | Z0010 | s2c_PU:x60 (Ring fenced fund or Matching Adjustment Portfolio) |
| Fund/Portfolio Number | Z0020 | MAP1111 |

Content of the submission

		C0010
Template Code - Template name		
SR.02.01.01 - Balance sheet	R0790	s2c_CN:x2 (Not repoted other reason)
SR.12.01.01 - Life and Health SLT Technical Provisions	R0800	s2c_CN:x1 (Repoted)
SR.17.01.01 - Non-Life Technical Provisions	R0810	s2c_CN:x1 (Repoted)

NOT OK

Filing indicators (only one per template):

```
<find:fIndicators>
  (...)
  <find:filingIndicator contextRef="c">SR.02.01</find:filingIndicator>
  <find:filingIndicator contextRef="c">SR.12.01</find:filingIndicator>
  <find:filingIndicator contextRef="c">SR.17.01</find:filingIndicator>
  (...)
</find:fIndicators>
```

- List of known issues #28 (BZ711) has been only partially fixed in 2.1.0 as identified in issue #102.
  - The part that has been fixed it triggering of validations for SR templates only when a template is marked as reported in the Content Template for a given RFF/MAP.
  - What has not been fixed is a check between Content Template and filing indicators (TV0 rules). Filing indicator must be reported for a SR template when this template is reported for at least one RFF/MAP. But for other RFF/MAP this template may be unreported which currently results in unsatisfied assertion and an ERROR. Therefore, validations vr-tv0-169 to vr-tv0-197 need to be deactivated due to technical reason.

- Solution in 2.2.0 PWD:
  - Redefine the TV0 checks by adding a general filter on the variable (e.g. v1) linking to the content template fact with a test checking if selected option is "Reported" (e.g. ". eq xs:QName('s2c_CN:x1')") and applying empty($v1) and not(empty($v1)) in the test expression (rather than current "$v1 eq xs:QName('s2c_CN:x1')" or "$v1 ne xs:QName('s2c_CN:x1')" respectively. Variable v1 would be bound as sequence, have matches="true" and link (optionally) to FN dimension.
  - As a result, the assertion would check if the filing indicator is reported for an SR template when at least one RFF/MAP has it marked as reported in the content template (and vice versa).

2.1.0 (deactivated for SR content templates)

2.2.0

assertion test expression:
(($a = xs:QName('s2c_CN:x1')) and not(empty($b)) and empty($c))
or
(($a ne xs:QName('s2c_CN:x1')) and (empty($b) or not(empty($c))))

assertion test expression:
(not(empty($a)) and not(empty($b)) and empty($c))
or
(empty($a) and (empty($b) or not(empty($c))))

variable name: a, bindAsSequence=false, fallbackValue = ()

variable name: a, bindAsSequence=true, fallbackValue = (), matches = true

conceptName: s2c_met:eiXXXX

conceptName: s2c_met:eiXXXX

additional dimensional filter for PO and FN

additional dimensional filter for PO and FN

test: . eq xs:QName('s2c_CN:x1')

variable name: b, bindAsSequence=true, fallbackValue = (), matches = true

variable name: b, bindAsSequence=true, fallbackValue = (), matches = true

conceptName: find:filingIndicator

conceptName: find:filingIndicator

test: (. eq 'S.NN.NN') and (not(@find:filed) or @find:filed = true())

test: (. eq 'S.NN.NN') and (not(@find:filed) or @find:filed = true())

parentFilter: find:fIndicators

parentFilter: find:fIndicators

variable name: c, bindAsSequence=true, fallbackValue = (), matches = true

variable name: c, bindAsSequence=true, fallbackValue = (), matches = true

conceptName: find:filingIndicator

conceptName: find:filingIndicator

test: (. eq 'S.NN.NN') and (@find:filed = false())

test: (. eq 'S.NN.NN') and (@find:filed = false())

parentFilter: find:fIndicators

parentFilter: find:fIndicators

- New TV0 checks if filing indicator is present when at least one RFF/MAP has it marked as reported or is missing when:
  - NO RFF/MAP has it marked as reported (this would work fine combined with the rule that all entries in the content template must be reported)
  - All RFF/MAP have it marked as not reported (requires one more variable with test . ne xs:QName('s2c_CN:x1') and change in test expression)

- In 2.1.0: 16.02.2017 – Technical Deactivation - Forces the SR content template to be reported even when an undertaking has no RFFs/MAPs

- in 2.2.0 PWD: If {S.01.02, r0150, c0010}=[s2c_PU:x4] or {S.01.02, r0170, c0010}=[s2c_PU:x51] then {SR.01.01, rNNN, c0010}<>empty

if Basic Information identifies that RFFs or MAPs are included in the report then SR template must be reported (at least once)

possible additional check: count number of all enumerated facts reported for each RFF/MAP and check if the number matches the number of rows in a content template

eiopa

- Extension of List of known issues #28 problem: SR templates can be reported multiple times for various RFFs, MAPs or remaining. However for each template there is only one filing indicator (technically, as item in a tuple). Therefore filing indicators are not proper representation of a report content with regard to the SR templates (which may be reported for one RFF/MAP but no necessarily for the other). As a result, in their current technical representation, filing indicators for SR templates cannot be used in preconditions on validation rules (which may depend on a specific RFF/MAP and are not generic for all RFFs/MAPs). Moreover a set of technical validations between the content template and filing indicators must be disabled (as currently it checks each filing indicator value against multiple entries in the content template that may mark template as reported for one RFF/MAP but as not reported for the other).

- Solution applied in 2.1.0: use content template entries as preconditions on assertions.
  - 2.0.1: {S.02.01, r0320,c0010}={S.12.01, r0020,c0210}+{S.12.01, r0080,c0210}
  - 2.1.0: If ({SR.01.01, r0790,c0010}=[s2c_CN:x1] and {SR.01.01, r0800,c0010}=[s2c_CN:x1]) and

    ((({SR.01.01, z0010}=[s2c_PU:x60] and {SR.02.01, z0020}=[s2c_PU:x40]

    and {SR.12.01, z0010}=[s2c_PU:x60]) or {SR.01.01, z0010}={SR.02.01, z0020}={SR.12.01, z0010})

    and {SR.01.01, z0020}={SR.02.01, z0030}={SR.12.01, z0020} then

    {SR.02.01, r0320,c0010}={SR.12.01, r0020,c0210}+{SR.12.01, r0080,c0210}

- Further problem: some rule must be evaluated only for a reported value of Article 112 as otherwise the rule may find at least one data point matching any variable from the expression reported in another table for the standard formula and the rule results in false evaluation e.g.
  - rule: {S.26.02, r0330, c0080}={S.26.02, r0400, c0080}-({S.26.02, r0100, c0080}+{S.26.02, r0300, c0080})
  - test: $a = $b - ($c + $d)
  - situation: S.26.02 is reported for Article 112
  - result: the rule is evaluated for Article 112
  - problem:
    – {S.26.02, r0400, c0080} is the same data point as {S.25.01, r0020, c0040}
    – S.25.01 is reported in standard formula with value „1000"
    – evaluation for this rule in standard formula:
      - 0 = 1000 – (0 +0)
      - assertion unsatisfied ERROR

SR.01.01.01.01
Appendix I: Quantitative reporting templates

| Ring-fenced fund/matching portfolio/remaining part | Z0010 | s2c_PU:x60 (Ring fenced fund or Matching Adjustment Portfolio) |
| Fund/Portfolio Number | Z0020 | RFF0001 |

Content of the submission

Template Code - Template name		C0010
SR.02.01.01 - Balance sheet	R0790	s2c_CN:x1 (Repoted)
SR.12.01.01 - Life and Health SLT Technical Provisions	R0800	s2c_CN:x1 (Repoted)
SR.17.01.01 - Non-Life Technical Provisions	R0810	s2c_CN:x1 (Repoted)

SR.01.01.01.01
Appendix I: Quantitative reporting templates

| Ring-fenced fund/matching portfolio/remaining part | Z0010 | s2c_PU:x60 (Ring fenced fund or Matching Adjustment Portfolio) |
| Fund/Portfolio Number | Z0020 | MAP1111 |

Content of the submission

Template Code - Template name		C0010
SR.02.01.01 - Balance sheet	R0790	s2c_CN:x2 (Not repoted other reason)
SR.12.01.01 - Life and Health SLT Technical Provisions	R0800	s2c_CN:x1 (Repoted)
SR.17.01.01 - Non-Life Technical Provisions	R0810	s2c_CN:x1 (Repoted)

Standard formula and Article 112 solution in 2.2.0 PWD

- content template dropdowns identify the reason for reporting separately for Article 112, e.g for „ S.26.02.01 - Solvency Capital Requirement - Counterparty default risk".

    1 – Reported [x1]

    2 - Not reported as risk not existent [x13]

    8 - Not reported as use of partial internal model [x12]

    9 - Not reported as use of full internal model [x11]

    11 - Not reported as reported at RFF/MAP level [x41]

    16 - Reported due to request of Article 112 of Directive 2009/138/EC [x60]

    17 - Reported twice due to use of PIM [x71]

    0 - Not reported other reason (in this case special justification is needed) [x2]

- technically the validation is split with appropriate filters:

    - If {S.01.01, r0510,c0010}=[s2c_CN:x1] or {S.01.01, r0510,c0010}=[s2c_CN:x71] then {S.26.02, r0330, c0080}={S.26.02, r0400, c0080}-({S.26.02, r0100, c0080}+{S.26.02, r0300, c0080}) for facts with standard formula

    - If {S.01.01, r0510,c0010}=[s2c_CN:x60] or {S.01.01, r0510,c0010}=[s2c_CN:x71] then {S.26.02, r0330, c0080}={S.26.02, r0400, c0080}-({S.26.02, r0100, c0080}+{S.26.02, r0300, c0080}) for facts with Article 112

Different severity for regular and non-regular reporting

- Basic information template identifies in R0010 if submission is Regular or ad-hoc:
  - 1 - Regular reporting [x35]
  - 2 - Ad-hoc reporting [x36]
  - 3 – Re-submission of S.30 templates in accordance with instructions of the template [x44]
  - 4 – Empty submission [x45]

- severity
  - for Regular reporting most of the rules (apart from those classified as non-blocking) are classified as ERROR by assertion severity
  - in case of non regular reporting all rules must be classified as WARNING by assertion severity

- this has been implemented by:
  - duplicating each Blocking rule with WARNING severity (with _w suffix) and their assignment in assertion sets
  - inclusion of two parameters in find-params.xml (val folder, where filing indicator parameters are defined)
    - `<variable:parameter xlink:type="resource" xlink:label="regularReporting" name="regularReporting" select="(s2md_met:ei1677 = xs:QName('s2c_CS:x35')) or (s2md_met:ei2503 = xs:QName('s2c_CS:x35'))" as="xs:boolean" id="regularReporting" />`
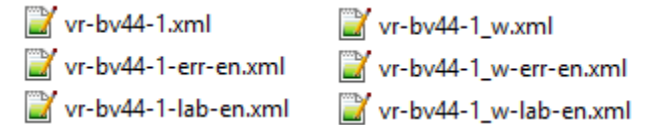   - `<variable:parameter xlink:type="resource" xlink:label="nonRegularReporting" name="nonRegularReporting" select="(s2md_met:ei1677 != xs:QName('s2c_CS:x35')) or (s2md_met:ei2503 != xs:QName('s2c_CS:x35'))" as="xs:boolean" id="nonRegularReporting" />`
  - use of the parameters as preconditions in find-prec.xml files for each module and rule:
    - `<variable:precondition xlink:type="resource" xlink:label="rp" test="$regularReporting"/>`
    - `<variable:precondition xlink:type="resource" xlink:label="nrp" test="$nonRegularReporting"/>`
    - `<gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition" xlink:from="loc_s2md_BV104-3" xlink:to="rp"/>`
    - `<gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition" xlink:from="loc_s2md_BV104-3_W" xlink:to="nrp"/>`

- concerns:
  - for performance reasons processors shall be profiled to treat there two new parameters in a similar manner as the filing indicator parameters

vr-bv44-1.xml   vr-bv44-1_w.xml
vr-bv44-1-err-en.xml   vr-bv44-1_w-err-en.xml
vr-bv44-1-lab-en.xml   vr-bv44-1_w-lab-en.xml

Deactivating assertions by means of XBRL

- solution as explained in http://eurofiling.info/portal/taxonomiesmechxml-blacklist/

- each module containing validations is associated with separate ignore-val.xml linkbase which defined a precondition:

  <variable:precondition xlink:type="resource" xlink:label="ignore" test="false()" />

- mechanism for deactivation:

  <link:loc xlink:type="locator" xlink:href="../val/vr-bv574_1-10.xml#s2md_BV574_1-10" xlink:label="loc_s2md_BV574_1-10"/>

  <gen:arc xlink:type="arc" xlink:arcrole="http://xbrl.org/arcrole/2008/variable-set-precondition" xlink:from="loc_s2md_BV574_1-10" xlink:to="ignore" />

- ignore-val.xml will be released with publication of lists of deactivations and may be replaced in taxonomy „mod" folder

Technical table

- introduced to enable reporting of data where two or more cells are wrongly classified as the same data point

- referenced by every module

T.99.01.01
Technical table

T.99.01.01.01
Technical table

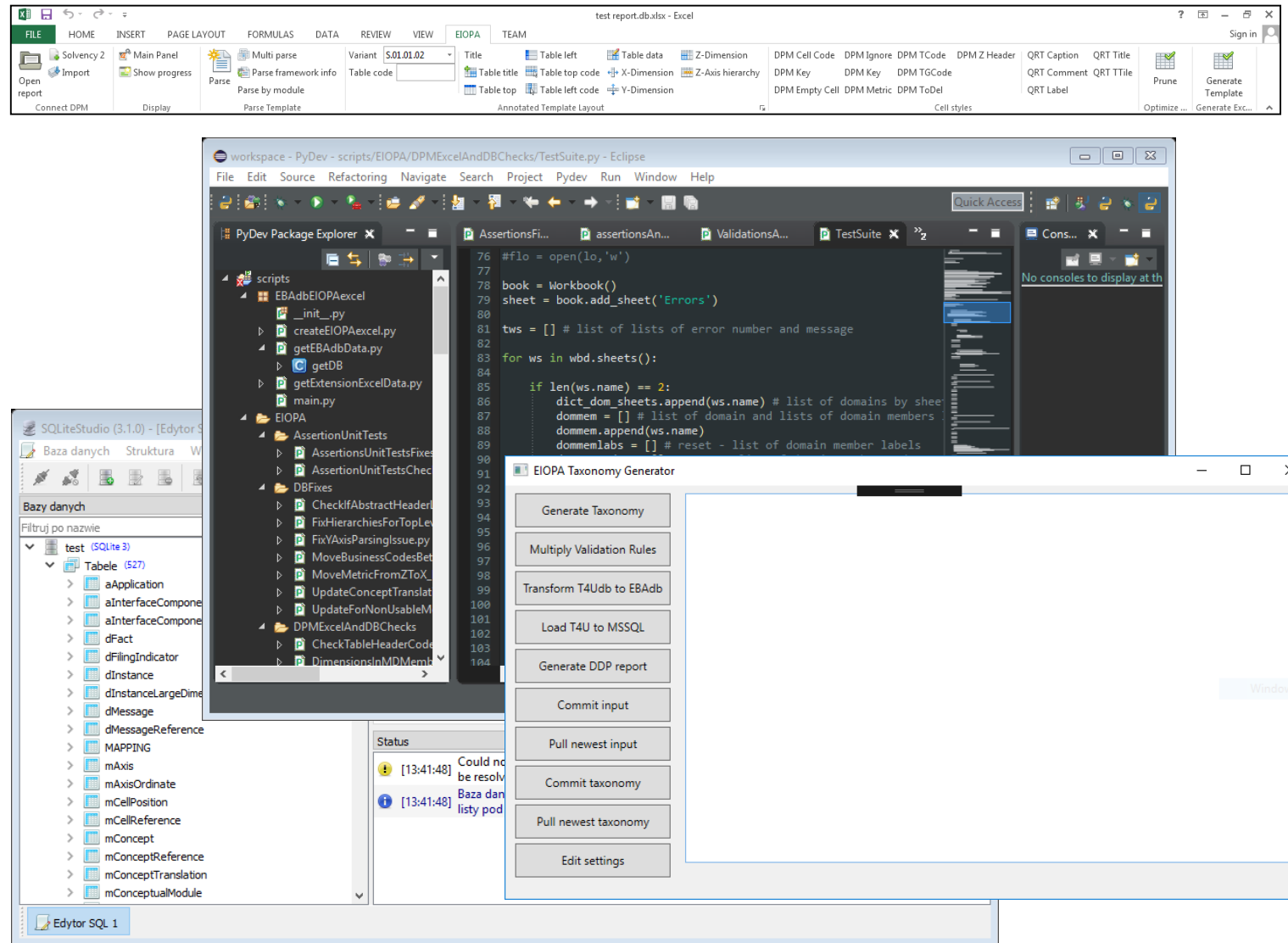| Table	X axis	Y axis	Z axis	Comment	Monetary	String	Date	Integer	Decimal	Pure	Boolean
C0010	C0020	C0030	C0040	C0050	C0060	C0070	C0080	C0090	C0100	C0110	C0120
•artificial key•|"mandatory" YM: T.99.01.01.01 line identification (Table)	•artificial key•|"optional" YN: T.99.01.01.01 line identification (X axis)	•artificial key•|"optional" YO: T.99.01.01.01 line identification (Y axis)	•artificial key•|"optional" YR: T.99.01.01.01 line identification (Z axis)	Metric: String|TS/Comment	Metric: Monetary	Metric: String	Metric: Date	Metric: Integer	Metric: Decimal	Metric: Pure	Metric: Boolean

Testing

- MS Excel
  - DPM Dictionary and Annotated Templates
    - 40 tests: repeated codes or labels, undefined/undeclared items, additional spaces or bad indentation, wrong value for a column, text outside named ranges, …
    - T4S parsing routines logs errors when not able to extract data
  - Validations:
    - Parsing to a format consumable by the DPM Architect engine – potential issues as Excel comments to each rule

- Database
  - Set of queries to check correctness (e.g. leaf ordinate missing R/C code)
  - Generate instance documents and test against the taxonomy
  - Testing in T4U:
    - Templates (visually)
    - Validations (SQL scripts with unit test of each rule)
  - Diff with previous version (MS Excel)

- Taxonomy
  - Validation with multiple tools,
  - Diff against last version (dictionary),
  - Unit tests for assertions,
  - Scrips checking „suspicious" things (e.g. some nuances of enumerations, rendering, assertions, etc.)

Thank you!