

Updates to Formula Specifications

Multi-instance and Chaining

Herm Fischer
Formula Working Group

XBRL XXV

HOSTED BY XBRL JAPAN

TO THE NEXT LEVEL OF BUSINESS REPORTING. AND BEYOND.

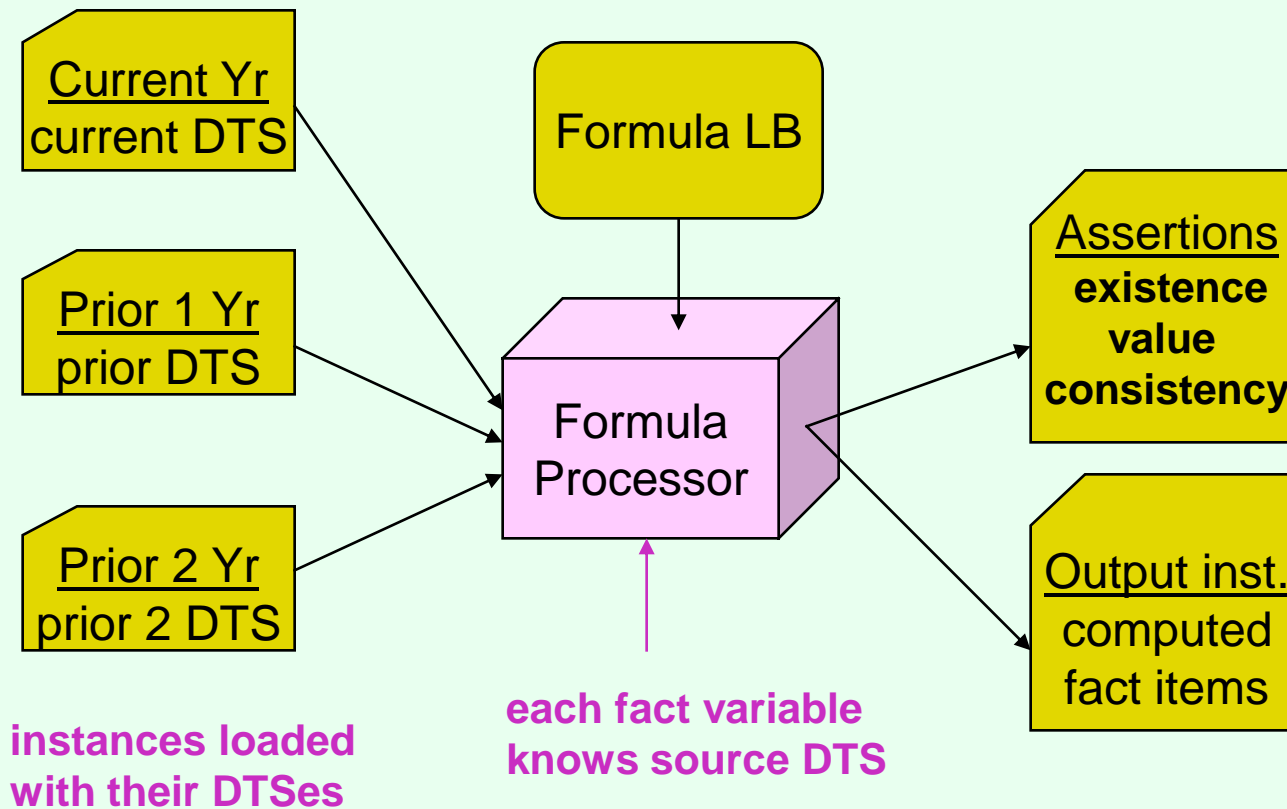
Handling multiple instances

- Issues to merge instances
 - Taxonomies differ?
 - Concepts change with changes in law, practice
 - Dimensions change
 - Tree relationship in presentation, definition change
 - Namespaces change
 - ContextRef's will be changed in merging
 - E.g., current-yr-consol, prior-yr-consol
 - May be constraints on altering contextRef's

Multi-instance Formulas

- Each instance loaded with its taxonomy
- Formula terms refer to nodes, which know their enclosing document
- Schemas and linkbases kept separate

Multi-instance formula inputs



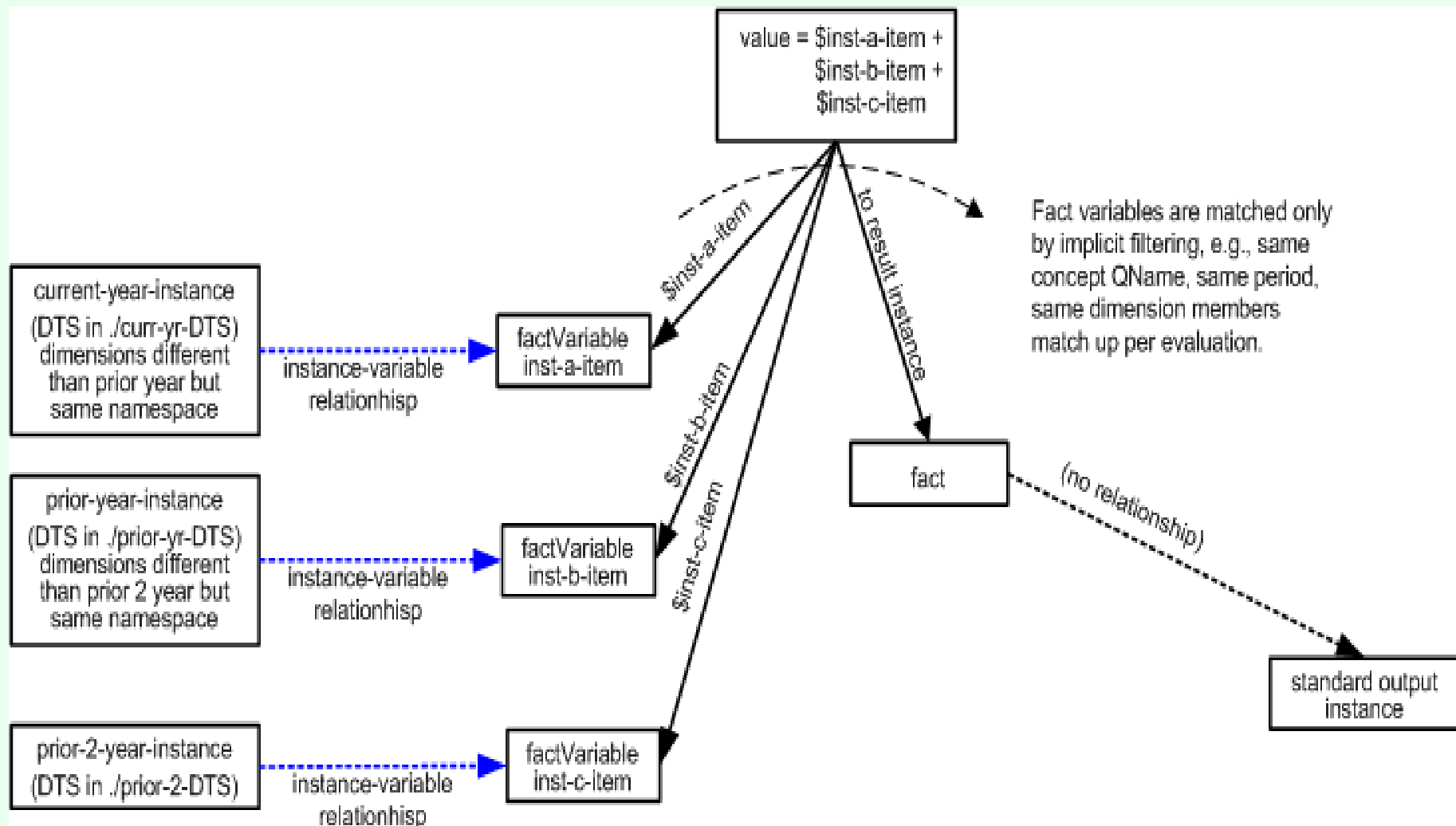
Multiple instances

- Multiple entity instances
 - Same period but different entities
 - Different company extension taxonomies
- Multiple period instances
 - Taxonomies change
 - Namespaces change
 - Linkbases and dimension aggregations change
- Multiple types of reports
 - Different taxonomies for each

Example with 3 input instances

- There are three input instances, representing
 - a current year,
 - a prior year, and
 - the second prior year
- Each instances has different DTSes
 - different member hierarchies in dimension
 - namespaces are the same
 - all aspects, including concept and dimension QNames can be matched up.

Model of 3-input instances



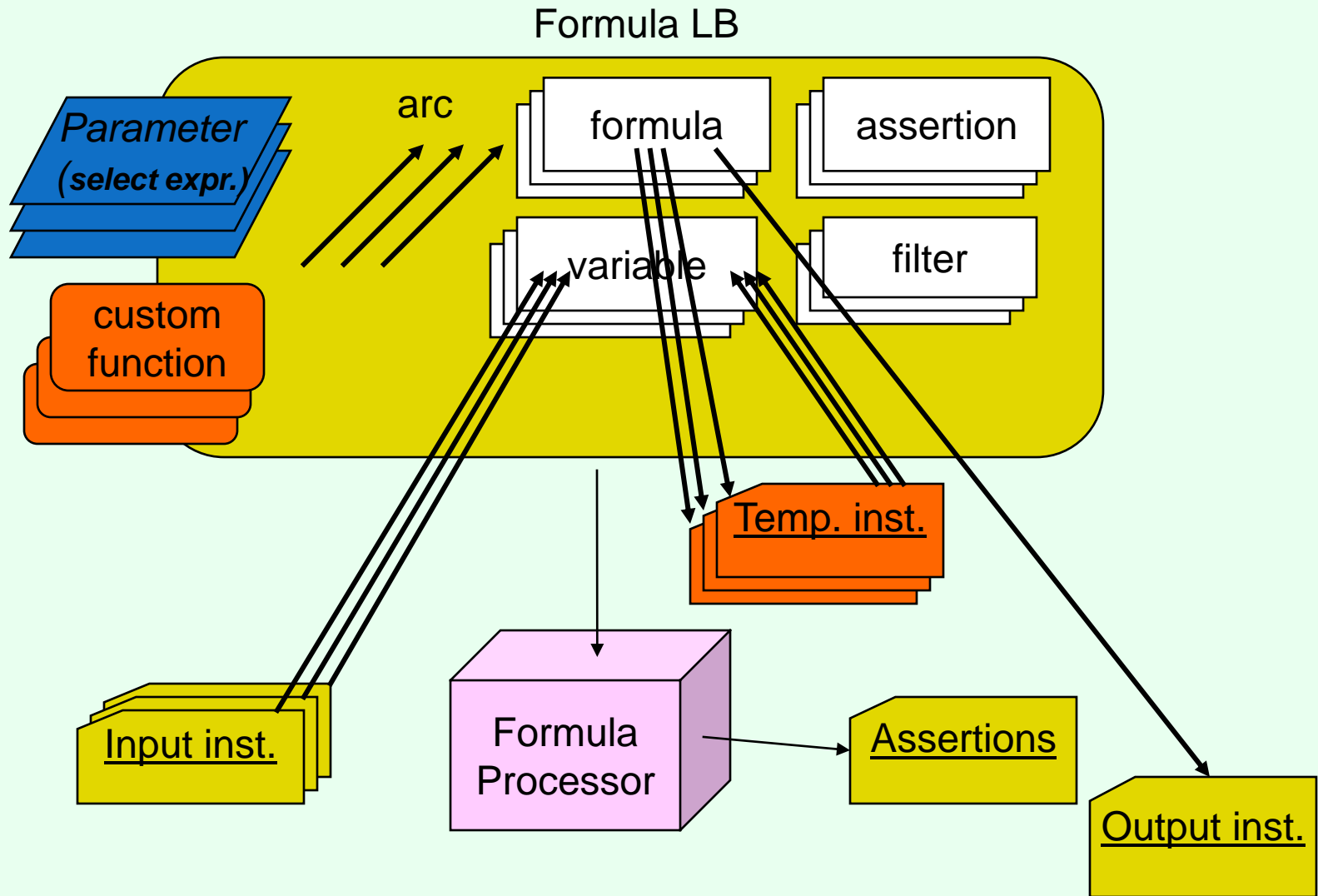
Linkbase Syntax	Meaning
<pre><instance:instance name="i:inst-b" xlink:type="resource" xlink:label="inst-b"/></pre>	<p>Input instance B is the second input instance</p>
<pre><instance:instance name="i:inst-c" xlink:type="resource" xlink:label="inst-c"/></pre>	<p>Input instance C is the third input instance</p>
<pre><formula:formula ... value="\$inst-a-item + \$inst-b-item + \$inst-c-item" source="inst-a-item" > <formula:decimals> 0</formula:decimals> </formula:formula></pre>	<p>The formula produces an output for each evaluation of implicitly-matched factVariable items</p>

Linkbase Syntax	Meaning
<pre><variable:factVariable xlink:type="resource" xlink:label="inst-a-item" bindAsSequence="false"/> <variable:variableArc xlink:type="arc" xlink:arcrole="...variable-set" name="inst-a-item" xlink:from="formula1" xlink:to="inst-a-item" order="1.0"/></pre>	<p>Fact variable inst-a-item bounds to facts from the standard input instance.</p>
<pre><variable:factVariable ... xlink:label="inst-b-item" /> <generic:arc xlink:type="arc" xlink:arcrole="... nstance-variable" xlink:from="inst-b" xlink:to="inst-b-item" order="1.0"/> <variable:variableArc xlink:type="arc" xlink:arcrole="...variable-set" name="inst-b-item" xlink:from="formula1" xlink:to="inst-b-item" order="1.0"/></pre>	<p>Fact variable inst-b-item bounds to facts from input instance B.</p>

Multi-instance solves chaining

- A simple approach to chaining
- Common solution to multi-instance and chaining
- Multi-instances can be ‘scratch-pads’ during computation
- Applies to very large instance solution

Multi-instances for chaining



*Formula LB is pat of DTS

Approach

- Instances are represented by a resource
- instance-variable arc to variable
 - If present, specifies non-default source instance
- formula-instance arc from formula
 - If present specifies the instance to receive fact
- Instance resources are files or temporary

Instance resources

- Could be loaded by processor
 - E.g., java code in a server loads primary instance and some prior-period or other-company instances
 - Or user of GUI adds ‘additional’ instances, such as loading prior-period or other-company instances
- Default implied source and result instances
- Can be temporary in memory only
 - Used for chaining and modularization

Aspect sources, implicit filtering

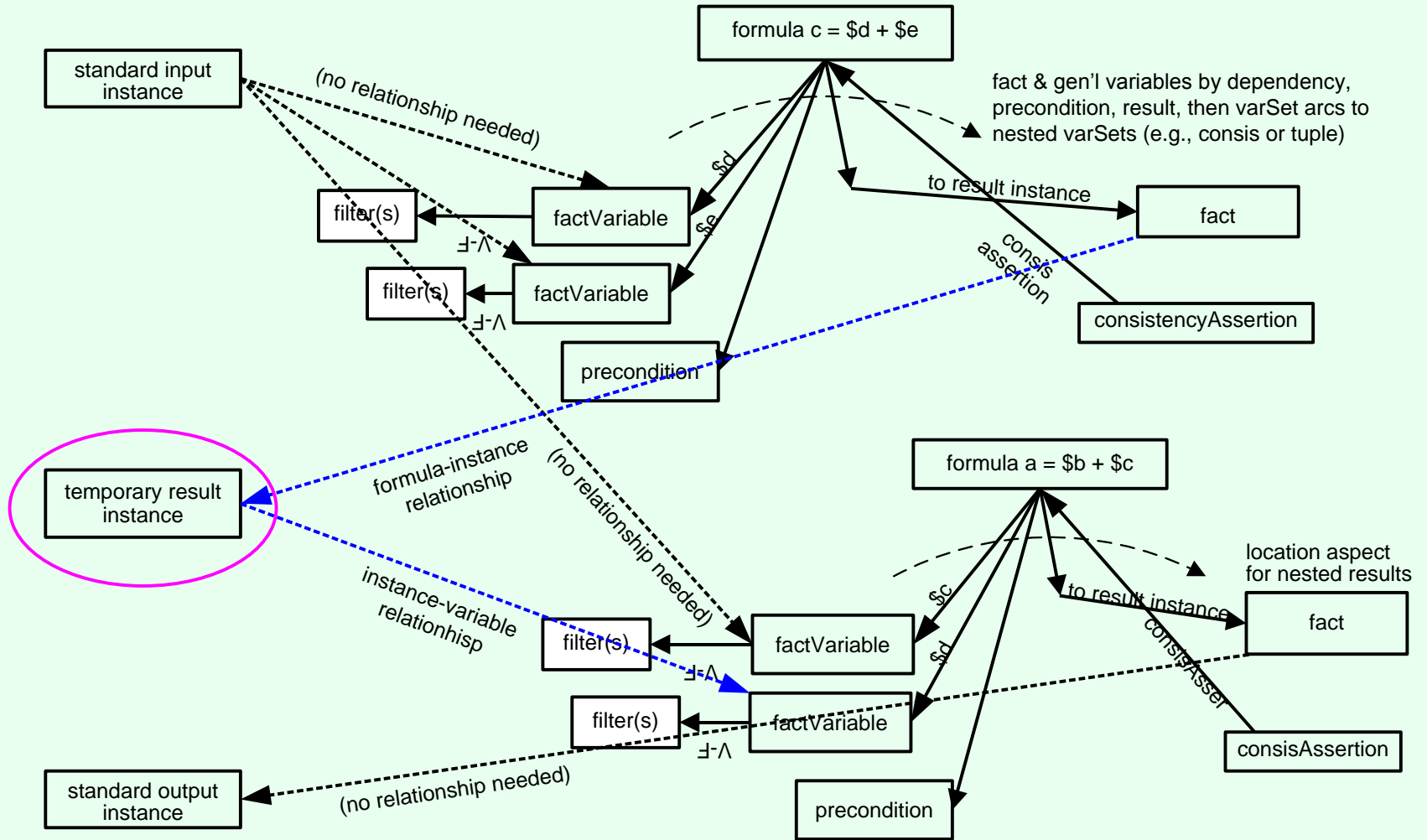
- Formula aspects come from its variables
- Variables from different instances contribute aspects
 - Aspects independent of the instances they come from
 - Aspect “covering” is by-aspect, not by-instance

A=B+C; C=D+E use case (Multi-instance chaining)

- Formula 1 (A=B+C)
 - Result is A, factVariables B & C
 - factVariable B is from source instance (default)
 - factVariable C is from temp instance (has an arc)
- Formula 2 (C=D+E)
 - Result is C, factVariables D & E, to temp instance
 - factVariables D & E are from the source instance

A=B+C; C=D+E (Example 0026 v-01)

Multi-instance chaining



COREP Use case 18: Weighted average of member children

- Weighted average of its dimensional children by another primary item

$$p_1(d_1) = \frac{\sum_i p_1(d_{1,i}) \times p_2(d_{1,i})}{\sum_i p_2(d_{1,i})}$$

Primary items ==>	PD Assigned to the obligor grade	Exposure value	Capital requirements
Total Exposures (dimension)	60%	26.750,00 €	
1 Assigned to obligor grade	27%	1.250,00 €	
Obligor grade 1	10%	100,00 €	
Obligor grade 2	20%	150,00 €	
Obligor grade 3	30%	1.000,00 €	
2 Specialized lending slotting	62%	25.500,00 €	
Risk weight 0%	50%	500,00 €	
Risk weight 10%	75%	20.000,00 €	
Risk weight 150%	10%	5.000,00 €	

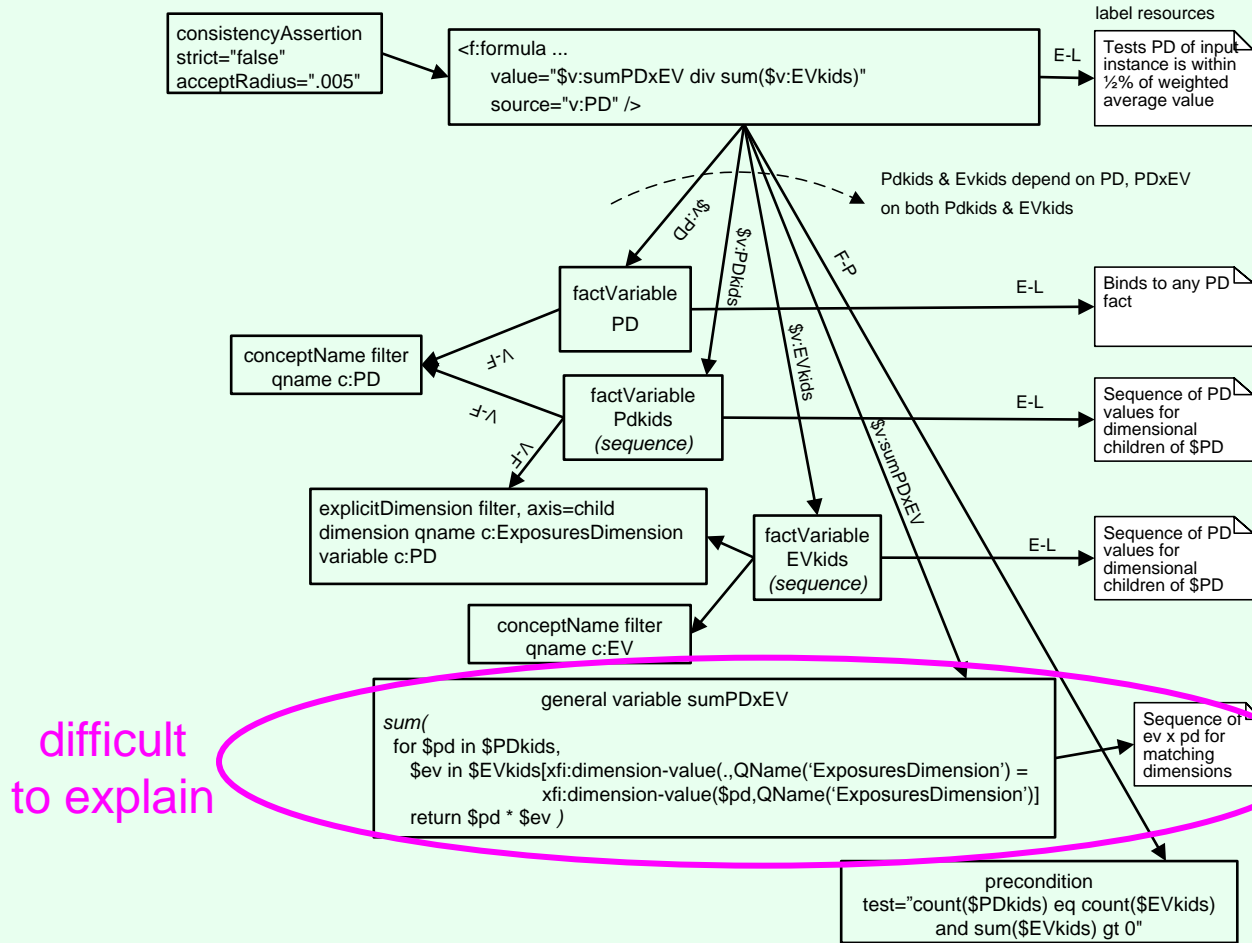
Current single-formula solution

- Excel formulas:

Primary items ==>	PD Assigned to the obligor grade	Exposure value	Capital requirements
Total Exposures (dimension)	$=(B3*C3+B7*C7)/C2$	$=C3+C7$	
1 Assigned to obligor grade	$=(B4*C4+B5*C5+B6*C6)/C3$	$=SUMA(C4:C6)$	
Obligor grade 1	0,1	100	
Obligor grade 2	0,2	150	
Obligor grade 3	0,3	1000	
2 Specialized lending slotting	$=(B8*C8+B9*C9+B10*C10)/C7$	$=SUMA(C8:C10)$	
Risk weight 0%	0,5	500	
Risk weight 10%	0,75	20000	
Risk weight 150%	0,1	5000	

- Make PD controlling fact, get PD and EV of dimensional children
- General variable for PDxEV member matching

Single formula (Example 0017 v-01)



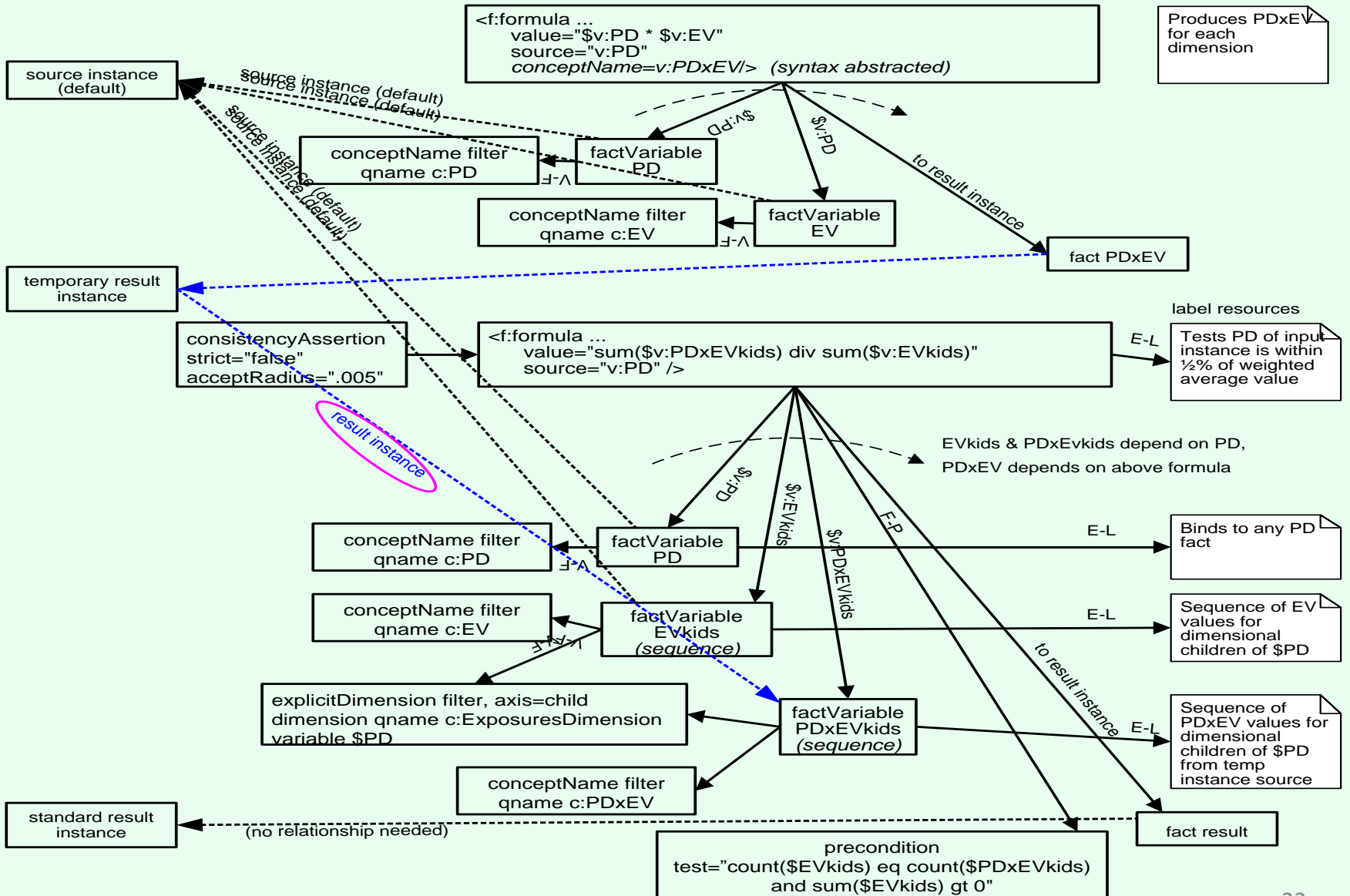
Exposure value formula

- Each PD x EV produced by one formula
 - Result factItem PDxEV is the product for each dimension value
- Second formula binds PDxEV's of dim-children to sequence and EV's of dim-children to second sequence, value assertion checks result

Multiple result instances

- The PDxEV result fact items aren't needed for a real result instance
- Only a value assertion is really needed
- A temporary-results instance in-memory
 - Like a scratch-pad
- Also a temporary facts DTS would be needed (to define the PDxEV result fact item)

Chained formulas (0026 v-20)



Tuple production

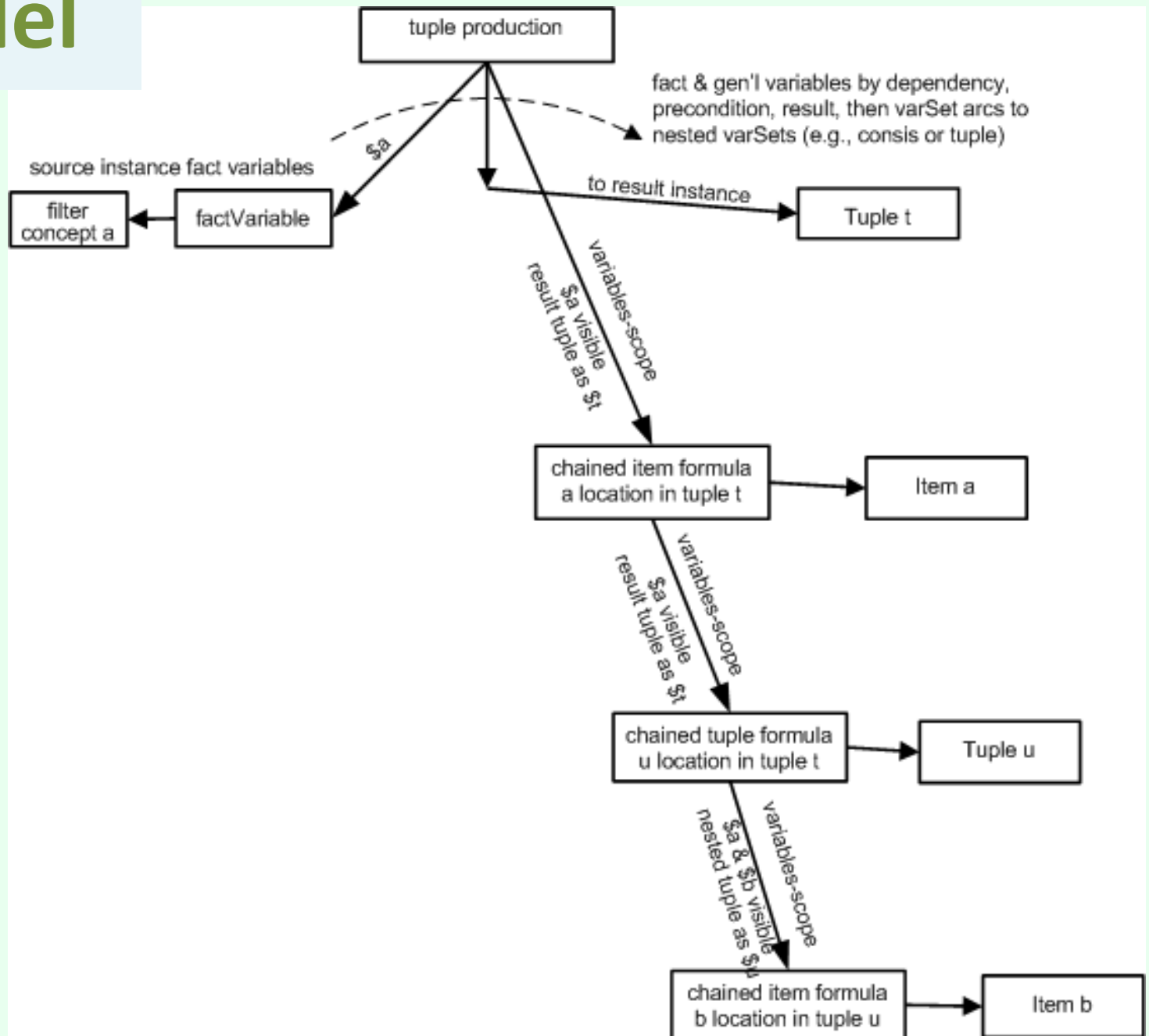
- <tuple:tuple> similar to <formula:formula>
 - No rules for value, accuracy, and item aspects
 - Nested elements have a variables-scope relationship from the parent <tuple:tuple>
 - Child tuple or item uses <tuple:location> rule source attribute, naming parent tuple

Example output tuple

double-nesting of tuples

```
<test:t>  
  <test:a contextRef="c1" unitRef="u1" decimals="0">  
    1001  
  </test:a>  
  <test:u>  
    <test:b contextRef="c1" unitRef="u1" decimals="0">  
      1002  
    </test:b>  
  </test:u>  
</test:t>
```


Model



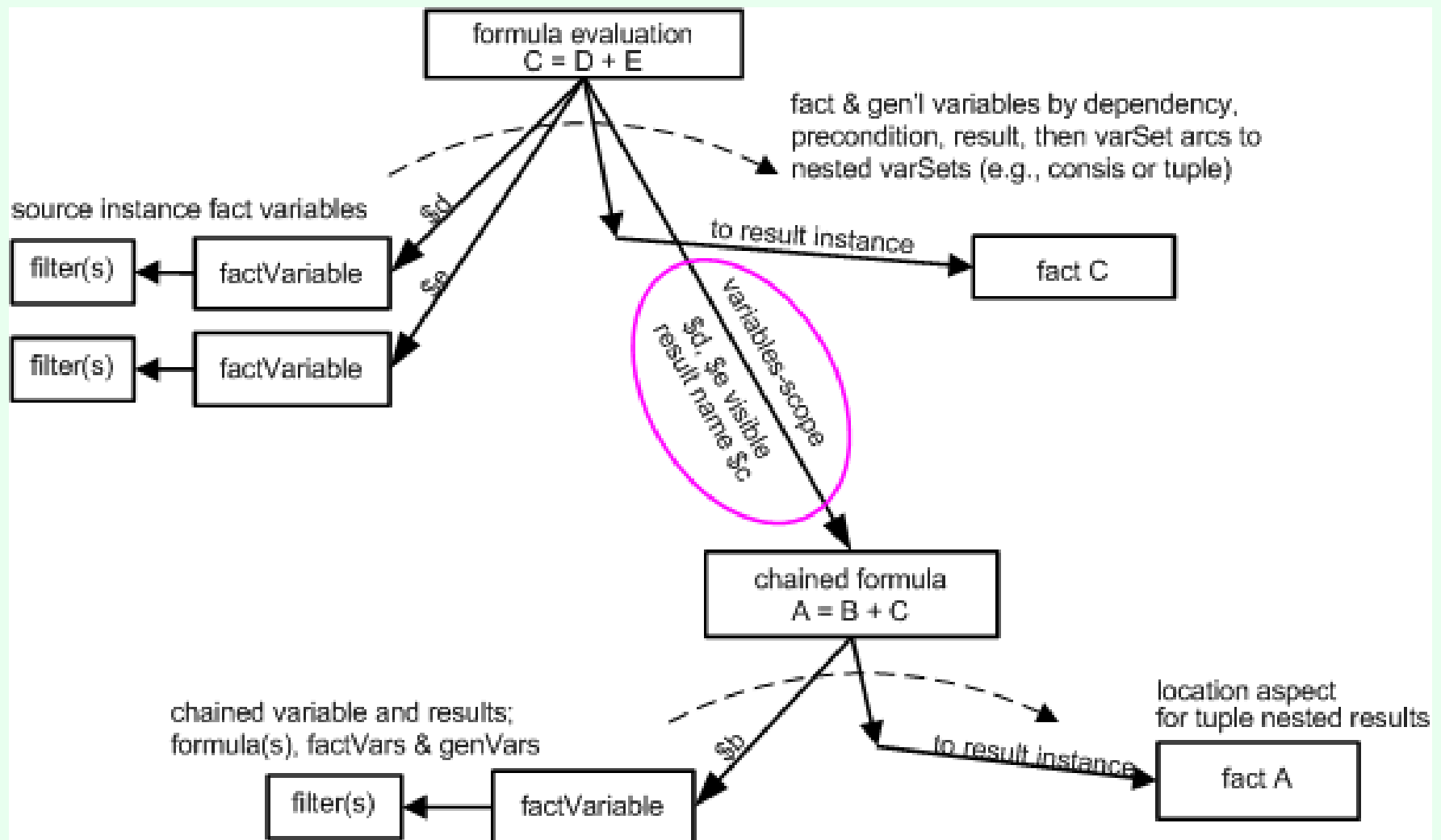
Linkbase Syntax	Meaning
<pre><tuple:tuple ...> <formula:aspects> <formula:concept> <formula:qname> test:t</formula:qname> </formula:concept> </formula:aspects> </tuple:tuple></pre>	(1) tuple t formula
<pre><varsscope:variablesScopeArc xlink:type="arc" xlink:arcrole="...variables-scope" name="outputInstanceTupleT" .../></pre>	Arc from tuple t to nested item A formula (2)
<pre><formula:formula > <formula:decimals> 0</formula:decimals> <formula:aspects> <formula:concept> <formula:qname> test:a</formula:qname> </formula:concept> <tuple:location source="outputInstanceTupleT"/> </formula:aspects> </formula:formula></pre>	(2) formula formulaItemA produces a single fact item <test:a> located inside tuple T.

Linkbase Syntax	Meaning
<pre><varscope:variablesScopeArc ... xlink:from="formulaItemA" xlink:to="formulaTupleU" order="1.0"/></pre>	<p>item A in tuple t visible to formula TupleU (3)</p>
<pre><tuple:tuple ...> <formula:aspects> <formula:concept> <formula:qname> test:u</formula:qname> </formula:concept> <tuple:location source="outputInstanceTupleT"/> </formula:aspects> </tuple:tuple></pre>	<p>Formula (3) formulaTupleU produces nested tuple U located inside tuple T in sequence after item a.</p>
<pre><formula:formula ...> <formula:aspects> <formula:concept> <formula:qname> test:b</formula:qname> </formula:concept> <tuple:location source="outputInstanceTupleU"/> </formula:aspects> </formula:formula></pre>	<p>(4) Formula to produce a single test:b fact item = \$itemA + 1, located inside tuple u.</p>

Chaining with variable scope relationships

- Formula 1 ($A=B+C$):
 - Result is A,
 - factVariables B & C factVariable B is from input XBRL instance factVariable C is variables-scope related to the result of formula (2), C
- Formula 2 ($C=D+E$):
 - Result is C,
 - factVariables D & E, result is variables-scope related to (1), the formula for A

Variables-scope chaining



Linkbase Syntax	Meaning
<pre> <formula:formula ... value="\$d + \$e" source="d" ...> <formula:aspects> <formula:concept> <formula:qname> test:c</formula:qname> </formula:concept> </formula:aspects> </formula:formula> </pre>	<p>(1) formula produces a single test:c result fact item</p>
<pre> <varsscope:variablesScopeArc ... xlink:from="formulaC" xlink:to="formulaA" order="1.0"/> </pre>	<p>formulaC's result (1) visible to formulaA (2)</p>
<pre> <formula:formula ... value="\$b + \$c" source="b" ...> <formula:aspects> <formula:concept> <formula:qname> test:a</formula:qname> </formula:concept> </formula:aspects> </formula:formula> </pre>	<p>(2) $a = b + c$, where the term c is output of formula (1) above</p>

Next Steps

- Sequences with variable scope chaining
 - Set of bound facts chained
 - Emulates behavior of instances chaining
 - Full set of bindings for each variable in sequence
 - Avoids need to specify full-aspect facts
- Large instances topics
 - Approaches to streamed instance processing
 - Constrained XPath feature use
- Future non-XML-DOM instances and processors