# pathogen-identified.xml

from:

<element>  
 <path value="Observation.component" />  
 <slicing>

To:

<element id="Observation.component">  
 <path value="Observation.component" />  
 <slicing>

# infection-condition.xml

from:

<element>  
 <path value="Condition.extension" />  
 <slicing>

To:

<element id="Condition.extension">  
 <path value="Condition.extension" />  
 <slicing>

# hai-surgical-site-infection-report-ssi.xml

from:

<element>  
 <path value="Composition.section" />  
 <slicing>

To:

<element id="Composition.section">  
 <path value="Composition.section" />  
 <slicing>

All value set files:

Remove meta element

# Edited ImplementationGuide file – the order of resources.

# hai-surgical-site-infection-report-ssi

Got differential does not have a slice: Composition.section.entry in profile <http://hl7.org/fhir/us/hai/StructureDefinition/hai-surgical-site-infection-report-ssi>

So did the following (this is something that always has to be done manually, Trifolia can’t do it):

<!-- SG: Commented out and moved pathogen-identified below to make it a simple choice rather than a slice -->  
 <!--<element id="Composition.section:section4.entry">  
 <path value="Composition.section.entry" />  
 <short value="entry" />  
 <definition value="MAY contain [0..3] Pathogen Identified (identifier: http://hl7.org/fhir/us/hai/StructureDefinition/pathogen-identified) (CONF:3347-125)." />  
 <min value="0" />  
 <max value="3" />  
 <type>  
 <code value="Reference" />  
 <targetProfile value="http://hl7.org/fhir/us/hai/StructureDefinition/pathogen-identified" />  
 </type>  
 </element>-->  
 <element id="Composition.section:section4.entry">  
 <path value="Composition.section.entry" />  
 <short value="entry" />  
 <definition value="MAY contain zero or one [0..1] MDRO/CDI Observation (identifier: http://hl7.org/fhir/us/hai/StructureDefinition/mdro-cdi-observation) (CONF:3347-136)." />  
 <min value="0" />  
 <max value="1" />  
 <type>  
 <code value="Reference" />  
 <targetProfile value="http://hl7.org/fhir/us/hai/StructureDefinition/mdro-cdi-observation" />  
 </type>  
 <!-- SG: Moved from commented out element above -->  
 <type>  
 <code value="Reference" />  
 <targetProfile value="http://hl7.org/fhir/us/hai/StructureDefinition/pathogen-identified" />  
 </type>  
 </element>

FHIR Validation Changes (nothing to do with Trifolia)

Observation.sch

Change:

<sch:assert test="not(exists(\*[starts-with(local-name(.), 'value')])) or not(count(for $coding in code/coding return parent::\*/component/code/coding[code/@value=$coding/code/@value and system/@value=$coding/system/@value])=0)">obs-7: If code is the same as a component code then the value element associated with the code SHALL NOT be present</sch:assert>

To: (have raised GForge Issue)

<sch:assert test="not(exists(f:\*[starts-with(local-name(.), 'value')])) or count(for $coding in f:code/f:coding return self::\*/f:component/f:code/f:coding[f:code/@value=$coding/f:code/@value and f:system/@value=$coding/f:system/@value])=0">obs-7: If code is the same as a component code then the value element associated with the code SHALL NOT be present</sch:assert>

Questionnaire.sch

Change:

<sch:rule context="f:Questionnaire/f:item/f:enableWhen">  
 <sch:assert test="count(f:\*[starts-with(local-name(.), 'answer')]|self::f:hasAnswer) = 1">que-7: enableWhen must contain either a 'answer' or a 'hasAnswer' element</sch:assert>  
 </sch:rule>

To (Sean raised GForge issue):

<sch:rule context="f:Questionnaire/f:item/f:enableWhen">  
 <sch:assert test="count(f:\*[starts-with(local-name(.), 'answer')]|f:hasAnswer) = 1">que-7: enableWhen must contain either a 'answer' or a 'hasAnswer' element</sch:assert>  
 </sch:rule>