

**Supplementary Appendix A. Table A1.** OCL-style invariants and their diagnostic role.

Table A1. OCL-style invariants and their diagnostic role.

| <b>T</b> | <b>OCL-style invariant</b>                                                                                                                                                 | <b>Reading</b>                                                                                                                                         |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| T0       | <pre>context LegalEconomicRelator inv T0_HasGrounding: self.groundingEvent- &gt;notEmpty() or self.underlyingObject- &gt;notEmpty() or self.evidence- &gt;notEmpty()</pre> | A reconstructed LER must have grounding support in an event, underlying object, or evidence.                                                           |
| T0       | <pre>context ObjectLER inv T0_ObjectLERDe- pendsOnUnderlyingObject: self.underly- ingObject-&gt;notEmpty()</pre>                                                           | An Object LER depends on an underlying object, whether physical, institutional, or objectified.                                                        |
| T1       | <pre>context ObjectLER inv T1_ObjectLER- Structure: self.underlyingObject- &gt;notEmpty() and self.others-&gt;notEmpty()</pre>                                             | An Object LER is object-grounded and holds with respect to Others; it is not the holder-specific position itself.                                      |
| T1       | <pre>context HoldingLER inv T1_Hold- ingLERStructure: self.objectLER- &gt;notEmpty() and self.holder-&gt;notEmpty()</pre>                                                  | A Holding LER mediates an Object LER and a Holder; it is externally dependent on the Object LER.                                                       |
| T1       | <pre>`context LegalEconomicRelator inv T1_HasCompati- bleLegalPositions: self.legalPosition-&gt;forAll(p</pre>                                                             | p.hasCompatibleCorrelative())`                                                                                                                         |
| T2       | <pre>context ReciprocityRelator inv T2_HasProfile: self.profile-&gt;notEmpty()</pre>                                                                                       | A reciprocity must be interpreted through a stated profile, such as bargaining, managerial/equity, or rationing.                                       |
| T2       | <pre>context ReciprocityRelator inv T2_HasEnablingAndResulting: self.ena- blingComponent-&gt;notEmpty() and self.re- sultingComponent-&gt;notEmpty()</pre>                 | A reciprocity has enabling and resulting component roles, interpreted through the profile patterns of Table 1.                                         |
| T2       | <pre>`context ReciprocityRelator inv T2_ComponentsTyped: self.component-&gt;forAll(c</pre>                                                                                 | c.ocIsKindOf(PerformanceLER) or c.ocIsKindOf(InstitutionalAct) or c.ocIsKindOf(AuthorityGovernedPosition) or c.ocIsKindOf(ObjectifiedPerformanceLER))` |
| T3       | <pre>context LERPhaseChange inv T3_Re- quiresTriggeringSituation: self.trig- geringSituation-&gt;notEmpty()</pre>                                                          | A phase change requires an explicit Triggering Situation.                                                                                              |
| T3       | <pre>`context LERPhaseChange inv T3_RequiresInstitu- tionalEffect: self.triggeringSituation-&gt;exists(ts</pre>                                                            | ts.institutionalEffect->notEmpty())`                                                                                                                   |
| T3       | <pre>context HolderChange inv T3_Existin- gObjectTransfer: self.transfersExistin- gObjectLER implies (self.terminated- HoldingLER-&gt;notEmpty() and</pre>                 | Transfer of an existing Object LER closes and opens Holding LERs over the same persistent Object LER.                                                  |

| T  | OCL-style invariant                                                                                                                                                                                                                                                                                                                         | Reading                                                                                                                                                       |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T3 | <pre> self.createdHoldingLER-&gt;notEmpty() and self.terminatedHoldingLER.objectLER = self.createdHoldingLER.objectLER)  context HolderChange inv T3_NewOb- jectDirectedRight: self.originates- ObjectLER implies (self.createdHold- ingLER-&gt;notEmpty() and self.created- HoldingLER.objectLER.underlyingObject- -&gt;notEmpty()) </pre> | <p>Origination or grant of a new object-directed right, such as a right-of-use, may create a new Object LER externally dependent on an underlying object.</p> |
| T4 | <pre> context Classification inv T4_Classi- fiedIdentifiedTarget: self.classi- fiedLER-&gt;notEmpty() or self.classi- fiedLERChange-&gt;notEmpty() or self.clas- sifiedInstitutionalAct-&gt;notEmpty() </pre>                                                                                                                               | <p>L2 classification must attach to an identified LER, LER change, or Institutional Act.</p>                                                                  |
| T5 | <pre> context RecognitionAssessment inv T5_RecognitionRequiresTarget: self.rec- ognizedLER-&gt;notEmpty() or self.recog- nizedInstitutionalAct-&gt;notEmpty() or self.recognizedRole-&gt;notEmpty() </pre>                                                                                                                                  | <p>Recognition must apply to an identified target; it must not constitute L1 existence.</p>                                                                   |
| T6 | <pre> context MeasurementAssessment inv T6_MeasurementRequiresTarget: self.measuredLER-&gt;notEmpty() or self.measuredInstitutionalAct- -&gt;notEmpty() or self.measuredRole- -&gt;notEmpty() </pre>                                                                                                                                        | <p>Measurement must attach to an identified LER, Institutional Act, or L2 role.</p>                                                                           |
| T6 | <pre> context MeasurementAssessment inv T6_DelimitationRequiredWhenNeeded: self.requiresDelimitation implies self.delimitationCriterion-&gt;notEmpty() </pre>                                                                                                                                                                               | <p>Measurement requiring re-tained/transferred, portion, extent, bundle, or unit-of-account allocation needs prior delimitation.</p>                          |
| T7 | <pre> context RepresentationScope inv T7_AnchorsRequired- Distinctions: self.requiredDistinction-&gt;forall(d </pre>                                                                                                                                                                                                                        | <p>self.representationAnchor-&gt;exists(a</p>                                                                                                                 |
| T7 | <pre> context Representation inv T7_Depend- sOnRepresentationScope: self.represen- tationScope-&gt;notEmpty() </pre>                                                                                                                                                                                                                        | <p>Representation is an L5 overlay dependent on a selected scope of lower-layer units; it does not create the L1 bearer.</p>                                  |