

Evaluation protocol ArchiMate language support

March 18, 2005

In order to assess the conformance of a modelling tool to the ArchiMate language, the following four aspects are evaluated:

1. Concept coverage
2. Relation coverage
3. Notation
4. Diagrams/viewpoints

Each of the aspects will be scored on a scale from 1 to 10. A minimum score of 7 for each aspect separately and a minimal total score of 30 are required.

1. *Concept coverage*

At least the basic concepts of the ArchiMate language should be supported (see checklist below; concepts marked with an asterisk have lower priority, but their support is recommended):

- Concepts from the business layer, application layer and technology layer.
- Concepts from the structure, behaviour and information aspects.
- Optionally, specialisations of the basic concepts may be defined. However, it should be clear how these relate to the basic concepts: additional concepts unrelated to the basic concepts are not allowed.

	Information	Behaviour	Structure
Business layer	<input type="checkbox"/> Business object <input type="checkbox"/> Representation* <input type="checkbox"/> Product <input type="checkbox"/> Meaning* <input type="checkbox"/> Value* <input type="checkbox"/> Contract*	<input type="checkbox"/> Business service <input type="checkbox"/> Business process <input type="checkbox"/> Business function* <input type="checkbox"/> Business interaction <input type="checkbox"/> Business event	<input type="checkbox"/> Business actor <input type="checkbox"/> Business role <input type="checkbox"/> Business interface* <input type="checkbox"/> Business collaboration
Application layer	<input type="checkbox"/> Data object	<input type="checkbox"/> Application service <input type="checkbox"/> Application function <input type="checkbox"/> Application interaction	<input type="checkbox"/> Application component <input type="checkbox"/> Application interface <input type="checkbox"/> Application collaboration
Technology layer	<input type="checkbox"/> Artifact	<input type="checkbox"/> Infrastructure service <input type="checkbox"/> System software	<input type="checkbox"/> Node* <input type="checkbox"/> Device <input type="checkbox"/> Infrastructure interface* <input type="checkbox"/> Network <input type="checkbox"/> Communication path

2. *Relation coverage*

- The following relation types should be supported:
 - Structural relations:

- composition*
 - aggregation
 - assignment
 - used by
 - realisation
 - access
 - association
 - Dynamic relations:
 - triggering
 - flow
 - Other relations:
 - grouping
 - junction
 - specialisation*
- It should be possible to model all direct relations between concepts as defined in the ArchiMate metamodel (see Appendix A), as well as all relations that can be derived with the composition rules (see D2.2.1v4.0, Section 7.4). Also, composition and aggregation relations should always be allowed between instances of the same concept.
- It should not be possible to model relations that do not conform to the ArchiMate metamodel and that cannot be derived with the composition rules; at least, a warning should be generated if such relations are modelled.
- Optionally, specialisations of the basic relation types may be defined. However, it should be clear how these relate to the basic relations: additional relation types unrelated to the basic relations are not allowed.

3. Notation

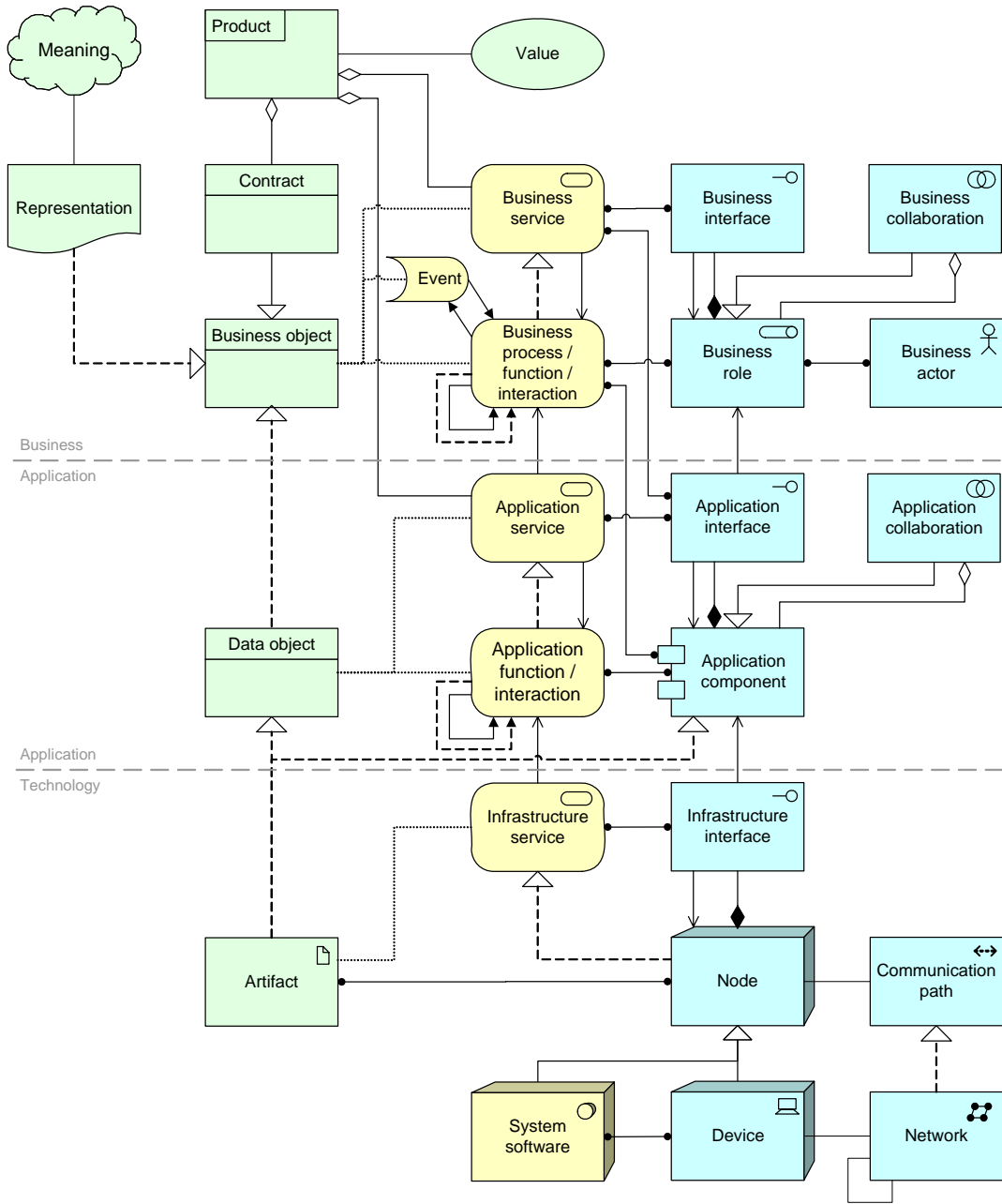
- Every concept and relation should have a precise graphical notation, with a sufficient resemblance the 'standard' ArchiMate notation. The notation in the Visio stencils can be used as a guideline (see Appendix B).
- Optionally, multiple notations may exist for a single concept.
- It should be possible to denote composition, aggregation and assignment both with their 'line' notation and with nesting.

4. Diagrams/viewpoints

- It should be possible to create models using different (predefined or user-defined) 'design viewpoints' (diagrams).
- Design viewpoints within layers, but also crossing layers, should be supported.
- The set of available design viewpoints should cover the whole ArchiMate language.
- It should at least be possible to reproduce the sample models in Appendix C (preferably in a single diagram, possibly split over multiple diagrams).

Appendix A - ArchiMate metamodel

Layered view



Appendix B - Notation

