BiT-ReNU Testsuite

**51 Validation Tests** (2 roundtripping, 20 Edit in UML, 29 Edit in NL)

**6 Performance Tests**

Contents

[1. Validation Tests 1](#_Toc143446322)

[1.1 Bidirectional Transformation 2](#_Toc143446323)

[1.2 Alterations in UML 2](#_Toc143446324)

[1.3 Alterations in NL 5](#_Toc143446325)

[1.3.1 Alterations in NL – adding and deleting sentences 5](#_Toc143446326)

[1.3.2 Alterations in NL – changing sentence elements 6](#_Toc143446327)

[2. Performance Tests 10](#_Toc143446328)

[3. Test Limitations 11](#_Toc143446329)

-----------------------------------------------------------------------------------------------------------------------------------------

# Validation Tests

The goal of the validation tests is to validate if:

* Bidirectional transformation works.
* Making changes after a transformation, and then transforming back into the original notation works as intended.

For the tests were UML elements, sentences or sentence elements are altered, restrictions were used to ensure that only the focused task was performed.

*Example:*

If an Association end was changed to another class, the old class has to be connected to other elements (attributes, associations, compositions, generalizations). This is done, because there is no sentence templates for individual classes. Therefore, if a class has no connection to another element, it would be deleted.

## Bidirectional Transformation

The tests in this category are used to validate that bidirectional transformation is possible in BiT-ReNU.

(Naming scheme: “BT” for “Bidirectional Transformation”)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test-ID** | **Test-Description** | **Expected Result** | **Deviations from expected result** |
| **BT.01** | Transform UML into NL and the resulting NL back into UML. | The original and final UML match. |  |
| **BT.02** | Transform NL into UML and the resulting UML back into NL. | The original and final NL match. |  |

\*Match = In NL, all sentences are the same (order can be different). In UML, all elements are the same with the same characteristics (x and y coordinates of the elements can change and the order in the plantuml notation can also be different)

## Alterations in NL

Input file from BT.01 is used for all AN tests

The changes to NL are all done within the test-functions in the test scripts.

The tests in this category are an extension of test BT.01.

The tests are based on the sentence templates, defined for the transformations in BiT-ReNU:

|  |  |
| --- | --- |
| **Template name** | **Template** |
| **T.1: Active Association** | <Subject> CAN |M U ST <Action> <Object> |
| **T.2: Passive Association** | <Subject> CAN |M U ST <PassiveAction> <Object> |
| **T.3: Attribute** | <Subject> HAS <Object> |
| **T.4: Generalization** | <Subject> IS <Object> |
| **T.5: Composition – Part of Whole** | <Subject> IS PART OF <Object> |
| **T.6: Composition – Whole has Part** | <Subject> CAN|MUST HAVE <Object> |

As NL is not similar to UML, where simply individual elements can be tested, the test-structure is different.

There are two types of alterations tests for NL. First, whole sentences can be changed; and second, individual elements can be changed.

### Alterations in NL – adding and deleting sentences

(Naming scheme: “AN” for “Alteration in Natural Language”)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test-ID** | **Test-Description** | **Expected Result in UML** | **Deviations from expected result** |
|  | **Add Active Association**  Add an Active Association sentence.  The subject and object class have to exist in other original sentences. | A new association will be created between the two classes. The multiplicity at the subject end will be empty. |  |
|  | **Delete Active Association**  Delete an Active Association sentence.  Subject and object have to appear in other sentences. | The multiplicity of the object end will be empty. |  |
|  | **Add Passive Association**  Add a Passive Association sentence.  The subject and object class have to exist in other original sentences. | A new association will be created between the two classes. The multiplicity at the subject end will be empty. |  |
|  | **Delete Passive Association**  Delete a Passive Association sentence.  Subject and object have to appear in other sentences. | The multiplicity of the subject end will be empty. |  |
|  | **Add Attribute**  Add an attribute sentence.  The subject has to exist in other original sentences. There cannot exist another attribute sentence with the same subject and object. | A new attribute will be added to the subject class. |  |
|  | **Delete Attribute**  Delete an attribute sentence.  The subject has to exist in other original sentence. | The attribute will be deleted from the subject. |  |
|  | **Add Generalization**  A generalization sentence is added.  The subject and object have to exist in other original sentences. There cannot exist another generalization sentence with the same subject and object. | A generalization will be added between the two classes. |  |
|  | **Delete Generalization**  Delete a generalization sentence.  Subject and object have to appear in other sentences. | The generalization will be deleted. |  |
|  | **Add Composition – Part of Whole**  A composition sentence based on T.5 is added.  The subject and object have to exist in other original sentences. There cannot exist another composition sentence with the same subject and object. | The composition will be added. The non-composite end will not have multiplcitiy. |  |
|  | **Delete Composition – Part of Whole**  Delete a T.5 composition sentence.  Subject and object have to appear in other sentences. | The composition will still be present if a ‘’Whole has Part’’ Composition sentence is still present.**WHY** |  |
|  | **Add Composition – Whole has Part**  A composition sentence based on T.6 is added.  The subject and object must exist in other original sentences. There cannot exist another composition sentence with the same subject and object. | The composition will be added. Both ends will have multiplicity. |  |
|  | **Delete Composition – Whole has Part**  Delete a T.6 composition sentence.  Subject and object have to appear in other sentences. | The multiplicity of the non-composite end will be deleted. If the corresponding T.5 sentence for the composition does not exist, the whole composition will be deleted. |  |

### Alterations in NL – changing sentence elements

|  |  |  |  |
| --- | --- | --- | --- |
| **Test-ID** | **Test-Description** | **Expected Result in UML** | **Deviations from expected result** |
|  | **Active Association: Edit Subject – new subject exists**  Edit the subject of an Active Association sentence.  The new subject exists in other original sentences.  The original subject has to exist in another sentence. | A new association will be added between the new subject and the object. The multiplicity of the subject end will be empty.  The multiplicity of the object end of the association between the original subject and object will be empty. |  |
|  | **Active Association: Edit Subject**  **– new subject does not exist**  Edit the subject of an Active Association sentence.  The new subject does not exist in another original sentence.  The original subject has to exist in another sentence. | A new association will be added between the new subject and the object. The multiplicity of the subject end will be empty.  The multiplicity of the object end of the association between the original subject and object will be empty. |  |
|  | **Active Association: Switch modal verb**  Switch the modal verb in an Active Association sentence. | The multiplicity of the object end of the association will be switched accordingly. |  |
|  | **Active Association: Edit Action**  Edit the action verb in an Active Association sentence. | A new association will be created between subject and object. The multiplicity of subject end will be empty.  The multiplicity of the object end of the association between the original subject and object will be empty. |  |
|  | **Active Association: Edit Object – new object exists**  Edit the object of an Active Association sentence.  The original object has to appear in another sentence.  The new object has to appear in another original sentence. | A new association will be added between the subject and the new object. The multiplicity of the subject end will be empty.  The multiplicity of the object end of the association between the subject and the original object will be empty. |  |
|  | **Active Association: Edit Object**  **– new object does not exist**  Edit the object of an Active Association sentence.  The new object does not exist in another original sentence.  The original object has to exist in another sentence. | **???** |  |
|  | **Passive Association: Edit Subject – new subject exists**  Edit the subject of a Passive Association sentence.  The original subject has to appear in another sentence.  The new subject has to appear in another original sentence. | A new association between new subject and object will be added. The multiplicity of the subject end will be empty.  The multiplicity of the object end of the original association will be empty. |  |
|  | **Passive Association: Edit Subject**  **– new subject does not exist**  Edit the subject of a Passive Association sentence.  The new subject does not exist in another original sentence.  The original subject has to exist in another sentence. | **???** |  |
|  | **Passive Association: Switch modal verb**  Switch the modal verb in a Passive Association sentence. | The multiplicity of the object end of the association sentence will be switched accordingly. |  |
|  | **Passive Association: Edit Action**  Edit the action verb of a Passive Association sentence. | A new association will be created between subject and object. The multiplicity of subject end will be empty.  The multiplicity of the object end of the association between the original subject and object will be empty. |  |
|  | **Passive Association: Edit Object – new object exists**  Edit the object of a Passive Association sentence.  The original object has to appear in another sentence.  The new object has to appear in another original sentence. | A new association will be added between the subject and the new object. The multiplicity of the subject end will be empty.  The multiplicity of the object end of the association between the subject and the original object will be empty. |  |
|  | **Passive Association: Edit Object**  **– new object does not exist**  Edit the object of a Passive Association sentence.  The new object does not exist in another original sentence.  The original object has to exist in another sentence. | **???** |  |
|  | **Attribute: Edit Subject – new subject exists**  Edit the subject in an attribute sentence.  The new subject exists in other original sentences.  The original subject has to exist in another sentence. | Attribute is transferred to other class. |  |
|  | **Attribute: Edit Subject – new subject does not exist**  Edit the subject in an attribute sentence.  The new subject does not exist in another original sentence.  The original subject has to exist in another sentence. | New class with the attribute is created |  |
|  | **Attribute: Edit Object**  Edit the object of an attribute sentence.  New object is not already an attribute of the class. | Attribute name changes |  |
|  | **Generalization: Edit Subject – new subject exists**  Edit the subject of a generalization sentence.  The new subject already exists in an original sentence.  The original subject exists in another sentence. | Generalization end is changed to the other class. |  |
|  | **Generalization: Edit Subject – new subject does not exist**  Edit the subject of a generalization sentence.  The new subject does not exist in an original sentence.  The old subject does exist in another sentence. | New class with the generalization is created |  |
|  | **Generalization: Edit Object – new object exists**  Edit the object of a generalization sentence.  The new object already exists in another original sentence.  The old object exists in another sentence. | Generalization end is changed to the other class. |  |
|  | **Generalization: Edit Object – new object does not exist**  Edit the object in a generalization sentence.  The new object does not exist in an original sentence.  The old object exists in another sentence. | New class with the generalization is created |  |
|  | **Composition T.5: Edit Subject – new subject exists**  Edit the subject in a T.5 composition sentence.  The new subject already exists in another original sentence.  The old subject also exists in another sentence. | Composition end is changed to the other class. |  |
|  | **Composition T.5: Edit Subject – new subject does not exist**  Edit the subject of a T.5 composition sentence.  The new subject does not already exist in another original sentence.  The old subject exists in another sentence. | New class with the composition is created. |  |
|  | **Composition T.5: Edit Object– new object exists**  Edit the object of a T.5 composition sentence.  The new object does already exist in another original sentence.  The old object does exist in another sentence. | Composition end is changed to the other class. |  |
|  | **Composition T.5: Edit Object – new object does not exist**  Edit the object of a T.5 composition sentence.  The new object does not already exist in another original sentence.  The old object does exist in another sentence. | New class with the composition is created. |  |
|  | **Composition T.6: Edit Subject – new subject exists**  Edit the subject in a T.6 composition sentence.  The new subject already exists in another original sentence.  The old subject also exists in another sentence. | Composition end is changed to the other class. |  |
|  | **Composition T.6: Edit Subject – new subject does not exist**  Edit the subject of a T.6 composition sentence.  The new subject does not already exist in another original sentence.  The old subject exists in another sentence. | New class with the composition is created. |  |
|  | **Composition T.6: Edit Object– new object exists**  Edit the object of a T.6 composition sentence.  The new object does already exist in another original sentence.  The old object does exist in another sentence. | Composition end is changed to the other class. |  |
|  | **Composition T.6: Edit Object – new object does not exist**  Edit the object of a T.6 composition sentence.  The new object does not already exist in another original sentence.  The old object does exist in another sentence. | New class with the composition is created. |  |
|  | **Composition T.6: Switch modal verb**  Switch the modal verb in a T.6 composition sentence. | The multiplicity of the non-composite end of the composition will be switched accordingly. |  |

## Alterations in UML

The tests in this category are an extension of test BT.02. The tests are used to validate, that if changes are made to the transformed UML class model, the final NL only differs from the original NL in ways that can be expected based on the changes made in the UML class model and the defined transformation rules for BiT-ReNU.

“Changes” are broken down into three different actions:

* **Add**: Add a new element
* **Edit**: Edit an existing element
* **Delete**: Delete an existing element

Changes can be made to the following UML class model elements in BiT-ReNU:

* Class
* Attribute
* Association
* Composition
* Multiplicity
* Generalization

Changes in the UML class model have do be done in a separate model editor. Because transforming the input NL into the UML class model, would result in the same transformation every test, this part is not performed in every test.

The intermediary UML class model from BT.02 is used as the transformed UML class model. This UML class model is then changed in a model editor and imported into the test. The test then performs the transformation of the changed UML class model into NL. This output NL can then be compared to the input NL of BT.02.

(Naming scheme: “AU” for “Alteration in UML”)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test-ID** | **Test-Description** | **Expected Result in NL** | **Deviations from expected result** |
| **AU.01** | **Add Class**  Add a new class, without any other connected elements (no attributes, associations, compositions, generalizations) | The class will not be transformed into NL, because there is no sentence template for single classes. |  |
| **AU.02** | **Edit Class name**  Edit the name of a class and edit the name in every connected generalizations, associations, compositions.  (Attributes and other connected elements are covered in different tests.) | The new class name will be transformed instead of the old class name. |  |
| **AU.03** | **Delete Class**  Also Delete all Attributes, Associations, Compositions, Generalizations, and connected Multiplicity. | All sentences that would be connected to the deleted elements will be deleted. |  |
| **AU.04** | **Add Attribute**  Add a single attribute to a class, that has a name, that is not already an attribute of the class. | A new attribute sentence will be created. |  |
| **AU.05** | **Edit Attribute name**  Edit the name of a single attribute to a name that no other attribute of the class has. | The attribute name will be changed in the corresponding sentence. |  |
| **AU.06** | **Delete Attribute**  Delete a single attribute of a class. The class still has to be connected to other elements after the deletion (attributes, associations, compositions, generalizations) | The attribute sentence will be deleted. |  |
| **AU.07** | **Add Association**  Add an association between two classes. The name of the association has to be different than existing names of other associations between the two classes.  Also add Multiplicities to the association ends. | An active association and passive association sentence will be created |  |
| **AU.08** | **Edit Association Name**  Edit the name of an association. The new name has to be different from other association names between the two classes. | The action verb in the corresponding active association and passive association sentences will be edited. |  |
| **AU.09** | **Edit Association End**  Change one end of the Association ends to another class.  The old class still has to be connected to other elements (Attributes, associations, compositions, generalizations. | The class name will be changed in the two corresponding sentences. |  |
| **AU.10** | **Edit Association End Multiplicity**  Change the Multiplicity of one end of an association.  The new Multiplicity can be either 0..\* or 1..\*. | The modal verb in the corresponding sentence templates will be switched. |  |
| **AU.11** | **Delete Association**  Delete an Association.  Only delete elements, such that all remaining classes are still connected to at least one other class. | The two corresponding sentence templates will be deleted. |  |
| **AU.12** | **Delete Association End Multiplicity**  Delete the Multiplicity of one association end. | **Unknown behaviour** |  |
| **AU.13** | **Add Composition**  Add a Composition between two classes that are not connected by a composition that has the same class as the “Part”. | A new sentence will be added. |  |
| **AU.14** | **Edit Composition End**  Change one end of the composition to another class.  The old class still has to be connected to other elements (Attributes, associations, compositions, generalizations. | The name of the new class will be used instead of the old class in the corresponding sentence. |  |
| **AU.15** | **Edit Composition End Multiplicity**  Change the Multiplicity of the non-composite end. | The modal verb in the corresponding sentence templates will be switched. |  |
| **AU.16** | **Delete Composition**  Delete a Composition.  Only delete elements, such that all remaining classes are still connected to at least one other class. | The corresponding sentence will be deleted |  |
| **AU.17** | **Delete Composition End Multiplicity**  Delete the Multiplicity of the non-composite end. | **Unknown behaviour** |  |
| **AU.18** | **Add Generalization**  Add a new Generalization between two classes. | A new sentence will be added. |  |
| **AU.19** | **Edit Generalization End**  Change one end of the generalization to another class.  The old class still has to be connected to other elements (Attributes, associations, compositions, generalizations. | The name of the new class will be used in the corresponding sentence. |  |
| **AU.20** | **Delete Generalization**  Delete a generalization between two classes.  Only delete elements, such that all remaining classes are still connected to at least one other class. | The corresponding sentence will be deleted. |  |

The entry “Unknown behaviour” is used because it is unknown before the test if and how exactly BiT-ReNU will create a result.

# Performance Tests

Performance Tests are used to measure the performance of BiT-ReNU, based on different sizes of input. Each test is performed 30 times.

The tests are focused on BiT-ReNUs performance on transforming various sizes of input. Therefore, only no-way transformations (T2M and M2T) are performed

There are three tests per direction, whereas the input size is doubled for the second tests and doubled again for the third tests.

The input size for NL is measured in sentences.

The input size for UML is measure in total number of elements

(Naming scheme: “PT” for “Performance Test”)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test-ID** | **Test-Description** | **Average execution time** | **Average memory usage** |
| **PT.01** | NL to UML: input size X |  |  |
| **PT.02** | NL to UML: input size X\*2 |  |  |
| **PT.03** | NL to UML: input size X\*4 |  |  |
| **PT.04** | UML to NL: input size X |  |  |
| **PT.05** | UML to NL: input size X\*2 |  |  |
| **PT.06** | UML to NL: input size X\*4 |  |  |

---------------------------------------------------------------------------------------------------------------------------------------

**BEISPIEL wie das script für die Performance Tests aussehen würde:**

Einzelne Performance tests als Funktionen und dann mit measure\_function() ausführen:

A computer screen shot of a program code

Description automatically generated

---------------------------------------------------------------------------------------------------------------------------------------

# Test Limitations

* **Add Multiplicity in UML:** 
  + Cannot be done without adding a new association, therefore there is no test to just add multiplicity alone.
* **Adding and deleting individual sentence elements**
  + not done, due to time constraints
  + Expected result would be that the whole sentence will be omitted. Therefore the same result as for Deletion of whole sentences:
    - AN.02 Delete Active Association
    - AN.04 Delete Passive Association
    - AN.06 Delete Attribute
    - AN.08 Delete Generalization
    - AN.10 Delete Composition

---------------------------------------------------------------------------------------------------------------------------------------

TEST-DOKU

**BT.01:** Done

**BT.02:** Done

-----------------

**AN.01:** A Customer must drive a vehicle.

**AN.02:**

**AN.03:** A vehicle can be driven by a customer.

**AN.04:**