

Documentation

ASSIST User Guide

Robert Hilbrich, Michael Behrisch

DLR German Aerospace Center
Institute of Transportation Systems

Berlin



DLR

Deutsches Zentrum
für Luft- und Raumfahrt
German Aerospace Center

DLR German Aerospace Center

Institute of Transportation Systems

Prof. Dr. Karsten Lemmer

Rutherfordstraße 2

12489 Berlin

Tel: +49 531 295-3401

Robert Hilbrich

Tel: +49 30 67055-582

Fax: +49 30 67055-291

E-Mail: robert.hilbrich@dlr.de

Document Identification:

Title	ASSIST User Guide
Subject	Documentation
Author(s)	Robert Hilbrich, Michael Behrisch
Filename	user-guide.tex
Last saved on	19th September 2015

Document History:

Version 0.1	Initial Version	18.08.2015
-----------------------	-----------------	------------

Contents

1. Introduction	1
2. Usage	3
3. Constraint Semantics	5
3.1. Basic Elements: RDCs, Connectors, Pins and Equipment Interfaces	5
3.2. Basic Interface Types Constraint	5
3.2.1. Intended semantics	5
3.2.2. Input Specification	5
3.2.3. Implementation Remarks	6
3.3. Configurable Interface Types Constraint	6
3.3.1. Intended semantics	6
3.3.2. Input Specification	6
3.3.3. Implementation Remarks	6
3.4. Protection Level Constrains	6
3.4.1. Intended semantics	6
3.4.2. Input Specification	6
3.4.3. Implementation Remarks	6
3.5. Connected Pins	6
3.5.1. Intended semantics	6
3.5.2. Input Specification	6
3.5.3. Implementation Remarks	6
A. Mapping Example	A
B. Input Specification Grammar	G

1. Introduction

2. Usage

3. Constraint Semantics

3.1. Basic Elements: RDCs, Connectors, Pins and Equipment Interfaces

- Hardware elements with *containment* relationships:
 - Compartments
 - Remote Data Concentrators - RDCs
 - Connectors
 - Pins
 - Equipment interfaces
- Allocation is generated between *equipment interfaces* and *pins*
- Allocation is also indirectly generated between *equipment interfaces* and all other elements

3.2. Basic Interface Types Constraint

3.2.1. Intended semantics

3.2.2. Input Specification

3.2.3. Implementation Remarks

3.3. Configurable Interface Types Constraint

3.3.1. Intended semantics

3.3.2. Input Specification

3.3.3. Implementation Remarks

3.4. Protection Level Constrains

3.4.1. Intended semantics

3.4.2. Input Specification

3.4.3. Implementation Remarks

3.5. Connected Pins

3.5.1. Intended semantics

3.5.2. Input Specification

3.5.3. Implementation Remarks

A. Mapping Example

In Listing A.1 an example specification is described. This example illustrates the specifications that are possible with the current version of ASSIST.

```
1  /*
   * GLOBAL PROPERTIES
   */
3
5  Global {
      Name = "System";
7
      Compatible Interface Types {
9        "EquipmentType0" -> "PinType0", "PinType1";
        "EquipmentType1" -> "PinType1", "PinType2";
11    }
13
      Cable Weights {
        "EquipmentType0" = 0.3232;
15        "EquipmentType1" = 1.3232;
        default          = 2.43;
17    }
19
      Protection Level Restrictions {
        RDC.Location = "LocationA" And Equipment.EmhZone1 = "LocationB" -> L1, L3;
21        RDC.Location = "LocationB" And Equipment.EmhZone1 = "LocationC" -> L5;
        }
23  }
25  /*
   * COMPARTMENTS, RDCs, CONNECTORS and PINS
   */
27
29  Compartments {
31
      Compartment Comp1 {
        Manufacturer = "ManufacturerName";
33        PowerSupply = "PowerSupplyName";
        Side        = "SideName";
35        Zone        = "ZoneName";
```

```

37     RDC RDC1 {
38         Manufacturer = "ManufacturerName";
39         PowerSupply1 = "PowerSupply1Name";
40         PowerSupply2 = "PowerSupply2Name";
41         Type         = "RDCTypeName";
42         ESS          = "ESSName";
43         Location     = "RDCLocationName";
44         ResourceX    = -120;
45         ResourceY    = 150;
46         ResourceZ    = -1200;
47
48         Connector Connector1{
49             // Available protection levels:
50             // None, L1, L2, L3, L4, L5, L6, L7, L8
51
52             Pin1: "PinType0"; // = protection level None
53             Pin2: "PinType0" protection level L5;
54             Pin3: "PinType1" protection level L8;
55         }
56
57         Connector Connector2{
58             Pin1: "PinType3";
59             Pin2: "PinType1" protection level L8;
60             Pin3: "PinType1" protection level L8;
61         }
62
63         // ... more Connectors possible ...
64
65         Connected Pins {
66             Comp1.RDC1.Connector1.Pin1, Comp1.RDC1.Connector1.Pin2 are connected;
67             Comp1.RDC1.Connector1.Pin3, Comp1.RDC1.Connector2.Pin2, Comp1.RDC1.Connector2
68             .Pin3 are connected;
69         }
70
71         Metric Parameters {
72             "RDC1Parameter1" = 54;
73
74             // ...
75         }
76     }
77
78     // ... more RDCs possible ...
79
80     Metric Parameters {
81         "Comp1Parameter1" = 12;
82         "Comp1Parameter2" = 6;
83         // ...
84     }

```

```

85     }
86 }
87
88 /*
89  * EQUIPMENT INTERFACES
90  */
91
92 Interfaces {
93
94     Interface EquipmentInterface1 {
95         System      = "SystemName";
96         SubAta      = "SubAtaName";
97         Resource     = "ResourceName";
98         LineName     = "LineName";
99         WiringLane   = "WiringLaneName";
100        GrpInfo      = "GrpInfoName";
101        Route        = "RouteName";
102        PwSup1       = "PwSup1Name";
103        EmhZone1     = "EmhZone1Name";
104        Type         = "EquipmentType0";
105        ResourceX    = -212;
106        ResourceY    = -12;
107        ResourceZ    = 55;
108    }
109
110    Interface EquipmentInterface2 {
111        System      = "SystemName";
112        SubAta      = "SubAtaName";
113        Resource     = "ResourceName";
114        LineName     = "LineName";
115        WiringLane   = "WiringLaneName";
116        GrpInfo      = "GrpInfoName";
117        Route        = "RouteName";
118        PwSup1       = "PwSup1Name";
119        EmhZone1     = "EmhZone1Name";
120        Type         = "EquipmentType1";
121        ResourceX    = -212;
122        ResourceY    = -12;
123        ResourceZ    = 55;
124    }
125 }
126
127 InterfaceGroups {
128
129     // Explicit member enumeration
130     Group G1 { EquipmentInterface1, EquipmentInterface2 };
131
132     // Implicit member enumeration based on attributes
133     Group G2 { interfaces with LineName = "LineName" };

```

```

135 // Implicit member enumerations can be combined with an And-filter
    Group G3 { interfaces with LineName = "LineName" and GrpInfo = "GrpInfoName" };
137
138 // Combinations of explicit and implicit member definitions are also possible
139 Group G4 { interfaces with LineName = "LineName", EquipmentInterface2 };
140
141 // Interfaces can also be excluded explicitly
    Group G5 { interfaces with LineName = "LineName" } without { EquipmentInterface1 };
143
144 // Interfaces can also be excluded implicitly
145 Group G6 { interfaces with LineName = "LineName" } without { interfaces with PwSup1 = "PwSup1" };
146
147 // Groups can be combined
    Group G7 combines G1 and G2 and G3;
149 }

151 /*
    * ADDITIONAL RESTRICTIONS
153 */

154 Restrictions {

155     /*
156     * DISLOCALITY
157     */

158     // EquipmentInterface1 and EquipmentInterface2 must
159     // be on separate Compartments (and RDCs and Connectors)
160     // (the level can be set to "Compartment" or "RDC" or "Connector")
161     EquipmentInterface1, EquipmentInterface2 dislocal up to Compartment;
162
163     // All members of G1 must be on separate RDCs
164     G1 dislocal up to RDC;

165     // The members of G1 and G2 must not share an RDC
166     // (while the members of G1 may be on the same RDC)
167     G1, G2 dislocal up to RDC;

168
169     /*
170     * COLOCALITY
171     */

172     // EquipmentInterface1 and EquipmentInterface2 must
173     // be mapped to the same Connector (or RDC or Compartment)
174     EquipmentInterface1, EquipmentInterface2 on same Connector;
175     EquipmentInterface1, EquipmentInterface2 on same RDC;
176
177 }

```



```

183 // This also works for groups
    G1 on same Connector;
185 // .. and combinations of groups and interfaces
    G4, EquipmentInterface2, G5 on same RDC;
187
189 /*
    * VALID DEPLOYMENTS
    */
191
193 // Valid allocations for EquipmentInterface1 are all
194 // pins on Connector1 of RDC1 in Comp1
195 // (explicit pin specification)
    Valid for EquipmentInterface1 are { Comp1.RDC1.Connector1 };
197
198 // This also works for groups
199 // (explicit pin specification)
    Valid for G1, EquipmentInterface1, G3 are { Comp1 };
201
202 // There are also implicit pin specifications possible
    Valid for EquipmentInterface1 are { connectors with Compartment.Name = "Comp1" };
203
204 // Or even implicit combinations ...
205 // EquipmentInterface1 can be allocated to all connectors
206 // where the hosting RDC has a powerSupply1 attribute of Sup1 or
207 // the hosting RDC has a powerSupply2 attribute of Sup2
    Valid for EquipmentInterface1 are { connectors with RDC.PowerSupply1 = "Sup1",
208                                     connectors with RDC.PowerSupply2 = "Sup2" };
209
211 /*
    * INVALID DEPLOYMENTS
    */
213
214 Invalid for EquipmentInterface1 are { Comp1.RDC1.Connector1 };
215
216 Invalid for G1, EquipmentInterface1, G3 are { Comp1 };
217
218 Invalid for EquipmentInterface1 are { connectors with Compartment.Name = "Comp1" };
219
220 Invalid for EquipmentInterface1 are { connectors with RDC.PowerSupply1 = "Sup1",
221                                     connectors with RDC.PowerSupply2 = "Sup2" };
222
223 }

```

Listing A.1: ASSIST Example Specification

B. Input Specification Grammar

In Listing B.1 the grammar for the input specification language is depicted. Please refer to this Listing when a specification contains syntax errors.

```
1  grammar ch.hilbri.assist.mappingsdsl.MappingDSL with org.eclipse.xtext.common.Terminals

3  import "ch.hilbri.assist.datamodel.model"
   import "http://www.eclipse.org/emf/2002/Ecore" as.ecore

5
   AssistModel:
7     (
9         globalBlock          = GlobalBlock          &
10        compartmentsBlock    = CompartmentsBlock    &
11        interfacesBlock       = InterfacesBlock       &
12        (interfaceGroupsBlock = InterfaceGroupsBlock)? &
13        ( restrictionsBlock    = RestrictionsBlock )?
14     )
15 ;

16 /* *****
17  * GLOBAL BLOCK
18  ***** */

19
   GlobalBlock: {GlobalBlock}
20     'Global' '{'
21     (
22         ('Name' '=' systemName=STRING ';' )? &
23         ( compatibleIoTypesBlock = CompatibleIoTypesBlock )? &
24         ( cableWeightDataBlock   = CableWeightDataBlock   )? &
25         ( protectionLevelDataBlock = ProtectionLevelDataBlock )?
26     )
27     '}',
28 ;

31 CompatibleIoTypesBlock:
   'Compatible Interface Types' '{' (compatibleIoTypes+=CompatibleIoTypeEntry)+ '}'
32 ;

35 CompatibleIoTypeEntry:
   eqInterfaceIoType=STRING '->' pinInterfaceIoTypes+=STRING (' , ' pinInterfaceIoTypes+=STRING)* ';' ;
```

```

37 ;

39 CableWeightDataBlock:
    'Cable Weights'          '{' (cableWeightEntries+=CableWeightEntry)+ '}',
41 ;

43 CableWeightEntry:
    (eqInterfaceloType=STRING | defaultEntry?='default') '=' weight=Double ';',
45 ;

47 ProtectionLevelDataBlock:
    'Protection Level Restrictions' '{' (protectionLevelEntries+=ProtectionLevelEntry)+ '}',
49 ;

51 ProtectionLevelEntry :
    'RDC.Location' '=' rdcLocation=STRING 'And' 'Equipment.EmhZone1' '=' emhZone1=STRING '->'
    ' protectionLevel+=ProtectionLevelType (' protectionLevel+=ProtectionLevelType)* ';',
53 ;

55
56 /* *****
57 * COMPARTMENTS
58 ***** */
59
60 CompartmentsBlock: {CompartmentsBlock}
61 'Compartments' '{'
    compartments+=Compartment+
63 '}',
65 ;

66 Compartment:
67 'Compartment' name=ID '{'
    (
68     ('Manufacturer'      '=' manufacturer=STRING ';')?      &
69     ('PowerSupply'       '=' powerSupply=STRING ';')?        &
70     ('Side'              '=' side=STRING ';')?               &
71     ('Zone'              '=' zone=STRING ';')?               &
72     (rdcs+=RDC+)
73     )
74     (metricParametersBlock=MetricParametersBlock)?
75 '}',
77 ;

79 RDC:
    'RDC' name=ID '{'
80     (
81         ('Manufacturer'      '=' manufacturer=STRING ';')?      &
82         ('PowerSupply1'       '=' powerSupply1=STRING ';')?      &
83         ('PowerSupply2'       '=' powerSupply2=STRING ';')?      &

```

```

85         ('Type'                      '=' rdcType=STRING ';' )?           &
86         ('ESS'                       '=' ess=STRING ';' )?              &
87         ('Location'                  '=' location=STRING ';' )?          &
88         ('ResourceX'                 '=' resourceX=SIGNEDINT ';' )?       &
89         ('ResourceY'                 '=' resourceY=SIGNEDINT ';' )?       &
90         ('ResourceZ'                 '=' resourceZ=SIGNEDINT ';' )?       &
91         (internalConnectedPinBlock=InternalConnectedPinBlock)?          &
92         (connectors+=Connector)+
93     )
94     (metricParametersBlock=MetricParametersBlock)?
95 '}',
96 ;
97 InternalConnectedPinBlock: {InternalConnectedPinBlock}
98 'Connected Pins' '{'
99     (connectedPins+=ConnectedPinEntry)*
100 '}',
101 ;
102
103 ConnectedPinEntry:
104     pins+=[PinQualifiedName] ', ' pins+=[PinQualifiedName] (', ' pins+=[PinQualifiedName])? 'are
105         connected' ';
106 ;
107
108 Connector:
109     'Connector' name=ID '{'
110         (pins+=Pin)*
111         (metricParametersBlock=MetricParametersBlock)?
112     '}',
113 ;
114
115 Pin:
116     name=ID ':' eqInterfaceType=STRING ('protection level' protectionLevel=ProtectionLevelType)? ';
117 ;
118
119 enum ProtectionLevelType:
120     NONE = 'None' |
121     L1   = 'L1'   |
122     L2   = 'L2'   |
123     L3   = 'L3'   |
124     L4   = 'L4'   |
125     L5   = 'L5'   |
126     L6   = 'L6'   |
127     L7   = 'L7'   |
128     L8   = 'L8'   |
129 ;
130
131

```

```

MetricParametersBlock: {MetricParametersBlock}
133     'Metric Parameters' '{'
        (metricParameters+=MetricParameter)*
135     '}',
;
137
MetricParameter:
139     name=STRING '=' value=INT ',';
;
141
/* *****
143 * INTERFACES
        ***** */
145
InterfacesBlock :
147     'Interfaces' '{'
        (eqInterfaces+=EqInterface)+
149     '}',
;
151
EqInterface :
153     'Interface' name=ID '{'
        (
155         ('System'      '=' system=STRING      ',';')?      &
156         ('SubAta'      '=' subAta=STRING      ',';')?      &
157         ('Resource'    '=' resource=STRING    ',';')?      &
158         ('LineName'    '=' lineName=STRING    ',';')?      &
159         ('WiringLane'  '=' wiringLane=STRING  ',';')?      &
160         ('GrpInfo'     '=' grpInfo=STRING     ',';')?      &
161         ('Route'       '=' route=STRING       ',';')?      &
162         ('PwSup1'      '=' pwSup1=STRING      ',';')?      &
163         ('EmhZone1'    '=' emhZone1=STRING    ',';')?      &
164         ('Type'        '=' ioType=STRING      ',';')?      &
165         ('ResourceX'   '=' resourceX=SIGNEDINT ',';')?      &
166         ('ResourceY'   '=' resourceY=SIGNEDINT ',';')?      &
167         ('ResourceZ'   '=' resourceZ=SIGNEDINT ',';')?
        )
169     '}',;

171
/* *****
173 * INTERFACE GROUPS
        ***** */
175

InterfaceGroupsBlock: {InterfaceGroupsBlock}
177     'InterfaceGroups' '{'
        (
179         eqInterfaceGroups+=EqInterfaceGroup
                                |
        eqInterfaceGroups+=EqInterfaceGroupWithCombinedDefinition

```

```

181     )*
182     '}',
183 ;
184
185 EqInterfaceGroup:
186     'Group' name=ID '{'
187         (
188             eqInterfaces+=[EqInterface] |
189             implicitMemberDefinitions+=ImplicitEqInterfaceMemberDefinition
190         )
191         (
192             ',' (
193                 eqInterfaces+=[EqInterface] |
194                 implicitMemberDefinitions+=ImplicitEqInterfaceMemberDefinition
195             )
196         )
197     )*
198     '}',
199
200     (
201         'without' '{'
202         (
203             withoutEqInterfaces+=[EqInterface] |
204             withoutImplicitMemberDefinitions+=ImplicitEqInterfaceMemberDefinition
205         )
206         (
207             ',' (
208                 withoutEqInterfaces+=[EqInterface] |
209                 withoutImplicitMemberDefinitions+=
210                     ImplicitEqInterfaceMemberDefinition
211             )
212         )
213     )*
214     '}',
215     )?
216     ';'
217 ;
218
219 ImplicitEqInterfaceMemberDefinition :
220     'interfaces with' entries+=ImplicitEqInterfaceMemberDefinitionAttributesAndValues ('and' entries
221         +=ImplicitEqInterfaceMemberDefinitionAttributesAndValues)*
222     ;
223
224 ImplicitEqInterfaceMemberDefinitionAttributesAndValues:
225     attribute =ImplicitEqInterfaceMemberDefinitionAttribute '=' value=STRING
226     ;
227

```

```

enum ImplicitEqInterfaceMemberDefinitionAttribute:
229     NAME      = 'Name'          |
        SYSTEM   = 'System'        |
231     SUBATA    = 'SubAta'        |
        LINENAME = 'LineName'      |
233     WIRINGLANE = 'WiringLane'   |
        GRPINFO  = 'GrpInfo'       |
235     ROUTE     = 'Route'         |
        PWSUP1   = 'PwSup1'        |
237     EMHZONE1  = 'EmhZone1'     |
        IOTYPE   = 'Type'          |
239     RESOURCE  = 'Resource'      |
        RESOURCE_X = 'ResourceX'   |
241     RESOURCE_Y = 'ResourceY'    |
        RESOURCE_Z = 'ResourceZ'   |
243 ;

245 EqInterfaceGroupWithCombinedDefinition:
        'Group' name=ID 'combines' combinedGroups+=[EqInterfaceGroup] ('and' combinedGroups+=[
            EqInterfaceGroup])+ ';'
247 ;

249
/* *****
251 * RESTRICTIONS
    * ***** */
253
RestrictionsBlock : { RestrictionsBlock }
255     'Restrictions' '{'
        (
257         dislocalityRelations += DislocalityRelation |
            colocalityRelations += ColocalityRelation |
259         validDeployments += ValidDeployment |
            invalidDeployments += InvalidDeployment
261     ) *
        '}',
263 ;

265 enum HardwareArchitectureLevelType :
        COMPARTMENT = 'Compartment' |
267         RDC        = 'RDC'         |
        CONNECTOR   = 'Connector'
269 ;

271 DislocalityRelation :
        eqInterfaceOrGroups+=[EqInterfaceOrGroup]
273     (',' eqInterfaceOrGroups+=[EqInterfaceOrGroup]) *
        'dislocal up to'
275     hardwareLevel=HardwareArchitectureLevelType ';' ;

```



```

277 ColocalityRelation :
279     eqInterfaceOrGroups+=[EqInterfaceOrGroup]
281     (',' eqInterfaceOrGroups+=[EqInterfaceOrGroup])*
283     'on same'
285     hardwareLevel=HardwareArchitectureLevelType ';'
287 ;

285 ValidDeployment:
287     'Valid for' eqInterfaceOrGroups+=[EqInterfaceOrGroup]
289     (',' eqInterfaceOrGroups+=[EqInterfaceOrGroup])*
291     'are' '{'
293         ( hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
295             DeploymentImplicitDefinition)
297         (',' (hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
299             DeploymentImplicitDefinition) ) *
301     '}' ',';

293 InvalidDeployment:
295     'Invalid for' eqInterfaceOrGroups+=[EqInterfaceOrGroup]
297     (',' eqInterfaceOrGroups+=[EqInterfaceOrGroup])*
299     'are' '{'
301         ( hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
303             DeploymentImplicitDefinition)
305         (',' (hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
307             DeploymentImplicitDefinition) ) *
309     '}' ',';

303 DeploymentImplicitDefinition :
305     'connectors with' entries+=DeploymentImplicitDefinitionAttributeAndValue
307     ('and' entries+=DeploymentImplicitDefinitionAttributeAndValue)*
309 ;

307 DeploymentImplicitDefinitionAttributeAndValue:
309     attribute = DeploymentImplicitDefinitionAttribute '=' value=STRING
311 ;

311 enum DeploymentImplicitDefinitionAttribute:
313     COMPARTMENT_NAME      = 'Compartment.Name'          |
315     COMPARTMENT_MANUFACTURER = 'Compartment.Manufacturer' |
317     COMPARTMENT_POWERSUPPLY = 'Compartment.PowerSupply'  |
319     COMPARTMENT_SIDE       = 'Compartment.Side'          |
321     COMPARTMENT_ZONE       = 'Compartment.Zone'          |
323     RDC_NAME               = 'RDC.Name'                  |
325     RDC_MANUFACTURER      = 'RDC.Manufacturer'          |

```

```

321   RDC_POWERSUPPLY1      = 'RDC.PowerSupply1'      |
      RDC_POWERSUPPLY2    = 'RDC.PowerSupply2'      |
323   RDC_SIDE             = 'RDC.Side'              |
      RDC_TYPE            = 'RDC.Type'              |
325   RDC_ESS              = 'RDC.ESS'               |
      RDC_RESOURCE_X      = 'RDC.ResourceX'         |
327   RDC_RESOURCE_Y      = 'RDC.ResourceY'         |
      RDC_RESOURCE_Z      = 'RDC.ResourceZ'         |
329   CONNECTOR_NAME       = 'Connector.Name'
331 ;

333
/* *****
335 * RESTRICTIONS
   * ***** */
337
Double returns ecore::EDouble:
339   INT '.' INT
   ;
341
Qualified Name:
343   ID ('.' ID)*
   ;
345
SIGNEDINT returns ecore::Elnt:
347   '-'? INT;

```

Listing B.1: ASSIST Input Specification Grammar

