

# 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 . . . . .	29th September 2015

**Document History:**

Version 0.1 . . . . .	Initial Version	18.08.2015
Version 0.2 . . . . .	Additional Structure	10.09.2015
Version 0.3 . . . . .	Introduction, Usage	28.09.2015



# Contents

<b>1. Introduction</b>	<b>1</b>
<b>2. Installation and Usage</b>	<b>3</b>
2.1. Installation . . . . .	3
2.2. Usage . . . . .	3
2.3. Useful Shortcuts . . . . .	5
<b>3. System Elements</b>	<b>7</b>
3.1. Overview . . . . .	7
3.2. Compartments . . . . .	7
3.3. RDCs . . . . .	7
3.4. Pins . . . . .	7
3.5. Equipment Interfaces . . . . .	7
<b>4. Constraint Semantics</b>	<b>9</b>
4.1. Basic Interface Types Constraint . . . . .	9
4.1.1. Intended semantics . . . . .	9
4.1.2. Input Specification . . . . .	9
4.1.3. Implementation Remarks . . . . .	9
4.2. Configurable Interface Types Constraint . . . . .	9
4.2.1. Intended semantics . . . . .	9
4.2.2. Input Specification . . . . .	9
4.2.3. Implementation Remarks . . . . .	9
4.3. Protection Level Constrains . . . . .	9
4.3.1. Intended semantics . . . . .	9
4.3.2. Input Specification . . . . .	9
4.3.3. Implementation Remarks . . . . .	10
4.4. Connected Pins . . . . .	10
4.4.1. Intended semantics . . . . .	10
4.4.2. Input Specification . . . . .	10
4.4.3. Implementation Remarks . . . . .	10

4.5. Restriction: Colocality . . . . .	10
4.5.1. Intended semantics . . . . .	10
4.5.2. Input Specification . . . . .	10
4.5.3. Implementation Remarks . . . . .	10
4.6. Restriction: Dislocality . . . . .	10
4.6.1. Intended semantics . . . . .	10
4.6.2. Input Specification . . . . .	10
4.6.3. Implementation Remarks . . . . .	10
4.7. Restriction: Valid Deployments . . . . .	10
4.7.1. Intended semantics . . . . .	11
4.7.2. Input Specification . . . . .	11
4.7.3. Implementation Remarks . . . . .	11
4.8. Restriction: Invalid Deployments . . . . .	11
4.8.1. Intended semantics . . . . .	11
4.8.2. Input Specification . . . . .	11
4.8.3. Implementation Remarks . . . . .	11
<b>A. Mapping Example</b>	<b>A</b>
<b>B. Input Specification Grammar</b>	<b>G</b>

# 1. Introduction

The ASSIST Toolsuite is for System Engineers and System Architects of safety-critical systems. It automates the construction of a mapping between equipment interfaces and the pins of an RDC while respecting safety and reliability requirements.

The ASSIST Toolsuite aims to automate this challenging, error prone and complex task. It requires the user to specify:

- the resource requirements of the equipment interfaces,
- the features and capabilities of the RDCs and
- safety and reliability requirements

in a textual domain specific language.

This specification of a mapping problem is automatically transferred into Constraint Satisfaction Problem (CSP). Solutions for this CSP represent correct deployments. They are automatically generated with the constraint solver *Choco3*. Solutions can also be evaluated and optimized based on pre-defined or customizable metrics.





## 2. Installation and Usage

### 2.1. Installation

1. Download the zip archive and extract it to some folder. (The binary build already includes a suitable Java Runtime Environment, so you do not need to install a JRE separately.)
2. Start ASSIST by double clicking on ASSIST.exe (Windows only).

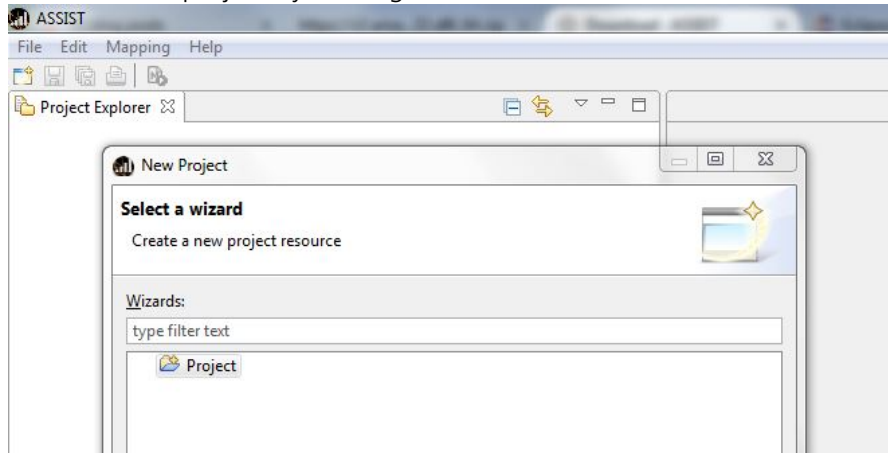
### 2.2. Usage

ASSIST has been developed to support the following workflow:

1. Import or specify your hardware and software architecture in an **mdsl** file
2. Import or Specify your resource and safety requirements in the same **mdsl** file
3. Generate a set of valid deployments
4. Sort these deployments according to customized metrics
5. Select an optimal deployment
6. Export the selected deployment

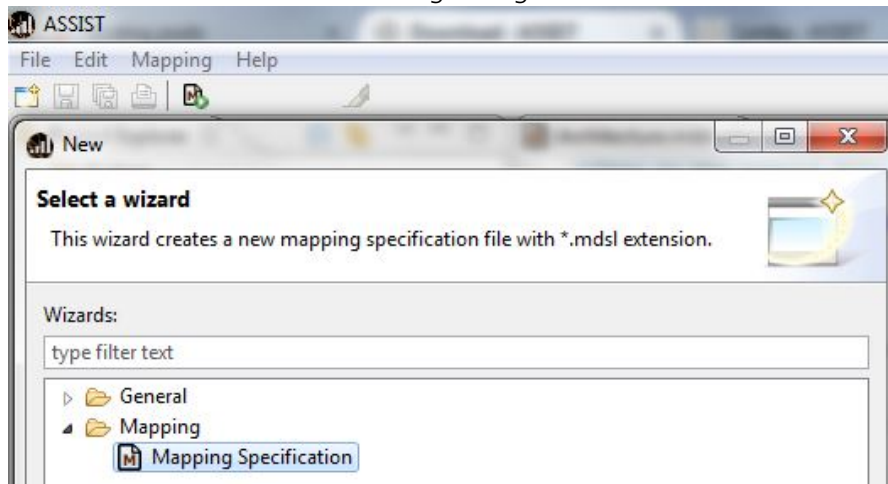
This can be achieved by:

1. Create a new project by clicking on FILE and NEW



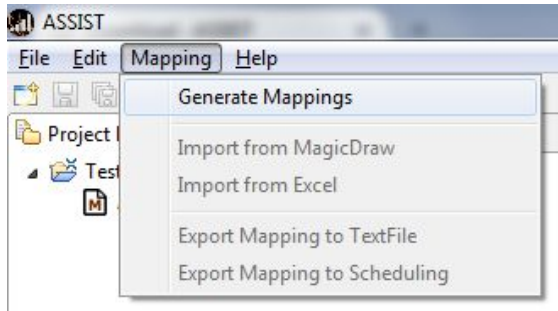
Please provide a suiteable name for your project and click on FINISH.

2. Create a new mapping specification by clicking on FILE and NEW. Please select MAPPING SPECIFICATION in the following dialog.



Please provide a suiteable name for your deployment specification and click on FINISH.

3. Mappings can be generated by clicking on MAPPING and GENERATE MAPPINGS



## 2.3. Useful Shortcuts

- ➔ **Ctrl-Shift-C** - Comment the current line or the currently selected lines
- ➔ **Ctrl-Shift-F** - Format the current file (indentation of sections, ...)



## **3. System Elements**

### **3.1. Overview**

### **3.2. Compartments**

### **3.3. RDCs**

### **3.4. Pins**

### **3.5. Equipment Interfaces**



## **4. Constraint Semantics**

### **4.1. Basic Interface Types Constraint**

#### **4.1.1. Intended semantics**

#### **4.1.2. Input Specification**

#### **4.1.3. Implementation Remarks**

### **4.2. Configurable Interface Types Constraint**

#### **4.2.1. Intended semantics**

#### **4.2.2. Input Specification**

#### **4.2.3. Implementation Remarks**

### **4.3. Protection Level Constrains**

#### **4.3.1. Intended semantics**

#### **4.3.2. Input Specification**

#### **4.3.3. Implementation Remarks**

### **4.4. Connected Pins**

#### **4.4.1. Intended semantics**

#### **4.4.2. Input Specification**

#### **4.4.3. Implementation Remarks**

### **4.5. Restriction: Colocality**

#### **4.5.1. Intended semantics**

#### **4.5.2. Input Specification**

#### **4.5.3. Implementation Remarks**

### **4.6. Restriction: Dislocality**

#### **4.6.1. Intended semantics**

#### **4.6.2. Input Specification**

#### **4.6.3. Implementation Remarks**

### **4.7. Restriction: Valid Deployments**





**4.7.1. Intended semantics**

**4.7.2. Input Specification**

**4.7.3. Implementation Remarks**

## **4.8. Restriction: Invalid Deployments**

**4.8.1. Intended semantics**

**4.8.2. Input Specification**

**4.8.3. 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. Comments are used to further document the intended specification. Single-line comments are prefixed with `//`. Multi-line comments have to be placed between `/*` and `*/`. The entire specification language is white space tolerant, so that additional white spaces can be added between keywords without violating its syntax.

```
1  /*
   * GLOBAL PROPERTIES
   */
3
5  Global {
   Name = "System";
7
   Compatible Interface Types {
8       "EquipmentType0" -> "PinType0", "PinType1";
9       "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
28 Compartments {
29  Compartment Comp1 {
   Manufacturer = "ManufacturerName";
31  PowerSupply = "PowerSupplyName";
```

```

33     Side          = "SideName";
34     Zone          = "ZoneName";

35     RDC RDC1 {
36         Manufacturer = "ManufacturerName";
37         PowerSupply1 = "PowerSupply1Name";
38         PowerSupply2 = "PowerSupply2Name";
39         Type          = "RDCTypeName";
40         ESS           = "ESSName";
41         Location      = "RDCLocationName";
42         ResourceX     = -120;
43         ResourceY     = 150;
44         ResourceZ     = -1200;

45         Connector Connector1{
46             // Available protection levels:
47             // None, L1, L2, L3, L4, L5, L6, L7, L8

48             Pin1: "PinType0"; // = protection level None
49             Pin2: "PinType0" protection level L5;
50             Pin3: "PinType1" protection level L8;

51         }

52         Connector Connector2{
53             Pin1: "PinType3";
54             Pin2: "PinType1" protection level L8;
55             Pin3: "PinType1" protection level L8;

56         }

57         // ... more Connectors possible ...

58         Connected Pins {
59             Comp1.RDC1.Connector1.Pin1, Comp1.RDC1.Connector1.Pin2 are connected;
60             Comp1.RDC1.Connector1.Pin3, Comp1.RDC1.Connector2.Pin2, Comp1.RDC1.Connector2
61             .Pin3 are connected;

62         }

63         Metric Parameters {
64             "RDC1Parameter1" = 54;

65             // ...

66         }

67     }

68     // ... more RDCs possible ...

69     Metric Parameters {
70         "Comp1Parameter1" = 12;
71     }

```

```

81         "Comp1Parameter2" = 6;
82         // ...
83     }

85     // Another empty compartment for demo purposes
86     Compartment Comp2 {
87         RDC RDC1 {
88             Connector Connector1{
89             }
90         }
91     }
92 }

93 /*
94  * EQUIPMENT INTERFACES
95  */
96 Interfaces {
97     Interface EquipmentInterface1 {
98         System      = "SystemName";
99         SubAta      = "SubAtaName";
100         Resource    = "ResourceName";
101         LineName    = "LineName";
102         WiringLane  = "WiringLaneName";
103         GrpInfo     = "GrpInfoName";
104         Route       = "RouteName";
105         PwSup1      = "PwSup1Name";
106         EmhZone1    = "EmhZone1Name";
107         Type        = "EquipmentType0";
108         ResourceX   = -212;
109         ResourceY   = -12;
110         ResourceZ   = 55;
111     }

112     Interface EquipmentInterface2 {
113         System      = "SystemName";
114         SubAta      = "SubAtaName";
115         Resource    = "ResourceName";
116         LineName    = "LineName";
117         WiringLane  = "WiringLaneName";
118         GrpInfo     = "GrpInfoName";
119         Route       = "RouteName";
120         PwSup1      = "PwSup1Name";
121         EmhZone1    = "EmhZone1Name";
122         Type        = "EquipmentType1";
123         ResourceX   = -212;
124         ResourceY   = -12;
125         ResourceZ   = 55;
126     }
127 }

```

```

129 }

131 InterfaceGroups {
    // Explicit member enumeration
133     Group G1 { EquipmentInterface1, EquipmentInterface2 };

135     // Implicit member enumeration based on attributes
    Group G2 { interfaces with LineName = "LineName" };
137
    // Implicit member enumerations can be combined with an And-filter
139     Group G3 { interfaces with LineName = "LineName" and GrpInfo = "GrpInfoName" };

141     // Combinations of explicit and implicit member definitions are also possible
    Group G4 { interfaces with LineName = "LineName", EquipmentInterface2 };
143
    // Interfaces can also be excluded explicitly
145     Group G5 { interfaces with LineName = "LineName" } without { EquipmentInterface1 };

147     // Interfaces can also be excluded implicitly
    Group G6 { interfaces with LineName = "LineName" } without { interfaces with PwSup1 = "PwSup1" };
149
    // Groups can be combined
151     Group G7 combines G1 and G2 and G3;
    }

153

155 /*
    * ADDITIONAL RESTRICTIONS
157 */

159 Restrictions {

161     /*
        * COLOCALITY
163     */

165     // EquipmentInterface1 and EquipmentInterface2 must
    // be mapped to the same Connector (or RDC or Compartment)

167     EquipmentInterface1, EquipmentInterface2 on same Connector;
169     EquipmentInterface1, EquipmentInterface2 on same RDC;

171     // This also works for groups

173     G1 on same Connector;

175     // .. and combinations of groups and interfaces

177     G4, EquipmentInterface2, G5 on same RDC;

```

```

179  /*
180      * DISLOCALITY
181  */

183  // EquipmentInterface1 and EquipmentInterface2 must
184  // be on separate Compartments (and RDCs and Connectors)
185  // (the level can be set to "Compartment" or "RDC" or "Connector")

187  // EquipmentInterface1, EquipmentInterface2 dislocal up to Connector;

189  // All members of G1 must be on separate Connector

191  // G1 dislocal up to Connector;

193  // The members of G1 and G2 must not share an Connector
194  // (while the members of G1 may be on the same Connector)

195  // G1, G2 dislocal up to Connector;

197

199  /*
200      * VALID DEPLOYMENTS
201  */

203  // Valid allocations for EquipmentInterface1 are all
204  // pins on Connector1 of RDC1 in Comp1
205  // (explicit pin specification)
206  Valid for EquipmentInterface1 is { Comp1.RDC1.Connector1 };

207  // This also works for groups
208  // (explicit pin specification)
209  Valid for G1, EquipmentInterface1, G3 is { Comp1 };

211  // There are also implicit pin specifications possible
212  Valid for EquipmentInterface1 is { pins with Compartment.Name = "Comp1" };

215  // Or even implicit combinations ...
216  // EquipmentInterface1 can be allocated to all connectors
217  // where the hosting RDC has a powerSupply1 attribute of Sup1 or
218  // the hosting RDC has a powerSupply2 attribute of Sup2
219  Valid for EquipmentInterface1 is { pins with RDC.PowerSupply1 = "PowerSupply1Name",
220                                     pins with RDC.PowerSupply2 = "PowerSupply2Name" };

221

222  /*
223      * INVALID DEPLOYMENTS
224  */

225  Invalid for EquipmentInterface1 is { Comp2.RDC1.Connector1 };

```

```
227 Invalid for G1, EquipmentInterface1, G3 is { Comp2 };
229
231 Invalid for EquipmentInterface1 is { pins with Compartment.Name = "UnsafeCompartment" };
233
235 Invalid for EquipmentInterface1 is { pins with RDC.PowerSupply1 = "UnsafeSupplier",
    pins with RDC.PowerSupply2 = "UnsafeSupplier" };
}
```

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.mappingdsl.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   (
      globalBlock          = GlobalBlock          &
9      compartmentsBlock   = CompartmentsBlock    &
      interfacesBlock      = InterfacesBlock       &
11     (interfaceGroupsBlock = InterfaceGroupsBlock)? &
      ( restrictionsBlock   = RestrictionsBlock )?
13   )
;

15 /* *****
17 * GLOBAL BLOCK
   ***** */

19
GlobalBlock: {GlobalBlock}
21   'Global' '{'
      (
23         ('Name' '=' systemName=STRING ';' )? &
         ( compatibleTypesBlock = CompatibleTypesBlock )? &
25         ( cableWeightDataBlock = CableWeightDataBlock )? &
         ( protectionLevelDataBlock = ProtectionLevelDataBlock )?
27       )
      '}',
29 ;

31 CompatibleTypesBlock:
      'Compatible' 'Interface' 'Types' '{' (compatibleTypes+=CompatibleTypeEntry)+ '}'
33 ;

35 CompatibleTypeEntry:
      eqInterfaceType=STRING '->' pinInterfaceTypes+=STRING (' , ' pinInterfaceTypes+=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 '}',
64 ;
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 '}',
76 ;
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
120 enum ProtectionLevelType:
121     NONE = 'None' |
122     L1   = 'L1'   |
123     L2   = 'L2'   |
124     L3   = 'L3'   |
125     L4   = 'L4'   |
126     L5   = 'L5'   |
127     L6   = 'L6'   |
128     L7   = 'L7'   |
129     L8   = 'L8'   |
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
221     ('and' entries+=ImplicitEqInterfaceMemberDefinitionAttributesAndValues)*
222 ;
223
224 ImplicitEqInterfaceMemberDefinitionAttributesAndValues:
225     attribute =ImplicitEqInterfaceMemberDefinitionAttribute '=' value=STRING
226 ;
227 ;

```

```

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

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

249

251 /* *****
      * RESTRICTIONS
253   * ***** */

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

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

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

275 ColocalityRelation :

```

```

277     eqInterfaceOrGroups+=[EqInterfaceOrGroup] (',' eqInterfaceOrGroups+=[EqInterfaceOrGroup])*
      'on' 'same' hardwareLevel=HardwareArchitectureLevelType '; '
279 ;

281 ValidDeployment:
      'Valid' 'for' eqInterfaceOrGroups+=[EqInterfaceOrGroup] (',' eqInterfaceOrGroups+=[
        EqInterfaceOrGroup])* 'is'
283     '{'
          (hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
            DeploymentImplicitDefinition)
285     (',' (hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
            DeploymentImplicitDefinition) ) *
      '}' ',';
287 ;

289 InvalidDeployment:
      'Invalid' 'for' eqInterfaceOrGroups+=[EqInterfaceOrGroup] (',' eqInterfaceOrGroups+=[
        EqInterfaceOrGroup])* 'is'
291     '{'
          ( hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
            DeploymentImplicitDefinition)
293     (',' (hardwareElements+=[HardwareElementIQualifiedName] | implicitHardwareElements+=
            DeploymentImplicitDefinition) ) *
      '}' ',';
295 ;

297 DeploymentImplicitDefinition :
      'pins' 'with' entries+=DeploymentImplicitDefinitionAttributeAndValue
299     ('and' entries+=DeploymentImplicitDefinitionAttributeAndValue)*
      ;
301
DeploymentImplicitDefinitionAttributeAndValue:
303     attribute = DeploymentImplicitDefinitionAttribute ' = ' value=STRING
      ;
305
enum DeploymentImplicitDefinitionAttribute:
307     COMPARTMENT_NAME      = 'Compartment.Name'          |
      COMPARTMENT_MANUFACTURER = 'Compartment.Manufacturer' |
309     COMPARTMENT_POWERSUPPLY = 'Compartment.PowerSupply' |
      COMPARTMENT_SIDE      = 'Compartment.Side'          |
311     COMPARTMENT_ZONE      = 'Compartment.Zone'          |
      RDC_NAME              = 'RDC.Name'                  |
313     RDC_MANUFACTURER      = 'RDC.Manufacturer'          |
      RDC_POWERSUPPLY1      = 'RDC.PowerSupply1'          |
315     RDC_POWERSUPPLY2      = 'RDC.PowerSupply2'          |
      RDC_SIDE              = 'RDC.Side'                  |
317     RDC_TYPE              = 'RDC.Type'                  |
      RDC_ESS               = 'RDC.ESS'                  |
319     RDC_RESOURCE_X        = 'RDC.ResourceX'             |

```

```

321     RDC_RESOURCE_Y      = 'RDC.ResourceY'      |
322     RDC_RESOURCE_Z      = 'RDC.ResourceZ'      |
323     CONNECTOR_NAME      = 'Connector.Name'      |
324     PIN_NAME            = 'Pin.Name'
325 ;

327 /* *****
328    * RESTRICTIONS
329    * ***** */

331 Double returns ecore::EDouble:
332     INT '.' INT
333 ;

335 QualifiedName:
336     ID ('.' ID)*
337 ;

339 SIGNEDINT returns ecore::Elint:
340     '-'? INT;

```

Listing B.1: ASSIST Input Specification Grammar

