

# Event-B formal models of ARINC 661 standard specification and weather radar system application

Ismail Mendil

Juin 30, 2021

# Contents

<b>Ontologies Theory (Section 5)</b>	<b>3</b>
Event-B theory of ontology description language . . . . .	3
<b>ARINC 661 Theory (Subsection 7.1)</b>	<b>9</b>
Ontological theory of a chunk of ARINC 661 specification . . . . .	9
<b>WXR Theory (Subsection 7.2)</b>	<b>17</b>
Event-B theory of contextual information used in WXR model . . . . .	17
<b>WXR Model (Subsection 7.3)</b>	<b>34</b>
WXR Event-B Model . . . . .	34

# **Ontologies Theory (Section 5)**

# Ontology description language Event-B Theory

## THEORY

**TYPE PARAMETERS**  $C, P, I$

### DATA TYPES

Ontology( $C, P, I$ )

### CONSTRUCTORS

consOntology(classes: $\mathbb{P}(C)$ , properties: $\mathbb{P}(P)$ , instances: $\mathbb{P}(I)$ ,  
classProperties: $\mathbb{P}(C \times P)$ , classInstances: $\mathbb{P}(C \times I)$ ,  
classAssociations: $\mathbb{P}(C \times P \times C)$ , instanceAssociations: $\mathbb{P}(I \times P$   
 $\times I)$ )

### OPERATORS

getClasses <expression> (o: Ontology( $C, P, I$ ))

**direct definition**

classes(o)

getProperties <expression> (o: Ontology( $C, P, I$ ))

**direct definition**

properties(o)

getInstances <expression> (o: Ontology( $C, P, I$ ))

**direct definition**

instances(o)

isWDClassProperites <predicate> (o: Ontology( $C, P, I$ ))

**direct definition**

classProperties(o)  $\in$  getClasses(o)  $\leftrightarrow$  getProperties(o)

getClassProperties <expression> (o: Ontology( $C, P, I$ ))

**well-definedness** isWDClassProperites(o)

**direct definition**

classProperties(o)

isWDClassInstances <predicate> (o: Ontology( $C, P, I$ ))

**direct definition**

classInstances(o)  $\in$  getClasses(o)  $\leftrightarrow$  getInstances(o)

getClassInstances <expression> (o: Ontology( $C, P, I$ ))

**well-definedness** isWDClassInstances(o)

**direct definition**

classInstances(o)

isWDClassAssociations <predicate> (o: Ontology( $C, P, I$ ))

**well-definedness** isWDClassProperites(o)

**direct definition**

classAssociations(o)  $\in$  getClassProperties(o)  $\rightarrow$  classes(o)

getClassAssociations <expression> (o: Ontology( $C, P, I$ ))

**well-definedness** isWDClassAssociations(o)

**direct definition**

classAssociations(o)

isWDInstancesAssociations <predicate> (o: Ontology( $C, P, I$ ))

**well-definedness** isWDClassProperites(o), isWDClassInstances(o),

isWDClassAssociations(o)

**direct definition**

instanceAssociations(o)  $\subseteq$  instances(o)  $\times$  properties(o)  $\times$

$$\begin{aligned}
& \text{instances}(o) \wedge \\
& \text{instanceAssociations}(o) \subseteq \{ i1 \mapsto p \mapsto i2 \mid i1 \in I \wedge p \in P \wedge \\
& \quad i2 \in I \wedge i1 \mapsto p \mapsto i2 \in \text{instances}(o) \times \text{properties}(o) \times \\
& \quad \text{instances}(o) \wedge \\
& \quad (\exists c1, c2 \cdot c1 \in C \wedge c2 \in C \wedge \{c1, c2\} \subseteq \\
& \quad \quad \text{getClasses}(o) \Rightarrow \\
& \quad \quad (c1 \mapsto p \mapsto c2 \in \text{getClassAssociations}(o) \wedge p \\
& \quad \quad \in \text{getClassProperties}(o)[\{c1\}] \wedge i1 \in \\
& \quad \quad \text{getClassInstances}(o)[\{c1\}] \wedge i2 \in \\
& \quad \quad \text{getClassInstances}(o)[\{c2\}])) \\
& \quad \} \\
\text{getInstanceAssociations} & \langle \textit{expression} \rangle (o: \text{Ontology}(C, P, I)) \\
& \textbf{well-definedness} \text{ isWDInstancesAssociations}(o) \\
& \textbf{direct definition} \\
& \quad \text{instanceAssociations}(o) \\
\text{isWDOntology} & \langle \textit{predicate} \rangle (o: \text{Ontology}(C, P, I)) \\
& \textbf{direct definition} \\
& \quad \text{isWDClassProperites}(o) \wedge \text{isWDClassInstances}(o) \wedge \\
& \quad \text{isWDClassAssociations}(o) \wedge \text{isWDInstancesAssociations}(o) \\
\text{isItWDOntology} & \langle \textit{predicate} \rangle (o: \text{Ontology}(C, P, I)) \\
& \textbf{well-definedness} \text{ isWDOntology}(o) \\
& \textbf{direct definition} \\
& \quad \top \\
\text{ontologyContainsClasses} & \langle \textit{predicate} \rangle (o: \text{Ontology}(C, P, I), cc: \mathbb{P}(C)) \\
& \textbf{well-definedness} \text{ isWDOntology}(o), cc \neq \emptyset \\
& \textbf{direct definition} \\
& \quad cc \subseteq \text{getClasses}(o) \\
\text{ontologyContainsProperties} & \langle \textit{predicate} \rangle (o: \text{Ontology}(C, P, I), pp: \mathbb{P}(P)) \\
& \textbf{well-definedness} \text{ isWDOntology}(o), pp \neq \emptyset \\
& \textbf{direct definition} \\
& \quad pp \subseteq \text{getProperties}(o) \\
\text{ontologyContainsInstances} & \langle \textit{predicate} \rangle (o: \text{Ontology}(C, P, I), ii: \mathbb{P}(I)) \\
& \textbf{well-definedness} \text{ isWDOntology}(o), ii \neq \emptyset \\
& \textbf{direct definition} \\
& \quad ii \subseteq \text{getInstances}(o) \\
\text{getPropertiesOfaClass} & \langle \textit{expression} \rangle (o: \text{Ontology}(C, P, I), c: C) \\
& \textbf{well-definedness} \text{ isWDOntology}(o), \text{ontologyContainsClasses}(o, \{c \\
& \quad \}) \\
& \textbf{direct definition} \\
& \quad \text{getClassProperties}(o)[\{c\}] \\
\text{getInstancesOfaClass} & \langle \textit{expression} \rangle (o: \text{Ontology}(C, P, I), c: C) \\
& \textbf{well-definedness} \text{ isWDOntology}(o), \text{ontologyContainsClasses}(o, \{c \\
& \quad \}) \\
& \textbf{direct definition} \\
& \quad \text{getClassInstances}(o)[\{c\}] \\
\text{isPropertyOfTheClass} & \langle \textit{predicate} \rangle (o: \text{Ontology}(C, P, I), c: C, p: P) \\
& \textbf{well-definedness} \text{ isWDOntology}(o), \text{ontologyContainsClasses}(o, \{c \\
& \quad \}), \text{ontologyContainsProperties}(o, \{p\}) \\
& \textbf{direct definition}
\end{aligned}$$

$p \in \text{getPropertiesOfaClass}(o, c)$   
**getPropertyRangeClasses** *<expression>* (o: Ontology(C, P, I), p: P)  
**well-definedness** isWDOntology(o), ontologyContainsProperties(o, {p})  
**direct definition**  
 $\{c2 \mid \forall c1 \cdot c1 \in \text{getClasses}(o) \Rightarrow c1 \mapsto p \mapsto c2 \in \text{getClassAssociations}(o)\}$   
**getPropertyRangeInstances** *<expression>* (o: Ontology(C, P, I), p: P)  
**well-definedness** isWDOntology(o), ontologyContainsProperties(o, {p})  
**direct definition**  
 $\{j \mid \forall i \cdot i \in \text{getInstances}(o) \Rightarrow i \mapsto p \mapsto j \in \text{getInstanceAssociations}(o)\}$   
**getValueOfAInstanceProperty** *<expression>* (o: Ontology(C, P, I), i: I, p: P)  
**well-definedness** isWDOntology(o), ontologyContainsProperties(o, {p}), ontologyContainsInstances(o, {i})  
**direct definition**  
 $\text{getInstanceAssociations}(o)[\{i \mapsto p\}]$   
**getClassesOfInstance** *<expression>* (o: Ontology(C, P, I), i: I)  
**well-definedness** isWDOntology(o), ontologyContainsInstances(o, {i})  
**direct definition**  
 $\text{getClassInstances}(o)^{-1}[\{i\}]$   
**classContainsProperties** *<predicate>* (o: Ontology(C, P, I), c: C, pp:  $\mathbb{P}(P)$ )  
**well-definedness** isWDOntology(o), ontologyContainsClasses(o, {c}), ontologyContainsProperties(o, pp)  
**direct definition**  
 $pp \subseteq \text{getPropertiesOfaClass}(o, c)$   
**classContainsInstances** *<predicate>* (o: Ontology(C, P, I), c: C, ii:  $\mathbb{P}(I)$ )  
**well-definedness** isWDOntology(o), ontologyContainsClasses(o, {c}), ontologyContainsInstances(o, ii)  
**direct definition**  
 $ii \subseteq \text{getInstancesOfaClass}(o, c)$   
**instanceHasPropertyValue** *<predicate>* (o: Ontology(C, P, I), i: I, p: P, v: I)  
**well-definedness** isWDOntology(o), ontologyContainsInstances(o, {i, v}), ontologyContainsProperties(o, {p})  
**direct definition**  
 $v \in \text{getValueOfAInstanceProperty}(o, i, p)$   
**isA** *<predicate>* (o: Ontology(C, P, I), c1: C, c2: C)  
**well-definedness** isWDOntology(o), ontologyContainsClasses(o, {c1, c2})  
**direct definition**  
 $\text{getInstancesOfaClass}(o, c1) \subseteq \text{getInstancesOfaClass}(o, c2)$   
**addInstancesToOntology** *<expression>* (o: Ontology(C, P, I), ii:  $\mathbb{P}(I)$ )  
**well-definedness** isWDOntology(o),  $\neg$  ontologyContainsInstances(o, ii)

**direct definition**  
 $\text{consOntology}(\text{getClasses}(o), \text{getProperties}(o), \text{getInstances}(o) \cup ii, \text{getClassProperties}(o), \text{getClassInstances}(o), \text{getClassAssociations}(o), \text{getInstanceAssociations}(o))$

**addInstancesToAClass** *<expression>* (o: Ontology(C, P, I), c: C, ii:  $\mathbb{P}(I)$ )  
**well-definedness**  $\text{isWDOntology}(o), \text{ontologyContainsClasses}(o, \{c\}), \text{ontologyContainsInstances}(o, ii), \neg \text{classContainsInstances}(o, c, ii)$

**direct definition**  
 $\text{consOntology}(\text{getClasses}(o), \text{getProperties}(o), \text{getInstances}(o), \text{getClassProperties}(o), \text{getClassInstances}(o) \cup (\{c\} \times ii), \text{getClassAssociations}(o), \text{getInstanceAssociations}(o))$

**ontologyContainsIpv** *<predicate>* (o: Ontology(C, P, I), ipvs:  $\mathbb{P}(I \times P \times I)$ )  
**well-definedness**  $\text{isWDOntology}(o)$

**direct definition**  
 $\text{ipvs} \subseteq \text{getInstanceAssociations}(o)$

**isWDAddValueOfAInstanceProperty** *<predicate>* (o: Ontology(C, P, I), ipvs:  $\mathbb{P}(I \times P \times I), i: I, p: P, v: I$ )  
**direct definition**  
 $\top$

**addValueOfAInstanceProperty** *<expression>* (o: Ontology(C, P, I), ipvs:  $\mathbb{P}(I \times P \times I), i: I, p: P, v: I$ )  
**well-definedness**  $\text{isWDAddValueOfAInstanceProperty}(o, \text{ipvs}, i, p, v)$

**direct definition**  
 $\text{ipvs} \cup \{i \mapsto p \mapsto v\}$

**isWDRemoveValueOfAInstanceProperty** *<predicate>* (o: Ontology(C, P, I), ipvs:  $\mathbb{P}(I \times P \times I), i: I, p: P, v: I$ )  
**direct definition**  
 $\top$

**removeValueOfAInstanceProperty** *<expression>* (o: Ontology(C, P, I), ipvs:  $\mathbb{P}(I \times P \times I), i: I, p: P, v: I$ )  
**well-definedness**  $\text{isWDRemoveValueOfAInstanceProperty}(o, \text{ipvs}, i, p, v)$

**direct definition**  
 $\text{ipvs} \setminus \{i \mapsto p \mapsto v\}$

**isVariableOfOntology** *<predicate>* (o: Ontology(C, P, I), ipvs:  $\mathbb{P}(I \times P \times I)$ )  
**well-definedness**  $\text{isWDOntology}(o)$

**direct definition**  
 $\text{ipvs} \subseteq \text{instances}(o) \times \text{properties}(o) \times \text{instances}(o) \wedge$   
 $\text{ipvs} \subseteq \{ i1 \mapsto p \mapsto i2 \mid i1 \in I \wedge p \in P \wedge i2 \in I \wedge i1 \mapsto p \mapsto i2 \in \text{instances}(o) \times \text{properties}(o) \times \text{instances}(o) \wedge$   
 $(\exists c1, c2 \cdot c1 \in C \wedge c2 \in C \wedge \{c1, c2\} \subseteq \text{getClasses}(o) \Rightarrow$   
 $(c1 \mapsto p \mapsto c2 \in \text{getClassAssociations}(o) \wedge p \in \text{getClassProperties}(o)[\{c1\}] \wedge i1 \in$

```

                                getClassInstances(o)[{c1}] ∧ i2 ∈
                                getClassInstances(o)[{c2}])
                                }
isWDInstanceHasPropertyValuei <predicate> (o: Ontology(C, P, I), ipvs:
    ℙ(I × P × I), i: I, p: P, v: I)
direct definition
    isWDOntology(o) ∧ isVariableOfOntology(o, ipvs) ∧
    ontologyContainsInstances(o, {i, v}) ∧
    ontologyContainsProperties(o, {p})
instanceHasPropertyValuei <predicate> (o: Ontology(C, P, I), ipvs: ℙ(I ×
    P × I), i: I, p: P, v: I)
well-definedness isWDInstanceHasPropertyValuei(o, ipvs, i, p, v
    )
direct definition
    v ∈ ipvs[{i ↦ p}]
THEOREMS
thm1:
    ∀o, c1, c2, c3. o ∈ Ontology(C, P, I) ∧ isWDOntology(o) ∧ c1 ∈
    C ∧ c2 ∈ C ∧ c3 ∈ C ∧ ontologyContainsClasses(o, {c1, c2,
    c3})
    ⇒ (isA(o, c1, c2) ∧ isA(o, c2, c3) ⇒ isA(o, c1, c3))
thm2:
    ∀o, cs1, cs2 . o ∈ Ontology(C, P, I) ∧ isWDOntology(o) ∧ cs1 ⊆
    C ∧ cs2 ⊆ C ∧ cs1 ≠ ∅ ∧ cs2 ≠ ∅ ∧ ontologyContainsClasses(o,
    cs1) ∧ ontologyContainsClasses(o, cs2)
    ⇒ ( ontologyContainsClasses(o, cs1 ∪ cs2) )
END

```

Listing 1: Event-B machine modelling the WXR user interface

# **ARINC 661 Theory (Subsection 7.1)**

# Ontological theory of a chunk of ARINC 661 specification

## THEORY

### IMPORT THEORY PROJECTS

```
/OntologiesTheoryFMICS THEORIES /OntologiesTheoryFMICS /  
  OntologiesTheory.dtf | org.eventb.theory.core.deployedTheoryRoot  
#OntologiesTheory
```

### AXIOMATIC DEFINITIONS ARINC661Axiomatisation:

**TYPES** ARINC661Classes, ARINC66Properties, ARINC661Instances

### OPERATORS

```
ARINC661_BOOL <expression> () : ARINC661Classes  
ARINC661_BOOL_EXTENDED <expression> () : ARINC661Classes  
A661_TRUE <expression> () : ARINC661Instances  
A661_FALSE <expression> () : ARINC661Instances  
A661_TRUE_WITH_VALIDATION <expression> () : ARINC661Instances  
ARINC661_STRING_CLASS <expression> () : ARINC661Classes  
ARINC661_STRING_INSTANCES <expression> () :  $\mathbb{P}$ (ARINC661Instances  
)  
A661_EDIT_BOX_NUMERIC_VALUES_CLASS <expression> () :  
  ARINC661Classes  
A661_EDIT_BOX_NUMERIC_VALUES <expression> () :  $\mathbb{P}$ (  
  ARINC661Instances)  
A661_EDIT_BOX_NUMERIC_ADMISSIBLE_VALUES <expression> () :  $\mathbb{P}$ (  
  ARINC661Instances)  
WidgetTypeClass <expression> () : ARINC661Classes  
WidgetIdentClass <expression> () : ARINC661Classes  
WidgetTypeInstances <expression> () :  $\mathbb{P}$ (ARINC661Instances)  
WidgetIdentInstances <expression> () :  $\mathbb{P}$ (ARINC661Instances)  
NoneIdent <expression> () : ARINC661Instances  
Label <expression> () : ARINC661Classes  
RadioBox <expression> () : ARINC661Classes  
CheckBox <expression> () : ARINC661Classes  
PushButton <expression> () : ARINC661Classes  
ToggleButton <expression> () : ARINC661Classes  
ToggleState <expression> () : ARINC661Classes  
EditBoxNumeric <expression> () : ARINC661Classes  
hasWidgetType <expression> () : ARINC66Properties  
hasWidgetIdent <expression> () : ARINC66Properties  
hasParentIdent <expression> () : ARINC66Properties  
hasVisible <expression> () : ARINC66Properties  
hasEnable <expression> () : ARINC66Properties  
hasLabelStringForLabel <expression> () : ARINC66Properties  
hasAnonymous <expression> () : ARINC66Properties  
hasChildrenForRadioBox <expression> () : ARINC66Properties  
hasCheckBoxState <expression> () : ARINC66Properties  
hasLabelStringForCheckBox <expression> () : ARINC66Properties  
hasLabelStringForPushButton <expression> () : ARINC66Properties  
hasToggleButtonState <expression> () : ARINC66Properties
```

**hasLabelStringForToggleButton**  $\langle expression \rangle$  () : ARINC66Properties  
**hasValue**  $\langle expression \rangle$  () : ARINC66Properties  
**LabelInstances**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances})$   
**RadioBoxInstances**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances})$   
**CheckBoxInstances**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances})$   
**PushButtonInstances**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances})$   
**ToggleButtonInstances**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances})$   
**EditBoxNumericInstances**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances})$   
**CheckBoxStateClass**  $\langle expression \rangle$  () : ARINC661Classes  
**ToggleButtonStateClass**  $\langle expression \rangle$  () : ARINC661Classes  
**SELECTED**  $\langle expression \rangle$  () : ARINC661Instances  
**UNSELECTED**  $\langle expression \rangle$  () : ARINC661Instances  
**WidgetsInstances**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances})$   
**wellBuiltClassProperties**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Classes} \times \text{ARINC66Properties})$   
**wellBuiltClassAssociations**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Classes} \times \text{ARINC66Properties} \times \text{ARINC661Classes})$   
**wellbuiltTypesElements**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Classes} \times \text{ARINC661Instances})$   
**isWDRadioBox**  $\langle predicate \rangle$  (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances) ) :  
**well-definedness** isWDOntology(o)  
**isWDEditBoxNumeric**  $\langle predicate \rangle$  (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances) ) :  
**well-definedness** isWDOntology(o)  
**isWDARINC661Ontology**  $\langle predicate \rangle$  (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances) ) :  
**consARINC661Ontology**  $\langle expression \rangle$  (ii:  $\mathbb{P}(\text{ARINC661Instances})$ ,cii:  $\mathbb{P}(\text{ARINC661Classes} \times \text{ARINC661Instances})$ ,ipvs:  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$ ) :  
Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances)  
**well-definedness** isWDARINC661Ontology(consOntology(ARINC661Classes, ARINC66Properties, ii, wellBuiltClassProperties, wellbuiltTypesElements  $\cup$  cii, wellBuiltClassAssociations, ipvs))  
**isVariableOfARINC661Ontology**  $\langle predicate \rangle$  (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances) ,ui:  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$ ) :  
**well-definedness** isWDARINC661Ontology(o)

## AXIOMS

axml:

partition(ARINC661Classes, {ARINC661\_BOOL}, {ARINC661\_STRING\_CLASS}, {WidgetTypeClass}, {WidgetIdentClass}, {A661\_EDIT\_BOX\_NUMERIC\_VALUES\_CLASS}, {Label}, {RadioBox}, {CheckBox}, {PushButton}, {ToggleButton}, {EditBoxNumeric}, {CheckBoxStateClass}, {

ToggleButtonStateClass })

*axm2:*

partition (ARINC66Properties, {hasWidgetType}, {hasWidgetIdent  
}, {hasParentIdent}, {hasVisible}, {hasEnable},  
{hasLabelStringForLabel}, {hasAnonymous}, {  
hasChildrenForRadioBox}, {hasCheckButtonState}, {  
hasToggleButtonState},  
{hasLabelStringForCheckButton}, {  
hasLabelStringForPushButton}, {  
hasLabelStringForToggleButton}, {hasParentIdent}, {  
hasValue})

*axm3:*

A661\_EDIT\_BOX\_NUMERIC\_VALUES  $\neq \emptyset$   $\wedge$  ARINC661\_STRING\_INSTANCES  
 $\neq \emptyset$   $\wedge$  WidgetIdentInstances  $\neq \emptyset$   $\wedge$  WidgetTypeInstances  $\neq \emptyset$   $\wedge$   
LabelInstances  $\neq \emptyset$   $\wedge$  RadioBoxInstances  $\neq \emptyset$   $\wedge$   
CheckButtonInstances =  $\emptyset$   $\wedge$  PushButtonInstances  $\neq \emptyset$   $\wedge$   
ToggleButtonInstances  $\neq \emptyset$   $\wedge$  EditTextNumericInstances  $\neq \emptyset$

*axm4:*

NoneIdent  $\in$  WidgetIdentInstances

*axm5:*

partition (ARINC661Instances, {A661\_TRUE}, {A661\_FALSE}, {  
A661\_TRUE\_WITH\_VALIDATION}, ARINC661\_STRING\_INSTANCES,  
WidgetTypeInstances, WidgetIdentInstances,  
A661\_EDIT\_BOX\_NUMERIC\_VALUES,  
LabelInstances, RadioBoxInstances, CheckButtonInstances  
, PushButtonInstances, EditTextNumericInstances, {  
SELECTED}, {UNSELECTED})

*axm6:*

WidgetsInstances = LabelInstances  $\cup$  RadioBoxInstances  $\cup$   
CheckButtonInstances  $\cup$  PushButtonInstances  $\cup$   
ToggleButtonInstances  $\cup$  EditTextNumericInstances

*axm7:*

A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES  $\subseteq$   
A661\_EDIT\_BOX\_NUMERIC\_VALUES  $\wedge$   
A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES  $\neq \emptyset$

*wellBuiltClassProperties:*

wellBuiltClassProperties = ({Label}  $\times$  {hasWidgetType,  
hasWidgetIdent, hasParentIdent, hasVisible,  
hasLabelStringForLabel, hasAnonymous})  $\cup$   
({RadioBox}  $\times$  {hasWidgetType, hasWidgetIdent,  
hasParentIdent, hasVisible, hasEnable,  
hasChildrenForRadioBox})  $\cup$   
({CheckButton}  $\times$  {hasWidgetType, hasWidgetIdent,  
hasParentIdent, hasVisible, hasEnable,  
hasCheckButtonState,  
hasLabelStringForCheckButton})  $\cup$   
({ToggleButton}  $\times$  {hasWidgetType,  
hasWidgetIdent, hasParentIdent, hasVisible,  
hasEnable, hasToggleButtonState,

hasLabelStringForToggleButton))  $\cup$   
 ({PushButton}  $\times$  {hasWidgetType, hasWidgetIdent,  
 hasParentIdent, hasVisible, hasEnable,  
 hasLabelStringForPushButton})  $\cup$   
 ({EditBoxNumeric}  $\times$  {hasWidgetType,  
 hasWidgetIdent, hasParentIdent, hasVisible,  
 hasEnable, hasValue})

*wellBuiltClassAssociations:*

wellBuiltClassAssociations = ({Label  $\mapsto$  hasWidgetType  $\mapsto$   
 WidgetTypeClass, Label  $\mapsto$  hasWidgetIdent  $\mapsto$   
 WidgetIdentClass, Label  $\mapsto$  hasParentIdent  $\mapsto$   
 WidgetIdentClass,  
 Label  $\mapsto$  hasVisible  $\mapsto$  ARINC661\_BOOL, Label  $\mapsto$   
 hasLabelStringForLabel  $\mapsto$   
 ARINC661\_STRING\_CLASS, Label  $\mapsto$   
 hasAnonymous  $\mapsto$  ARINC661\_BOOL})  
 $\cup$   
 ({RadioButton  $\mapsto$  hasWidgetType  $\mapsto$  WidgetTypeClass,  
 RadioButton  $\mapsto$  hasWidgetIdent  $\mapsto$   
 WidgetIdentClass, RadioButton  $\mapsto$   
 hasParentIdent  $\mapsto$  WidgetIdentClass,  
 RadioButton  $\mapsto$  hasVisible  $\mapsto$  ARINC661\_BOOL,  
 RadioButton  $\mapsto$  hasEnable  $\mapsto$   
 ARINC661\_BOOL\_EXTENDED, RadioButton  $\mapsto$   
 hasChildrenForRadioButton  $\mapsto$  WidgetIdentClass  
 })  
 $\cup$   
 ({CheckBox  $\mapsto$  hasWidgetType  $\mapsto$   
 WidgetTypeClass, CheckBox  $\mapsto$   
 hasWidgetIdent  $\mapsto$  WidgetIdentClass,  
 CheckBox  $\mapsto$  hasParentIdent  $\mapsto$   
 WidgetIdentClass,  
 CheckBox  $\mapsto$  hasVisible  $\mapsto$  ARINC661\_BOOL,  
 CheckBox  $\mapsto$  hasEnable  $\mapsto$   
 ARINC661\_BOOL\_EXTENDED, CheckBox  $\mapsto$   
 hasCheckBoxState  $\mapsto$   
 CheckBoxStateClass,  
 CheckBox  $\mapsto$  hasLabelStringForCheckBox  
 $\mapsto$  ARINC661\_STRING\_CLASS})  
 $\cup$   
 ({ToggleButton  $\mapsto$  hasWidgetType  $\mapsto$   
 WidgetTypeClass, ToggleButton  $\mapsto$   
 hasWidgetIdent  $\mapsto$  WidgetIdentClass,  
 ToggleButton  $\mapsto$  hasParentIdent  $\mapsto$   
 WidgetIdentClass,  
 ToggleButton  $\mapsto$  hasVisible  $\mapsto$  ARINC661\_BOOL,  
 ToggleButton  $\mapsto$  hasEnable  $\mapsto$   
 ARINC661\_BOOL\_EXTENDED, ToggleButton  $\mapsto$   
 hasToggleButtonState  $\mapsto$

ToggleButtonStateClass ,  
 ToggleButton  $\mapsto$  hasLabelStringForToggleButton  
 $\mapsto$  ARINC661\_STRING\_CLASS })  
 $\cup$   
 ({ PushButton  $\mapsto$  hasWidgetType  $\mapsto$   
 WidgetTypeClass , PushButton  $\mapsto$   
 hasWidgetIdent  $\mapsto$  WidgetIdentClass ,  
 PushButton  $\mapsto$  hasParentIdent  $\mapsto$   
 WidgetIdentClass ,  
 PushButton  $\mapsto$  hasVisible  $\mapsto$  ARINC661\_BOOL ,  
 PushButton  $\mapsto$  hasEnable  $\mapsto$   
 ARINC661\_BOOL\_EXTENDED ,  
 PushButton  $\mapsto$  hasLabelStringForPushButton  $\mapsto$   
 ARINC661\_STRING\_CLASS })  
 $\cup$   
 ({ EditTextNumeric  $\mapsto$  hasWidgetType  $\mapsto$   
 WidgetTypeClass , EditTextNumeric  $\mapsto$   
 hasWidgetIdent  $\mapsto$  WidgetIdentClass ,  
 EditTextNumeric  $\mapsto$  hasParentIdent  $\mapsto$   
 WidgetIdentClass ,  
 EditTextNumeric  $\mapsto$  hasVisible  $\mapsto$   
 ARINC661\_BOOL , EditTextNumeric  $\mapsto$   
 hasEnable  $\mapsto$  ARINC661\_BOOL\_EXTENDED ,  
 EditTextNumeric  $\mapsto$  hasValue  $\mapsto$   
 A661\_EDIT\_BOX\_NUMERIC\_VALUES\_CLASS })

*wellbuiltTypesElements:*

wellbuiltTypesElements = ({ ARINC661\_BOOL }  $\times$  { A661\_TRUE ,  
 A661\_FALSE })  $\cup$   
 ({ ARINC661\_BOOL\_EXTENDED }  $\times$  { A661\_TRUE ,  
 A661\_FALSE , A661\_TRUE\_WITH\_VALIDATION })  $\cup$   
 ({ ARINC661\_STRING\_CLASS }  $\times$   
 ARINC661\_STRING\_INSTANCES)  $\cup$   
 ({ WidgetTypeClass }  $\times$  WidgetTypeInstances)  $\cup$   
 ({ WidgetIdentClass }  $\times$  WidgetIdentInstances)  $\cup$   
 ({ A661\_EDIT\_BOX\_NUMERIC\_VALUES\_CLASS }  $\times$   
 A661\_EDIT\_BOX\_NUMERIC\_VALUES)  $\cup$   
 ({ CheckButtonStateClass }  $\times$  { SELECTED ,  
 UNSELECTED })

*consARINC661Ontology:*

$\forall ii , cii , ipvs \cdot ii \in \mathbb{P}(\text{ARINC661Instances}) \wedge cii \in \mathbb{P}(\text{ARINC661Classes} \times \text{ARINC661Instances}) \wedge ipvs \in \mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$   
 $\wedge$   
 wellbuiltTypesElements  $\cap$  cii =  $\emptyset \wedge ii \subseteq$   
 WidgetsInstances  
 $\Rightarrow$   
 consARINC661Ontology(ii , cii , ipvs) = consOntology(  
 ARINC661Classes , ARINC66Properties , ii ,  
 wellBuiltClassProperties , wellbuiltTypesElements

$$\cup \text{cii}, \text{wellBuiltClassAssociations}, \text{ipvs})$$

*isWDRadioButton:*

$$\forall o. o \in \text{Ontology}(\text{ARINC661Classes}, \text{ARINC66Properties}, \text{ARINC661Instances}) \Rightarrow$$

$$(\text{isWDRadioButton}(o) \Leftrightarrow$$

$$(\forall \text{rb}, \text{b1}, \text{b2}. \text{rb} \in \text{RadioButtonInstances} \wedge \text{b1} \in \text{CheckBoxInstances} \wedge \text{b2} \in \text{CheckBoxInstances} \wedge$$

$$\text{rb} \mapsto \text{hasChildrenForRadioButton} \mapsto \text{b1} \in \text{getInstanceAssociations}(o) \wedge \text{rb} \mapsto \text{hasChildrenForRadioButton} \mapsto \text{b2} \in \text{getInstanceAssociations}(o)$$

$$\Rightarrow (\text{b1} \mapsto \text{hasCheckBoxState} \mapsto \text{SELECTED} \in \text{getInstanceAssociations}(o) \wedge \text{b2} \mapsto \text{hasCheckBoxState} \mapsto \text{SELECTED} \in \text{getInstanceAssociations}(o) \Rightarrow \text{b1} = \text{b2}))$$

$$) \wedge$$

$$(\forall \text{rb}, \text{b1}, \text{b2}. \text{rb} \in \text{RadioButtonInstances} \wedge \text{b1} \in \text{ToggleButtonInstances} \wedge \text{b2} \in \text{ToggleButtonInstances} \wedge$$

$$\text{rb} \mapsto \text{hasChildrenForRadioButton} \mapsto \text{b1} \in \text{getInstanceAssociations}(o) \wedge \text{rb} \mapsto \text{hasChildrenForRadioButton} \mapsto \text{b2} \in \text{getInstanceAssociations}(o)$$

$$\Rightarrow (\text{b1} \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{getInstanceAssociations}(o) \wedge \text{b2} \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{getInstanceAssociations}(o) \Rightarrow \text{b1} = \text{b2}))$$

$$))$$

*isWDEditBoxNumeric:*

$$\forall o. o \in \text{Ontology}(\text{ARINC661Classes}, \text{ARINC66Properties}, \text{ARINC661Instances}) \Rightarrow$$

$$(\text{isWDRadioButton}(o) \Leftrightarrow (\forall \text{ed}, \text{v}. \text{ed} \mapsto \text{hasValue} \mapsto \text{v} \in \text{getInstanceAssociations}(o) \Rightarrow \text{v} \in \text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES}))$$

*isWDARINC661Ontology:*

$$\forall o. o \in \text{Ontology}(\text{ARINC661Classes}, \text{ARINC66Properties}, \text{ARINC661Instances}) \Rightarrow$$

$$(\text{isWDOntology}(o) \wedge \text{isWDRadioButton}(o) \wedge \text{isWDEditBoxNumeric}(o) \Rightarrow \text{isWDARINC661Ontology}(o))$$

*isVariableOfARINC661Ontology:*

$$\forall o, \text{ipvs}. o \in \text{Ontology}(\text{ARINC661Classes}, \text{ARINC66Properties}, \text{ARINC661Instances}) \wedge \text{ipvs} \in \mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances}) \Rightarrow$$

$$(\text{isWDARINC661Ontology}(\text{consOntology}(\text{getClasses}(o), \text{getProperties}(o), \text{getInstances}(o), \text{getClassProperties}(o), \text{getClassInstances}(o), \text{getClassAssociations}(o), \text{getInstanceAssociations}(o)))$$

$$\Rightarrow \text{isVariableOfARINC661Ontology}(o, \text{ipvs}))$$

## THEOREMS

*thm1:*

$$\begin{aligned} \forall ii, cii, ipvs \cdot ii \in \mathbb{P}(\text{ARINC661Instances}) \wedge cii \in \mathbb{P}(\text{ARINC661Classes} \times \text{ARINC661Instances}) \wedge ipvs \in \mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances}) \wedge \\ \text{wellbuiltTypesElements} \cap cii = \emptyset \wedge ii \subseteq \text{WidgetsInstances} \\ \Rightarrow \\ \text{isWDClassProperites}(\text{consARINC661Ontology}(ii, cii, ipvs)) \end{aligned}$$

*thm2:*

$$\begin{aligned} \forall ii, cii, ipvs \cdot ii \in \mathbb{P}(\text{ARINC661Instances}) \wedge cii \in \mathbb{P}(\text{ARINC661Classes} \times \text{ARINC661Instances}) \wedge ipvs \in \mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances}) \wedge \\ \text{wellbuiltTypesElements} \cap cii = \emptyset \wedge ii \subseteq \text{WidgetsInstances} \\ \Rightarrow \\ \text{isWDClassAssociations}(\text{consARINC661Ontology}(ii, cii, ipvs)) \end{aligned}$$

*thm3:*

$$\begin{aligned} \forall ii, cii, ipvs \cdot ii \in \mathbb{P}(\text{ARINC661Instances}) \wedge cii \in \mathbb{P}(\text{ARINC661Classes} \times \text{ARINC661Instances}) \wedge ipvs \in \mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances}) \wedge \\ \text{wellbuiltTypesElements} \cap cii = \emptyset \wedge ii \subseteq \text{WidgetsInstances} \\ \Rightarrow \\ (\text{isWDClassProperites}(\text{consARINC661Ontology}(ii, cii, ipvs)) \wedge \text{isWDClassAssociations}(\text{consARINC661Ontology}(ii, cii, ipvs))) \end{aligned}$$

**END**

Listing 2: Event-B machine modelling the WXR user interface

## **WXR Theory (Subsection 7.2)**

## Event-B theory of contextual information used in WXR model

### THEORY

#### IMPORT THEORY PROJECTS

/WidgetTheory **THEORIES** /WidgetTheory/ARINC661Theory.dtf|org.  
eventb.theory.core.deployedTheoryRoot#ARINC661Theory

#### AXIOMATIC DEFINITIONS WXRUIDescriptoinAxiomatisationon:

#### OPERATORS

**A661WXRontology**  $\langle expression \rangle ()$  :  $\text{Ontology}(\text{ARINC661Classes}, \text{ARINC66Properties}, \text{ARINC661Instances})$   
**Instances**  $\langle expression \rangle ()$  :  $\mathbb{P}(\text{ARINC661Instances})$   
**ClassInstances**  $\langle expression \rangle ()$  :  $\mathbb{P}(\text{ARINC661Classes} \times \text{ARINC661Instances})$   
**InstanceAssociation**  $\langle expression \rangle ()$  :  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$   
**MODESELECTIONLabel**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**OFF1Label**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**STDBYLabel**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**TSTLabel**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**WXONLabel**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**WXALabel**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**TITLSELECTIONLabel**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**CRTL1Label**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**AUTOLabel**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**STABILIZATIONLabel**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**CRTL2Label**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**OFF2Label**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**TITLANGLELabel**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**MODESELECTIONString**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**STDBYString**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**TSTString**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**WXONString**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**WXAStrng**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**OFF1String**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**TITLSELECTIONString**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**CRTL1String**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**AUTOString**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**STABILIZATIONString**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**CRTL2String**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**OFF2String**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**TITLANGLEString**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**OFF1CheckBox**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**STDBYCheckBox**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**TSTCheckBox**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**TiltSectionCTRLToggleButon**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**TiltSectionAUTOToggleButon**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**StabilizationCRTLToggleButon**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$   
**StabilizationOFFToggleButon**  $\langle expression \rangle ()$  :  $\text{ARINC661Instances}$

**WXONCheckBox** <*expression*> () : ARINC661Instances  
**WXACheckButton** <*expression*> () : ARINC661Instances  
**TITLANGLEEditNumericBox** <*expression*> () : ARINC661Instances  
**WXRLabels** <*expression*> () :  $\mathbb{P}$ (ARINC661Instances)  
**WXRcheckButtons** <*expression*> () :  $\mathbb{P}$ (ARINC661Instances)  
**WXRToggleButtons** <*expression*> () :  $\mathbb{P}$ (ARINC661Instances)  
**WXRradioBoxes** <*expression*> () :  $\mathbb{P}$ (ARINC661Instances)  
**WXRradioBoxModeSelection** <*expression*> () : ARINC661Instances  
**WXRradioBoxTiltSelection** <*expression*> () : ARINC661Instances  
**WXRradioBoxStabilization** <*expression*> () : ARINC661Instances  
**MODESELECTIONLabelWidgetIdent** <*expression*> () :  
ARINC661Instances  
**OFF1LabelWidgetIdent** <*expression*> () : ARINC661Instances  
**STDBYLabelWidgetIdent** <*expression*> () : ARINC661Instances  
**TSTLabelWidgetIdent** <*expression*> () : ARINC661Instances  
**WXONLabelWidgetIdent** <*expression*> () : ARINC661Instances  
**WXALabelWidgetIdent** <*expression*> () : ARINC661Instances  
**TITLLabelWidgetIdent** <*expression*> () : ARINC661Instances  
**CRTL1LabelWidgetIdent** <*expression*> () : ARINC661Instances  
**AUTOSLabelWidgetIdent** <*expression*> () : ARINC661Instances  
**STABILIZATIONLabelWidgetIdent** <*expression*> () : ARINC661Instances  
**CRTL2LabelWidgetIdent** <*expression*> () : ARINC661Instances  
**OFF2LabelWidgetIdent** <*expression*> () : ARINC661Instances  
**TITLANGLELabelWidgetIdent** <*expression*> () : ARINC661Instances  
**OFF1CheckBoxWidgetIdent** <*expression*> () : ARINC661Instances  
**STDBYCheckBoxWidgetIdent** <*expression*> () : ARINC661Instances  
**TSTCheckBoxWidgetIdent** <*expression*> () : ARINC661Instances  
**TITLSELECTIONLabelWidgetIdent** <*expression*> () : ARINC661Instances  
**TiltSectionCTRLToggleButtonWidgetIdent** <*expression*> () :  
ARINC661Instances  
**TiltSectionAUTOToggleButtonWidgetIdent** <*expression*> () :  
ARINC661Instances  
**StabilizationCTRLToggleButtonWidgetIdent** <*expression*> () :  
ARINC661Instances  
**StabilizationOFFToggleButtonWidgetIdent** <*expression*> () :  
ARINC661Instances  
**WXONCheckBoxWidgetIdent** <*expression*> () : ARINC661Instances  
**WXACheckButtonWidgetIdent** <*expression*> () : ARINC661Instances  
**MODESELECTIONRadioBoxWidgetIdent** <*expression*> () :  
ARINC661Instances  
**TITLANGLEEditNumericBoxWidgetIdent** <*expression*> () :  
ARINC661Instances  
**WXRradioBoxModeSelectionWidgetIdent** <*expression*> () :  
ARINC661Instances  
**WXRradioBoxTiltSelectionWidgetIdent** <*expression*> () :  
ARINC661Instances  
**WXRradioBoxStabilizationWidgetIdent** <*expression*> () :  
ARINC661Instances  
**WXRStrings** <*expression*> () :  $\mathbb{P}$ (ARINC661Instances)

**WXRWidgetIdents**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances})$   
**WXREditNumericBoxes**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances})$   
**iaLabels**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances} \times$   
 $\text{ARINC66Properties} \times \text{ARINC661Instances})$   
**iaCheckBttons**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances} \times$   
 $\text{ARINC66Properties} \times \text{ARINC661Instances})$   
**iaToggleButtons**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances} \times$   
 $\text{ARINC66Properties} \times \text{ARINC661Instances})$   
**ioRadioBoxes**  $\langle expression \rangle$  () :  $\mathbb{P}(\text{ARINC661Instances} \times$   
 $\text{ARINC66Properties} \times \text{ARINC661Instances})$   
**initiator**  $\langle expression \rangle$  (o:  $\text{Ontology}(\text{ARINC661Classes},$   
 $\text{ARINC66Properties}, \text{ARINC661Instances})$ ) :  $\mathbb{P}(\text{ARINC661Instances} \times$   
 $\text{ARINC66Properties} \times \text{ARINC661Instances})$   
**well-definedness** isWDARINC661Ontology(o)

## AXIOMS

**WXRLabels:**  
 $\text{partition}(\text{WXRLabels}, \{\text{MODESELECTIONLabel}\}, \{\text{OFF1Label}\}, \{\text{STDBYLabel}\}, \{\text{TSTLabel}\}, \{\text{WXONLabel}\}, \{\text{WXALabel}\}, \{\text{TITLSELECTIONLabel}\}, \{\text{CRTL1Label}\}, \{\text{AUTOLabel}\}, \{\text{STABILIZATIONLabel}\}, \{\text{CRTL2Label}\}, \{\text{OFF2Label}\}, \{\text{TITLANGLELabel}\})$

**WXRStrings:**  
 $\text{partition}(\text{WXRStrings}, \{\text{MODESELECTIONString}\}, \{\text{OFF1String}\}, \{\text{STDBYString}\}, \{\text{TSTString}\}, \{\text{WXONString}\}, \{\text{WXAStrng}\}, \{\text{TITLSELECTIONString}\}, \{\text{CRTL1String}\}, \{\text{AUTOStrng}\}, \{\text{STABILIZATIONString}\}, \{\text{CRTL2String}\}, \{\text{OFF2String}\}, \{\text{TITLANGLEString}\})$

**WXRcheckButtons:**  
 $\text{partition}(\text{WXRcheckButtons}, \{\text{OFF1CheckButton}\}, \{\text{STDBYCheckButton}\}, \{\text{TSTCheckButton}\}, \{\text{WXONCheckButton}\}, \{\text{WXACheckButton}\})$

**WXRToggleButtons:**  
 $\text{partition}(\text{WXRToggleButtons}, \{\text{TiltSectionCTRLToogleButton}\}, \{\text{TiltSectionAUTOToogleButton}\}, \{\text{StabilizationCRTLToogleButton}\}, \{\text{StabilizationOFFToogleButton}\})$

**WXRradioBoxes:**  
 $\text{partition}(\text{WXRradioBoxes}, \{\text{WXRradioBoxModeSelection}\}, \{\text{WXRradioBoxTiltSelection}\}, \{\text{WXRradioBoxStabilization}\})$

**WXREditNumericBoxes:**  
 $\text{WXREditNumericBoxes} = \{\text{TITLANGLEEditNumericBox}\}$

**WXRWidgetIdents:**  
 $\text{partition}(\text{WXRWidgetIdents}, \{\text{MODESELECTIONLabelWidgetIdent}\}, \{\text{OFF1LabelWidgetIdent}\}, \{\text{STDBYLabelWidgetIdent}\}, \{\text{TITLANGLELabelWidgetIdent}\}, \{\text{TSTLabelWidgetIdent}\}, \{\text{WXONLabelWidgetIdent}\}, \{\text{WXALabelWidgetIdent}\}, \{\text{TITLSELECTIONLabelWidgetIdent}\}, \{\text{CRTL1LabelWidgetIdent}\}, \{\text{AUTOLabelWidgetIdent}\}, \{\text{STABILIZATIONLabelWidgetIdent}\}, \{\text{CRTL2LabelWidgetIdent}\}, \{\text{OFF2LabelWidgetIdent}\}, \{\text{TITLANGLELabelWidgetIdent}\})$

```

TSTLabelWidgetIdent }, { WXONLabelWidgetIdent }, {
WXALabelWidgetIdent },
{ TITLLabelWidgetIdent }, { CRTL1LabelWidgetIdent }, {
AUTOSLabelWidgetIdent }, { STABILIZATIONLabelWidgetIdent },
{ CRTL2LabelWidgetIdent },
{ OFF2LabelWidgetIdent }, { TITLANGLELabelWidgetIdent }, {
OFF1CheckBoxWidgetIdent }, {
STDBYCheckBoxWidgetIdent },
{ TSTCheckBoxWidgetIdent }, {
TiltSectionCTRLToggleButtonWidgetIdent }, {
TiltSectionAUTOToggleButtonWidgetIdent }, {
TITLSELECTIONLabelWidgetIdent },
{ StabilizationCRTLToggleButtonWidgetIdent },
{ StabilizationOFFToggleButtonWidgetIdent }, {
WXONCheckBoxWidgetIdent }, { WXACheckButtonWidgetIdent
}, { MODESELECTIONRadioBoxWidgetIdent },
{ TITLANGLEEditNumericBoxWidgetIdent }, {
WXRradioBoxStabilizationWidgetIdent }, {
WXRradioBoxModeSelectionWidgetIdent }, {
WXRradioBoxTiltSelectionWidgetIdent }
)

```

*instancesAsSubsets:*

```

WXRLabels ⊆ LabelInstances ∧
WXRcheckboxButtons ⊆ CheckButtonInstances ∧
WXRStrings ⊆ ARINC661_STRING_INSTANCES ∧
WXRToggleButtonButtons ⊆ ToggleButtonInstances ∧
WXReditNumericBoxes ⊆ EditBoxNumericInstances ∧
WXRradioBoxes ⊆ RadioBoxInstances ∧
WXRWidgetIdents ⊆ WidgetIdentInstances

```

*Instances:*

```

partition(Instances , WXRLabels , WXRcheckboxButtons , WXRStrings ,
WXRToggleButtonButtons , WXReditNumericBoxes , WXRradioBoxes ,
WXRWidgetIdents )

```

*ClassInstances:*

```

ClassInstances = ({ Label } × WXRLabels) ∪
({ ARINC661_STRING_CLASS } × WXRStrings) ∪
({ CheckButton } × WXRcheckboxButtons) ∪
({ ToggleButton } × WXRToggleButtonButtons) ∪
({ RadioBox } × WXRradioBoxes) ∪
({ WidgetIdentClass } × WXRWidgetIdents)

```

*iaLabels:*

```

iaLabels = { MODESELECTIONLabel ↦ hasLabelStringForLabel ↦
MODESELECTIONString , OFF1Label ↦ hasLabelStringForLabel ↦
OFF1String ,
STDBYLabel ↦ hasLabelStringForLabel ↦ STDBYString ,
TSTLabel ↦ hasLabelStringForLabel ↦ TSTString ,
STDBYLabel ↦ hasLabelStringForLabel ↦ STDBYString ,

```

$$\begin{aligned}
& \text{TSTLabel} \mapsto \text{hasLabelStringForLabel} \mapsto \text{TSTString}, \\
& \text{WXONLabel} \mapsto \text{hasLabelStringForLabel} \mapsto \text{WXONString}, \\
& \text{WXALabel} \mapsto \text{hasLabelStringForLabel} \mapsto \text{WXAStrng}, \\
& \text{TITLSELECTIONLabel} \mapsto \text{hasLabelStringForLabel} \mapsto \\
& \quad \text{TITLSELECTIONString}, \text{CRTL1Label} \mapsto \\
& \quad \text{hasLabelStringForLabel} \mapsto \text{CRTL1String}, \\
& \text{AUTOLabel} \mapsto \text{hasLabelStringForLabel} \mapsto \text{AUTOStrng}, \\
& \text{STABILIZATIONLabel} \mapsto \text{hasLabelStringForLabel} \mapsto \\
& \quad \text{STABILIZATIONString}, \\
& \text{CRTL2Label} \mapsto \text{hasLabelStringForLabel} \mapsto \text{CRTL2String}, \\
& \text{OFF2Label} \mapsto \text{hasLabelStringForLabel} \mapsto \text{OFF2String}, \\
& \text{TITLANGLELabel} \mapsto \text{hasLabelStringForLabel} \mapsto \\
& \quad \text{TITLANGLEString} \\
& \} \cup \\
& (\text{WXRLabels} \times \{\text{hasVisible}\} \times \{\text{A661\_TRUE}\}) \cup (\text{WXRLabels} \\
& \quad \times \{\text{hasAnonymous}\} \times \{\text{A661\_FALSE}\}) \cup \\
& \{ \text{MODESELECTIONLabel} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{MODESELECTIONLabelWidgetIdent}, \text{OFF1Label} \mapsto \\
& \quad \text{hasWidgetIdent} \mapsto \text{OFF1LabelWidgetIdent}, \\
& \text{STDBYLabel} \mapsto \text{hasWidgetIdent} \mapsto \text{STDBYLabelWidgetIdent}, \\
& \quad \text{TSTLabel} \mapsto \text{hasWidgetIdent} \mapsto \text{TSTLabelWidgetIdent}, \\
& \text{STDBYLabel} \mapsto \text{hasWidgetIdent} \mapsto \text{STDBYLabelWidgetIdent}, \\
& \quad \text{TSTLabel} \mapsto \text{hasWidgetIdent} \mapsto \text{TSTLabelWidgetIdent}, \\
& \text{WXONLabel} \mapsto \text{hasWidgetIdent} \mapsto \text{WXONLabelWidgetIdent}, \\
& \quad \text{WXALabel} \mapsto \text{hasWidgetIdent} \mapsto \text{WXALabelWidgetIdent}, \\
& \text{TITLSELECTIONLabel} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{TITLSELECTIONLabelWidgetIdent}, \text{CRTL1Label} \mapsto \\
& \quad \text{hasWidgetIdent} \mapsto \text{CRTL1LabelWidgetIdent}, \\
& \text{AUTOLabel} \mapsto \text{hasWidgetIdent} \mapsto \text{AUTOStrng}, \\
& \quad \text{STABILIZATIONLabel} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{STABILIZATIONString}, \\
& \text{CRTL2Label} \mapsto \text{hasWidgetIdent} \mapsto \text{CRTL2LabelWidgetIdent}, \\
& \quad \text{OFF2Label} \mapsto \text{hasWidgetIdent} \mapsto \text{OFF2LabelWidgetIdent}, \\
& \text{TITLANGLELabel} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{TITLANGLELabelWidgetIdent} \} \cup \\
& (\{ \text{MODESELECTIONLabel}, \text{OFF1Label}, \text{STDBYLabel}, \\
& \quad \text{TSTLabel}, \text{STDBYLabel}, \text{TSTLabel}, \text{WXONLabel}, \\
& \quad \text{WXALabel}, \text{TITLSELECTIONLabel}, \text{CRTL1Label}, \\
& \text{AUTOLabel}, \text{STABILIZATIONLabel}, \text{CRTL2Label}, \text{OFF2Label}, \\
& \quad \text{TITLANGLELabel} \} \times \{\text{hasParentIdent}\} \times \{\text{NoneIdent}\}) \\
\text{iaCheckBttons:} \\
& \text{iaCheckBttons} = (\{ \text{OFF1CheckButton}, \text{STDBYCheckButton}, \\
& \quad \text{TSTCheckButton}, \text{WXONCheckButton}, \text{WXACheckButton} \} \times \{ \\
& \quad \text{hasVisible}, \text{hasEnable} \} \times \{\text{A661\_TRUE}\}) \cup \\
& (\{ \text{OFF1CheckButton}, \text{STDBYCheckButton}, \text{TSTCheckButton}, \\
& \quad \text{WXONCheckButton}, \text{WXACheckButton} \} \times \{ \\
& \quad \text{hasCheckButtonState} \} \times \{\text{UNSELECTED}\}) \cup
\end{aligned}$$

$$\begin{aligned}
& (\{ \text{OFF1CheckButton} \} \times \{ \text{hasCheckButtonState} \} \times \{ \\
& \quad \text{SELECTED} \}) \cup \\
& (\{ \text{OFF1CheckButton} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{OFF1CheckButtonWidgetIdent} , \\
& \quad \text{STDBYCheckButtonWidgetIdent} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{STDBYCheckButtonWidgetIdent} , \\
& \quad \text{TSTCheckButton} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \quad \text{TSTCheckButtonWidgetIdent} , \text{WXONCheckButton} \mapsto \\
& \quad \quad \text{hasWidgetIdent} \mapsto \text{WXONCheckButtonWidgetIdent} , \\
& \quad \text{WXACheckButton} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \quad \text{WXACheckButtonWidgetIdent} \}) \cup \\
& (\{ \text{OFF1CheckButton} , \text{STDBYCheckButton} , \text{TSTCheckButton} , \\
& \quad \text{WXONCheckButton} , \text{WXACheckButton} \} \times \{ \text{hasParentIdent} \} \\
& \quad \times \{ \text{WXRradioBoxModeSelectionWidgetIdent} \})
\end{aligned}$$

*iaToggleButtons:*

$$\begin{aligned}
\text{iaToggleButtons} = & (\{ \text{TiltSectionCTRLToggle} , \\
& \quad \text{TiltSectionAUTOToggle} , \text{StabilizationCTRLToggle} , \\
& \quad \text{StabilizationOFFToggle} \} \times \{ \text{hasVisible} , \text{hasEnable} \} \\
& \quad \times \{ \text{A661\_TRUE} \}) \cup \\
& (\{ \text{TiltSectionCTRLToggle} , \\
& \quad \text{StabilizationOFFToggle} \} \times \{ \\
& \quad \text{hasToggleButtonState} \} \times \{ \text{SELECTED} \}) \cup \\
& (\{ \text{TiltSectionCTRLToggle} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{TiltSectionCTRLToggleWidgetIdent} , \\
& \quad \text{TiltSectionAUTOToggle} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{TiltSectionAUTOToggleWidgetIdent} , \\
& \quad \text{StabilizationCTRLToggle} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \quad \text{StabilizationCTRLToggleWidgetIdent} , \\
& \quad \text{StabilizationOFFToggle} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \quad \text{StabilizationOFFToggleWidgetIdent} \}) \cup \\
& (\{ \text{TiltSectionCTRLToggle} , \\
& \quad \text{TiltSectionAUTOToggle} \} \times \{ \text{hasParentIdent} \} \\
& \quad \times \{ \text{WXRradioBoxTiltSelection} \}) \cup \\
& (\{ \text{StabilizationCTRLToggle} , \\
& \quad \text{StabilizationOFFToggle} \} \times \{ \text{hasParentIdent} \} \\
& \quad \times \{ \text{WXRradioBoxStabilizationWidgetIdent} \})
\end{aligned}$$

*ioRadioBoxes:*

$$\begin{aligned}
\text{ioRadioBoxes} = & (\{ \text{WXRradioBoxModeSelection} , \\
& \quad \text{WXRradioBoxTiltSelection} , \text{WXRradioBoxStabilization} \} \times \{ \\
& \quad \text{hasVisible} , \text{hasEnable} \} \times \{ \text{A661\_TRUE} \}) \cup \\
& (\{ \text{WXRradioBoxModeSelection} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{WXRradioBoxModeSelectionWidgetIdent} , \\
& \quad \text{WXRradioBoxTiltSelection} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \text{WXRradioBoxTiltSelectionWidgetIdent} , \\
& \quad \text{WXRradioBoxStabilization} \mapsto \text{hasWidgetIdent} \mapsto \\
& \quad \quad \text{WXRradioBoxStabilizationWidgetIdent} \}) \cup \\
& (\{ \text{WXRradioBoxModeSelection} , \text{WXRradioBoxTiltSelection} , \\
& \quad \text{WXRradioBoxStabilization} \} \times \{ \text{hasParentIdent} \} \times
\end{aligned}$$

$$\begin{aligned} & \{ \text{NoneIdent} \} \cup \\ & (\{ \text{WXRradioBoxModeSelection} \} \times \{ \\ & \quad \text{hasChildrenForRadioBox} \} \times \{ \text{OFF1CheckBox}, \\ & \quad \text{STDBYCheckBox}, \text{TSTCheckBox}, \text{WXONCheckBox}, \\ & \quad \text{WXACheckBox} \}) \cup \\ & (\{ \text{WXRradioBoxTiltSelection} \} \times \{ \\ & \quad \text{hasChildrenForRadioBox} \} \times \{ \\ & \quad \text{TiltSectionCTRLToogleButton}, \\ & \quad \text{TiltSectionAUTOToogleButton} \}) \cup \\ & (\{ \text{WXRradioBoxStabilization} \} \times \{ \\ & \quad \text{hasChildrenForRadioBox} \} \times \{ \\ & \quad \text{StabilizationCTRLToogleButton}, \\ & \quad \text{StabilizationOFFToogleButton} \}) \end{aligned}$$

*InstanceAssociation:*

$$\text{InstanceAssociation} = \text{iaLabels} \cup \text{iaCheckBttons} \cup \\ \text{iaToggleButtons} \cup \text{ioRadioBoxes}$$

*A661WXRontology:*

$$\text{A661WXRontology} = \text{consARINC661ontology}(\text{Instances}, \\ \text{ClassInstances}, \text{InstanceAssociation})$$

*initiator:*

$$\forall o \cdot o \in \text{ontology}(\text{ARINC661Classes}, \text{ARINC66Properties}, \\ \text{ARINC661Instances}) \wedge \text{isWDARINC661ontology}(o) \Rightarrow \text{initiator}(o) = \text{InstanceAssociation}$$

EventsAffectingWidgetsAxiomatisation:

## OPERATORS

**isWDChangeModeSelection** *<predicate>* (o: ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$ , mode: ARINC661Instances) :

**changeModeSelection** *<expression>* (o: ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$ , mode: ARINC661Instances) :  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$

**well-definedness** isWDChangeModeSelection(o, ui, mode)

**isWDSetTiltSelectionCTRL** *<predicate>* (o: ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$ ) :

**setTiltSelectionCTRL** *<expression>* (o: ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$ ) :  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$

**well-definedness** isWDSetTiltSelectionCTRL(o, ui)

**isWDSetTiltSelectionAUTO** *<predicate>* (o: ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances})$ ) :

**setTiltSelectionAUTO** <*expression*> (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances)) :  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances)  
**well-definedness** isWDSsetTiltSelectionAUTO(o, ui)  
**isWDSsetStabilizationCTRL** <*predicate*> (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances)) :  
**setStabilizationCTRL** <*expression*> (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances)) :  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances)  
**well-definedness** isWDSsetStabilizationCTRL(o, ui)  
**isWDSsetStabilizationOFF** <*predicate*> (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances)) :  
**setStabilizationOFF** <*expression*> (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances)) :  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances)  
**well-definedness** isWDSsetStabilizationOFF(o, ui)  
**isWDChangeTitlAngle** <*predicate*> (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances), newAngle: ARINC661Instances) :  
**changeTitlAngle** <*expression*> (o: Ontology(ARINC661Classes, ARINC66Properties, ARINC661Instances), ui:  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances), newAngle: ARINC661Instances) :  $\mathbb{P}$ (ARINC661Instances  $\times$  ARINC66Properties  $\times$  ARINC661Instances)  
**well-definedness** isWDChangeTitlAngle(o, ui, newAngle)

#### AXIOMS

*isWDChangeTitlAngle* :  
 $\forall o, ui, newAngle \cdot o \in \text{Ontology}(\text{ARINC661Classes}, \text{ARINC66Properties}, \text{ARINC661Instances}) \wedge ui \in \mathbb{P}(\text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances}) \wedge newAngle \in \text{ARINC661Instances} \Rightarrow$   
 $(\text{isWDChangeTitlAngle}(o, ui, newAngle) \Leftrightarrow$   
 $\text{isVariableOfARINC661Ontology}(o, ui) \wedge newAngle \in$   
 $\text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES} \wedge$   
 $\text{TiltSectionCTRLToggleButton} \mapsto \text{hasToggleButtonState} \mapsto$   
 $\text{SELECTED} \in ui \wedge$   
 $\text{StabilizationCTRLToggleButton} \mapsto \text{hasToggleButtonState} \mapsto$   
 $\text{SELECTED} \in ui)$

*changeTitlAngle:*

$$\begin{aligned} & \forall o, ui, newAngle \cdot o \in \text{Ontology}(\text{ARINC661Classes}, \\ & \quad \text{ARINC66Properties}, \text{ARINC661Instances}) \wedge ui \in \mathbb{P}( \\ & \quad \text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances}) \\ & \quad \wedge newAngle \in \text{ARINC661Instances} \Rightarrow \\ & (\text{changeTitlAngle}(o, ui, newAngle) = (ui \setminus \{ \\ & \quad \text{TITLAngleEditNumericBox} \mapsto \text{hasValue} \mapsto i \mid \\ & \quad \text{TITLAngleEditNumericBox} \mapsto \text{hasValue} \mapsto i \in ui \}) \cup \{ \\ & \quad \text{TITLAngleEditNumericBox} \mapsto \text{hasValue} \mapsto newAngle \}) \end{aligned}$$

*isWDChangeModeSelection:*

$$\begin{aligned} & \forall o, ui, mode \cdot o \in \text{Ontology}(\text{ARINC661Classes}, \text{ARINC66Properties} \\ & \quad , \text{ARINC661Instances}) \wedge ui \in \mathbb{P}(\text{ARINC661Instances} \times \\ & \quad \text{ARINC66Properties} \times \text{ARINC661Instances}) \wedge mode \in \\ & \quad \text{ARINC661Instances} \Rightarrow \\ & (\text{isWDChangeModeSelection}(o, ui, mode) \Leftrightarrow \\ & \quad \text{isVariableOfARINC661Ontology}(o, ui) \wedge mode \in \\ & \quad \text{WXRcheckButtons}) \end{aligned}$$

*changeModeSelection:*

$$\begin{aligned} & \forall o, ui, mode \cdot o \in \text{Ontology}(\text{ARINC661Classes}, \text{ARINC66Properties} \\ & \quad , \text{ARINC661Instances}) \wedge ui \in \mathbb{P}(\text{ARINC661Instances} \times \\ & \quad \text{ARINC66Properties} \times \text{ARINC661Instances}) \wedge mode \in \\ & \quad \text{ARINC661Instances} \Rightarrow \\ & (\text{changeModeSelection}(o, ui, mode) = (ui \setminus \{i \mapsto \\ & \quad \text{hasCheckBoxButtonState} \mapsto \text{UNSELECTED} \mid i \mapsto \\ & \quad \text{hasCheckBoxButtonState} \mapsto \text{SELECTED} \in ui \wedge i \in ( \\ & \quad \text{WXRcheckButtons} \setminus \{mode\}) \}) \cup \{mode \mapsto \\ & \quad \text{hasCheckBoxButtonState} \mapsto \text{SELECTED}\}) \end{aligned}$$

*isWDSetTiltSelectionCTRL:*

$$\begin{aligned} & \forall o, ui, newAngle \cdot o \in \text{Ontology}(\text{ARINC661Classes}, \\ & \quad \text{ARINC66Properties}, \text{ARINC661Instances}) \wedge ui \in \mathbb{P}( \\ & \quad \text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances}) \\ & \quad \wedge newAngle \in \text{ARINC661Instances} \Rightarrow \\ & (\text{isWDSetTiltSelectionCTRL}(o, ui) \Leftrightarrow \\ & \quad \text{isVariableOfARINC661Ontology}(o, ui)) \end{aligned}$$

*setTiltSelectionAUTO:*

$$\begin{aligned} & \forall o, ui, newAngle \cdot o \in \text{Ontology}(\text{ARINC661Classes}, \\ & \quad \text{ARINC66Properties}, \text{ARINC661Instances}) \wedge ui \in \mathbb{P}( \\ & \quad \text{ARINC661Instances} \times \text{ARINC66Properties} \times \text{ARINC661Instances}) \\ & \quad \wedge newAngle \in \text{ARINC661Instances} \Rightarrow \\ & (\text{setTiltSelectionAUTO}(o, ui) = (ui \setminus \{ \\ & \quad \text{TiltSectionCTRLToggle} \mapsto \text{hasToggleButtonState} \mapsto \\ & \quad \text{SELECTED}, \text{TiltSectionAUTOToggle} \mapsto \\ & \quad \text{hasToggleButtonState} \mapsto \text{UNSELECTED}\}) \cup \\ & \quad \{ \text{TiltSectionAUTOToggle} \mapsto \\ & \quad \quad \text{hasToggleButtonState} \mapsto \text{SELECTED}, \\ & \quad \text{TiltSectionCTRLToggle} \mapsto \\ & \quad \quad \text{hasToggleButtonState} \mapsto \text{UNSELECTED} \}) \end{aligned}$$

*isWDSetStabilizationCTRL:*

$$\forall o, ui, newAngle \cdot o \in \text{Ontology}(\text{ARINC661Classes},$$

$$\begin{aligned} & \text{ARINC666Properties, ARINC661Instances}) \wedge \text{ui} \in \mathbb{P}( \\ & \text{ARINC661Instances} \times \text{ARINC666Properties} \times \text{ARINC661Instances}) \\ & \wedge \text{newAngle} \in \text{ARINC661Instances} \Rightarrow \\ & (\text{isWDSsetStabilizationCTRL}(\text{o}, \text{ui}) \Leftrightarrow \\ & \text{isVariableOfARINC661Ontology}(\text{o}, \text{ui})) \end{aligned}$$

*setStabilizationCTRL:*

$$\begin{aligned} \forall \text{o}, \text{ui}, \text{newAngle} \cdot \text{o} \in \text{Ontology}(\text{ARINC661Classes}, \\ \text{ARINC666Properties}, \text{ARINC661Instances}) \wedge \text{ui} \in \mathbb{P}( \\ \text{ARINC661Instances} \times \text{ARINC666Properties} \times \text{ARINC661Instances}) \\ \wedge \text{newAngle} \in \text{ARINC661Instances} \Rightarrow \\ (\text{setStabilizationCTRL}(\text{o}, \text{ui}) = (\text{ui} \setminus \{ \\ \text{StabilizationOFFTogglebutton} \mapsto \text{hasToggleButtonState} \mapsto \\ \text{SELECTED}, \text{StabilizationCTRLTogglebutton} \mapsto \\ \text{hasToggleButtonState} \mapsto \text{UNSELECTED}\}) \cup \\ \{ \text{StabilizationCTRLTogglebutton} \mapsto \\ \text{hasToggleButtonState} \mapsto \text{SELECTED}, \\ \text{StabilizationOFFTogglebutton} \mapsto \\ \text{hasToggleButtonState} \mapsto \text{UNSELECTED}\} ) \end{aligned}$$

*isWDSsetStabilizationOFF:*

$$\begin{aligned} \forall \text{o}, \text{ui}, \text{newAngle} \cdot \text{o} \in \text{Ontology}(\text{ARINC661Classes}, \\ \text{ARINC666Properties}, \text{ARINC661Instances}) \wedge \text{ui} \in \mathbb{P}( \\ \text{ARINC661Instances} \times \text{ARINC666Properties} \times \text{ARINC661Instances}) \\ \wedge \text{newAngle} \in \text{ARINC661Instances} \Rightarrow \\ (\text{isWDSsetStabilizationOFF}(\text{o}, \text{ui}) \Leftrightarrow \\ \text{isVariableOfARINC661Ontology}(\text{o}, \text{ui})) \end{aligned}$$

*setStabilizationOFF:*

$$\begin{aligned} \forall \text{o}, \text{ui}, \text{newAngle} \cdot \text{o} \in \text{Ontology}(\text{ARINC661Classes}, \\ \text{ARINC666Properties}, \text{ARINC661Instances}) \wedge \text{ui} \in \mathbb{P}( \\ \text{ARINC661Instances} \times \text{ARINC666Properties} \times \text{ARINC661Instances}) \\ \wedge \text{newAngle} \in \text{ARINC661Instances} \Rightarrow \\ (\text{setStabilizationOFF}(\text{o}, \text{ui}) = (\text{ui} \setminus \{ \\ \text{StabilizationCTRLTogglebutton} \mapsto \text{hasToggleButtonState} \mapsto \\ \text{SELECTED}, \text{StabilizationOFFTogglebutton} \mapsto \\ \text{hasToggleButtonState} \mapsto \text{UNSELECTED}\}) \cup \\ \{ \text{StabilizationOFFTogglebutton} \mapsto \\ \text{hasToggleButtonState} \mapsto \text{SELECTED}, \\ \text{StabilizationCTRLTogglebutton} \mapsto \\ \text{hasToggleButtonState} \mapsto \text{UNSELECTED}\} ) \end{aligned}$$

## THEOREMS

*isWDARINC661Ontology:*

$$\text{isWDARINC661Ontology}(\text{A661WXRontology})$$

*initiatorSafety:*

$$\forall \text{o}, \text{ipvs} \cdot \text{isVariableOfARINC661Ontology}(\text{o}, \text{ipvs}) \wedge (\text{ipvs} = \\ \text{initiator}(\text{o}))$$

$\Rightarrow$

$$\left( \begin{aligned} \forall \text{rb}, \text{b1}, \text{b2} \cdot \text{rb} \in \text{RadioBoxInstances} \wedge \text{b1} \in \\ \text{CheckButtonInstances} \wedge \text{b2} \in \text{CheckButtonInstances} \wedge \\ \text{rb} \mapsto \text{hasChildrenForRadioBox} \mapsto \text{b1} \in \text{ipvs} \wedge \text{rb} \mapsto \end{aligned} \right.$$

$$\begin{aligned}
& \text{hasChildrenForRadioBox} \mapsto b2 \in \text{ipvs} \\
\Rightarrow & (b1 \mapsto \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge b2 \mapsto \\
& \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow b1 = b2) \\
& ) \wedge \\
& ( \\
\forall & rb, b1, b2. rb \in \text{RadioBoxInstances} \wedge b1 \in \\
& \text{ToggleButtonInstances} \wedge b2 \in \text{ToggleButtonInstances} \wedge \\
& rb \mapsto \text{hasChildrenForRadioBox} \mapsto b1 \in \text{ipvs} \wedge rb \mapsto \\
& \text{hasChildrenForRadioBox} \mapsto b2 \in \text{ipvs} \\
\Rightarrow & (b1 \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge b2 \mapsto \\
& \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow b1 = b2) \\
& ) \wedge \\
& ( \forall ed, v. ed \mapsto \text{hasValue} \mapsto v \in \text{ipvs} \Rightarrow v \in \\
& \text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES} ) \\
\text{initiatorisVariableOfARINC661Ontology:} \\
\forall & o, \text{ipvs} \cdot \text{isWDARINC661Ontology}(o) \wedge \text{ipvs} \in \mathbb{P}(\text{ARINC661Instances} \\
& \times \text{ARINC66Properties} \times \text{ARINC661Instances}) \wedge (\text{ipvs} = \\
& \text{initiator}(o)) \\
\Rightarrow & \text{isVariableOfARINC661Ontology}(o, \text{ipvs}) \\
\text{changeModeSelectionSafety:} \\
\forall & o, \text{ipvs} \cdot \text{isVariableOfARINC661Ontology}(o, \text{ipvs}) \wedge \\
& ( \exists \text{uiArg} \cdot \\
& ( \exists m \cdot \text{isWDChangeModeSelection}(\text{A661WXROntology}, \text{uiArg}, m) \\
& \wedge \text{ipvs} = \text{changeModeSelection}(\text{A661WXROntology}, \text{uiArg}, m) \\
& ) ) ) \\
\Rightarrow & \\
& ( \\
\forall & rb, b1, b2. rb \in \text{RadioBoxInstances} \wedge b1 \in \\
& \text{CheckButtonInstances} \wedge b2 \in \text{CheckButtonInstances} \wedge \\
& rb \mapsto \text{hasChildrenForRadioBox} \mapsto b1 \in \text{ipvs} \wedge rb \mapsto \\
& \text{hasChildrenForRadioBox} \mapsto b2 \in \text{ipvs} \\
\Rightarrow & (b1 \mapsto \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge b2 \mapsto \\
& \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow b1 = b2) \\
& ) \wedge \\
& ( \\
\forall & rb, b1, b2. rb \in \text{RadioBoxInstances} \wedge b1 \in \\
& \text{ToggleButtonInstances} \wedge b2 \in \text{ToggleButtonInstances} \wedge \\
& rb \mapsto \text{hasChildrenForRadioBox} \mapsto b1 \in \text{ipvs} \wedge rb \mapsto \\
& \text{hasChildrenForRadioBox} \mapsto b2 \in \text{ipvs} \\
\Rightarrow & (b1 \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge b2 \mapsto \\
& \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow b1 = b2) \\
& ) \wedge \\
& ( \forall ed, v. ed \mapsto \text{hasValue} \mapsto v \in \text{ipvs} \Rightarrow v \in \\
& \text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES} ) \\
\text{changeModeSelectionisVariableOfARINC661Ontology:} \\
\forall & o, \text{ipvs} \cdot \text{isWDARINC661Ontology}(o) \wedge \\
& ( \exists \text{uiArg} \cdot ( \exists m \cdot \text{isWDChangeModeSelection}(\text{A661WXROntology}, \\
& \text{uiArg}, m) \wedge \text{ipvs} = \text{changeModeSelection}(\text{A661WXROntology}, \\
& \text{uiArg}, m) ) )
\end{aligned}$$

$$\Rightarrow \text{isVariableOfARINC661Ontology}(o, \text{ipvs})$$

*changeTitlAngleSafety:*

$$\forall o, \text{ipvs} \cdot \text{isVariableOfARINC661Ontology}(o, \text{ipvs}) \wedge$$

$$(\exists \text{uiArg} \cdot (\exists v \cdot \text{isWDChangeTitlAngle}(\text{A661WXRontology}, \text{uiArg}, v) \wedge \text{ipvs} = \text{changeTitlAngle}(\text{A661WXRontology}, \text{uiArg}, v)))$$

$$\Rightarrow$$

$$($$

$$\forall \text{rb}, \text{b1}, \text{b2} \cdot \text{rb} \in \text{RadioBoxInstances} \wedge \text{b1} \in$$

$$\text{CheckButtonInstances} \wedge \text{b2} \in \text{CheckButtonInstances} \wedge$$

$$\text{rb} \mapsto \text{hasChildrenForRadioBox} \mapsto \text{b1} \in \text{ipvs} \wedge \text{rb} \mapsto$$

$$\text{hasChildrenForRadioBox} \mapsto \text{b2} \in \text{ipvs}$$

$$\Rightarrow (\text{b1} \mapsto \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge \text{b2} \mapsto$$

$$\text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow \text{b1} = \text{b2})$$

$$) \wedge$$

$$($$

$$\forall \text{rb}, \text{b1}, \text{b2} \cdot \text{rb} \in \text{RadioBoxInstances} \wedge \text{b1} \in$$

$$\text{ToggleButtonInstances} \wedge \text{b2} \in \text{ToggleButtonInstances} \wedge$$

$$\text{rb} \mapsto \text{hasChildrenForRadioBox} \mapsto \text{b1} \in \text{ipvs} \wedge \text{rb} \mapsto$$

$$\text{hasChildrenForRadioBox} \mapsto \text{b2} \in \text{ipvs}$$

$$\Rightarrow (\text{b1} \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge \text{b2} \mapsto$$

$$\text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow \text{b1} = \text{b2})$$

$$) \wedge$$

$$(\forall \text{ed}, v \cdot \text{ed} \mapsto \text{hasValue} \mapsto v \in \text{ipvs} \Rightarrow v \in$$

$$\text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES} )$$

*changeTitlAngleisVariableOfARINC661Ontology:*

$$\forall o, \text{ipvs} \cdot \text{isWDARINC661Ontology}(o) \wedge$$

$$(\exists \text{uiArg} \cdot (\exists v \cdot \text{isWDChangeTitlAngle}(\text{A661WXRontology}, \text{uiArg}, v) \wedge \text{ipvs} = \text{changeTitlAngle}(\text{A661WXRontology}, \text{uiArg}, v)))$$

$$\Rightarrow \text{isVariableOfARINC661Ontology}(o, \text{ipvs})$$

*SetTiltSelectionCTRLSafety:*

$$\forall o, \text{ipvs} \cdot \text{isVariableOfARINC661Ontology}(o, \text{ipvs}) \wedge$$

$$(\exists \text{uiArg} \cdot (\text{isWDSetTiltSelectionCTRL}(o, \text{ipvs}) \wedge \text{uiArg} =$$

$$\text{setTiltSelectionCTRL}(o, \text{uiArg})))$$

$$\Rightarrow$$

$$($$

$$\forall \text{rb}, \text{b1}, \text{b2} \cdot \text{rb} \in \text{RadioBoxInstances} \wedge \text{b1} \in$$

$$\text{CheckButtonInstances} \wedge \text{b2} \in \text{CheckButtonInstances} \wedge$$

$$\text{rb} \mapsto \text{hasChildrenForRadioBox} \mapsto \text{b1} \in \text{ipvs} \wedge \text{rb} \mapsto$$

$$\text{hasChildrenForRadioBox} \mapsto \text{b2} \in \text{ipvs}$$

$$\Rightarrow (\text{b1} \mapsto \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge \text{b2} \mapsto$$

$$\text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow \text{b1} = \text{b2})$$

$$) \wedge$$

$$($$

$$\forall \text{rb}, \text{b1}, \text{b2} \cdot \text{rb} \in \text{RadioBoxInstances} \wedge \text{b1} \in$$

$$\text{ToggleButtonInstances} \wedge \text{b2} \in \text{ToggleButtonInstances} \wedge$$

$$\text{rb} \mapsto \text{hasChildrenForRadioBox} \mapsto \text{b1} \in \text{ipvs} \wedge \text{rb} \mapsto$$

$$\text{hasChildrenForRadioBox} \mapsto \text{b2} \in \text{ipvs}$$

$$\Rightarrow (\text{b1} \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge \text{b2} \mapsto$$

$$\begin{aligned}
& \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow \text{b1} = \text{b2}) \\
& ) \wedge \\
& ( \forall \text{ed}, \text{v} \cdot \text{ed} \mapsto \text{hasValue} \mapsto \text{v} \in \text{ipvs} \Rightarrow \text{v} \in \\
& \quad \text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES} ) \\
\textit{SetTiltSelectionCTRLisVariableOfARINC661Ontology:} \\
& \forall \text{o}, \text{ipvs} \cdot \text{isWDARINC661Ontology}(\text{o}) \wedge \\
& ( \exists \text{uiArg} \cdot ( \text{isWDSetTiltSelectionCTRL}(\text{o}, \text{uiArg}) \wedge \text{ipvs} = \\
& \quad \text{setTiltSelectionCTRL}(\text{o}, \text{uiArg})) ) \\
& \Rightarrow \text{isVariableOfARINC661Ontology}(\text{o}, \text{ipvs}) \\
\textit{TiltSelectionAUTOSafety:} \\
& \forall \text{o}, \text{ipvs} \cdot \text{isVariableOfARINC661Ontology}(\text{o}, \text{ipvs}) \wedge \\
& ( \exists \text{uiArg} \cdot ( \text{isWDSetTiltSelectionAUTO}(\text{A661WXRontology}, \text{uiArg}) \\
& \quad \wedge \text{ipvs} = \text{setTiltSelectionAUTO}(\text{A661WXRontology}, \text{uiArg})) ) \\
& \Rightarrow \\
& ( \\
& \quad \forall \text{rb}, \text{b1}, \text{b2} \cdot \text{rb} \in \text{RadioBoxInstances} \wedge \text{b1} \in \\
& \quad \quad \text{CheckButtonInstances} \wedge \text{b2} \in \text{CheckButtonInstances} \wedge \\
& \quad \text{rb} \mapsto \text{hasChildrenForRadioBox} \mapsto \text{b1} \in \text{ipvs} \wedge \text{rb} \mapsto \\
& \quad \quad \text{hasChildrenForRadioBox} \mapsto \text{b2} \in \text{ipvs} \\
& \quad \Rightarrow (\text{b1} \mapsto \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge \text{b2} \mapsto \\
& \quad \quad \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow \text{b1} = \text{b2}) \\
& ) \wedge \\
& ( \\
& \quad \forall \text{rb}, \text{b1}, \text{b2} \cdot \text{rb} \in \text{RadioBoxInstances} \wedge \text{b1} \in \\
& \quad \quad \text{ToggleButtonInstances} \wedge \text{b2} \in \text{ToggleButtonInstances} \wedge \\
& \quad \text{rb} \mapsto \text{hasChildrenForRadioBox} \mapsto \text{b1} \in \text{ipvs} \wedge \text{rb} \mapsto \\
& \quad \quad \text{hasChildrenForRadioBox} \mapsto \text{b2} \in \text{ipvs} \\
& \quad \Rightarrow (\text{b1} \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge \text{b2} \mapsto \\
& \quad \quad \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow \text{b1} = \text{b2}) \\
& ) \wedge \\
& ( \forall \text{ed}, \text{v} \cdot \text{ed} \mapsto \text{hasValue} \mapsto \text{v} \in \text{ipvs} \Rightarrow \text{v} \in \\
& \quad \text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES} ) \\
\textit{TiltSelectionAUTOisVariableOfARINC661Ontology:} \\
& \forall \text{o}, \text{ipvs} \cdot \text{isWDARINC661Ontology}(\text{o}) \wedge \\
& ( \exists \text{uiArg} \cdot ( \text{isWDSetTiltSelectionAUTO}(\text{A661WXRontology}, \text{ipvs}) \wedge \\
& \quad \text{ipvs} = \text{setTiltSelectionAUTO}(\text{A661WXRontology}, \text{uiArg})) ) \\
& \Rightarrow \text{isVariableOfARINC661Ontology}(\text{o}, \text{ipvs}) \\
\textit{SetStabilizationCTRLSafety:} \\
& \forall \text{o}, \text{ipvs} \cdot \text{isVariableOfARINC661Ontology}(\text{o}, \text{ipvs}) \wedge \\
& ( \exists \text{uiArg} \cdot ( \text{isWDSetStabilizationCTRL}(\text{A661WXRontology}, \text{ipvs}) \\
& \quad \wedge \text{ipvs} = \text{setStabilizationCTRL}(\text{A661WXRontology}, \text{uiArg})) ) \\
& \Rightarrow \\
& ( \\
& \quad \forall \text{rb}, \text{b1}, \text{b2} \cdot \text{rb} \in \text{RadioBoxInstances} \wedge \text{b1} \in \\
& \quad \quad \text{CheckButtonInstances} \wedge \text{b2} \in \text{CheckButtonInstances} \wedge \\
& \quad \text{rb} \mapsto \text{hasChildrenForRadioBox} \mapsto \text{b1} \in \text{ipvs} \wedge \text{rb} \mapsto \\
& \quad \quad \text{hasChildrenForRadioBox} \mapsto \text{b2} \in \text{ipvs} \\
& \quad \Rightarrow (\text{b1} \mapsto \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge \text{b2} \mapsto \\
& \quad \quad \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow \text{b1} = \text{b2}) \\
& ) \\
\end{aligned}$$

$$\begin{aligned}
& ) \wedge \\
& ( \\
& \forall rb, b1, b2. rb \in \text{RadioBoxInstances} \wedge b1 \in \\
& \quad \text{ToggleButtonInstances} \wedge b2 \in \text{ToggleButtonInstances} \wedge \\
& \quad rb \mapsto \text{hasChildrenForRadioBox} \mapsto b1 \in \text{ipvs} \wedge rb \mapsto \\
& \quad \text{hasChildrenForRadioBox} \mapsto b2 \in \text{ipvs} \\
& \Rightarrow (b1 \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge b2 \mapsto \\
& \quad \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow b1 = b2) \\
& ) \wedge \\
& ( \forall ed, v. ed \mapsto \text{hasValue} \mapsto v \in \text{ipvs} \Rightarrow v \in \\
& \quad \text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES} ) \\
& \textit{SetStabilizationCTRLIsVariableOfARINC661Ontology:} \\
& \forall o, \text{ipvs}. \text{isWDARINC661Ontology}(o) \wedge \\
& \quad (\exists \text{uiArg}. (( \text{isWDSetStabilizationCTRL}(\text{A661WXRontology}, \text{ipvs}) \wedge \\
& \quad \text{ipvs} = \text{setStabilizationCTRL}(\text{A661WXRontology}, \text{uiArg})) )) \\
& \Rightarrow \text{isVariableOfARINC661Ontology}(o, \text{ipvs}) \\
& \textit{SetStabilizationOFFSafety:} \\
& \forall o, \text{ipvs}. \text{isVariableOfARINC661Ontology}(o, \text{ipvs}) \wedge \\
& \quad (\exists \text{uiArg}. ( \text{isWDSetStabilizationOFF}(\text{A661WXRontology}, \text{ipvs}) \wedge \\
& \quad \text{ipvs} = \text{setStabilizationOFF}(\text{A661WXRontology}, \text{uiArg})) ) \\
& \Rightarrow \\
& \quad ( \\
& \quad \forall rb, b1, b2. rb \in \text{RadioBoxInstances} \wedge b1 \in \\
& \quad \quad \text{CheckButtonInstances} \wedge b2 \in \text{CheckButtonInstances} \wedge \\
& \quad \quad rb \mapsto \text{hasChildrenForRadioBox} \mapsto b1 \in \text{ipvs} \wedge rb \mapsto \\
& \quad \quad \text{hasChildrenForRadioBox} \mapsto b2 \in \text{ipvs} \\
& \quad \Rightarrow (b1 \mapsto \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge b2 \mapsto \\
& \quad \quad \text{hasCheckButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow b1 = b2) \\
& \quad ) \wedge \\
& \quad ( \\
& \quad \forall rb, b1, b2. rb \in \text{RadioBoxInstances} \wedge b1 \in \\
& \quad \quad \text{ToggleButtonInstances} \wedge b2 \in \text{ToggleButtonInstances} \wedge \\
& \quad \quad rb \mapsto \text{hasChildrenForRadioBox} \mapsto b1 \in \text{ipvs} \wedge rb \mapsto \\
& \quad \quad \text{hasChildrenForRadioBox} \mapsto b2 \in \text{ipvs} \\
& \quad \Rightarrow (b1 \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge b2 \mapsto \\
& \quad \quad \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow b1 = b2) \\
& \quad ) \wedge \\
& \quad ( \forall ed, v. ed \mapsto \text{hasValue} \mapsto v \in \text{ipvs} \Rightarrow v \in \\
& \quad \quad \text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES} ) \\
& \textit{SetStabilizationOFFIsVariableOfARINC661Ontology:} \\
& \forall o, \text{ipvs}. \text{isWDARINC661Ontology}(o) \wedge \\
& \quad (\exists \text{uiArg}. ( \text{isWDSetStabilizationOFF}(\text{A661WXRontology}, \text{ipvs}) \wedge \\
& \quad \text{ipvs} = \text{setStabilizationOFF}(\text{A661WXRontology}, \text{uiArg})) )) \\
& \Rightarrow \text{isVariableOfARINC661Ontology}(o, \text{ipvs}) \\
& \textit{AllOperatorsSafety:} \\
& \forall o, \text{ipvs}. \text{isVariableOfARINC661Ontology}(o, \text{ipvs}) \wedge \\
& \quad (\exists \text{uiArg}. ( (\text{ipvs} = \text{initiator}(o)) \vee \\
& \quad \quad \exists m. \text{isWDChangeModeSelection}(o, \text{uiArg}, m) \wedge \text{ipvs} = \\
& \quad \quad \text{changeModeSelection}(o, \text{uiArg}, m)) \vee
\end{aligned}$$

$$\begin{aligned}
& (\exists v \cdot \text{isWDChangeTitlAngle}(o, \text{uiArg}, v) \wedge \text{ipvs} = \\
& \quad \text{changeTitlAngle}(o, \text{uiArg}, v)) \vee \\
& (\text{isWDSetTiltSelectionCTRL}(o, \text{uiArg}) \wedge \text{ipvs} = \\
& \quad \text{setTiltSelectionCTRL}(o, \text{uiArg})) \vee \\
& (\text{isWDSetTiltSelectionAUTO}(o, \text{uiArg}) \wedge \text{ipvs} = \\
& \quad \text{setTiltSelectionAUTO}(o, \text{uiArg})) \vee \\
& (\text{isWDSetStabilizationCTRL}(o, \text{uiArg}) \wedge \text{ipvs} = \\
& \quad \text{setStabilizationCTRL}(o, \text{uiArg})) \vee \\
& (\text{isWDSetStabilizationOFF}(o, \text{uiArg}) \wedge \text{ipvs} = \\
& \quad \text{setStabilizationOFF}(o, \text{uiArg})) \\
\Rightarrow & \\
& ( \\
& \quad \forall rb, b1, b2 \cdot rb \in \text{RadioBoxInstances} \wedge b1 \in \\
& \quad \quad \text{CheckBoxInstances} \wedge b2 \in \text{CheckBoxInstances} \wedge \\
& \quad rb \mapsto \text{hasChildrenForRadioBox} \mapsto b1 \in \text{ipvs} \wedge rb \mapsto \\
& \quad \quad \text{hasChildrenForRadioBox} \mapsto b2 \in \text{ipvs} \\
& \Rightarrow (b1 \mapsto \text{hasCheckBoxState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge b2 \mapsto \\
& \quad \text{hasCheckBoxState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow b1 = b2) \\
& ) \wedge \\
& ( \\
& \quad \forall rb, b1, b2 \cdot rb \in \text{RadioBoxInstances} \wedge b1 \in \\
& \quad \quad \text{ToggleButtonInstances} \wedge b2 \in \text{ToggleButtonInstances} \wedge \\
& \quad rb \mapsto \text{hasChildrenForRadioBox} \mapsto b1 \in \text{ipvs} \wedge rb \mapsto \\
& \quad \quad \text{hasChildrenForRadioBox} \mapsto b2 \in \text{ipvs} \\
& \Rightarrow (b1 \mapsto \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \wedge b2 \mapsto \\
& \quad \text{hasToggleButtonState} \mapsto \text{SELECTED} \in \text{ipvs} \Rightarrow b1 = b2) \\
& ) \wedge \\
& ( \forall ed, v \cdot ed \mapsto \text{hasValue} \mapsto v \in \text{ipvs} \Rightarrow v \in \\
& \quad \text{A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES} )
\end{aligned}$$

*AllOperatorsIsVariableOfARINC661Ontology:*

$$\begin{aligned}
& \forall o, \text{ipvs} \cdot \text{isWDARINC661Ontology}(o) \wedge \\
& (\exists \text{uiArg} \cdot ( (\text{ipvs} = \text{initiator}(o)) \vee \\
& \quad \exists m \cdot \text{isWDChangeModeSelection}(o, \text{uiArg}, m) \wedge \text{ipvs} = \\
& \quad \quad \text{changeModeSelection}(o, \text{uiArg}, m)) \vee \\
& (\exists v \cdot \text{isWDChangeTitlAngle}(o, \text{uiArg}, v) \wedge \text{ipvs} = \\
& \quad \text{changeTitlAngle}(o, \text{uiArg}, v)) \vee \\
& (\text{isWDSetTiltSelectionCTRL}(o, \text{uiArg}) \wedge \text{ipvs} = \\
& \quad \text{setTiltSelectionCTRL}(o, \text{uiArg})) \vee \\
& (\text{isWDSetTiltSelectionAUTO}(o, \text{uiArg}) \wedge \text{ipvs} = \\
& \quad \text{setTiltSelectionAUTO}(o, \text{uiArg})) \vee \\
& (\text{isWDSetStabilizationCTRL}(o, \text{uiArg}) \wedge \text{ipvs} = \\
& \quad \text{setStabilizationCTRL}(o, \text{uiArg})) \vee \\
& (\text{isWDSetStabilizationOFF}(o, \text{uiArg}) \wedge \text{ipvs} = \\
& \quad \text{setStabilizationOFF}(o, \text{uiArg})) \\
\Rightarrow & \text{isVariableOfARINC661Ontology}(o, \text{ipvs})
\end{aligned}$$

**END**

Listing 3: Event-B machine modelling the WXR user interface

## **WXR Model (Subsection 7.3)**

## WXR Model Definition

### MACHINE

WXRModel

**VARIABLES** *ui*

**INVARIANTS**

*inv1* :

$$\begin{aligned} & \exists ui Arg \cdot ((ui = initiator(A661WXRontology)) \vee \\ & \quad \exists m \cdot isWDChangeModeSelection(A661WXRontology, ui Arg, m) \wedge ui = changeModeSelection(A661WXRontology, ui Arg, m)) \vee \\ & \quad (\exists v \cdot isWDChangeTitlAngle(A661WXRontology, ui Arg, v) \wedge ui = changeTitlAngle(A661WXRontology, ui Arg, v)) \vee \\ & \quad (isWDSerTiltSelectionCTRL(A661WXRontology, ui Arg) \wedge ui = serTiltSelectionCTRL(A661WXRontology, ui Arg)) \vee \\ & \quad (isWDSerTiltSelectionAUTO(A661WXRontology, ui Arg) \wedge ui = serTiltSelectionAUTO(A661WXRontology, ui Arg)) \vee \\ & \quad (isWDSerStabilizationCTRL(A661WXRontology, ui Arg) \wedge ui = serStabilizationCTRL(A661WXRontology, ui Arg)) \vee \\ & \quad (isWDSerStabilizationOFF(A661WXRontology, ui Arg) \wedge ui = serStabilizationOFF(A661WXRontology, ui Arg)) \end{aligned}$$

*thm1* : *isVariableOfARINC661Ontology*(A661WXRontology, *ui*)

*thm2* :

$$\begin{aligned} & (\forall rb, b1, b2 \cdot rb \in RadioBoxInstances \wedge b1 \in CheckButtonInstances \wedge b2 \in CheckButtonInstances \wedge \\ & \quad rb \mapsto hasChildrenForRadioBox \mapsto b1 \in ui \wedge rb \mapsto hasChildrenForRadioBox \mapsto b2 \in ui \\ & \quad \Rightarrow (b1 \mapsto hasCheckButtonState \mapsto SELECTED \in ui \wedge b2 \mapsto hasCheckButtonState \mapsto SELECTED \in ui)) \wedge \\ & \quad ( \\ & \quad \quad (\forall rb, b1, b2 \cdot rb \in RadioBoxInstances \wedge b1 \in ToggleButtonInstances \wedge b2 \in ToggleButtonInstances \wedge \\ & \quad \quad \quad rb \mapsto hasChildrenForRadioBox \mapsto b1 \in ui \wedge rb \mapsto hasChildrenForRadioBox \mapsto b2 \in ui \\ & \quad \quad \quad \Rightarrow (b1 \mapsto hasToggleButtonState \mapsto SELECTED \in ui \wedge b2 \mapsto hasToggleButtonState \mapsto SELECTED \in ui)) \\ & \quad ) \\ & \quad \wedge ( \\ & \quad \quad \forall ed, v \cdot ed \mapsto hasValue \mapsto v \in ui \Rightarrow v \in A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES \\ & \quad ) \\ & ) \end{aligned}$$

### EVENTS

**INITIALISATION**

**THEN**

*act1* : *ui* := *initiator*(A661WXRontology)

**END**

**changeModeSelection**

**ANY** *mode*

**WHERE**

grd1 : *mode* ∈ *WXRcheckButtons*

grd2 : *isWDChangeModeSelection(A661WXROntology, ui, mode)*

**THEN**

act1 : *ui* := *changeModeSelection(A661WXROntology, ui, mode)*

**END**

**setTiltSelectionCTRL**

**WHERE**

grd1 : *isWDSetTiltSelectionCTRL(A661WXROntology, ui)*

**THEN**

act1 : *ui* := *setTiltSelectionCTRL(A661WXROntology, ui)*

**END**

**setTiltSelectionAUTO**

**WHERE**

grd1 : *isWDSetTiltSelectionAUTO(A661WXROntology, ui)*

**THEN**

act1 : *ui* := *setTiltSelectionAUTO(A661WXROntology, ui)*

**END**

**setStabilizationCTRL**

**WHERE**

grd1 : *isWDSetStabilizationCTRL(A661WXROntology, ui)*

**THEN**

act1 : *ui* := *setStabilizationCTRL(A661WXROntology, ui)*

**END**

**setStabilizationOFF**

**WHERE**

grd1 : *isWDSetStabilizationOFF(A661WXROntology, ui)*

**THEN**

act1 : *ui* := *setStabilizationOFF(A661WXROntology, ui)*

**END**

**changeTitlAngle**

**ANY** *newAngle*

**WHERE**

grd1 : *newAngle* ∈ *A661\_EDIT\_BOX\_NUMERIC\_VALUES*

grd2 : *newAngle* ∈ *A661\_EDIT\_BOX\_NUMERIC\_ADMISSIBLE\_VALUES*

grd3 : *isWDChangeTitlAngle(A661WXROntology, ui, newAngle)*

**THEN**

act1 : *ui* := *changeTitlAngle(A661WXROntology, ui, newAngle)*

**END**

**END**

Listing 4: Event-B machine modelling the WXR user interface