

AADL-BA-FrontEnd

Copyright © 2011 TELECOM ParisTech and CNRS

TELECOM ParisTech/LTCI

Authors: see AUTHORS

This program is free software: you can redistribute it and/or modify it under the terms of the Eclipse Public License as published by Eclipse, either version 1.0 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Public License for more details. You should have received a copy of the Eclipse Public License along with this program. If not, see <http://www.eclipse.org/org/documents/epl-v10.php>

Draft 0.7 (04/06/2011)

Notes:

- \_ Implementation language is Java.
- \_ The type of the attribute is inferred from the name of the attribute. If the attribute name is not explicite, or if the inferred type is ambiguous, the type is given.
- \_ Plural attribute name means collection.
- \_ The implicate super type of all the classes is fr.tpt.aadl.annex.behavior.Element.
- \_ The classes not described here just inherit from fr.tpt.aadl.annex.behavior.Element and have no attribute (ex: TimeoutCatch).
- \_ Operators are just Java's enumerations and are not described here.
- \_ The classe names in emphasis font refer to pure interfaces.
- \_ The classe names underlined refer to abstract classes.

Terminology:

- \_ HAS: means is composed of the following attribute(s)
- \_ IS: means derived from
- \_ : list the derived classes or interfaces.
- \_ suffix \* means that the attribute can be null (cardinality 1) or unset (collections)
- \_ | means or
- \_ & means and

Element HAS:

- \_ Element baRef (binded objects from aadlba EMF model)
- \_ Element aadlRef (binded objets from aadl2 EMF model)

BehaviorAnnex HAS:

- \_ behaviorVariables\*
- \_ behaviorStates\*
- \_ behaviorTransitions\*

BehaviorVariable HAS:

- \_ declarators
- \_ uniqueComponentClassifierReference

UniqueComponentClassifierReference IS: NamedElement

NamedElement HAS:

- \_ String name

- \_ String qualifiedName
- \_ String namespaceSeparator\*
- \_ String namespace\*

Declarator HAS:

- \_ identifierOwned
- \_ *IntegerValueConstant* arraySizes\*

BehaviorState HAS:

- \_ identifiers
- \_ boolean initial
- \_ boolean complete
- \_ boolean final

Identifier HAS:

- \_ String id

BehaviorTransition HAS:

- \_ transitionIdentifier\*
- \_ Numeral behaviorTransitionPriority\*
- \_ sourceStateIdentifiers
- \_ behaviorConditionOwned\*
- \_ destinationStateIdentifier
- \_ behaviorActionBlockOwned\*

Numeral HAS:

- \_ Integer value

*BehaviorCondition*: *DispatchCondition* | *ExecuteCondition*

*ExecuteCondition*: *ValueExpression* | *TimeoutCatch* | *Otherwise*

*DispatchCondition* HAS :

- \_ dispatchTriggerConditionOwned\*
- \_ Identifier frozenPorts\*

*DispatchTriggerCondition*: *DispatchTriggerLogicalExpression* | Identifier | *DispatchTriggerConditionStop* | *CompletionRelativeTimeoutConditionAndCatch* | *TimeoutCatch*

*CompletionRelativeTimeoutConditionAndCatch* IS: *BehaviorTime* & *DispatchTriggerCondition*

*BehaviorTime* HAS:

- \_ integerValueOwned
- \_ unitIdentifier

*IntegerValue*: *IntegerValueVariable* | *IntegerValueConstant*

*IntegerValueVariable*: *ValueVariable*

*IntegerValueConstant*: *ValueConstant*

*DispatchTriggerLogicalExpression* HAS:

- \_ dispatchConjunctions

*DispatchConjunction* HAS:

- \_ Identifier dispatchTriggers

BehaviorActionBlock IS: *BehaviorAction*

BehaviorActionBlock HAS:

- \_ behaviorActionsOwned
- \_ behaviorTimeOwned\*

*BehaviorActions*: *BehaviorAction* | BehaviorActionCollection

BehaviorActionCollection: BehaviorActionSet | BehaviorActionSequence

BehaviorActionCollection HAS:

- \_ *BehaviorAction* behaviorActions

*BehaviorAction*: *BasicAction* | BehaviorActionBlock | *CondStatement*

*CondStatement*: IfStatement | LoopStatement

IfStatement HAS:

- \_ LogicalValueExpressions
- \_ *BehaviorActions* behaviorActionsOwned
- \_ boolean hasElse

LoopStatement: ForOrForAllStatement | WhileOrDoUntilStatement

LoopStatement HAS:

- \_ *BehaviorActions* behaviorActionsOwned

WhileOrDoUntilStatement HAS:

- \_ LogicalValueExpression
- \_ boolean doUntil

ForOrForAllStatement HAS:

- \_ elementIdentifier
- \_ dataUniqueComponentClassifierReference
- \_ elementValuesOwned
- \_ boolean forAll

*ElementValues*: IntegerRange | Name | DataComponentReference

IntegerRange HAS:

- \_ lowerIntegerValue
- \_ upperIntegerValue

Name IS: *ElementValues* & *Target* & *ValueVariable*

Name HAS:

- \_ identifierOwned
- \_ *IntegerValueVariable* arrayIndexes\*

DataComponentReference IS: *Target* & *ValueVariable* & *ElementValues*

DataComponentReference HAS:

- \_ names

*BasicAction*: AssignmentAction | *CommunicationAction* | TimedAction

AssignmentAction HAS:

- \_ targetOwned
- \_ ValueExpressionOwned\*
- \_ boolean any

*Target* IS: *ParameterLabel*

*Target*: Name | DataComponentReference

TimedAction HAS:

- \_ lowerBehaviorTime
- \_ upperBehaviorTime\*

*CommunicationAction*: SubprogramCallAction | PortSendAction | PortDequeueAction | PortFreezeAction | LockAction | UnlockAction

*ParameterLabel*: ValueExpression | *Target*

SubprogramCallAction HAS:

- \_ nameOwned\*
- \_ UniqueComponentClassifierReference subprogramReference\*
- \_ parameterLabels\*

PortSendAction HAS:

- \_ portName
- \_ valueExpressionOwned\*

PortDequeueAction HAS:

- \_ portName
- \_ targetOwned\*

PortFreezeAction IS: Name & *CommunicationAction*

LockAction HAS:

- \_ dataAccessName\* (a null value means all data are locked)

UnlockAction HAS:

- \_ dataAccessName\* (a null value means all data are unlocked)

ValueExpression IS: *ParameterLabel*

ValueExpression HAS:

- \_ relations

Relation HAS:

- \_ simpleExpressionOwned
- \_ simpleExpressionSdOwned\*

SimpleExpression HAS:

- \_ terms
- \_ unaryAddingOperatorOwned\*
- \_ binaryAddingOperatorOwned\*

Term HAS:

- \_ factors
- \_ multiplyingOperatorOwned\*

Factor HAS:

- \_ valueOwned
- \_ valueSdOwned\*
- \_ unaryNumericOperatorOwned\*
- \_ binaryNumericOperatorOwned\*
- \_ unaryBooleanOperatorOwned\*

*Value*: ValueVariable | ValueConstant | ValueExpression

*ValueVariable*: Name | DataComponentReference | PortCountValue | PortFreshValue | PortDequeueValue

PortCountValue IS: Name & ValueVariable

PortFreshValue IS: Name & ValueVariable

PortDequeueValue IS: Name & ValueVariable

*ValueConstant*: PropertyConstant | PropertyValue | Literal

PropertyConstant IS: NamedElement & ValueConstant

PropertyValue IS: NamedElement & ValueConstant

*Literal*: BooleanLiteral | NumericLiteral | StringLiteral

BooleanLiteral HAS:

\_ boolean value

NumericLiteral: RealLiteral | IntegerLiteral

NumericLiteral HAS:

\_ String valueString

RealLiteral HAS:

\_ double value

IntegerLiteral HAS:

\_ int value

\_ int base