EAST-ADL Example: RegenerativeBrakingSystem

- EAST-ADL is used to specify the architecture of the RegenerativeBrakingSystem. RegenerativeBrakingSystem is an innovative brake by wire distributed architecture, where the kinetic energy produced by braking is converted to electrical energy and stored in capacitor or/and battery. Use of EAST-ADL covers model analysis, integration with implementation in Simulink and Verification and Validation tasks.
- Tools used include MetaEdit+ for EAST-ADL, UPPAAL, Simulink
- Parts of EAST-ADL applied:
    - Requirements
    - Features
    - Functional Analysis
    - Functional Design
    - Hardware Architecture
    - Dependability
    - Error model
    - Allocations
    - Behavior
    - Timing
    - Verification & Validation
- Sample models on behavior constraints and verification & validation

Window 1: BehaviorConstraintComputationConstraint: RegenBraking_CompConstraint, 12. June 2013, 15:05

Menu: Graph Edit View Types Format Help

Toolbar: f(x) LogicalResp LogicalStimu

Tree:
- f(x) LogicalTransformation
  - WriteBuffer
- LogicalTimeCondition
  - EBSTimerCondition_2
- LogicalPath
  - path_1
  - path_2
  - path_3
- LogicalEvent
- TransformationOccurrence
  - Sending_BrakingOff
  - Sending_BrakingOn

Property / Value:
- Graph type — BehaviorConstrai
  - Short name — RegenBraking_Co
  - Name
  - Description

Diagram:

LogicalPath
path_1
TransformationOccurrence :=
{Sending_BrakingOn }

TransformationOccurrence
Sending_BrakingOn
inQuantification:={
Default_BrakingControlComChannel
BrakeCtrl_On }
invokedTransformation:={ WriteBuffer( ) }

LogicalPath
path_3
TransformationOccurrence := {}

f(x) *LogicalTransformation*
void **WriteBuffer(ComChannelNr BrakeCtrlValue)**

LogicalPath
path_2
TransformationOccurrence :=
{Sending_BrakingOff }

TransformationOccurrence
Sending_BrakingOff
inQuantification:={
Default_BrakingControlComChannel
BrakeCtrl_Off }
invokedTransformation:={ WriteBuffer( ) }

LogicalTimeCondition
EBSTimerCondition_2
upper := 10 msec
lower := 10 msec
width := 10 msec
startPointReference := EBSTriggeringOccurrence

Active: None   Grid: 10 @ 10   Snap   Show   100%

Window 2: VerificationValidation: RegenBraking_VV_1, 14. October 2013, 11:45

Menu: Graph Edit View Types Format Help

Tree:
- VVCase
  - Requested_torque
  - Slip_rate
- VVTarget
  - Requested_torque
  - Slip_rate
- Requirement
  - Req2_1_BBW_V_4
  - Req2_2_BBW_V_5
  - Req3_1_BBW_V_1
  - Req3_2_BBW_V_2
  - Req3_3_BBW_V_3
  - Req3_4_BBW_V_6
- QualityRequirement

Property / Value:
- Graph type — VerificationValida
  - Short name — RegenBraking_VV
  - Name
  - Description
  - TestTarget — Attach New...

Diagram boxes:

Req3_1_BBW_V_1
The "slip rate" for a certain wheel shall be defined as (vehicle speed - wheel speed * wheel radius)/vehicle speed.

Req3_2_BBW_V_2
If the brake pedal is pressed, vehicle speed > 10km/h and a wheel has a slip rate > 20%, then the brake torque for that wheel shall be set to 0Nm.

Req3_3_BBW_V_3
If the brake pedal is pressed and [a certain wheel has a slip rate <= 20% or the vehicle speed <= 10km/h], then the brake torque for that wheel shall be proportional to the angle of the brake pedal.

Req2_1_BBW_V_4
If the brake pedal is not pressed, then the requested torque shall be set to 0Nm.

Req2_2_BBW_V_5
If the brake pedal is pressed to its maximum angle, then the requested brake torque shall be set to its maximum allowed value.

Req3_4_BBW_V_6
The requested brake torque on each wheel shall not exceed 3000Nm.

Verify Verify Verify → *VVCase* **Slip_rate**

Verify Verify Verify → *VVCase* **Requested_torque**

VVTarget **Slip_rate**

VVTarget **Requested_torque**

Active: None   Grid: 10 @ 10   Snap   Show   100%