Specification of Variant-Specific Component Fault Behaviors using AADL Error Modeling Notation

1 – Purpose

This document presents the detailed description of the specification variant fault behaviors for the Powertrain Battery component from the variant-rich automotive Hybrid Braking System (HBS) [1] (Figure 1) using the AADL extension named Error Annex. AADL Error Annex [2] provides a modeling notation to support the specification system and component error behaviors and their relationships. An overview of AADL Error Annex modeling notation can be found in Delange and Feiler [2].

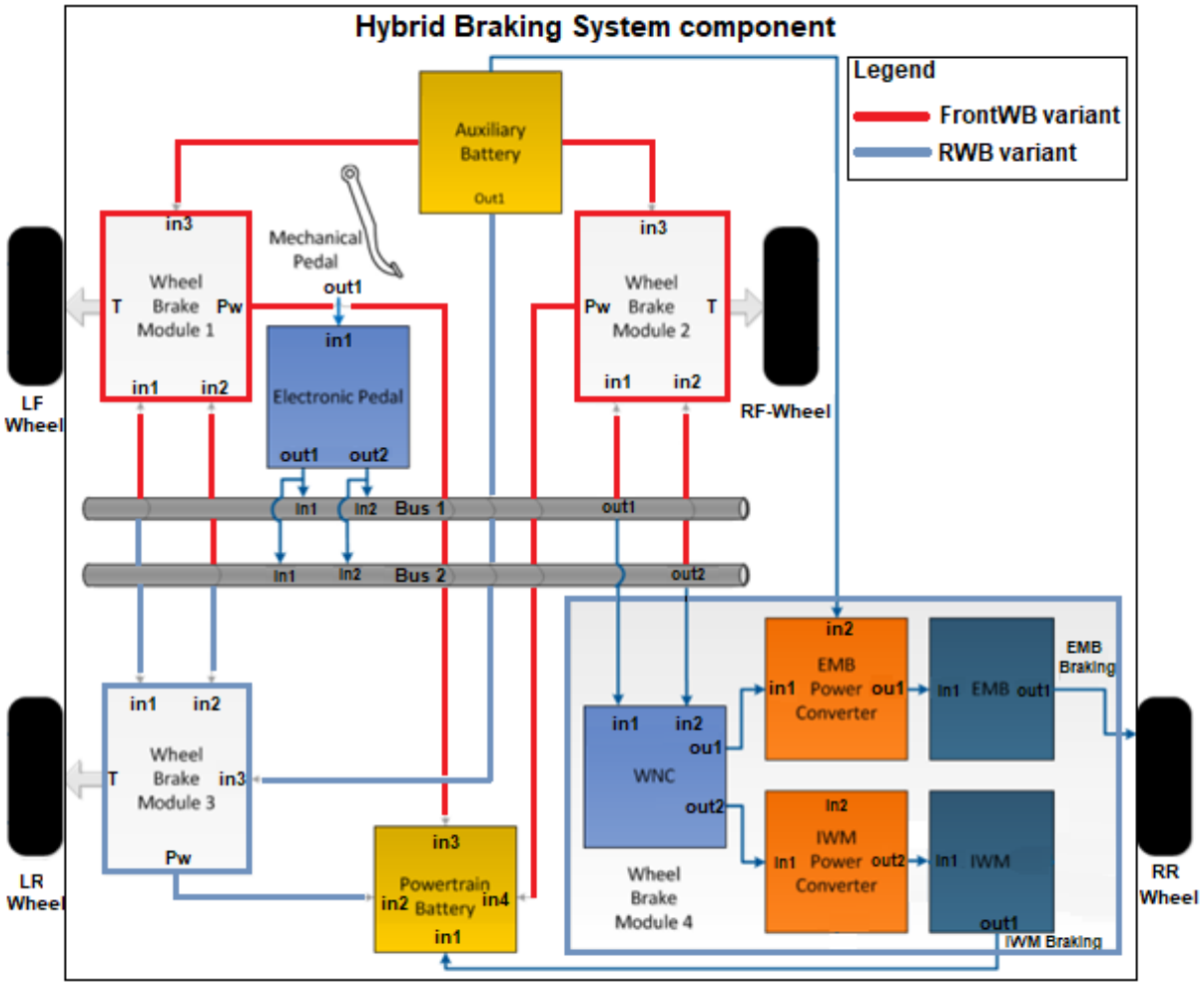

Figure 1. An excerpt of the hybrid braking system design [1].

2 - AADL Error Annex Notation

In the AADL error modeling notation, we specify a component fault behavior by describing **error propagations**, **error flows** (**source**, **sink**, and/or **path**), and component **error behaviors**. An **error** can propagate through the component itself, its input and/or outputs. A component ***error behavior*** may include ***error events***, ***transition events***, e.g., from a *healthy* to a *failing* state, and descriptions of ***propagations*** of *internal* and/or *input failures* through component *outputs*. Each variant ***fault behavior*** may contain different ***error***

propagations, ***flows***, and ***behaviors***. A variant-specific component error behavior is stored into a component implementation inside an error `annex emv2 {**}` declaration.

3 - HBS Component Fault Modeling using AADL Error Annex

Considering the *PowertrainBattery* component from the HBS shown in Figure 1, specifications of variant-specific fault behaviors for this component are stored into different component implementations. Listing 1 shows an excerpt of *PowertrainBattery.fwb* component implementation and its fault behavior operating in a **Front Wheel Braking** (FWB) system variant. This fault behavior model comprises the propagation of ***no value*** and ***bad value*** errors throughout ***out1*** output port (line 307), ***in3*** and ***in4*** component input ports (lines 308-309). PowertrainBattery component error model also comprises the specification of the source for "*no*" and "*bad value*" ***errors*** (lines 311-312).

```
302 process implementation Powertrain_Battery.fwb
303   annex emv2{**
304      use types ErrorModelLibrary;
305      use behavior ErrorModelLibrary::Simple;
306      error propagations
307       out1: out propagation {NoValue, BadValue};
308       in3: in propagation {NoValue, BadValue};
309       in4: in propagation {NoValue, BadValue};
310          flows
311             o_pw_Fail: error source out1{NoValue} when Failed;
312             v_pw_Fail: error source out1{BadValue} when Failed;
313      end propagations;
314      component error behavior
315          transitions
316             t0:Operational-[in3{NoValue} and in4{NoValue}]→Failed;
317             t1:Operational-[in3{BadValue} and in4{BadValue}]→Failed;
318          propagations
319             o_out1: Operational-[]→out1{NoValue};
320             v_out1: Operational-[]→out1{BadValue};
321       end component;
322  **};
323 end Powertrain_Battery.fwb;
```

Listing 1. Powertrain battery error model in the HBSFWB system variant.

The **component error behavior** (Listing 1) shows two transitions to a failing state, and their propagation through **out1** component output (lines 319-320). The occurrence of "no value" errors in both **in3** and **in4** input ports (line 316) triggers **t0** state transition. The occurrence of "bad value" **errors** in both component inputs (line 317) trigger **t1**.

The fault behavior for the PowertrainBattery operating in a **FourWB** system variant is illustrated in Listing 2. It contains two additional **input error propagations** (lines 308-311) and changes on **transition conditions** (lines 318-319), with the addition of **errors** on **in1** and **in2** input ports (lines 318-319) that receive the power from rear-wheel brakes (see Figure 1). HBSFourWB e HBSFWB variant fault behaviors are stored into two different *PowertrainBattery* component implementations.

```
process implementation Powertrain_Battery.fourwb
  annex emv2{**
     use types ErrorModelLibrary;
     use behavior ErrorModelLibrary::Simple;
     error propagations
         out1: out propagation {NoValue, BadValue};
         in1: in propagation {NoValue, BadValue};
         in2: in propagation {NoValue, BadValue};
         in3: in propagation {NoValue, BadValue};
         in4: in propagation {NoValue, BadValue};
         flows
              o_pw_Fail: error source out1{NoValue} when Failed;
              v_pw_Fail: error source out1{BadValue} when Failed;
     end propagations;
     component error behavior
          transitions
               t0:Operational-[in1{NoValue} and in2{NoValue} and in3{NoValue} and in4{NoValue}]→Failed;
               t1:Operational-[in1{BadValue} and in2{BadValue} and in3{BadValue} and in4{BadValue}]→Failed;
         propagations
              o_out1: Operational-[]→out1{NoValue};
               v_out1: Operational-[]→out1{BadValue};
      end component;
  **};
end Powertrain_Battery.fourwb;
```

Listing 2. Powertrain battery error model for the HBSFourWB system variant.

References

[1] R. de Castro, R. E. Araújo, D. Freitas, D. Hybrid ABS with Electric motor and friction Brakes. 22nd Int. Symposium on Dynamics of Vehicles on Roads and Tracks, Manchester, UK, 2011.

[2] J. Delange and P. Feiler, Architecture Fault Modeling with the AADL Error-Model Annex, 40th EUROMICRO Conference on Software Engineering and Advanced Applications, Verona, Italy, 2014, pp. 361-368.