MOVEP 2012 Tutorial
Safety, Dependability and Performance Analysis of Extended AADL Models

Part 2: System Modeling Using AADL

European Space Agency
European Space Research and Technology Centre

RWTH Aachen University
Software Modeling and Verification Group
Thomas Noll

Fondazione Bruno Kessler
Centre for Scientific and Technological Research
Alessandro Cimatti

MOVEP 2012 School; December 7, 2012; Marseille, France

Contents

Outline

The Industry Standard AADL

- **1989** MetaH
- **1998** SAE AS-2C
- **2004** AADL 1.0
- **2006** Error Annex 1.0
- **2009** AADL 2.0
- **2010** Error Annex 2.0

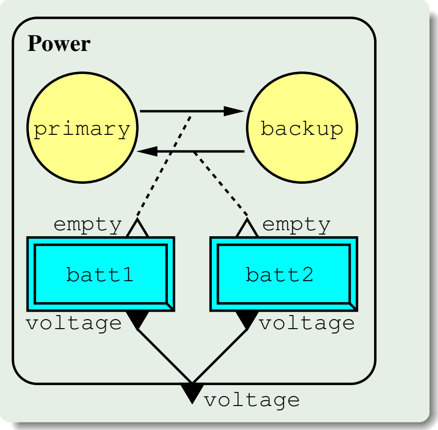
Paradigm

- Architecture-based and model-driven top-down and bottom-up engineering
- Real-time and performance critical distributed systems
- Complements component-based product-line development

AADL Example: Redundant Power System

Redundant power system:

- contains two batteries batt1/batt2
- used in primary/backup mode
- power switches from primary to backup (and back) when batt1 (batt2) empty
- additionally provides voltage information



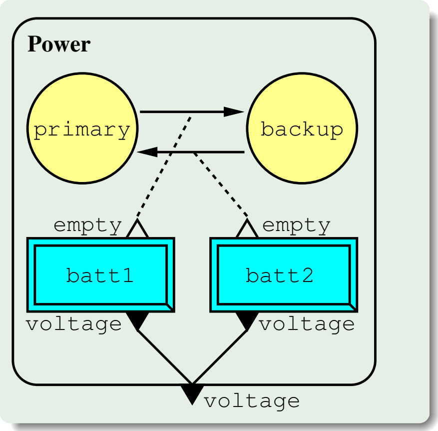
AADL Example: Redundant Power System

Redundant power system:

- contains two batteries batt1/batt2
- used in primary/backup mode
- power switches from primary to backup (and back) when batt1 (batt2) empty
- additionally provides voltage information

We shall show:

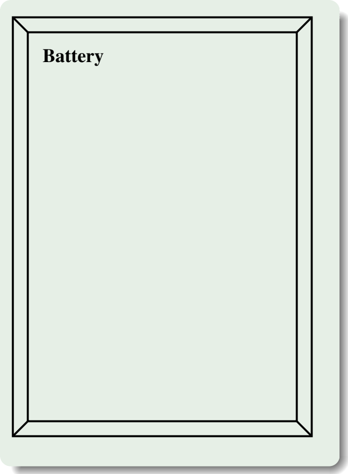
- hybrid behavior of the batteries
- composition of the power system
- semantics as transition systems
- interweaving of errors



Modeling a Battery

Component type and implementation:

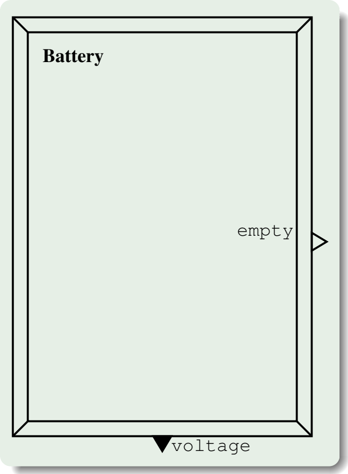
```
device Battery


end Battery;

device implementation Battery.Imp




end Battery.Imp;
```



Modeling a Battery

Type defines the interface:

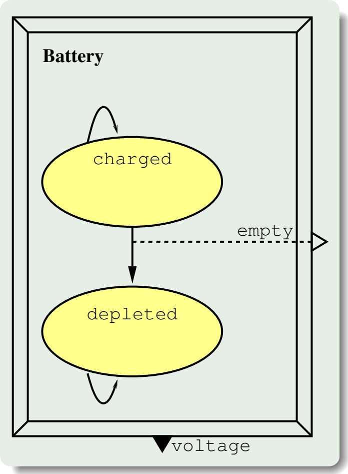
```
device Battery
  features
    empty: out event port;
    voltage: out data port real default 6.0;
end Battery;

device implementation Battery.Imp








end Battery.Imp;
```



Modeling a Battery

Adding modes behavior:

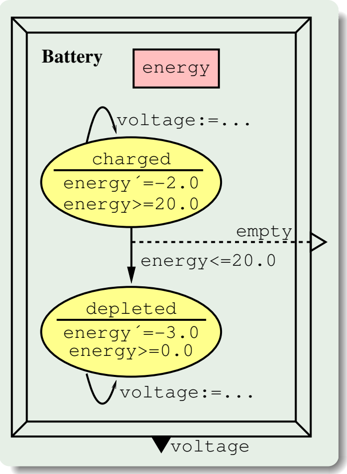
```
device Battery
  features
    empty: out event port;
    voltage: out data port real default 6.0;
end Battery;

device implementation Battery.Imp


  modes
    charged: activation mode

    depleted: mode

  transitions
    charged -[]-> charged;
    charged -[empty]-> depleted;
    depleted -[]-> depleted;
end Battery.Imp;
```



Modeling a Battery

Adding hybrid behavior:

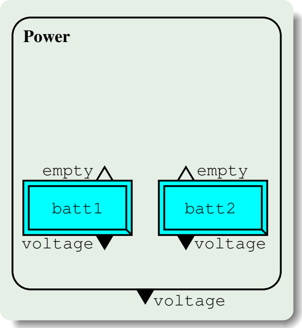
```
device Battery
  features
    empty: out event port;
    voltage: out data port real default 6.0;
end Battery;

device implementation Battery.Imp
  subcomponents
    energy: data continuous default 100.0;
  modes
    charged: activation mode
      while energy'=-2.0 and energy>=20.0;
    depleted: mode
      while energy'=-3.0 and energy>=0.0;
  transitions
    charged -[then voltage:=energy/50.0+4.0]-> charged;
    charged -[empty when energy<=20.0]-> depleted;
    depleted -[then voltage:=energy/50.0+4.0]-> depleted;
end Battery.Imp;
```



Modeling the Redundant Power System

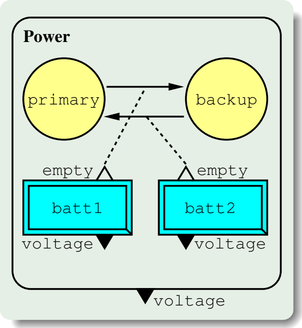
Power system with battery subcomponents:

```
system Power
  features
    voltage: out data port real;
end Power;

system implementation Power.Imp
  subcomponents
    batt1: device Battery.Imp
    batt2: device Battery.Imp








end Power.Imp;
```

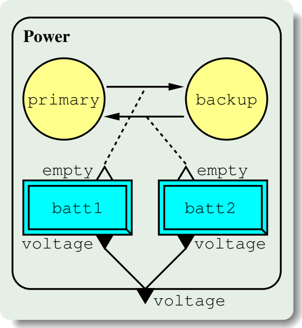


Adding dynamic reconfiguration:

```
system Power
  features
    voltage: out data port real;
end Power;

system implementation Power.Imp
  subcomponents
    batt1: device Battery.Imp in modes (primary);
    batt2: device Battery.Imp in modes (backup);


  modes
    primary: initial mode;
    backup: mode;
  transitions
    primary -[batt1.empty]-> backup;
    backup -[batt2.empty]-> primary;
end Power.Imp;
```



Modeling the Redundant Power System



Adding port connections:

```
system Power
  features
    voltage: out data port real;
end Power;

system implementation Power.Imp
  subcomponents
    batt1: device Battery.Imp in modes (primary);
    batt2: device Battery.Imp in modes (backup);
  connections
    data port batt1.voltage -> voltage in modes (primary);
    data port batt2.voltage -> voltage in modes (backup);
  modes
    primary: initial mode;
    backup: mode;
  transitions
    primary -[batt1.empty]-> backup;
    backup -[batt2.empty]-> primary;
end Power.Imp;
```

Outline

Operational Semantics of Single Components

- States: (mode, data values)
- Transitions: timed or internal or synchronized

Operational Semantics of Single Components

- States: (mode, data values)
- Transitions: timed or internal or synchronized

Example (Battery)

$$\langle \mathbf{mode} = \mathtt{charged}, \mathtt{energy} = 100.0, \mathtt{voltage} = 6.0 \rangle$$

Operational Semantics of Single Components

- States: (mode, data values)
- Transitions: timed or internal or synchronized

Example (Battery)

$$\langle \mathbf{mode} = \texttt{charged}, \texttt{energy} = 100.0, \texttt{voltage} = 6.0 \rangle$$

$$\downarrow 30.0$$

$$\langle \mathbf{mode} = \texttt{charged}, \texttt{energy} = 40.0, \texttt{voltage} = 6.0 \rangle$$

Operational Semantics of Single Components

- States: (mode, data values)
- Transitions: timed or internal or synchronized

Example (Battery)

$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 100.0, \texttt{voltage} = 6.0 \rangle$$
$$\downarrow 30.0$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 40.0, \texttt{voltage} = 6.0 \rangle$$
$$\downarrow \tau \langle \texttt{voltage:=...} \rangle$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 40.0, \texttt{voltage} = 4.8 \rangle$$

Operational Semantics of Single Components

- States: (mode, data values)
- Transitions: timed or internal or synchronized

Example (Battery)

$$\langle \mathtt{mode} = \mathtt{charged}, \mathtt{energy} = 100.0, \mathtt{voltage} = 6.0 \rangle$$
$$\downarrow 30.0$$
$$\langle \mathtt{mode} = \mathtt{charged}, \mathtt{energy} = 40.0, \mathtt{voltage} = 6.0 \rangle$$
$$\downarrow \tau \langle \mathtt{voltage:=}\ldots \rangle$$
$$\langle \mathtt{mode} = \mathtt{charged}, \mathtt{energy} = 40.0, \mathtt{voltage} = 4.8 \rangle$$
$$\downarrow 10.0$$
$$\langle \mathtt{mode} = \mathtt{charged}, \mathtt{energy} = 20.0, \mathtt{voltage} = 4.8 \rangle$$

Operational Semantics of Single Components

- States: (mode, data values)
- Transitions: timed or internal or synchronized

Example (Battery)

$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 100.0, \texttt{voltage} = 6.0\rangle$$
$$\downarrow 30.0$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 40.0, \texttt{voltage} = 6.0\rangle$$
$$\downarrow \tau\langle \texttt{voltage:=...}\rangle$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 40.0, \texttt{voltage} = 4.8\rangle$$
$$\downarrow 10.0$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 20.0, \texttt{voltage} = 4.8\rangle$$
$$\downarrow \tau\langle \texttt{voltage:=...}\rangle$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 20.0, \texttt{voltage} = 4.4\rangle$$

Operational Semantics of Single Components

- States: (mode, data values)
- Transitions: timed or internal or synchronized

Example (Battery)

$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 100.0, \texttt{voltage} = 6.0\rangle$$
$$\downarrow 30.0$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 40.0, \texttt{voltage} = 6.0\rangle$$
$$\downarrow \tau\langle \texttt{voltage:=...}\rangle$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 40.0, \texttt{voltage} = 4.8\rangle$$
$$\downarrow 10.0$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 20.0, \texttt{voltage} = 4.8\rangle$$
$$\downarrow \tau\langle \texttt{voltage:=...}\rangle$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 20.0, \texttt{voltage} = 4.4\rangle$$
$$\downarrow \texttt{empty}$$
$$\langle \texttt{mode} = \texttt{depleted}, \texttt{energy} = 20.0, \texttt{voltage} = 4.4\rangle$$

Operational Semantics of Single Components

- States: (mode, data values)
- Transitions: timed or internal or synchronized

Example (Battery)

$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 100.0, \texttt{voltage} = 6.0\rangle$$
$$\downarrow 30.0$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 40.0, \texttt{voltage} = 6.0\rangle$$
$$\downarrow \tau\langle \texttt{voltage:=}\ldots\rangle$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 40.0, \texttt{voltage} = 4.8\rangle$$
$$\downarrow 10.0$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 20.0, \texttt{voltage} = 4.8\rangle$$
$$\downarrow \tau\langle \texttt{voltage:=}\ldots\rangle$$
$$\langle \texttt{mode} = \texttt{charged}, \texttt{energy} = 20.0, \texttt{voltage} = 4.4\rangle$$
$$\downarrow \texttt{empty}$$
$$\langle \texttt{mode} = \texttt{depleted}, \texttt{energy} = 20.0, \texttt{voltage} = 4.4\rangle$$
$$\downarrow \cdots$$

The Power System Revisited

```
system Power
  features
    voltage: out data port real;
end Power;

system implementation Power.Imp
  subcomponents
    batt1: device Battery.Imp in modes (primary);
    batt2: device Battery.Imp in modes (backup);
  connections
    data port batt1.voltage -> voltage
      in modes (primary);
    data port batt2.voltage -> voltage
      in modes (backup);
  modes
    primary: initial mode;
    backup: mode;
  transitions
    primary -[batt1.empty]-> backup;
    backup -[batt2.empty]-> primary;
end Power.Imp;
```

Operational Semantics of Systems

- States: (mode, data values)$^+$
- Transitions determined by active components:
  1. Perform local transitions:
     - timed local transition in all components or
     - internal transition in component or
     - multi-way event communication from component to ≥ 1 connected components
  2. Initialize (re-)activated subcomponents
  3. Evaluate data connections (copy source $\rightarrow$ target data port)

Operational Semantics of Systems

- States: (mode, data values)$^{+}$
- Transitions determined by <u>active</u> components:
  1. Perform local transitions:
     - timed local transition in all components or
     - internal transition in component or
     - multi-way event communication from component to ≥ 1 connected components
  2. Initialize (re-)activated subcomponents
  3. Evaluate data connections (copy source $\rightarrow$ target data port)

Example (Power system)

$\langle \mathtt{m} = \underline{\mathtt{primary}}, \mathtt{v} = 6.0\rangle \,|\, \langle \mathtt{m} = \underline{\mathtt{charged}}, \mathtt{e} = 100.0, \mathtt{v} = 6.0\rangle \,|\, \langle \mathtt{m} = \mathtt{charged}, \mathtt{e} = 100.0, \mathtt{v} = 6.0\rangle$

Operational Semantics of Systems

- States: (mode, data values)$^{+}$
- Transitions determined by active components:
  1. Perform local transitions:
     - timed local transition in all components or
     - internal transition in component or
     - multi-way event communication from component to ≥ 1 connected components
  2. Initialize (re-)activated subcomponents
  3. Evaluate data connections (copy source $\rightarrow$ target data port)

Example (Power system)

$$\langle \mathtt{m} = \underline{\mathtt{primary}}, \mathtt{v} = 6.0 \rangle \,|\, \langle \mathtt{m} = \underline{\mathtt{charged}}, \mathtt{e} = 100.0, \mathtt{v} = 6.0 \rangle \,|\, \langle \mathtt{m} = \mathtt{charged}, \mathtt{e} = 100.0, \mathtt{v} = 6.0 \rangle$$

$$\Downarrow 40.0$$

$$\langle \mathtt{m} = \underline{\mathtt{primary}}, \mathtt{v} = 6.0 \rangle \,|\, \langle \mathtt{m} = \underline{\mathtt{charged}}, \mathtt{e} = 20.0, \mathtt{v} = 6.0 \rangle \,|\, \langle \mathtt{m} = \mathtt{charged}, \mathtt{e} = 100.0, \mathtt{v} = 6.0 \rangle$$

Operational Semantics of Systems

- States: (mode, data values)$^+$
- Transitions determined by active components:
  1. Perform local transitions:
     - timed local transition in all components or
     - internal transition in component or
     - multi-way event communication from component to ≥ 1 connected components
  2. Initialize (re-)activated subcomponents
  3. Evaluate data connections (copy source $\rightarrow$ target data port)

Example (Power system)

$$\langle \mathtt{m}=\underline{\mathtt{primary}}, \mathtt{v}=6.0\rangle | \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle | \langle \mathtt{m}=\mathtt{charged}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$

$$\Downarrow 40.0$$

$$\langle \mathtt{m}=\underline{\mathtt{primary}}, \mathtt{v}=6.0\rangle | \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=20.0, \mathtt{v}=6.0\rangle | \langle \mathtt{m}=\mathtt{charged}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$

$$\Downarrow \tau\langle \mathtt{voltage:=}\ldots\rangle$$

$$\langle \mathtt{m}=\underline{\mathtt{primary}}, \mathtt{v}=4.4\rangle | \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=20.0, \mathtt{v}=4.4\rangle | \langle \mathtt{m}=\mathtt{charged}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$

Operational Semantics of Systems

- States: (mode, data values)$^+$
- Transitions determined by <u>active</u> components:
  1. Perform local transitions:
     - timed local transition in all components or
     - internal transition in component or
     - multi-way event communication from component to ≥ 1 connected components
  2. Initialize (re-)activated subcomponents
  3. Evaluate data connections (copy source $\rightarrow$ target data port)

Example (Power system)

$$\langle \mathtt{m}=\underline{\mathtt{primary}}, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\mathtt{charged}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$
$$\Downarrow 40.0$$
$$\langle \mathtt{m}=\underline{\mathtt{primary}}, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=20.0, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\mathtt{charged}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$
$$\Downarrow \tau\langle \mathtt{voltage:=...}\rangle$$
$$\langle \mathtt{m}=\underline{\mathtt{primary}}, \mathtt{v}=4.4\rangle \,|\, \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=20.0, \mathtt{v}=4.4\rangle \,|\, \langle \mathtt{m}=\mathtt{charged}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$
$$\Downarrow \tau\langle \mathtt{empty}\rangle$$
$$\langle \mathtt{m}=\underline{\mathtt{backup}}, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\mathtt{depleted}, \mathtt{e}=20.0, \mathtt{v}=4.4\rangle \,|\, \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$

Operational Semantics of Systems

- States: (mode, data values)$^{+}$
- Transitions determined by <u>active</u> components:
  1. Perform local transitions:
     - timed local transition in all components or
     - internal transition in component or
     - multi-way event communication from component to ≥ 1 connected components
  2. Initialize (re-)activated subcomponents
  3. Evaluate data connections (copy source $\rightarrow$ target data port)

Example (Power system)

$$\langle \mathtt{m}=\underline{\mathtt{primary}}, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\mathtt{charged}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$
$$\Downarrow 40.0$$
$$\langle \mathtt{m}=\underline{\mathtt{primary}}, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=20.0, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\mathtt{charged}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$
$$\Downarrow \tau\langle \mathtt{voltage:=...}\rangle$$
$$\langle \mathtt{m}=\underline{\mathtt{primary}}, \mathtt{v}=4.4\rangle \,|\, \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=20.0, \mathtt{v}=4.4\rangle \,|\, \langle \mathtt{m}=\mathtt{charged}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$
$$\Downarrow \tau\langle \mathtt{empty}\rangle$$
$$\langle \mathtt{m}=\underline{\mathtt{backup}}, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\mathtt{depleted}, \mathtt{e}=20.0, \mathtt{v}=4.4\rangle \,|\, \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=100.0, \mathtt{v}=6.0\rangle$$
$$\Downarrow 40.0$$
$$\langle \mathtt{m}=\underline{\mathtt{backup}}, \mathtt{v}=6.0\rangle \,|\, \langle \mathtt{m}=\mathtt{depleted}, \mathtt{e}=20.0, \mathtt{v}=4.4\rangle \,|\, \langle \mathtt{m}=\underline{\mathtt{charged}}, \mathtt{e}=20.0, \mathtt{v}=6.0\rangle$$

Operational Semantics of Systems

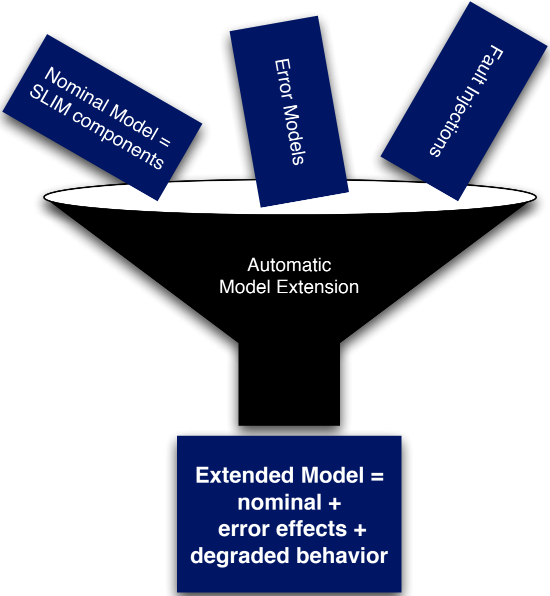
- States: (mode, data values)$^+$
- Transitions determined by <u>active</u> components:
  1. Perform local transitions:
     - timed local transition in all components or
     - internal transition in component or
     - multi-way event communication from component to ≥ 1 connected components
  2. Initialize (re-)activated subcomponents
  3. Evaluate data connections (copy source $\rightarrow$ target data port)

Example (Power system)

$$\langle \mathtt{m} = \underline{\mathtt{primary}}, \mathtt{v} = 6.0 \rangle | \langle \mathtt{m} = \underline{\mathtt{charged}}, \mathtt{e} = 100.0, \mathtt{v} = 6.0 \rangle | \langle \mathtt{m} = \mathtt{charged}, \mathtt{e} = 100.0, \mathtt{v} = 6.0 \rangle$$
$$\Downarrow 40.0$$
$$\langle \mathtt{m} = \underline{\mathtt{primary}}, \mathtt{v} = 6.0 \rangle | \langle \mathtt{m} = \underline{\mathtt{charged}}, \mathtt{e} = 20.0, \mathtt{v} = 6.0 \rangle | \langle \mathtt{m} = \mathtt{charged}, \mathtt{e} = 100.0, \mathtt{v} = 6.0 \rangle$$
$$\Downarrow \tau \langle \mathtt{voltage:=...} \rangle$$
$$\langle \mathtt{m} = \underline{\mathtt{primary}}, \mathtt{v} = 4.4 \rangle | \langle \mathtt{m} = \underline{\mathtt{charged}}, \mathtt{e} = 20.0, \mathtt{v} = 4.4 \rangle | \langle \mathtt{m} = \mathtt{charged}, \mathtt{e} = 100.0, \mathtt{v} = 6.0 \rangle$$
$$\Downarrow \tau \langle \mathtt{empty} \rangle$$
$$\langle \mathtt{m} = \underline{\mathtt{backup}}, \mathtt{v} = 6.0 \rangle | \langle \mathtt{m} = \mathtt{depleted}, \mathtt{e} = 20.0, \mathtt{v} = 4.4 \rangle | \langle \mathtt{m} = \underline{\mathtt{charged}}, \mathtt{e} = 100.0, \mathtt{v} = 6.0 \rangle$$
$$\Downarrow 40.0$$
$$\langle \mathtt{m} = \underline{\mathtt{backup}}, \mathtt{v} = 6.0 \rangle | \langle \mathtt{m} = \mathtt{depleted}, \mathtt{e} = 20.0, \mathtt{v} = 4.4 \rangle | \langle \mathtt{m} = \underline{\mathtt{charged}}, \mathtt{e} = 20.0, \mathtt{v} = 6.0 \rangle$$
$$\Downarrow \cdots$$

Outline

Integrating Erroneous and Nominal Behavior



Error Modeling

```
error model BatteryFailure
  features
    ok: initial state;
    dead:  error state;
    batteryDied:  out error propagation;
end BatteryFailure;

error model implementation BatteryFailure.Imp
  events
    fault:  error event occurrence poisson 0.01;
  transitions
    ok -[fault]-> dead;
    dead -[batteryDied]-> dead;
end BatteryFailure.Imp;
```

Error Modeling

```
error model BatteryFailure
  features
    ok: initial state;
    dead: error state;
    batteryDied: out error propagation;
end BatteryFailure;

error model implementation BatteryFailure.Imp
  events
    fault: error event occurrence poisson 0.01;
  transitions
    ok -[fault]-> dead;
    dead -[batteryDied]-> dead;
end BatteryFailure.Imp;
```

Fault injection

An error model does not influence the nominal behavior unless they are linked through fault injection: (s, d, a) means that on entering error state s, the assignment $d := a$ is performed, where d is a data element and a the fault effect.

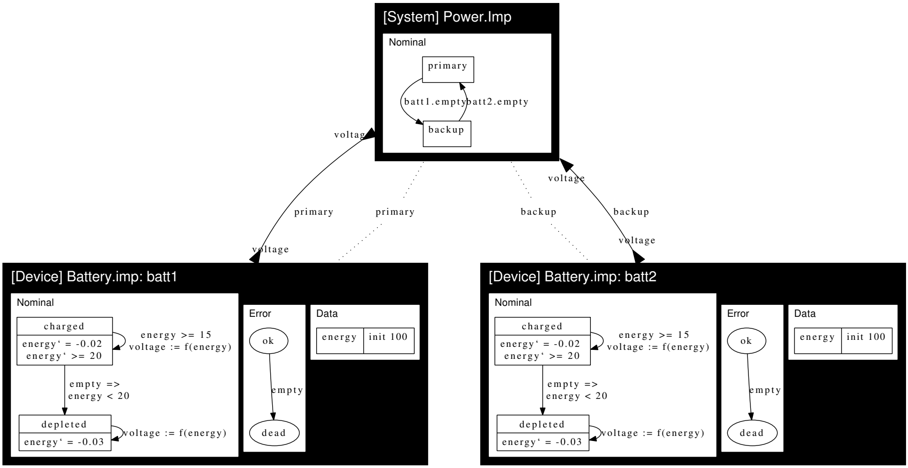
Error Modeling

```
error model BatteryFailure
  features
    ok: initial state;
    dead: error state;
    batteryDied: out error propagation;
end BatteryFailure;

error model implementation BatteryFailure.Imp
  events
    fault: error event occurrence poisson 0.01;
  transitions
    ok -[fault]-> dead;
    dead -[batteryDied]-> dead;
end BatteryFailure.Imp;
```

Fault injection

In error state `dead`, `voltage := 0`

The Complete Power System Model



Model Extension

Nominal model + error model + fault injections = extended model

- Modes are pairs of nominal modes and error model states
  - Starting mode = (original starting mode, starting error state)
- Event ports +:= error propagations
- Event port connections +:= propagation port connections
- Transition relation := all possible interleavings and interactions between nominal and error model, taking failure effects into account
- Other elements (e.g., mode invariants) are unaffected

Model Extension

Nominal model $+$ error model $+$ fault injections $=$ extended model

- Modes are pairs of nominal modes and error model states
  - Starting mode $=$ (original starting mode, starting error state)
- Event ports $+:=$ error propagations
- Event port connections $+:=$ propagation port connections
- Transition relation $:=$ all possible interleavings and interactions between nominal and error model, taking failure effects into account
- Other elements (e.g., mode invariants) are unaffected

Probabilistic error transitions

As an error model has probabilistic transitions, our semantical model has to be equipped with such transitions.

This yields interactive Markov chains $:=$ LTS $+$ Markov chains.

Battery Component

Nominal specification

```
device Battery
  features
    empty: out event port;
    voltage: out data port real default 6.0;

end Battery;

device implementation Battery.Imp
  subcomponents
    energy: data continuous default 100.0;
  modes
    charged: activation mode while ...;
    depleted: mode while ...;
  transitions
    charged -[then voltage:=...]-> charged;
    charged -[empty when energy<=20.0]-> depleted;
    depleted -[then voltage:=...]-> depleted;






end Battery.Imp;
```

Battery Component After Model Extension

Product construction for modes

```
device Battery
  features
    empty: out event port;
    voltage: out data port real default 6.0;

end Battery;

device implementation Battery.Imp
  subcomponents
    energy: data continuous default 100.0;
  modes
    charged#ok: activation mode while ...;
    depleted#ok, charged#dead, depleted#dead: mode while ...;
  transitions
    charged -[then voltage:=...]-> charged;
    charged -[empty when energy<=20.0]-> depleted;
    depleted -[then voltage:=...]-> depleted;

end Battery.Imp;
```

Battery Component After Model Extension

Integration of nominal transitions

```
device Battery
  features
    empty: out event port;
    voltage: out data port real default 6.0;

end Battery;

device implementation Battery.Imp
  subcomponents
    energy: data continuous default 100.0;
  modes
    charged#ok: activation mode while ...;
    depleted#ok, charged#dead, depleted#dead: mode while ...;
  transitions
    charged#ok -[then voltage:=...]-> charged#ok;
    charged#ok -[empty when energy<=20.0]-> depleted#ok;
    depleted#ok -[then voltage:=...]-> depleted#ok;




end Battery.Imp;
```

Battery Component After Model Extension

Fault injection

```
device Battery
  features
    empty: out event port;
    voltage: out data port real default 6.0;

end Battery;

device implementation Battery.Imp
  subcomponents
    energy: data continuous default 100.0;
  modes
    charged#ok: activation mode while ...;
    depleted#ok, charged#dead, depleted#dead: mode while ...;
  transitions
    charged#ok -[then voltage:=...]-> charged#ok;
    charged#ok -[empty when energy<=20.0]-> depleted#ok;
    depleted#ok -[then voltage:=...]-> depleted#ok;
    charged#ok -[prob 0.01 then voltage:=0]-> charged#dead;
    depleted#ok -[prob 0.01 then voltage:=0]-> depleted#dead;



end Battery.Imp;
```

Battery Component After Model Extension

Original transitions with fault effects

```
device Battery
  features
    empty: out event port;
    voltage: out data port real default 6.0;

end Battery;

device implementation Battery.Imp
  subcomponents
    energy: data continuous default 100.0;
  modes
    charged#ok: activation mode while ...;
    depleted#ok, charged#dead, depleted#dead: mode while ...;
  transitions
    charged#ok -[then voltage:=...]-> charged#ok;
    charged#ok -[empty when energy<=20.0]-> depleted#ok;
    depleted#ok -[then voltage:=...]-> depleted#ok;
    charged#ok -[prob 0.01 then voltage:=0]-> charged#dead;
    depleted#ok -[prob 0.01 then voltage:=0]-> depleted#dead;
    charged#dead -[then voltage:=0]-> charged#dead;
    charged#dead -[empty when energy<=20.0]-> depleted#dead;
    depleted#dead -[then voltage:=0]-> depleted#dead;

end Battery.Imp;
```

Battery Component After Model Extension

Addition of error propagations

```
device Battery
  features
    empty: out event port;
    voltage: out data port real default 6.0;
    batteryDied: out event port;
end Battery;

device implementation Battery.Imp
  subcomponents
    energy: data continuous default 100.0;
  modes
    charged#ok: activation mode while ...;
    depleted#ok, charged#dead, depleted#dead: mode while ...;
  transitions
    charged#ok -[then voltage:=...]-> charged#ok;
    charged#ok -[empty when energy<=20.0]-> depleted#ok;
    depleted#ok -[then voltage:=...]-> depleted#ok;
    charged#ok -[prob 0.01 then voltage:=0]-> charged#dead;
    depleted#ok -[prob 0.01 then voltage:=0]-> depleted#dead;
    charged#dead -[then voltage:=0]-> charged#dead;
    charged#dead -[empty when energy<=20.0]-> depleted#dead;
    depleted#dead -[then voltage:=0]-> depleted#dead;
    depleted#dead -[batteryDied]-> depleted#dead;
end Battery.Imp;
```

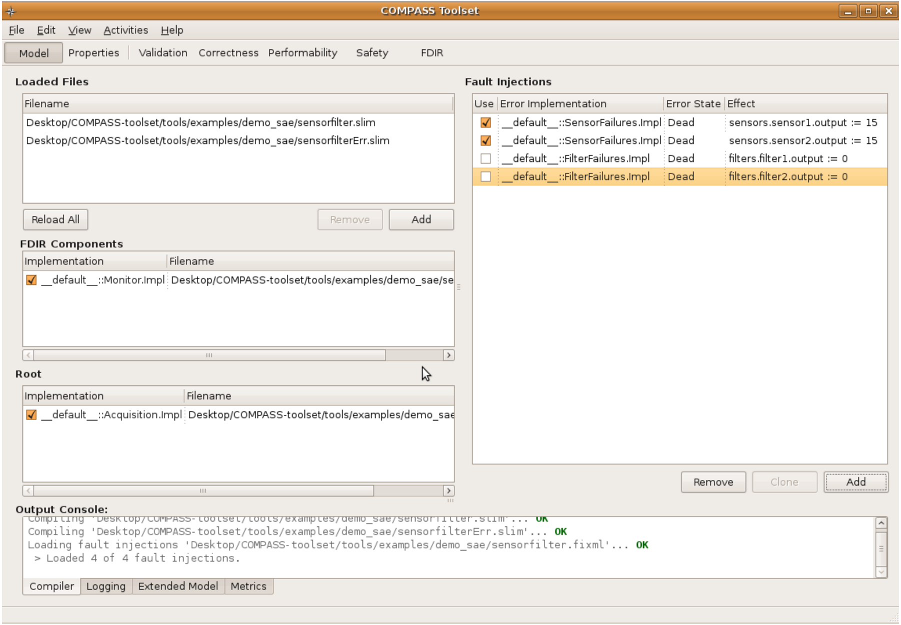
Specifying Observability

- Specification of observables for diagnosability analysis (later)
  - for outgoing data ports of type **bool**
- Example:

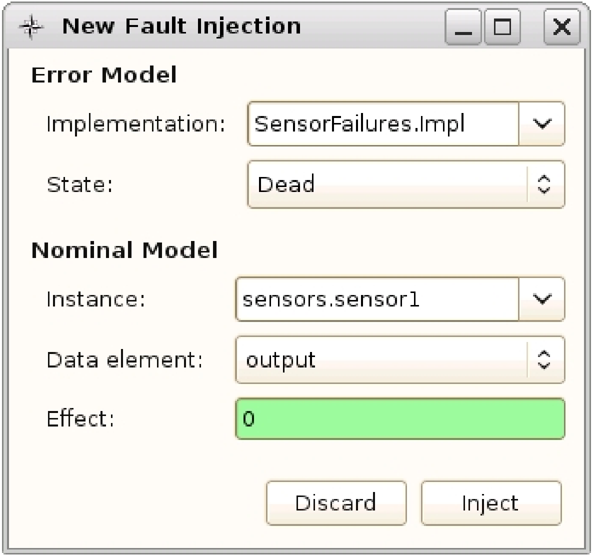
```
system PowerSystem
  features
    voltage: out data port real;
    alarm: out data port bool initially false observable;
end PowerSystem;

system implementation PowerSystem.Imp
  subcomponents
    pow: system Power.Imp;
  connections
    data port pow.voltage -> voltage;
  modes
    normal: initial mode;
    critical: mode;
  transitions
    normal -[when voltage<4.5 then alarm:=true]-> critical;
    critical -[when voltage>5.5 then alarm:=false]-> normal;
end PowerSystem.Imp;
```

Outline

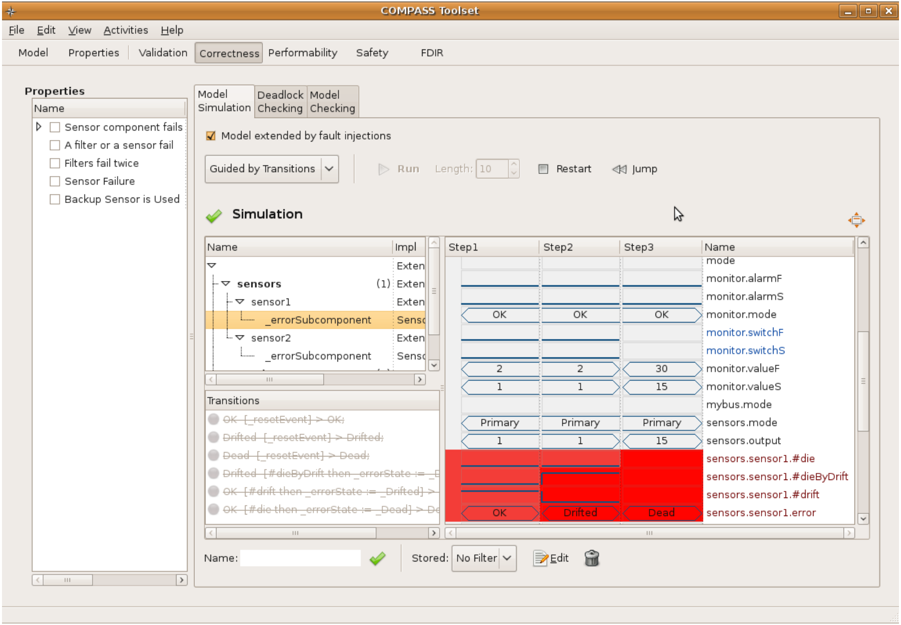
Loading Models



Defining Fault Injections



Simulating System Behavior



Outline

References & Ongoing Activities

References

- Our AADL variant (Bozzano et. al, MEMOCODE 2009)
- AADL formal semantics (Bozzano et. al, Computer J. 2010)
- Relation to attribute grammars (Noll, FACS 2011)

References & Ongoing Activities

References

- Our AADL variant (Bozzano et. al, MEMOCODE 2009)
- AADL formal semantics (Bozzano et. al, Computer J. 2010)
- Relation to attribute grammars (Noll, FACS 2011)

Ongoing activities

- Contribution to AADL standardization
- Modeling features for launcher systems
- Security aspects in AADL (D-MILS Project)