iFM & ABZ 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
Joost-Pieter Katoen & Thomas Noll

Fondazione Bruno Kessler
Centre for Scientific and Technological Research
Marco Bozzano & Alessandro Cimatti

iFM & ABZ 2012; June 18, 2012; Pisa, Italy

Contents

Outline

The Battery Revisited

```
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;
```

Event-Data Automata

Definition (Event-data automaton)

An event-data automaton (EDA) is a tuple

$$\mathfrak{A} = (M, X, V, \iota, E, \rightarrow)$$

with

- M finite set of modes
- $X = IX \uplus OX \uplus LX$ finite set of input/output/local variables
- $V := \{v \mid v : X \rightarrow \ldots\}$ valuations
- $\iota : M \rightarrow (V \rightarrow \mathbb{B})$ mode invariants
- $E = IE \uplus OE$ finite set of input/output events
- $\rightarrow \subseteq M \times \underbrace{(E \cup \{\tau\})}_{\text{trigger}} \times \underbrace{(V \rightarrow \mathbb{B})}_{\text{guard}} \times \underbrace{(V \rightarrow V)}_{\text{effect}} \times M$ transition relation

Event-Data Automata

Definition (Event-data automaton)

An event-data automaton (EDA) is a tuple

$$\mathfrak{A} = (M, X, V, \iota, E, \rightarrow)$$

with

$\{\texttt{charged}, \texttt{depleted}\}$

- M finite set of modes
- $X = IX \uplus OX \uplus LX$ finite set of input/output/local variables
- $V := \{v \mid v : X \rightarrow \ldots\}$ valuations
- $\iota : M \rightarrow (V \rightarrow \mathbb{B})$ mode invariants
- $E = IE \uplus OE$ finite set of input/output events
- $\rightarrow \subseteq M \times \underbrace{(E \cup \{\tau\})}_{\text{trigger}} \times \underbrace{(V \rightarrow \mathbb{B})}_{\text{guard}} \times \underbrace{(V \rightarrow V)}_{\text{effect}} \times M$ transition relation

Event-Data Automata

Definition (Event-data automaton)

An event-data automaton (EDA) is a tuple

$$\mathfrak{A} = (M, X, V, \iota, E, \rightarrow)$$

with

- M finite set of modes
- $X = IX \uplus OX \uplus LX$ finite set of input/output/local variables
- $V := \{v \mid v : X \rightarrow \ldots\}$ valuations
- $\iota : M \rightarrow (V \rightarrow \mathbb{B})$ mode invariants
- $E = IE \uplus OE$ finite set of input/output events
- $\rightarrow \subseteq M \times \underbrace{(E \cup \{\tau\})}_{\text{trigger}} \times \underbrace{(V \rightarrow \mathbb{B})}_{\text{guard}} \times \underbrace{(V \rightarrow V)}_{\text{effect}} \times M$ transition relation

Event-Data Automata

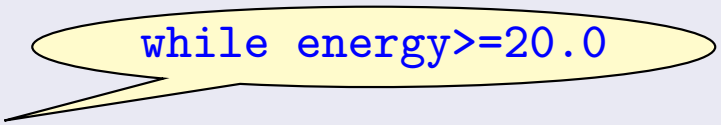
Definition (Event-data automaton)

An event-data automaton (EDA) is a tuple

$$\mathfrak{A} = (M, X, V, \iota, E, \rightarrow)$$

with

- M finite set of modes
- $X = IX \uplus OX \uplus LX$ finite set of input/output/local variables
- $V := \{v \mid v : X \rightarrow \ldots\}$ valuations
- $\iota : M \rightarrow (V \rightarrow \mathbb{B})$ mode invariants



- $E = IE \uplus OE$ finite set of input/output events
- $\rightarrow \subseteq M \times \underbrace{(E \cup \{\tau\})}_{\text{trigger}} \times \underbrace{(V \rightarrow \mathbb{B})}_{\text{guard}} \times \underbrace{(V \rightarrow V)}_{\text{effect}} \times M$ transition relation

Event-Data Automata

Definition (Event-data automaton)

An event-data automaton (EDA) is a tuple

$$\mathfrak{A} = (M, X, V, \iota, E, \rightarrow)$$

with

- M finite set of modes
- $X = IX \uplus OX \uplus LX$ finite set of input/output/local variables
- $V := \{v \mid v : X \rightarrow \ldots\}$ valuations
- $\iota : M \rightarrow (V \rightarrow \mathbb{B})$ mode invariants
- $E = IE \uplus OE$ finite set of input/output events

empty

- $\rightarrow \subseteq M \times \underbrace{(E \cup \{\tau\})}_{\text{trigger}} \times \underbrace{(V \rightarrow \mathbb{B})}_{\text{guard}} \times \underbrace{(V \rightarrow V)}_{\text{effect}} \times M$ transition relation

Event-Data Automata

Definition (Event-data automaton)

An event-data automaton (EDA) is a tuple

$$\mathfrak{A} = (M, X, V, \iota, E, \rightarrow)$$

with

- M finite set of modes
- $X = IX \uplus OX \uplus LX$ finite set of input/output/local variables
- $V := \{v \mid v : X \rightarrow \ldots\}$ valuations
- $\iota : M \rightarrow (V \rightarrow \mathbb{B})$ mode invariants
- $E = IE \uplus OE$ finite set of input/output events
- $\rightarrow \subseteq M \times \underbrace{(E \cup \{\tau\})}_{\text{trigger}} \times \underbrace{(V \rightarrow \mathbb{B})}_{\text{guard}} \times \underbrace{(V \rightarrow V)}_{\text{effect}} \times M$ transition relation

`when energy<=20.0`

`empty`

`then voltage:=...`

Operational Semantics of EDAs

- States: $M \times V$
- Transitions: timed or internal or synchronized

Operational Semantics of EDAs

- States: $M \times V$
- Transitions: timed or internal or synchronized

Example (Battery)

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

Operational Semantics of EDAs

- States: $M \times V$
- Transitions: timed or internal or synchronized

Example (Battery)

$$\langle \mathbf{mode} = \mathtt{charged}, \mathtt{energy} = 100.0, \mathtt{voltage} = 6.0 \rangle$$
$$\downarrow 30.0$$
$$\langle \mathbf{mode} = \mathtt{charged}, \mathtt{energy} = 40.0, \mathtt{voltage} = 6.0 \rangle$$

Operational Semantics of EDAs

- States: $M \times V$
- Transitions: timed or internal or synchronized

Example (Battery)

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

Operational Semantics of EDAs

- States: $M \times V$
- Transitions: timed or internal or synchronized

Example (Battery)

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

Operational Semantics of EDAs

- States: $M \times V$
- Transitions: timed or internal or synchronized

Example (Battery)

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

Operational Semantics of EDAs

- States: $M \times V$
- Transitions: timed or internal or synchronized

Example (Battery)

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

Operational Semantics of EDAs

- States: $M \times V$
- Transitions: timed or internal or synchronized

Example (Battery)

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

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;
```

Networks of Event-Data Automata

Dynamic reconfiguration

$\Rightarrow$ component activity and port connections mode dependent

Definition (Networks of Event-Data Automata)

A network of event-data automata (NEDA) is a tuple

$$\mathfrak{N} = ((\mathfrak{A}_i)_{i \in [n]}, \alpha, EC, DC)$$

with $n \geq 1$, $[n] := \{1, \ldots, n\}$, and

- each $\mathfrak{A}_i$ an EDA $\mathfrak{A}_i = (M_i, X_i, V_i, \iota_i, E_i, \rightarrow_i)$
- $M := \prod_{i=1}^{n} M_i$ set of global modes
- $\alpha : M \rightarrow 2^{[n]}$ activation mapping
- $EC : M \rightarrow (\{i.e \mid i \in [n], e \in E_i\})^2$ event connection mapping
- $DC : M \rightarrow (\{i.x \mid i \in [n], x \in X_i\})^2$ data connection mapping

The Activation Mapping

Definition (Activation mapping)

α determined by subcomponent declarations:

- Root component always active
- Component c active and in mode m, subcomponent c' of c activated in m $\Rightarrow$ c' active

The Activation Mapping

Definition (Activation mapping)

α determined by subcomponent declarations:

- Root component always active
- Component c active and in mode m, subcomponent c' of c activated in m $\Rightarrow$ c' active

Example (Power system)

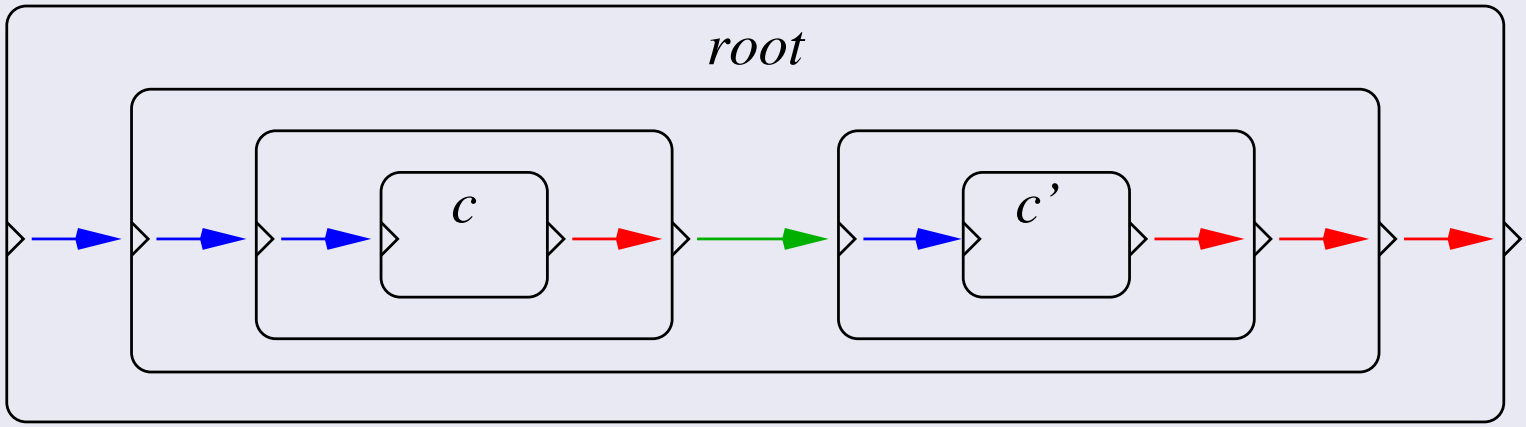
For $\texttt{Power}/\texttt{Battery1}/\texttt{Battery2}$ ($m_1, m_2 \in \{\texttt{charged}, \texttt{depleted}\}$):

$$\alpha(\texttt{primary}, m_1, m_2) = \{\texttt{Power}, \texttt{Battery1}\}$$
$$\alpha(\texttt{backup}, m_1, m_2) = \{\texttt{Power}, \texttt{Battery2}\}$$

The Connection Mappings

Definition (Event/data connection mappings)

EC/*DC* determined by following all end-to-end chains of port connections:



in−to−in

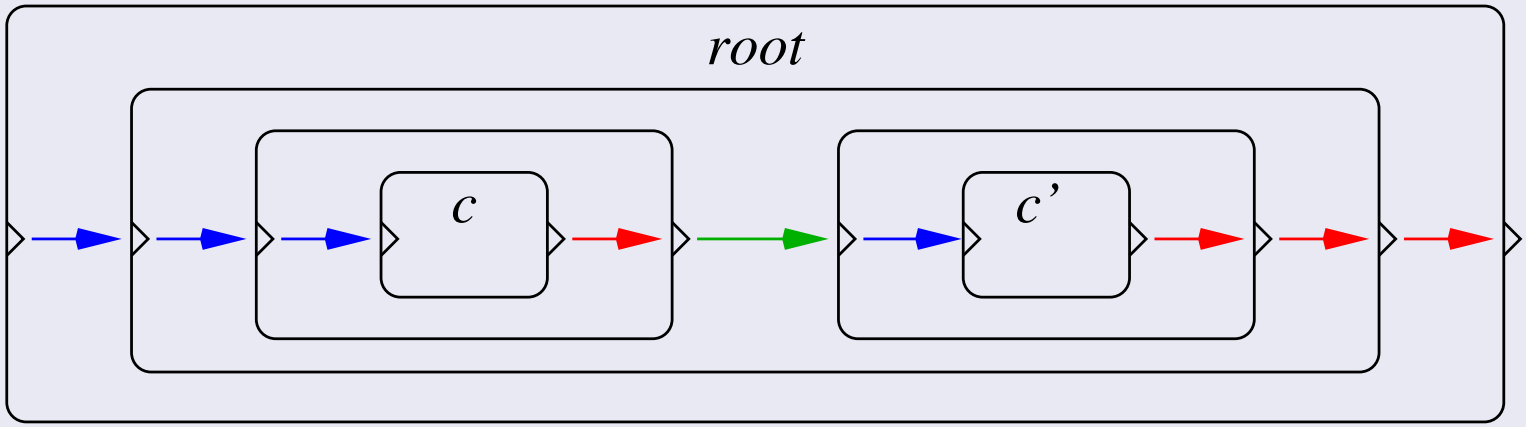
out−to−in

out−to−out

The Connection Mappings

Definition (Event/data connection mappings)

EC/*DC* determined by following all end-to-end chains of port connections:



Example (Power system)

For Power/Battery1/Battery2 ($m_1, m_2 \in \{\texttt{charged}, \texttt{depleted}\}$):

$$EC(\texttt{primary}, m_1, m_2) = \{\texttt{Battery1.empty} \rightarrow \texttt{Power.empty}\}$$
$$EC(\texttt{backup}, m_1, m_2) = \{\texttt{Battery2.empty} \rightarrow \texttt{Power.empty}\}$$
$$DC(\texttt{primary}, m_1, m_2) = \{\texttt{Battery1.voltage} \rightarrow \texttt{Power.voltage}\}$$
$$DC(\texttt{backup}, m_1, m_2) = \{\texttt{Battery2.voltage} \rightarrow \texttt{Power.voltage}\}$$

Operational Semantics of Networks of EDAs

- States $:= (M_1 \times V_1) \times \ldots \times (M_n \times V_n)$
- Transitions determined by active EDAs:
  1. Perform local transitions:
     - timed local transition in all EDAs or
     - internal transition in EDA or
     - multi-way event communication from EDA to ≥ 1 connected EDAs
  2. Initialize (re-)activated subcomponents
  3. Establish consistency w.r.t. DC (copy source $\rightarrow$ target data port)

Operational Semantics of Networks of EDAs

- States := $(M_1 \times V_1) \times \ldots \times (M_n \times V_n)$
- Transitions determined by active EDAs:
  1. Perform local transitions:
     - timed local transition in all EDAs or
     - internal transition in EDA or
     - multi-way event communication from EDA to ≥ 1 connected EDAs
  2. Initialize (re-)activated subcomponents
  3. Establish consistency w.r.t. DC (copy source $\rightarrow$ target data port)

Example (Power system)

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

Operational Semantics of Networks of EDAs

- States := $(M_1 \times V_1) \times \ldots \times (M_n \times V_n)$
- Transitions determined by active EDAs:
  1. Perform local transitions:
     - timed local transition in all EDAs or
     - internal transition in EDA or
     - multi-way event communication from EDA to ≥ 1 connected EDAs
  2. Initialize (re-)activated subcomponents
  3. Establish consistency w.r.t. DC (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 Networks of EDAs

- States := $(M_1 \times V_1) \times \ldots \times (M_n \times V_n)$
- Transitions determined by active EDAs:
  1. Perform local transitions:
     - timed local transition in all EDAs or
     - internal transition in EDA or
     - multi-way event communication from EDA to ≥ 1 connected EDAs
  2. Initialize (re-)activated subcomponents
  3. Establish consistency w.r.t. DC (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 Networks of EDAs

- States := $(M_1 \times V_1) \times \ldots \times (M_n \times V_n)$
- Transitions determined by active EDAs:
  1. Perform local transitions:
     - timed local transition in all EDAs or
     - internal transition in EDA or
     - multi-way event communication from EDA to ≥ 1 connected EDAs
  2. Initialize (re-)activated subcomponents
  3. Establish consistency w.r.t. DC (copy source $\rightarrow$ target data port)

Example (Power system)

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

Operational Semantics of Networks of EDAs

- States := $(M_1 \times V_1) \times \ldots \times (M_n \times V_n)$
- Transitions determined by active EDAs:
  1. Perform local transitions:
     - timed local transition in all EDAs or
     - internal transition in EDA or
     - multi-way event communication from EDA to ≥ 1 connected EDAs
  2. Initialize (re-)activated subcomponents
  3. Establish consistency w.r.t. DC (copy source $\rightarrow$ target data port)

Example (Power system)

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

Operational Semantics of Networks of EDAs

- States := $(M_1 \times V_1) \times \ldots \times (M_n \times V_n)$
- Transitions determined by active EDAs:
  1. Perform local transitions:
     - timed local transition in all EDAs or
     - internal transition in EDA or
     - multi-way event communication from EDA to ≥ 1 connected EDAs
  2. Initialize (re-)activated subcomponents
  3. Establish consistency w.r.t. DC (copy source $\rightarrow$ target data port)

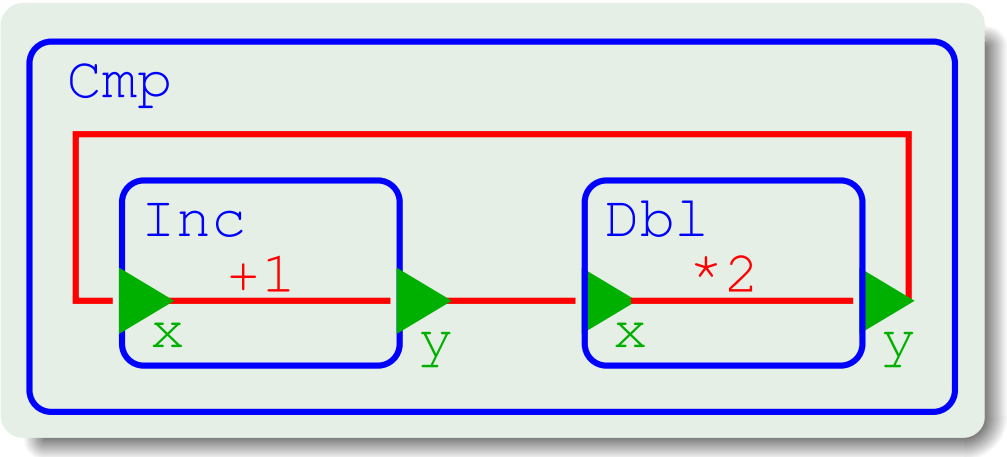
Example (Power system)

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

AADL Specifications and Attribute Grammars

Problem: data port dependencies can be cyclic

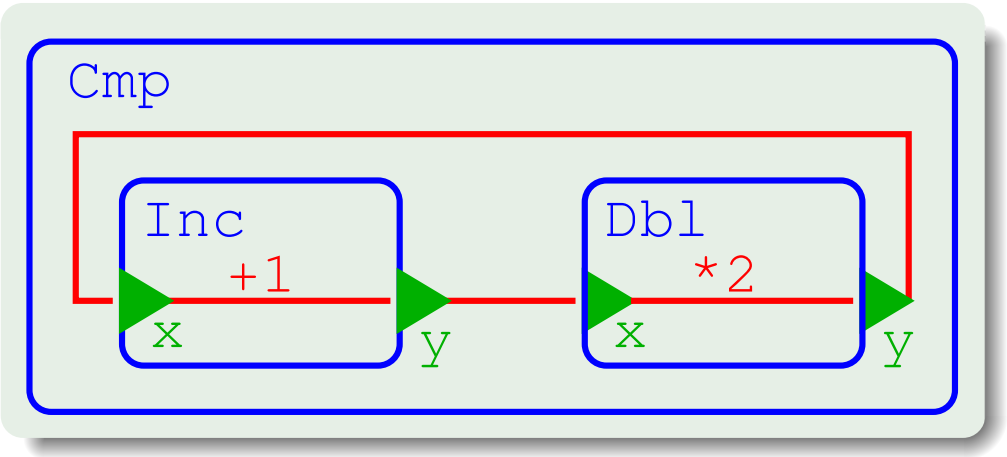
$\Rightarrow$ data port values possibly inconsistent



AADL Specifications and Attribute Grammars

Problem: data port dependencies can be cyclic

$\Rightarrow$ data port values possibly inconsistent

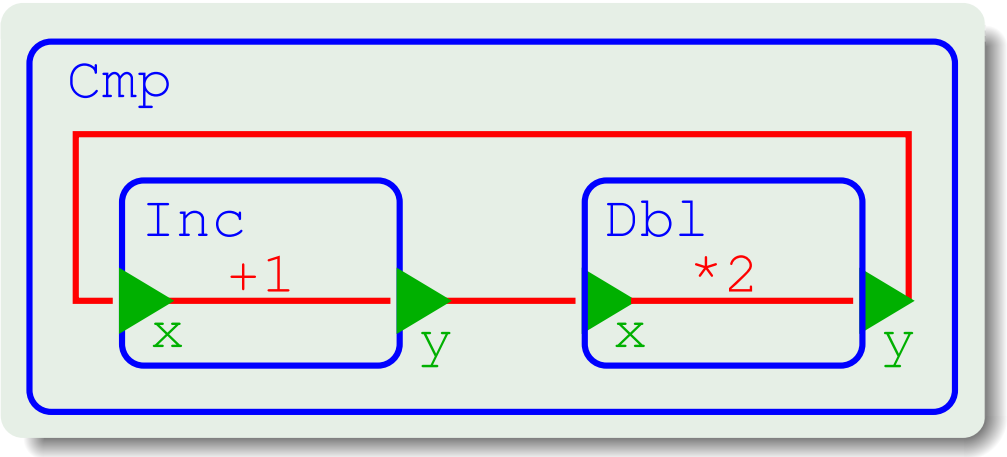


$\Rightarrow \texttt{Inc.x} = 2 \cdot (\texttt{Inc.x} + 1)$ ↯

AADL Specifications and Attribute Grammars

Problem: data port dependencies can be cyclic

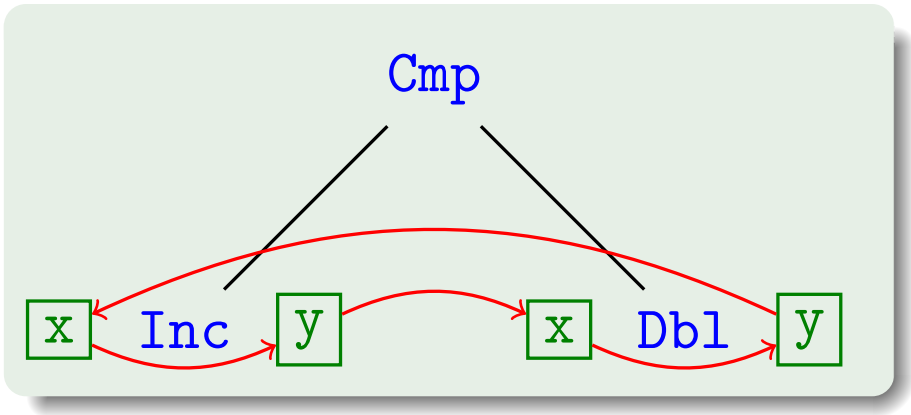
$\Rightarrow$ data port values possibly inconsistent



$\Rightarrow \texttt{Inc.x} = 2 \cdot (\texttt{Inc.x} + 1)$ ↯

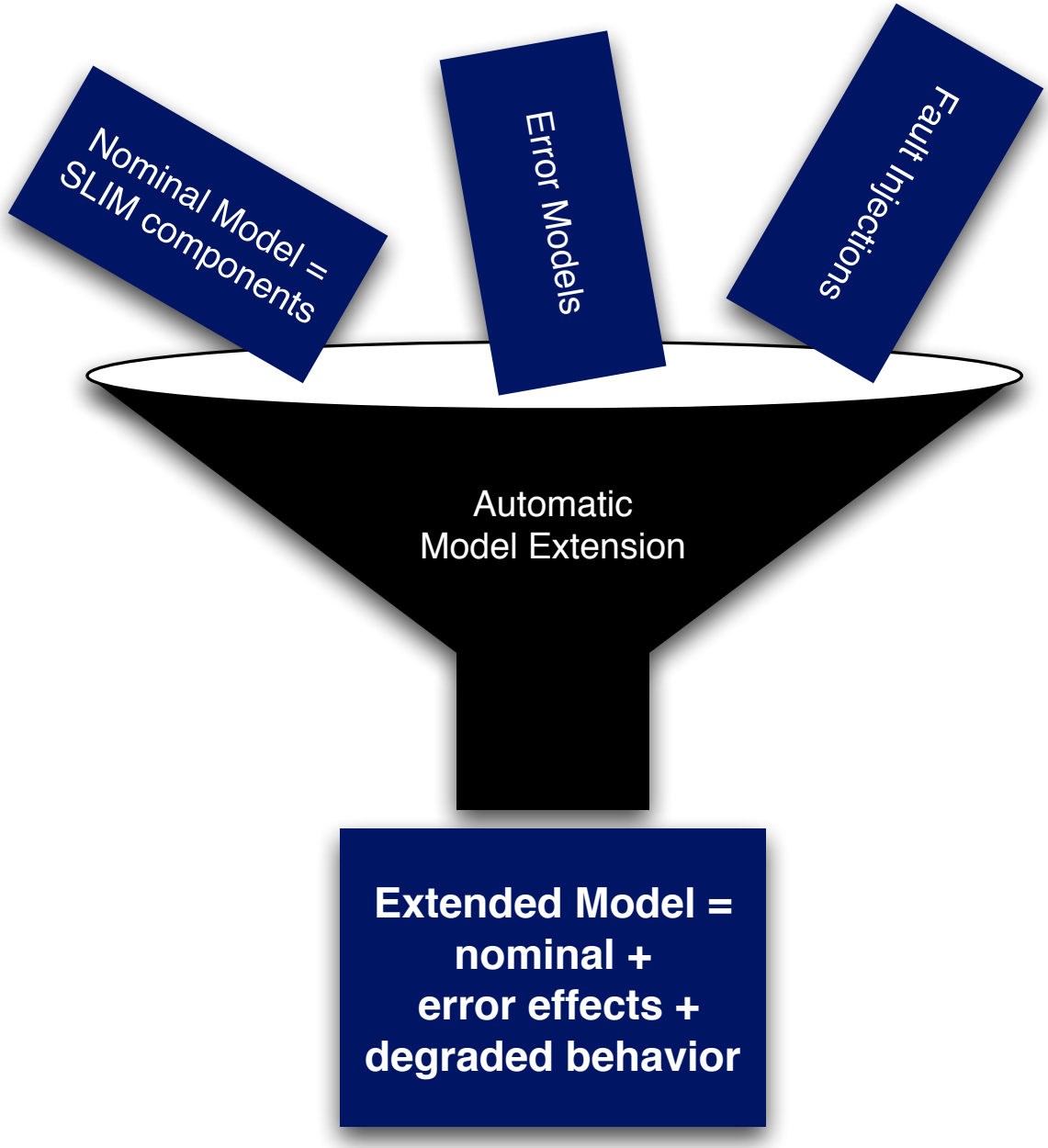
Equivalent: circularity of attribute grammars

AADL	AGs
Configuration	Derivation tree
Mode	Production
Component	Symbol
Data port	Attribute
Connection	Dependency



Outline

Integrating Erroneous and Nominal Behavior



The Battery Error Model Revisited

```
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;
```

The Battery Error Model Revisited

```
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

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

Integrate 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

Nominal 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

Add 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;
```

Outline

References

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

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
- Security aspects in AADL