educed
ENGINEERING MADE BETTER

Final Milestone

Extending PROVA to AADL

Fábio Fernandes
Telma Correia

Universidade do Minho, 03 de Julho

Silence!!!!

I kill you!!!

Outline

- AADL
  - Brief overview of AADL concepts, structure and use
- PROVA
  - Entities Model
    - Boilerplates

Outline

- AADL to PROVA
  - Our approach (with examples)

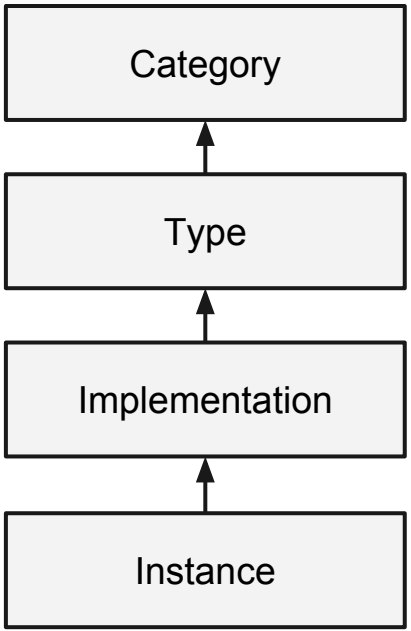
- DEMOS
  - Osate
  - aadl2prova

AADL

- Architecture Analysis and Design Language
- A language to model the interactions of software and hardware components of embedded real-time systems

AADL

In AADL we have Components that have a **Category.**



AADL

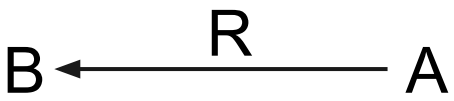
- A **Component type** contains
  - *features*
  - *flows specifications*
  - *property associations*

AADL

- A **Component Implementation** contains
  - *subcomponents*
  - *connections*
  - *flows*
  - *modes*
  - *properties*

PROVA - Entities Model

- We have
  - **Entities**
  - **Relations** between them



PROVA - Boilerplates

- We only use Boilerplates of Entity Model
  - They describe the **structural** **information**
    - The **Entities**
      - Their *attributes* and *properties*
    - **Relations** between them

Boilerplates

- We will focus in five **Boilerplates**:
  - **Mult**
    - “there are ***m rel***”
  - **Assoc**
    - “every ***A*** shall have ***m fixed? r B***”
  - **Gen**
    - “every ***r1*** is a ***r2***”

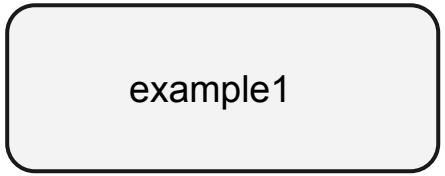
Boilerplates

- **Abstract**
  - “***r*** is abstract”
- **Extends**
  - “***r*** extends ***s***”

AADL to PROVA

AADL

system example1
end example1;



PROVA

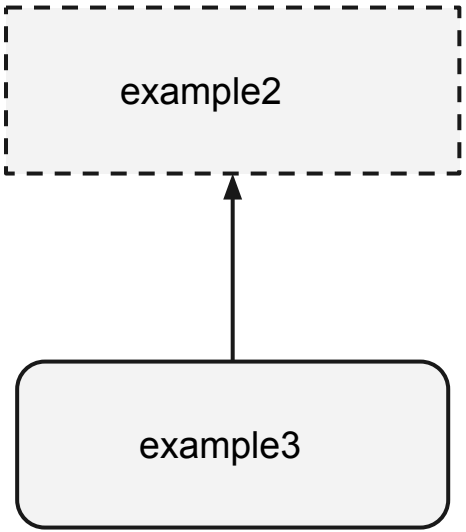
there are 1 ***SYSTEM_example1***

AADL to PROVA

AADL

```
abstract example2
end example2;

system example3 extends example2
end example3;
```

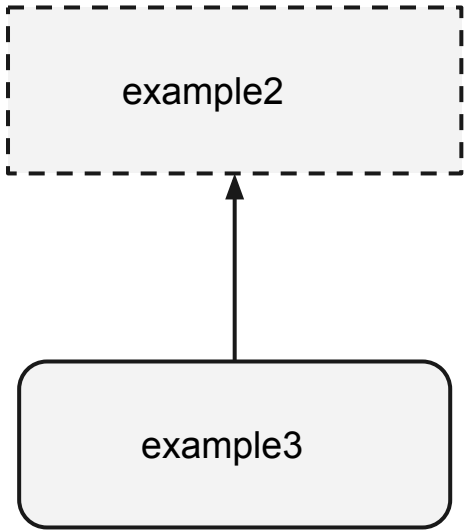


AADL to PROVA

PROVA

ABSTRACT_example2 is abstract

System_example3 extends ***ABSTRACT_example2***



AADL to PROVA

AADL

```
system example1
        features
                feat1 : feature ft_example1;
      flows
                flow1 : flow path fl_exampl1;
end example1;
```

AADL to PROVA

PROVA

there are 1 ***SYSTEM_example1***;

every **SYSTEM_exampl1** shall have exactly 1 ***FEATURES_feat1 FEATURE_ft_example1***;

every **SYSTEM_exampl1** shall have exactly 1 ***FLOWS_flow1 FLOW-PATH_fl_exampl1;***

AADL to PROVA

AADL

```
system example1
end example1;

system implementation example1.impl
        connections
                conA : port B -> C;
    end example1.impl;
```

AADL to PROVA

PROVA

SYSTEM_example1 is abstract;

there are 1 ***SYSTEM-IMPLEMENTATION_example1.impl;***

SYSTEM-IMPLEMENTATION_example1.impl extends ***SYSTEM_example1***

every **SYSTEM_example1** shall have exactly 1 ***CONNECTIONS_CA PORT_B***;

every ***PORT_B*** shall have exactly 1 ***CONNECTIONS_CA1 C***;

Now… It’s time for a DEMO!!

OSATE

And… Another Demo!!

aadl2prova

Goals Achieved

- Study and analysis of **AADL** and existing tool support
- Study and analysis **PROVA** (Boilerplates)
- Implementation of backend transformation between **PROVA** and **AADL**

Future work

- Improve our tool to support all the static parts of **AADL**
- Possible implementation of **behaviour annex** in **PROVA**

Conclusions

- AADL is a very powerful language to describe Architectures
- Like alloy, Prova allow us to predict behaviour and early identification of errors, we had some difficulties in the analyse of the boilerplate’s syntax but thanks to our external supervisor, that help us when we needed, we could understood the boilerplate’s syntax and use them in our tool

Questions??

educed
ENGINEERING MADE BETTER

Final Milestone

Extending PROVA to AADL

Fábio Fernandes
Telma Correia

Universidade do Minho, 03 de Julho