# Using AADL in Model Driven Development

**Didier Delanote, Stefan Van Baelen, Wouter Joosen and Yolande Berbers**
**Katholieke Universiteit Leuven**
**Belgium**


KULeuven
DistriNet
Research Group

KU Leuven DistriNet Research Group

## Contents

- **Introduction**
- Overview of AADL
- Usability assessment of AADL
- A general approach to improve usability of AADL using MDD
- Conclusion



## Introduction

- Software-intensive embedded systems
- Verification of functional and non-functional properties
- Non-functional properties
  - timeliness
  - performance
  - safety
- Enable verification of properties
  - modeling language
  - Model Driven Development (MDD)



## Introduction

- Modeling language
  - UML
    - SysML UML Profile
    - MARTE UML Profile
  - Architecture Description Language (ADL)
    - SAE AADL standard
- Model Driven Development
  - models as primary artifact
  - models with functional and non-functional properties

KU Leuven DistriNet Research Group

## Contents

- Introduction
- **Overview of AADL**
- Usability assessment of AADL
- A general approach to improve usability of AADL using MDD
- Conclusion



## Overview of AADL

- AADL provides **six dimensions** for specifying components
  - component analysis and design
    - analyze existing systems in terms of components
    - design new systems as black box components
    - two separate usages with specific modeling concepts
  - component categories
  - component type and implementation
  - component properties
  - component composition
  - component connection and binding



## Overview of AADL

- AADL provides **six dimensions** for specifying components
  - component analysis and design
  - component categories
    - execution platform components
    - application software components
    - composite components
  - component type and implementation
  - component properties
  - component composition
  - component connection and binding



## Overview of AADL

- AADL provides **six dimensions** for specifying components
  - component analysis and design
  - component categories
  - component type and implementation
    - component type describes interface
    - component implementation describes contents
  - component properties
  - component composition
  - component connection and binding



## Overview of AADL

- AADL provides **six dimensions** for specifying components
  - component analysis and design
  - component categories
  - component type and implementation
  - component properties
    - specific information about modeling concepts
    - functional and non-functional properties
  - component composition
  - component connection and binding



## Overview of AADL

- AADL provides **six dimensions** for specifying components
  - component analysis and design
  - component categories
  - component type and implementation
  - component properties
  - component composition
    - legality rules for component composition
    - specific to component category
    - specific to component type and implementation
  - component connection and binding



## Overview of AADL

- AADL provides **six dimensions** for specifying components
  - component analysis and design
  - component categories
  - component type and implementation
  - component properties
  - component composition
  - component connection and binding
    - connections for communication of data and events
    - binding application software and execution platform components through properties

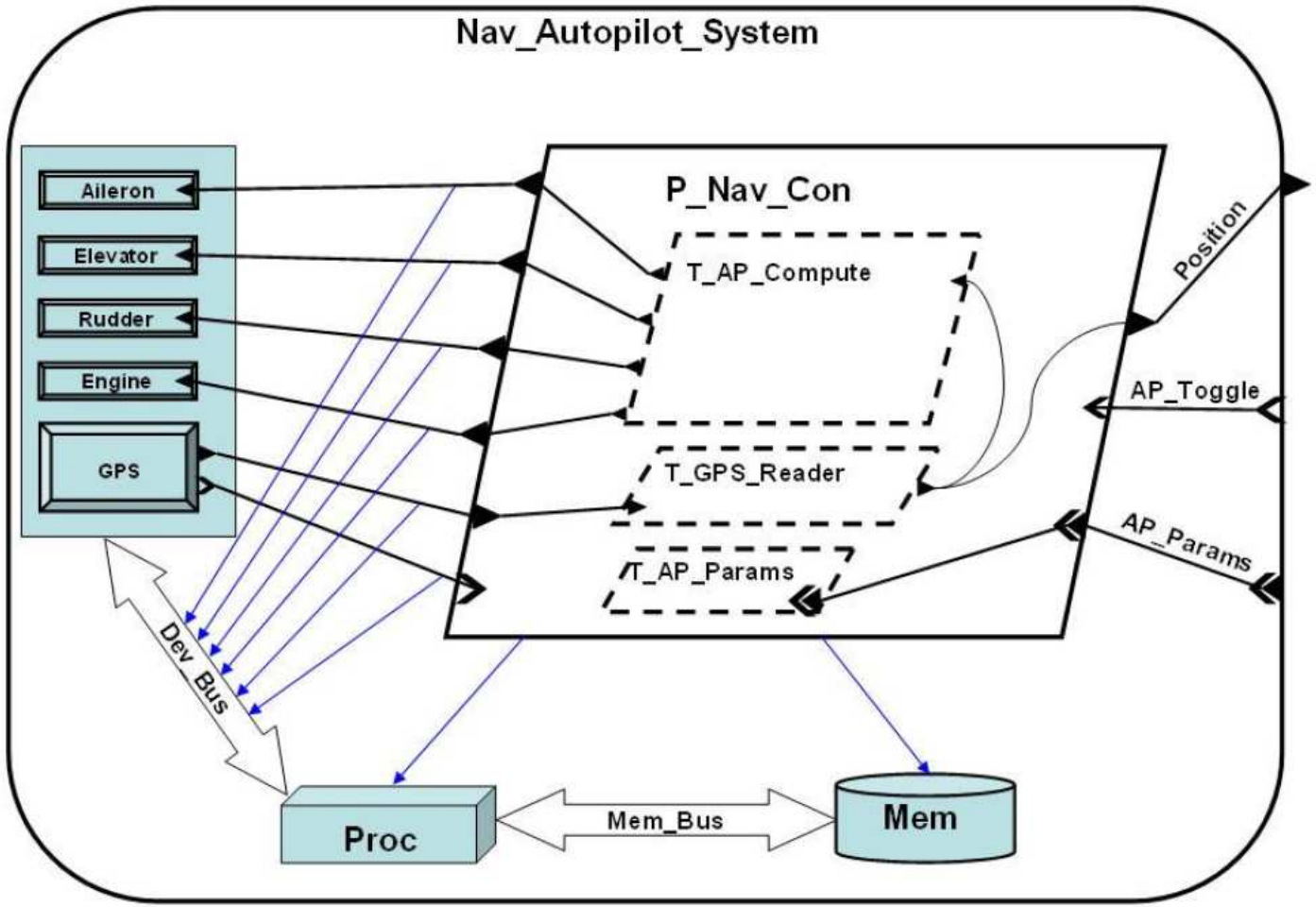
KU Leuven DistriNet Research Group

## Contents

- Introduction
- Overview of AADL
- **Usability assessment of AADL**
- A general approach to improve usability of AADL using MDD
- Conclusion

## Usability assessment of AADL

- Navigation control example





## Usability assessment of AADL

- Issues in the usability of AADL as a modeling language
  - System versus software level
  - Complex component composition
  - Property ambiguity



## Usability assessment of AADL

- System versus software level
  - component-based and object-oriented paradigm share many properties
    - distinction between type and implementation of modeling concepts
  - components are at a higher level of abstraction than classes
  - components can be both hardware and software
  - gap between AADL modeling concepts and implementation concepts



## Usability assessment of AADL

- Complex component composition
  - legality rules specific to component category

| Category | Type | Implementation |
|---|---|---|
| **`device`** | Features<br>• port<br>• port group<br>• subprogram<br>• requires bus access<br>Flow specifications: yes<br>Properties: yes | Subcomponents: none<br>Subprogram calls: no<br>Connections: no<br>Flows: yes<br>Modes: yes<br>Properties: yes |

  - component extension adds complexity to composition



## Usability assessment of AADL

- Property ambiguity
  - properties specific to component categories
  - no clearly defined relations between property sets and model analyses



## Contents

- Introduction
- Overview of AADL
- Usability assessment of AADL
- **A general approach to improve usability of AADL using MDD**
- Conclusion



## A general approach to improve usability of AADL using MDD

- AADL and runtime environment considered a platform
- Considerable amount of platform-specific (semantic) knowledge
- MDD approach to reduce requirement of this knowledge

KU Leuven DistriNet Research Group

## A general approach to improve usability of AADL using MDD

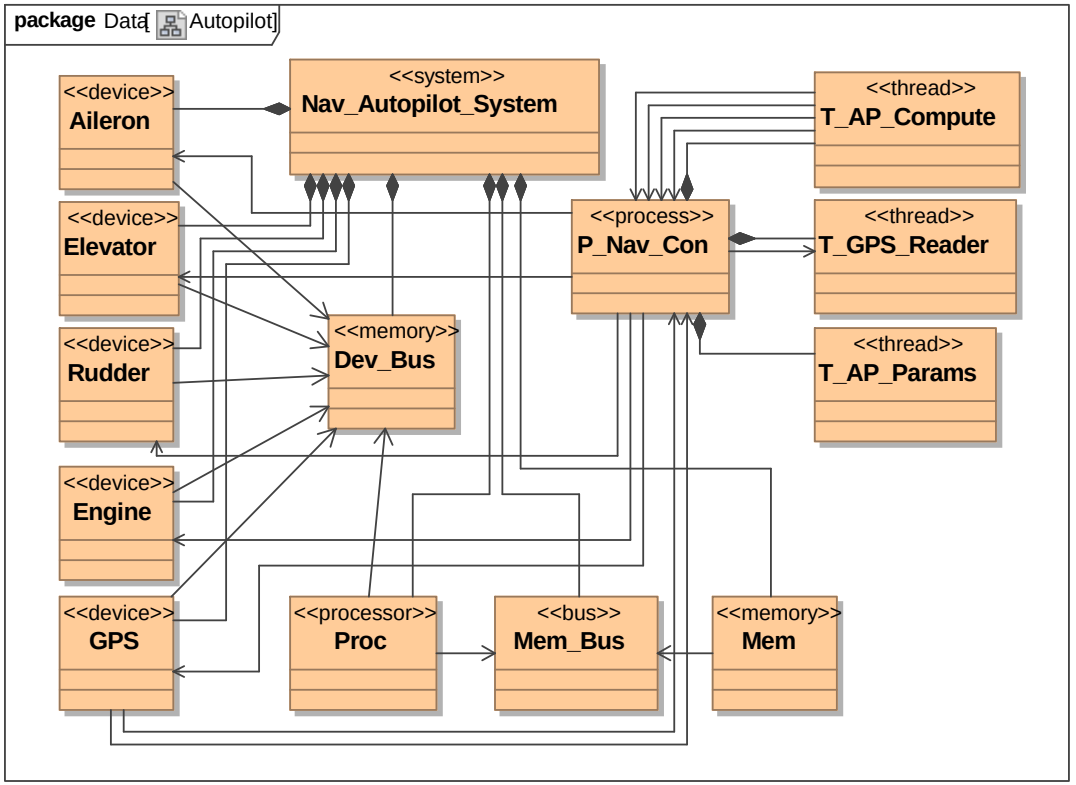
- Models
  - AADL PIM
    - UML model of application
    - annotated with component category stereotypes
    - does not necessarily comply to AADL legality rules
  - AADL PSM
    - AADL Ecore model of application
    - components of specific category
    - does comply to AADL legality rules
  - AADL Analysis Model
    - AADL Ecore model of application
    - AADL PSM annotated with properties
    - properties have default value


## A general approach to improve usability of AADL using MDD

- Steps
  - Obtaining an AADL PIM
  - Model transformations
  - Execute analyses in AADL runtime environment

# A general approach to improve usability of AADL using MDD

- Obtaining an AADL PIM
  - design application from scratch, or
  - reverse engineer application and annotate with stereotypes



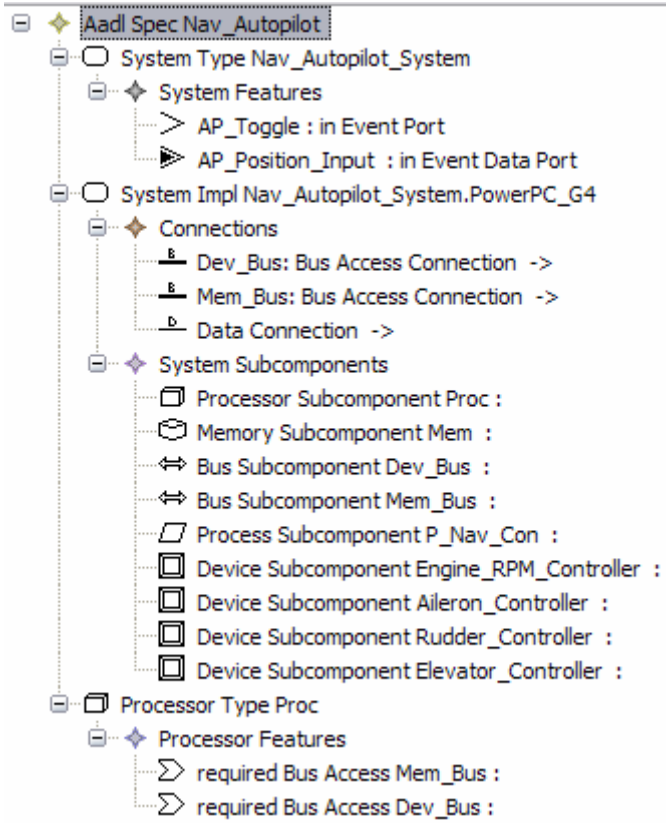


## A general approach to improve usability of AADL using MDD

- Model transformations
  - Functional transformations
    - Encapsulate platform specific knowledge of SAE AADL legality rules
    - Transform UML to DSL
    - Transformation rules
      - Transform UML class with <<component category>> stereotype to component type of same category
      - Provide component implementation if applicable
      - Transform UML composition links between UML classes into subcomponents
      - Transform directed associations between UML classes into *in*, *out* or *in out* port features and provide connections
      - Transform dependency relations between UML classes into bus access features

# A general approach to improve usability of AADL using MDD

- AADL PSM



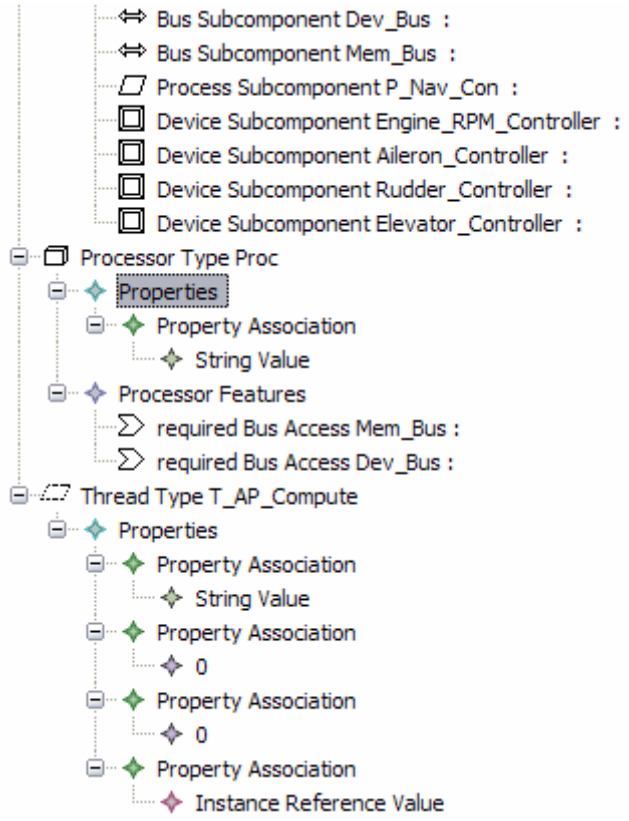


## A general approach to improve usability of AADL using MDD

- Model transformations
  - Non-functional transformations
    - Add set of properties allowing **analysis** of AADL PSM
    - Towards **exactly one** non-functional property of system
    - Platform specific knowledge concerning relation between AADL properties and analysis of AADL model built into non-functional transformations
    - Subsequent non-functional transformations possible on same model
    - Transformation rules
      - Add specific set of properties to each component category
      - Provide properties with default value
    - Example of schedulability analysis

# A general approach to improve usability of AADL using MDD

- AADL Analysis Model



KU Leuven
DistriNet
Research Group

## Contents

- Introduction
- Overview of AADL
- Usability assessment of AADL
- A general approach to improve usability of AADL using MDD
- **Conclusion**



## Conclusion

- Number of issues in usability of AADL
  - System versus software level
  - Complex component composition
  - Property ambiguity
- MDD process to ease these issues
    - Obtaining AADL PIM
  - AADL PIM
    - Functional transformations
  - AADL PSM
    - Non-functional transformations
  - AADL Analysis Model
    - Analysis tool



## Conclusion

- Questions?
- Discussion