

Integrating AADL within a multi-domain modeling framework

Ivano Malavolta, Henry Muccini, Patrizio Pelliccione

Computer Science Department

University of L'Aquila - Italy

Roadmap

- » Background
- » Main goal
- » DUALLY
- » Integrating AADL within DUALLY
- » Illustrative example
- » Conclusions



Background

- There exist today many languages for specifying software architectures: formal ADLs, UML profiles, other proprietary notations.
 - Increasingly, ADLs are defined by stakeholder concerns
 - different degrees of **formality**,
 - at different levels of **abstraction**,
 - **domain** specific (vehicular, electronics...) or generic
 - different **analysis**
 - Researchers acknowledge that a unique universal language **cannot** exist
- proliferation of architectural languages and UML-based approaches

Main goal

- » New real-time systems have increasingly complex architectures → many different aspects must be considered at design time.

AADL covers many of them, but not all (e.g. reachability analysis).

- » A possible solution is to complement AADL-specific capabilities with other notations' capabilities.

To achieve this goal, we experiment the integration of AADL (along with its behavioral annex) within **DUALy** (a multi-domain modeling framework), allowing the translation of AADL models into other notations.

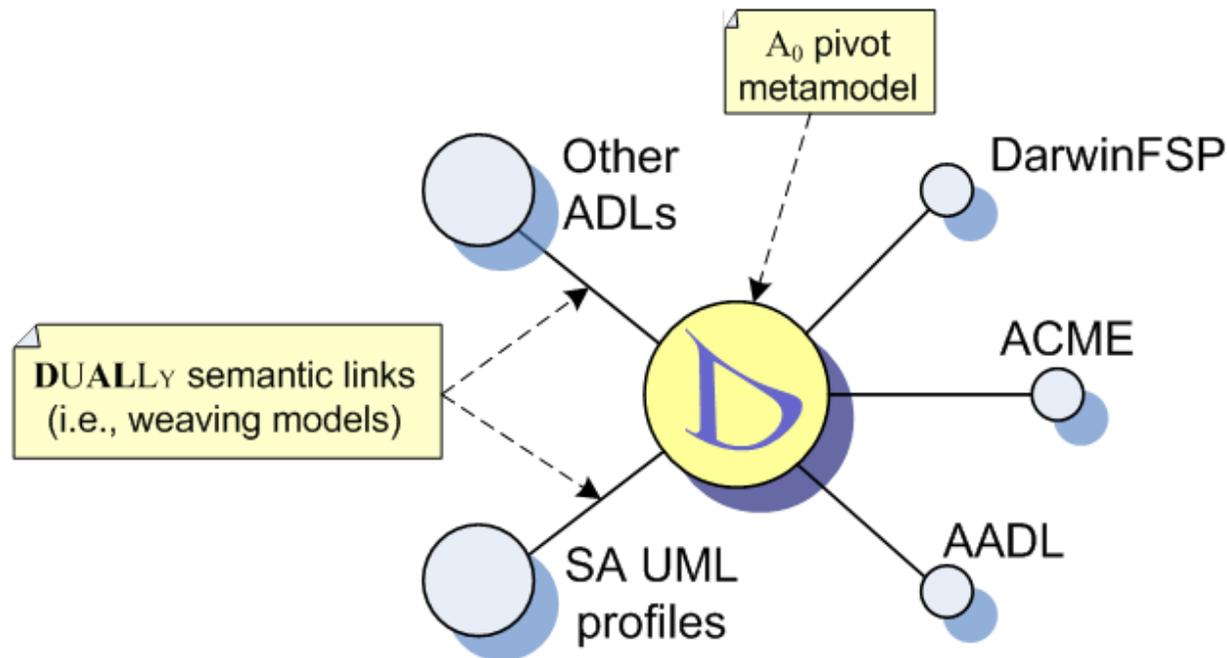
Roadmap

- » Background
- » Main goal
- » **DUALLY**
- » Integrating AADL within DUALLY
- » Illustrative example
- » Conclusions



What is DUALLy

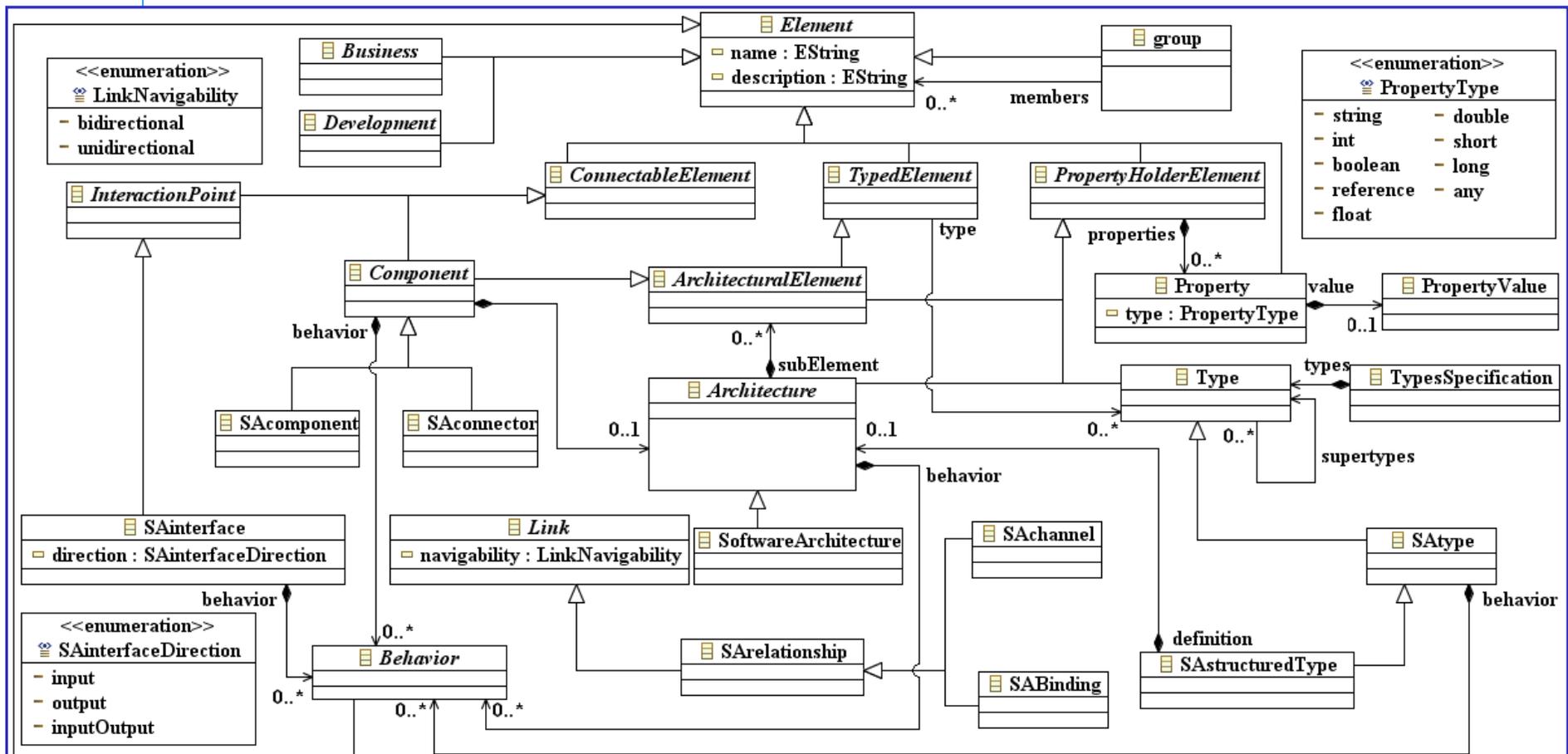
An automated framework that allows architectural languages and tools **interoperability**, including both ADLs and UML-based notations.



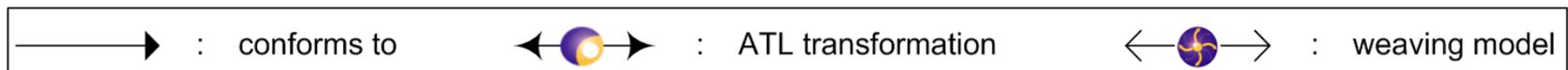
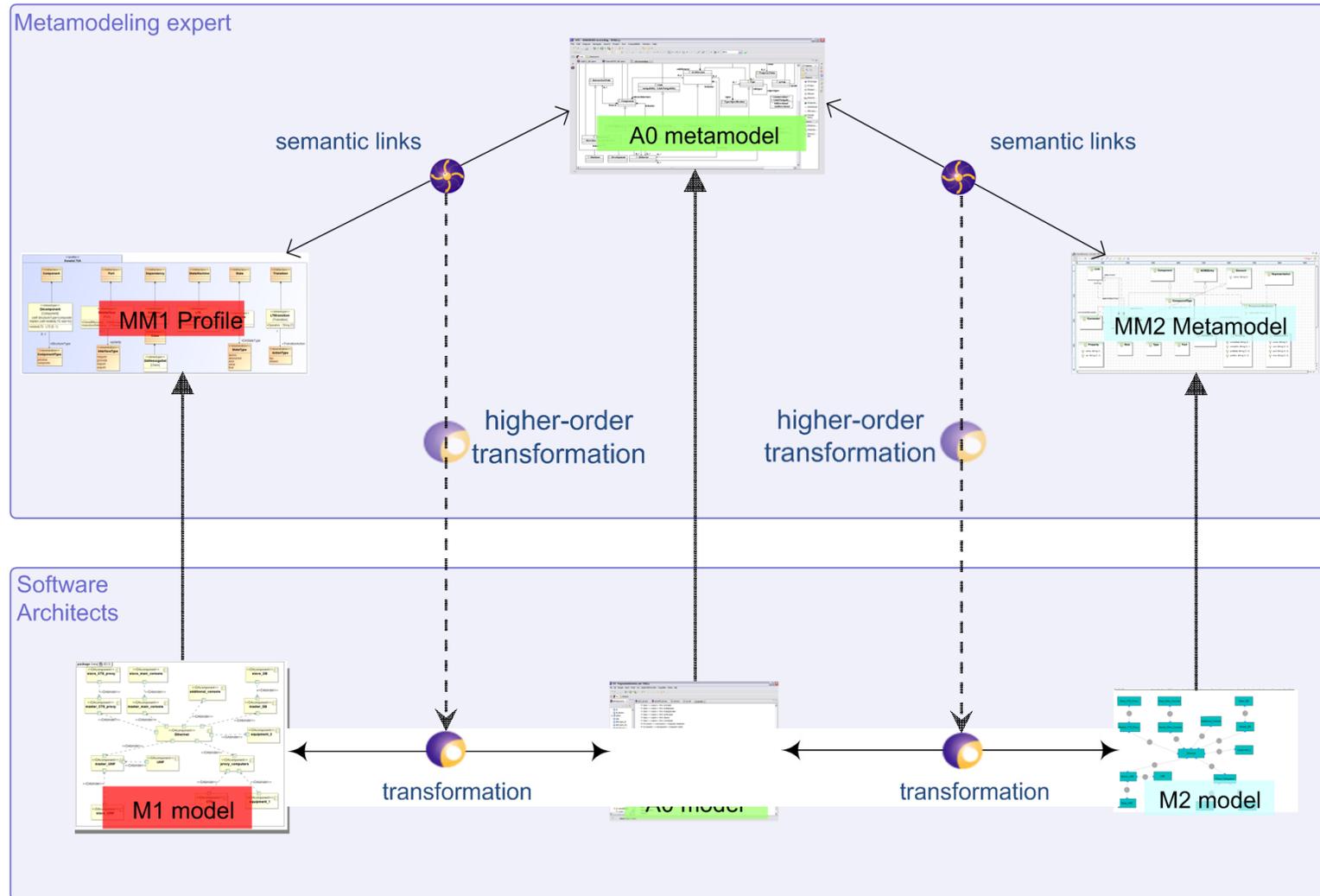
Languages interoperate thanks to automated model transformation techniques.

A₀ : the pivot metamodel

» A₀ is specific to the software architectures domain



The DUALy framework



Further informations on DUALLy

Official home page: <http://dually.di.univaq.it>

Technical details:

Ivano Malavolta, Henry Muccini, Patrizio Pelliccione, and Damien Tamburri "Providing Architectural Languages and Tools Interoperability through Model Transformation Technologies". IEEE Transactions on Software Engineering (TSE). IEEE computer Society. April 2009.



Roadmap

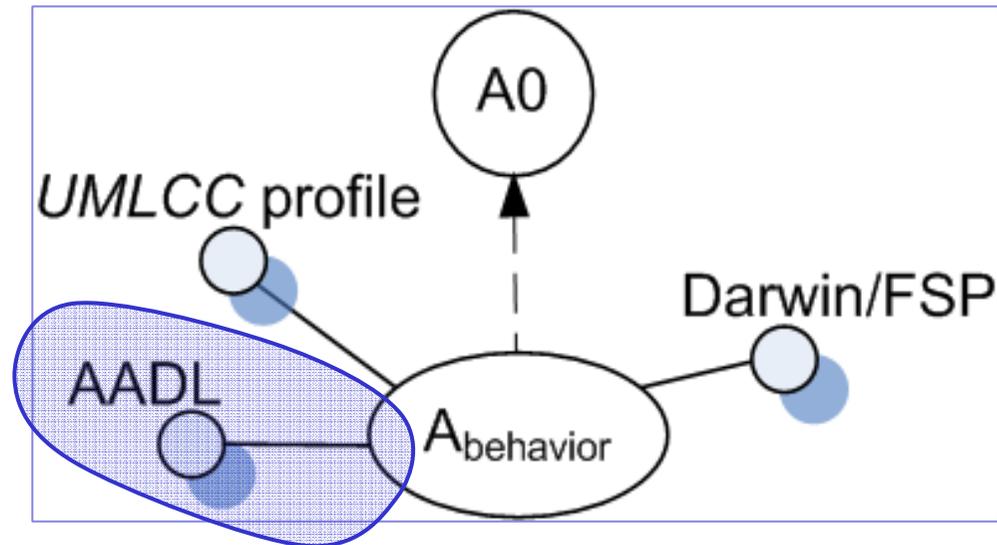
- » Background
- » Main goal
- » DUALLy
- » **Integrating AADL within DUALLy**
- » Illustrative example
- » Conclusions



Integrating AADL within DUALLy

Initial step: extend $A_0 \rightarrow A_{\text{behavior}}$

$A_{\text{behavior}} = A_0 + \text{minimal metamodel for state machines}$

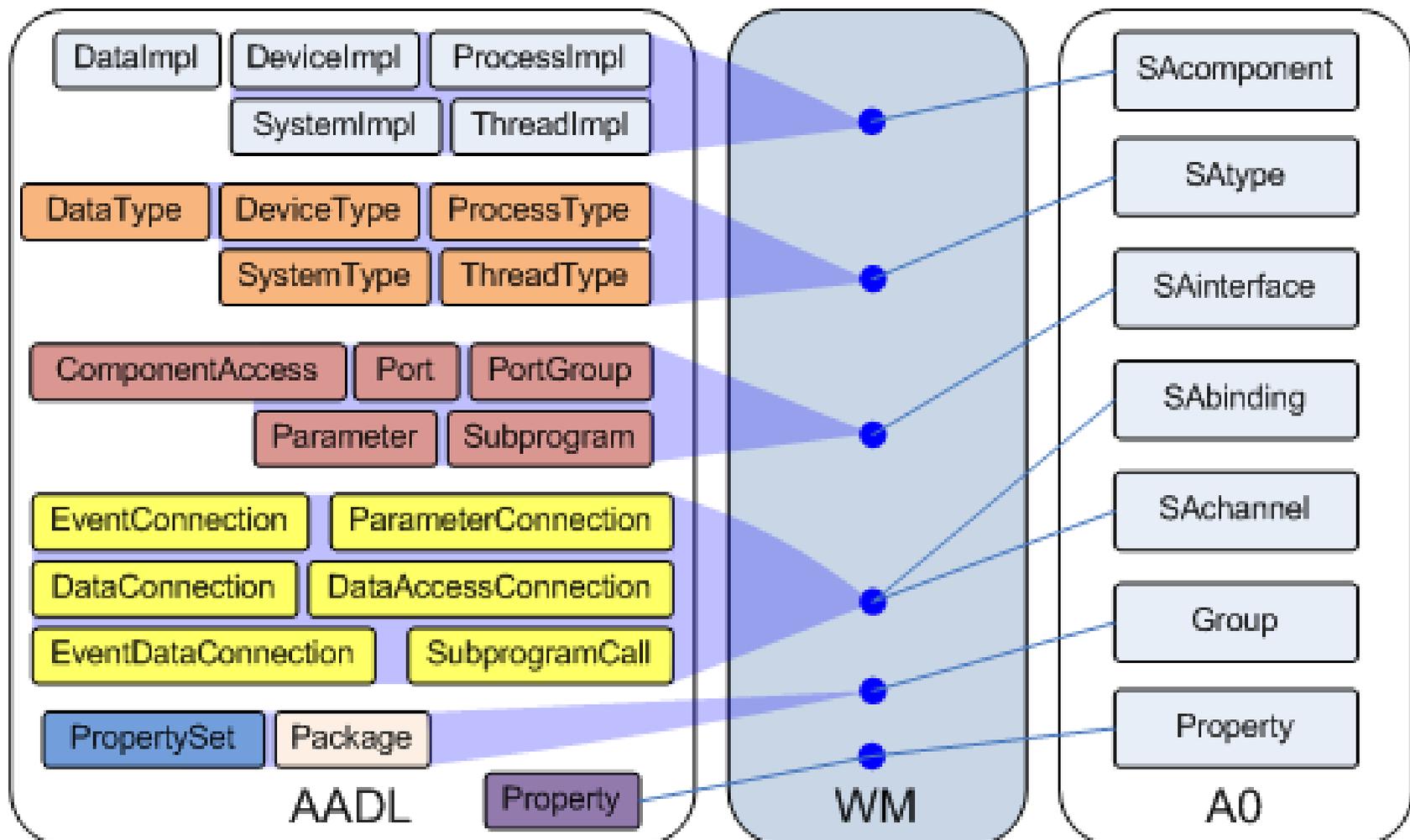


The DUALLy-zation of AADL is composed of two main steps:

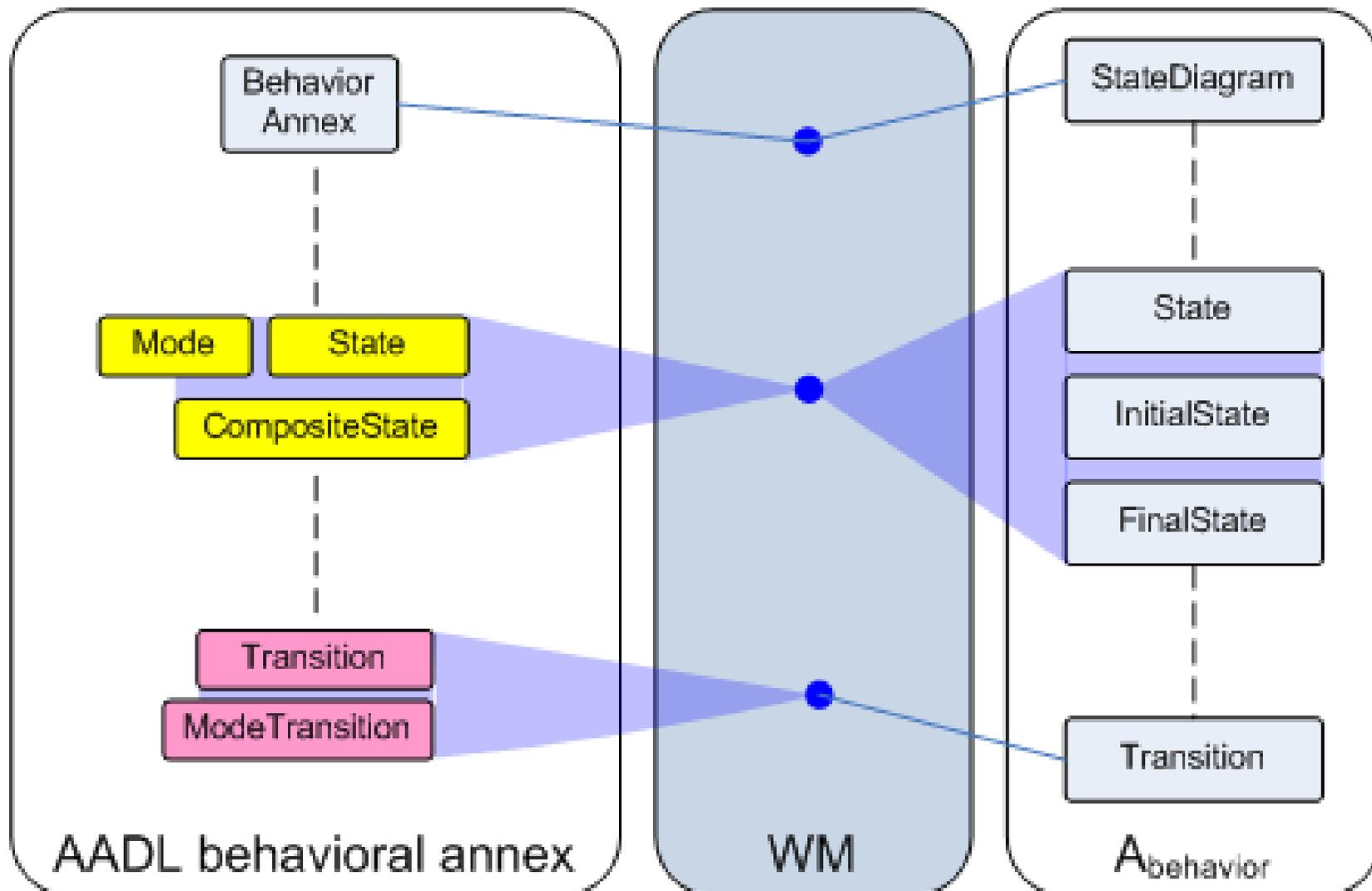
- 1) Specify the weaving model between AADL and A_0
- 2) Generate model-to-model transformations (e.g. $Aadl2A_0$)



Mapping core AADL concepts



Mapping behavioral AADL concepts

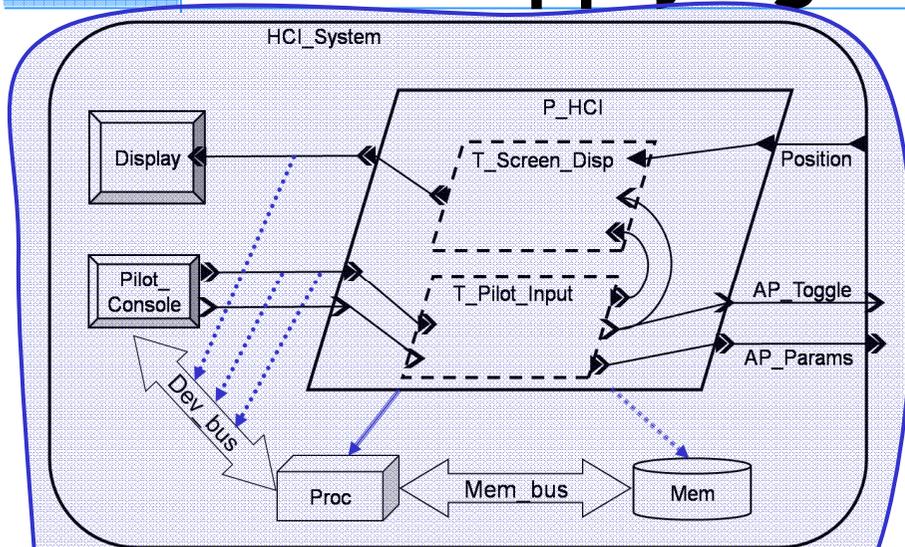


Roadmap

- » Background
- » Main goal
- » DUALLY
- » Integrating AADL within DUALLY
- » **Illustrative example**
- » Conclusions



Applying the



thread implementation T_Pilot_Input.PowerPC_G4

properties

Dispatch_Protocol => Background;

annex behavior specification {**

state variables

aux : Nav_Types::Position.GPS;

states

s_Off : initial complete state;

s_On : complete state;

s1, s2, s3, s4, s5 : state;

transitions

s_Off -[AP_Position_In?]-> s2;

s2 -[]-> s_Off {AP_Position_Out_Dispatch!};

s_Off -[AP_Toggle_In?]-> s1;

s1 -[]-> s_On {AP_Toggle_Out!};

s_On -[AP_Toggle_In?]-> s3;

s3 -[]-> s_Off {AP_Toggle_Out!};

s_On -[AP_Position_In?(aux)]-> s4;

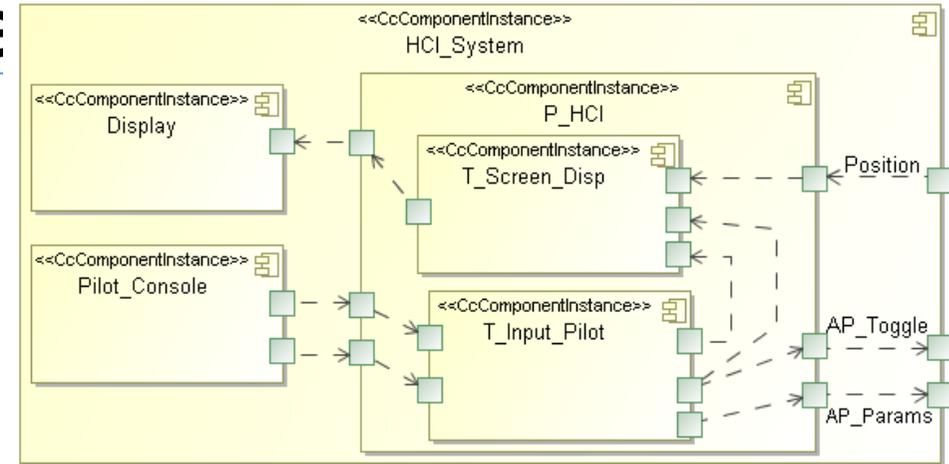
s4 -[]-> s5 {AP_Position_Out_Dispatch!(aux)};

s5 -[]-> s_On {AP_Position_Out_Nav!(aux)};

Sf**);

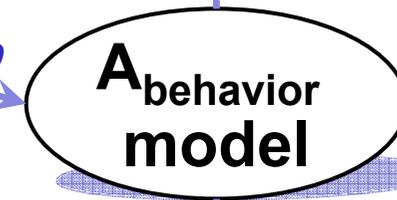
end Pilot_Input_Thread.PowerPC_G4;

13

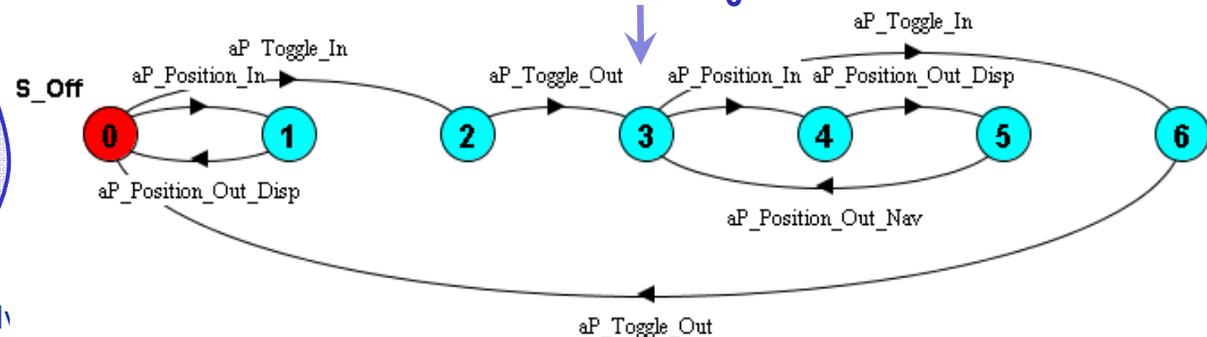


Aadl2A0

A02umlCC



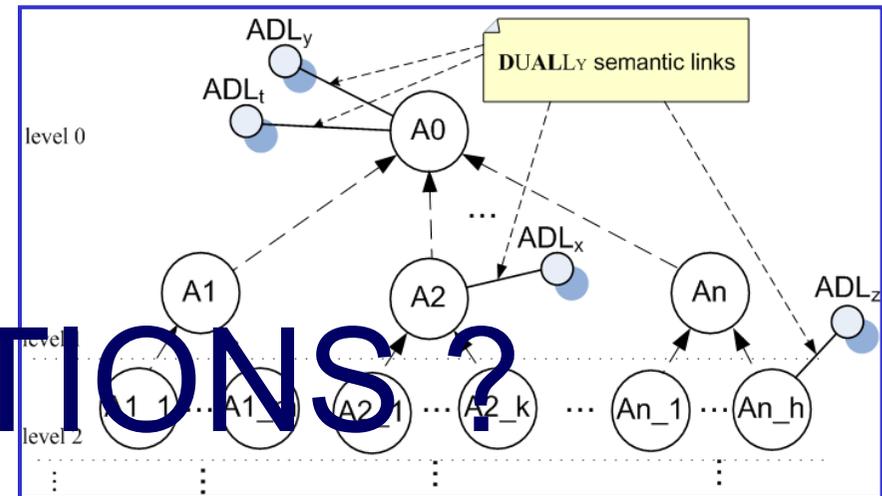
A02darwinFSP



Conclusions and future work

We analyzed the feasibility of integrating AADL in DUALLY

A_0 has been designed as general as possible, but it is important to customize it for each specific domain



- **Next step:** formalize the extension mechanism through well defined, property-preserving extension operators

Open issues and future work:

- Automatic generation of weaving models
- Round-trip engineering of modifications of target models
- Techniques to check the quality of the transformations

