

# Facilitating AADL Model Processing and Analysis with OSATE-DIM

Rakshit Mittal <sup>1,2</sup> and Dominique Blouin <sup>2</sup>

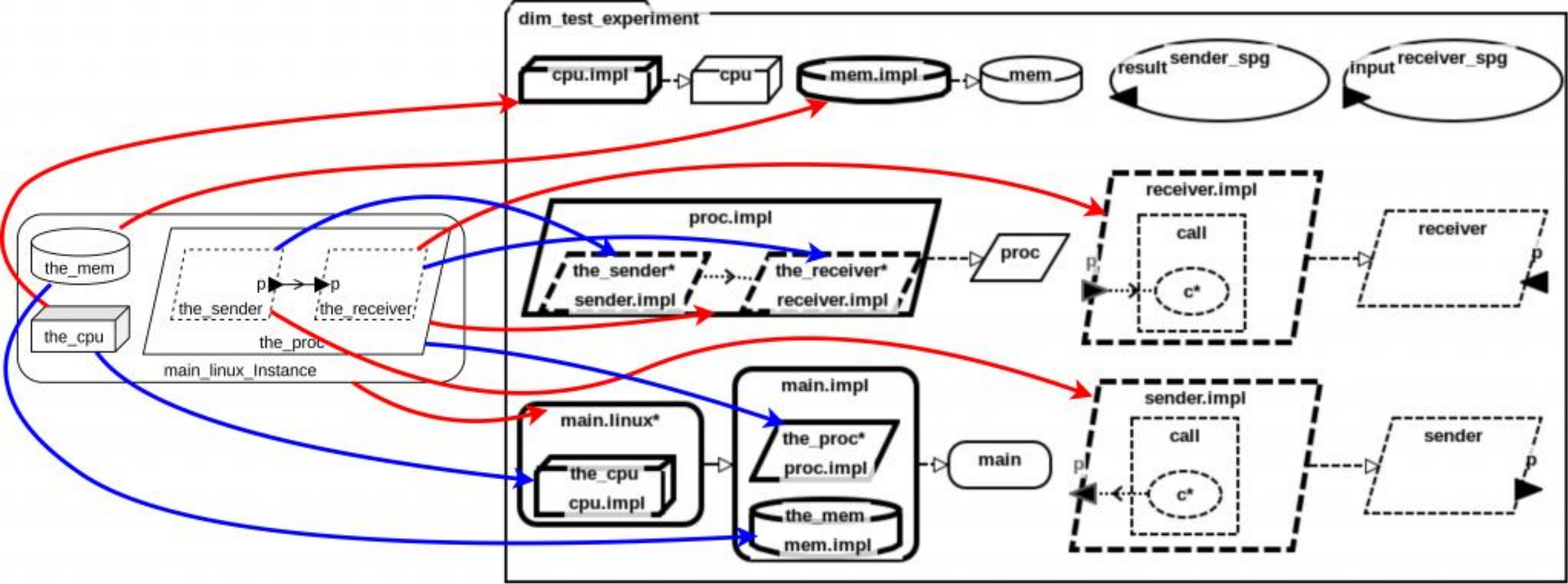
<sup>1</sup> Universiteit Antwerpen, Antwerp, Belgium

<sup>2</sup> Telecom Paris, Palaiseau, France



# OSATE Instance Model

blue: subcomponent reference  
 red: classifier reference

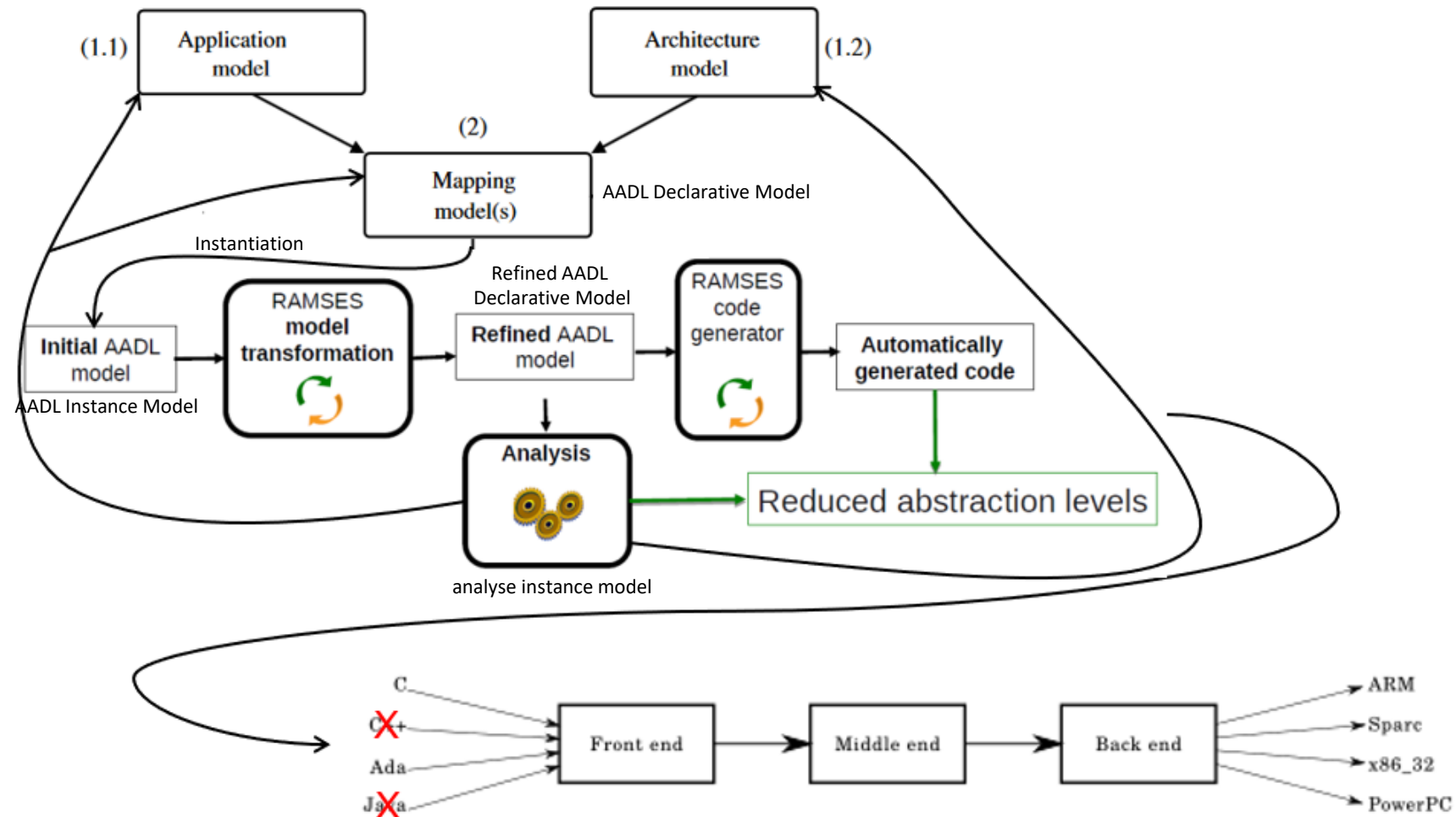


Instance Model

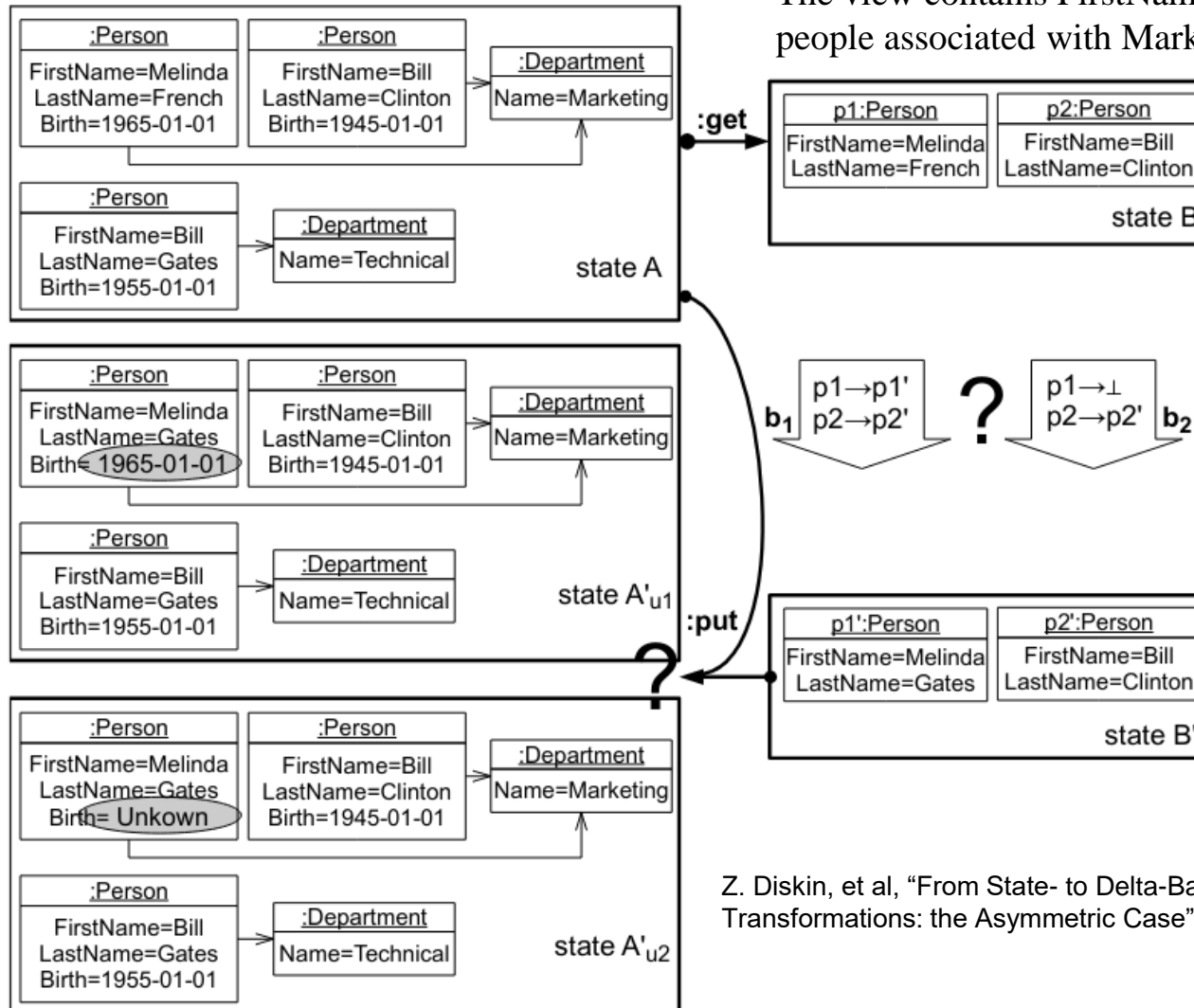
Declarative Model

# RAMSES Workflow

(Refinement of AADL Models for the Synthesis of Embedded Systems)



# View-Update Problem



The view contains FirstName and LastName only, of all people associated with Marketing department

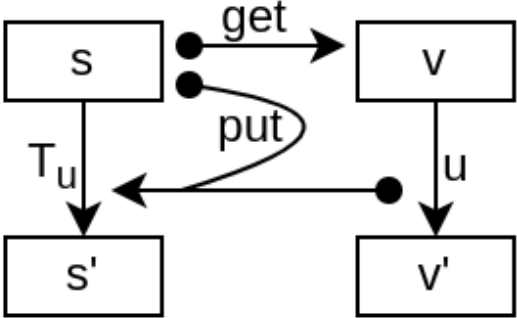
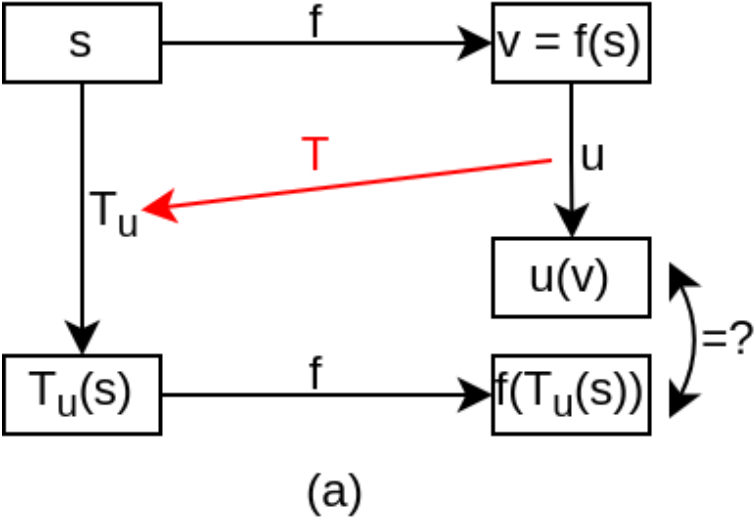
- Q. Change name of Melinda French to Melinda Gates
- b1 > Change the LastName property of p1
- b2 > Delete p1 and add p1' with FirstName, LastName

b2 causes loss of Birth data of Melinda.

Even though the state B' is the same for b1 and b2. How to capture such a difference in the base model?

Z. Diskin, et al, "From State- to Delta-Based Bidirectional Model Transformations: the Asymmetric Case", Journal of Object Technology, 2011

# Solution: Delta-based Lens



(GetPut)  $s = \text{put}(\Phi, s)(s)$   
 (PutGet)  $\text{get}(\text{put}(u, s)(s)) = v'$   
 (PutPut)  $\text{put}(u', \text{put}(u, s)(s)) = \text{put}(u \circ u', s)$

$s$  : Model State  
 $f$  : View-Generating Function  
 $v$  : View-State  
 $u$  : View-Update  
 $T$  : Translation

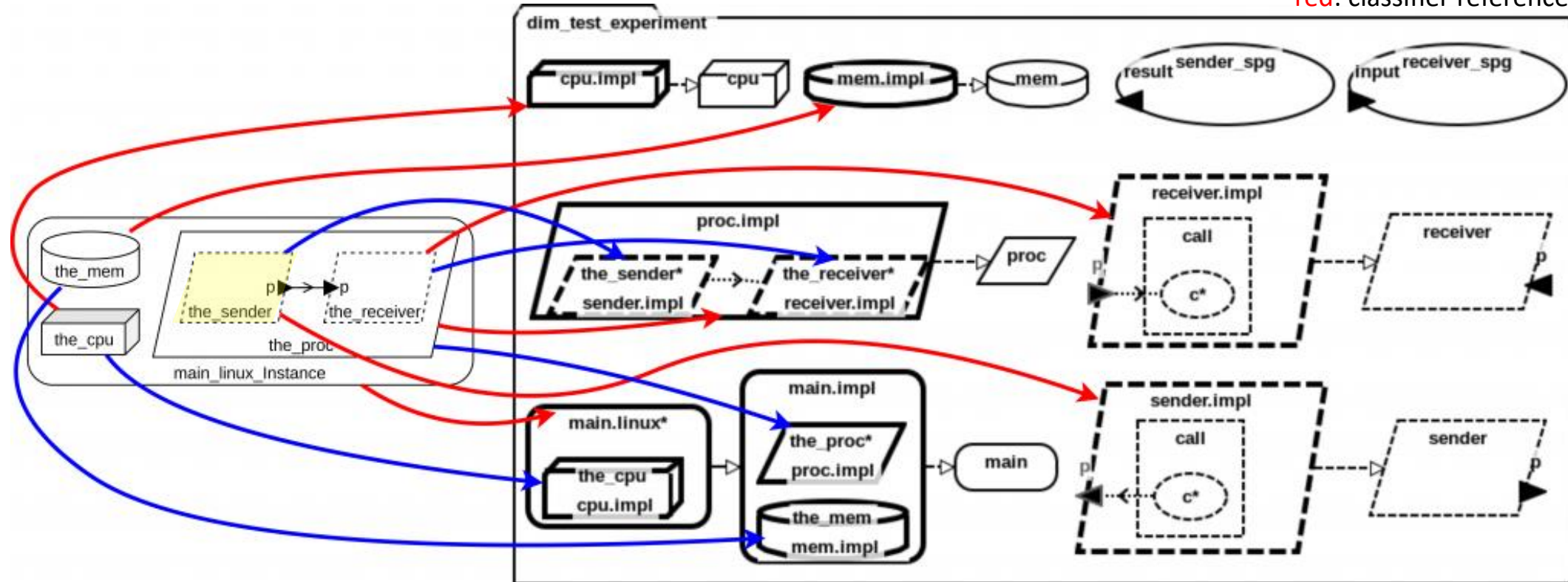
$\Rightarrow$  Declarative Model  
 $\Rightarrow$  Instantiation  
 $\Rightarrow$  Instance Model  
 $\Rightarrow$  Updates of Instance Model  
 $\Rightarrow$  Deinstantiation

$f$  is not a bijection!

# Complications in AADL View-Updates

blue: subcomponent reference

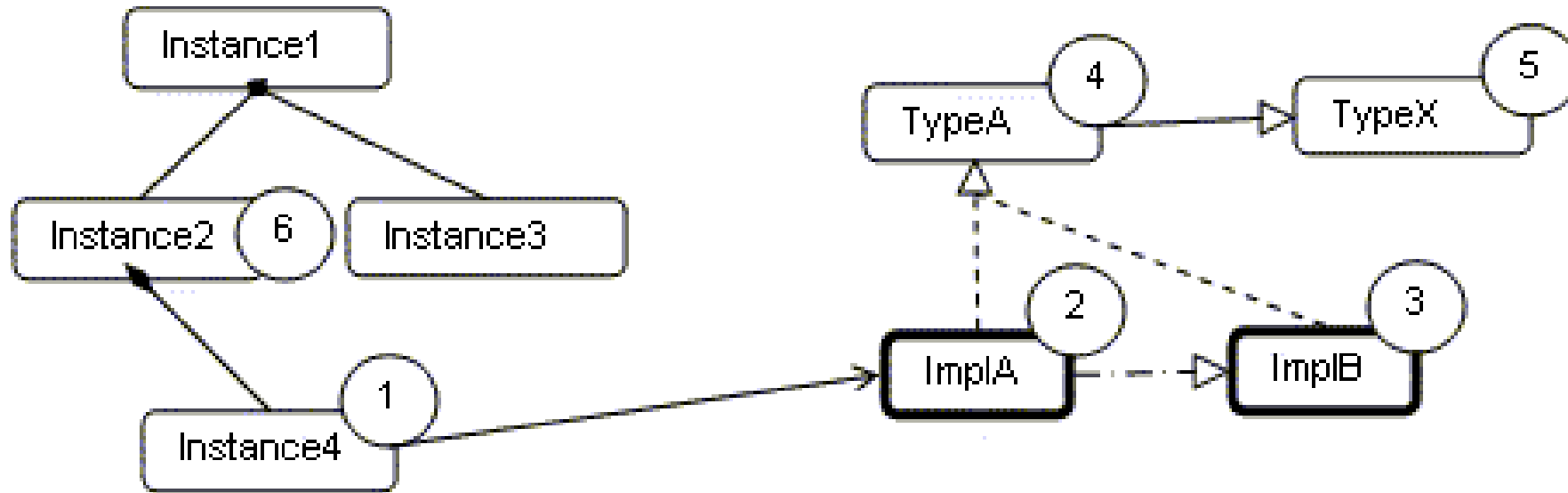
red: classifier reference



Q. Add a property to *the\_sender*

# Complications in AADL View-Updates

## AADL Property Value Determination

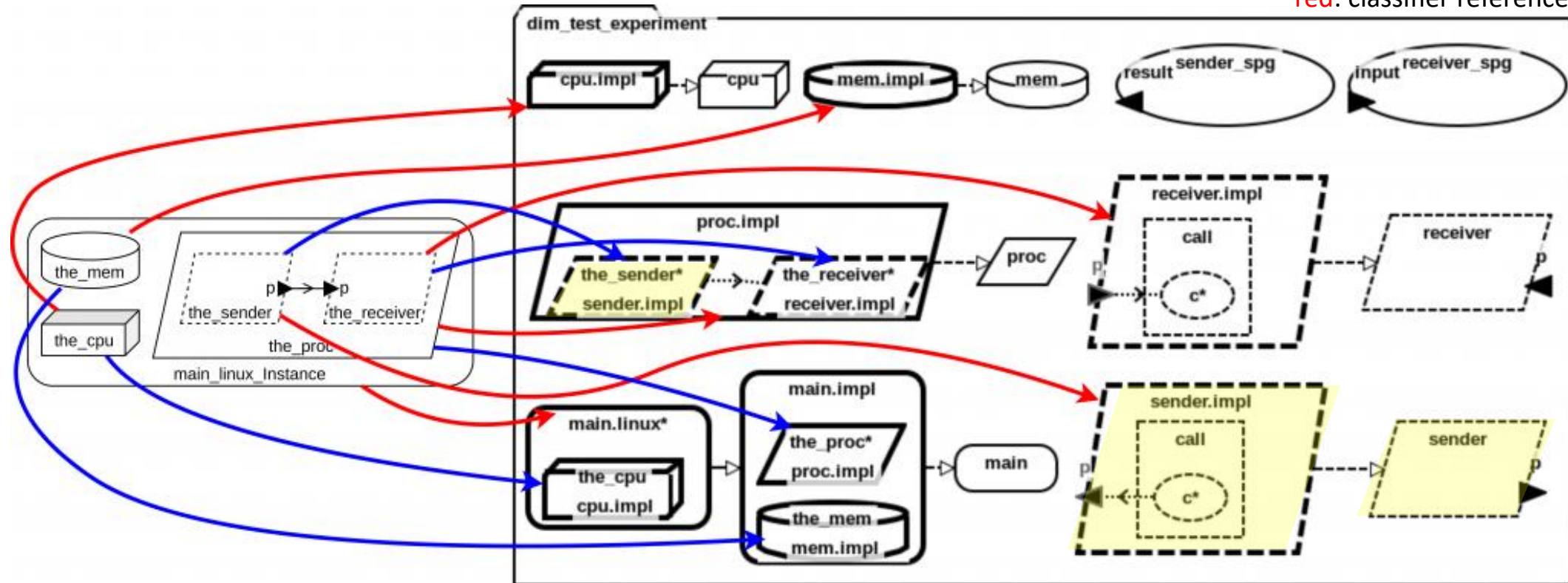


\*from OSATE2 Help Contents > AADL2.0 reference manual

# Complications in AADL View-Updates

blue: subcomponent reference

red: classifier reference



Q. Add a property to *the\_sender*

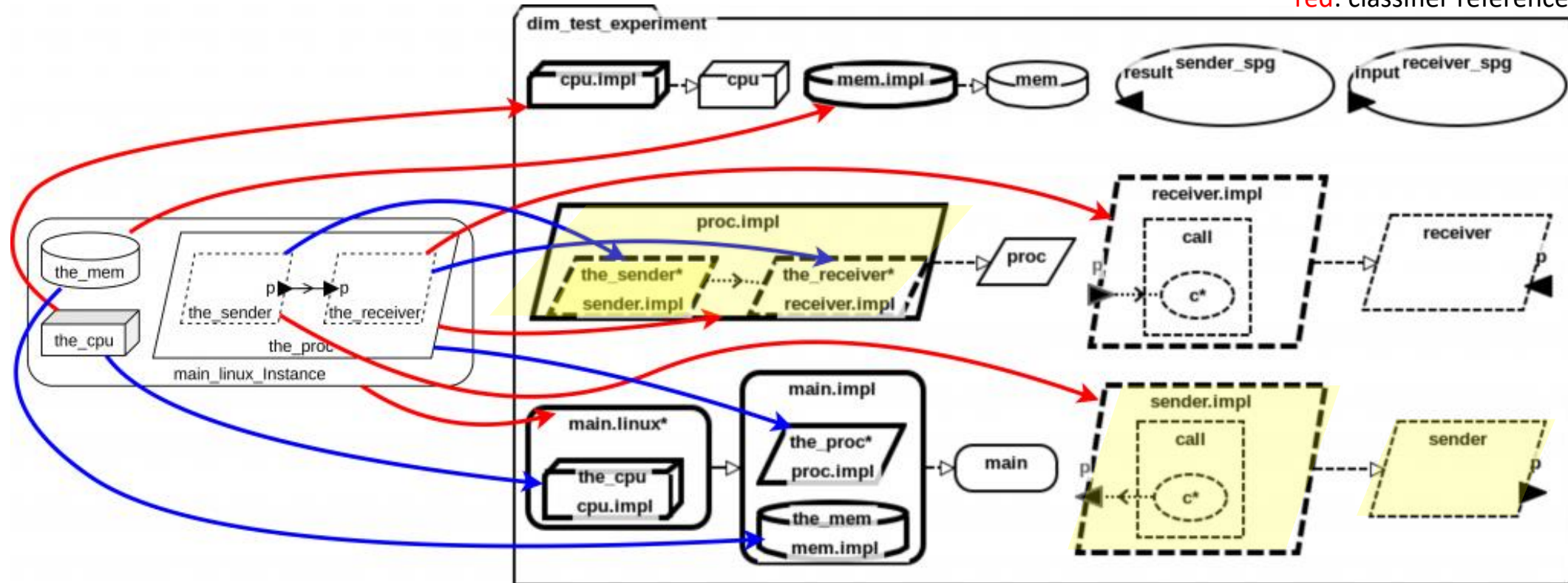
Soln. Add property to *the\_sender\** or *sender.impl* or *sender* ??



# Complications in AADL View-Updates

blue: subcomponent reference

red: classifier reference



Q. Add a property to *the\_sender*

Soln. Add property to *the\_sender\** or *sender.impl* or *sender*

or to *proc.impl* (applies to *the\_sender\**) ??

Too many choices and decisions for de-instantiation in AADL-OSATE!

# Another Example: Need for automated de-instantiation

```
package demo_models
public
  system main
  end main;

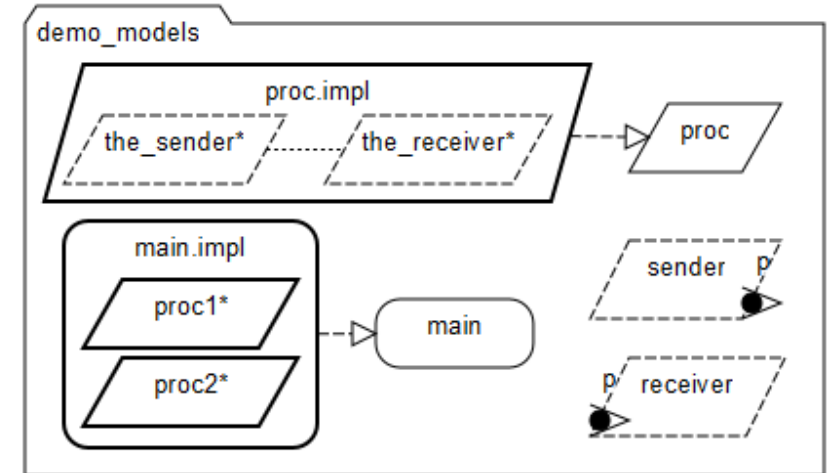
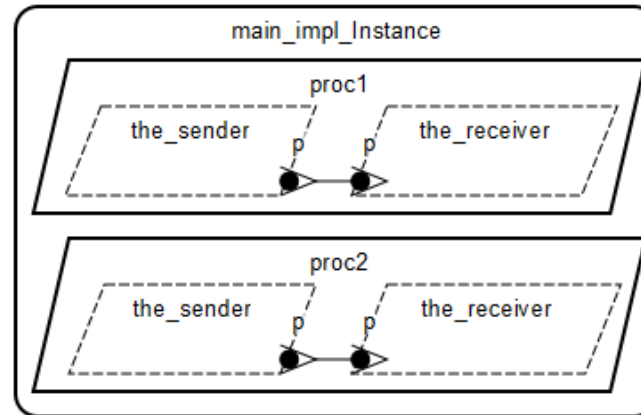
  system implementation main.impl
  subcomponents
    proc1: process proc.impl;
    proc2: process proc.impl;
  end main.impl;

  process proc
  end proc;

  process implementation proc.impl
  subcomponents
    the_sender: thread sender;
    the_receiver: thread receiver;
  connections
    cnx: feature the_sender.p -> the_receiver.p;
  end proc.impl;

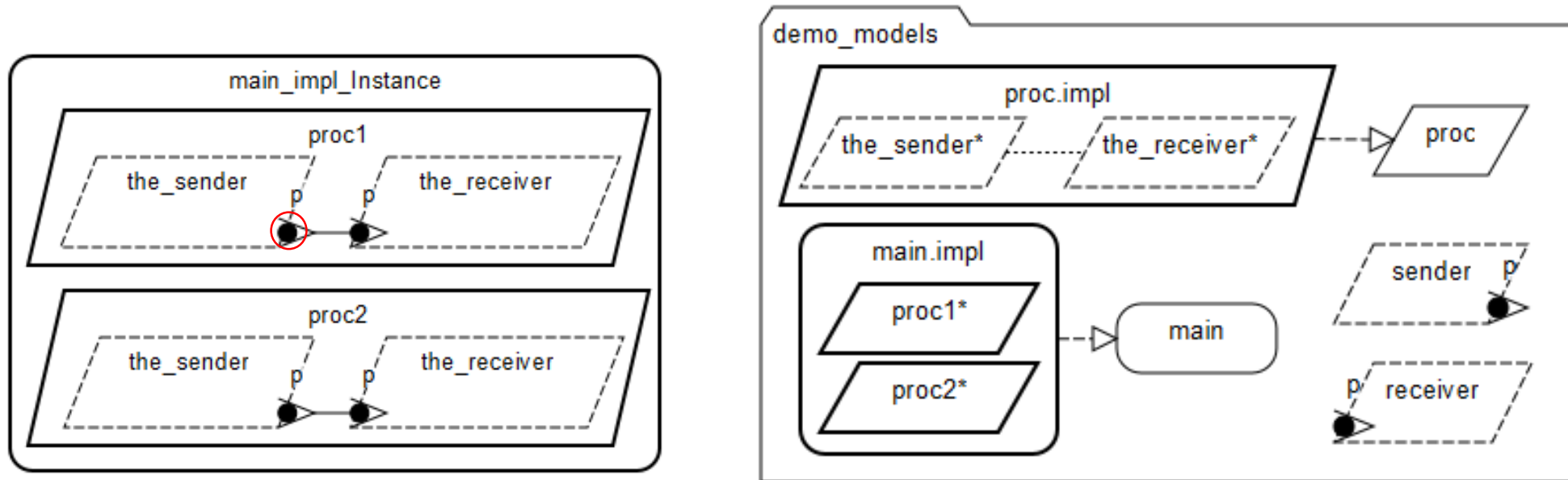
  thread sender
  features
    p: out feature;
  end sender;

  thread receiver
  features
    p: in feature;
  end receiver;
end demo_models;
```



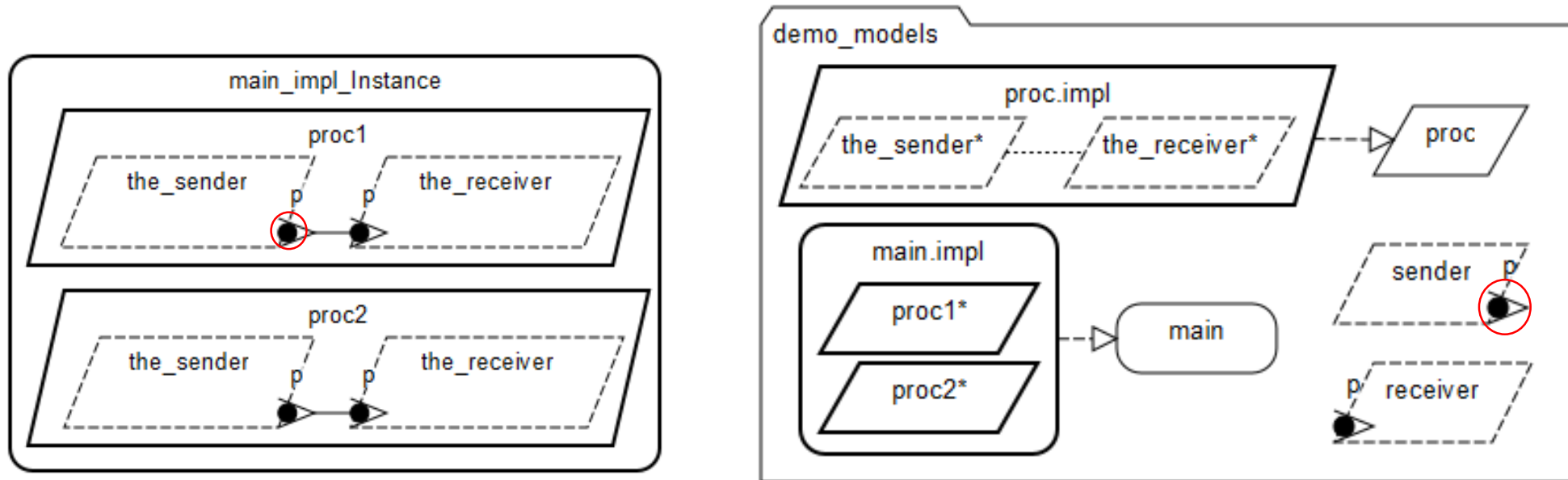
*proc1* and *proc2* have the same classifier *proc.impl*

# Another Example: Need for automated de-instantiation



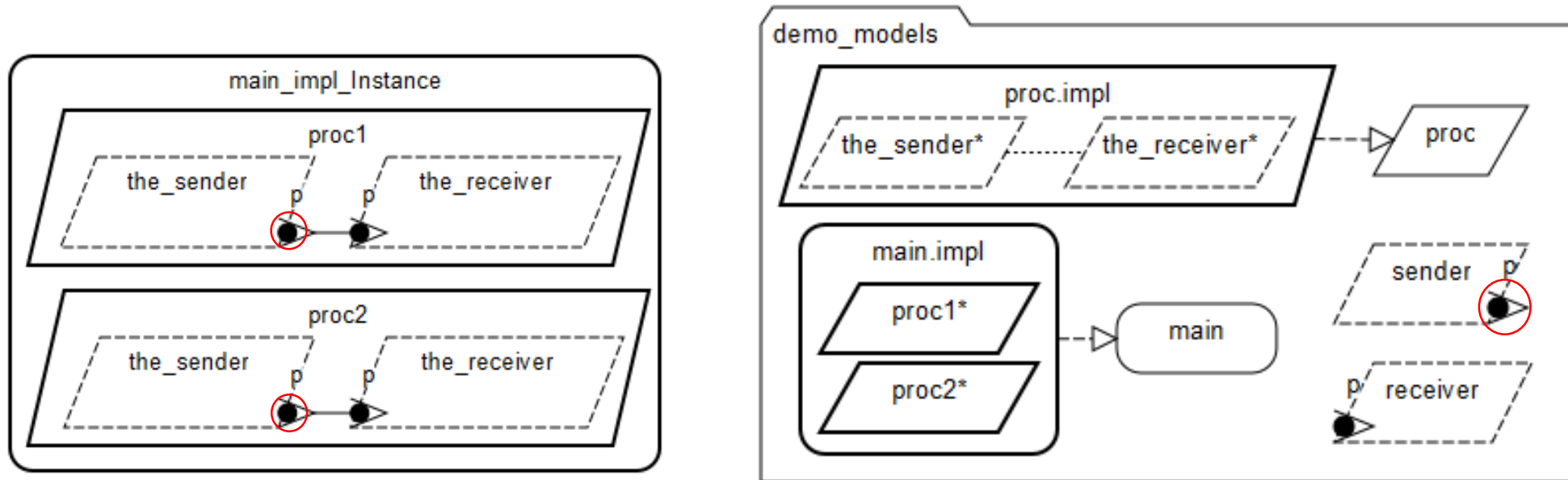
We want to change the abstract feature  $p$  in `proc1.the_sender` to data port

# Example: Need for automated de-instantiation



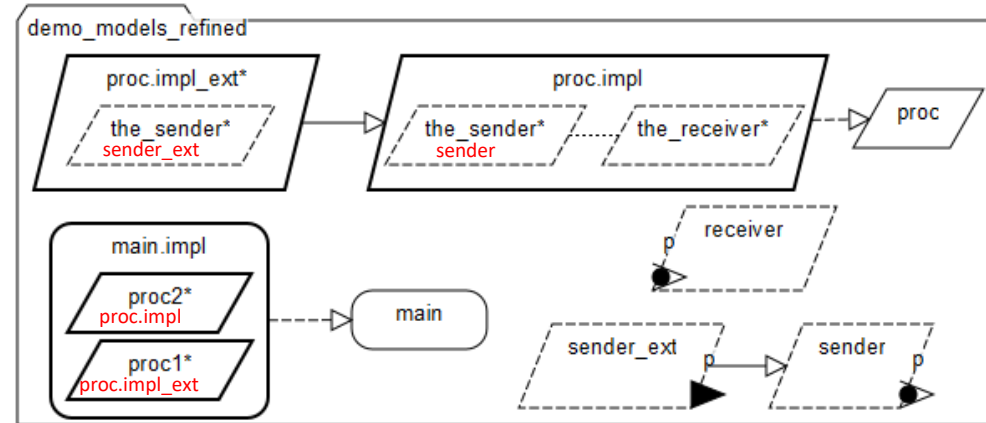
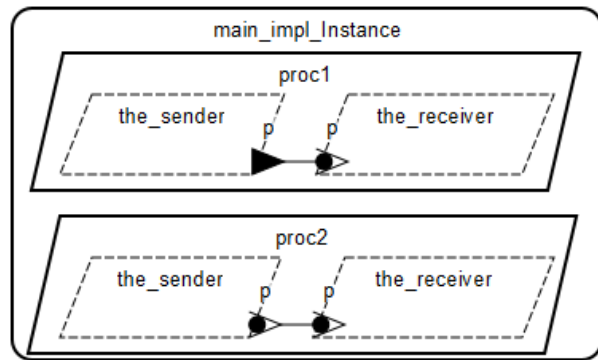
Simple! Change the corresponding feature.

# Example: Need for automated de-instantiation



But this also changes  $p$  in `proc2.the_sender` !!

# Example: Need for automated de-instantiation



```

package demo_models_refined
public
  system main
  end main;

  system implementation main.impl
  subcomponents
    proc1: process proc.impl_ext;
    proc2: process proc.impl;
  end main.impl;

  process proc
  end proc;

  process implementation proc.impl
  subcomponents
    the_sender: thread sender;
    the_receiver: thread receiver;
  connections
    cnx: feature the_sender.p -> the_receiver.p;
  end proc.impl;

  process implementation proc.impl_ext extends proc.impl
  subcomponents
    the_sender: refined to thread sender_ext;
  end proc.impl_ext;

  thread sender
  features
    p: out feature;
  end sender;

  thread sender_ext extends sender
  features
    p: refined to out data port;
  end sender_ext;

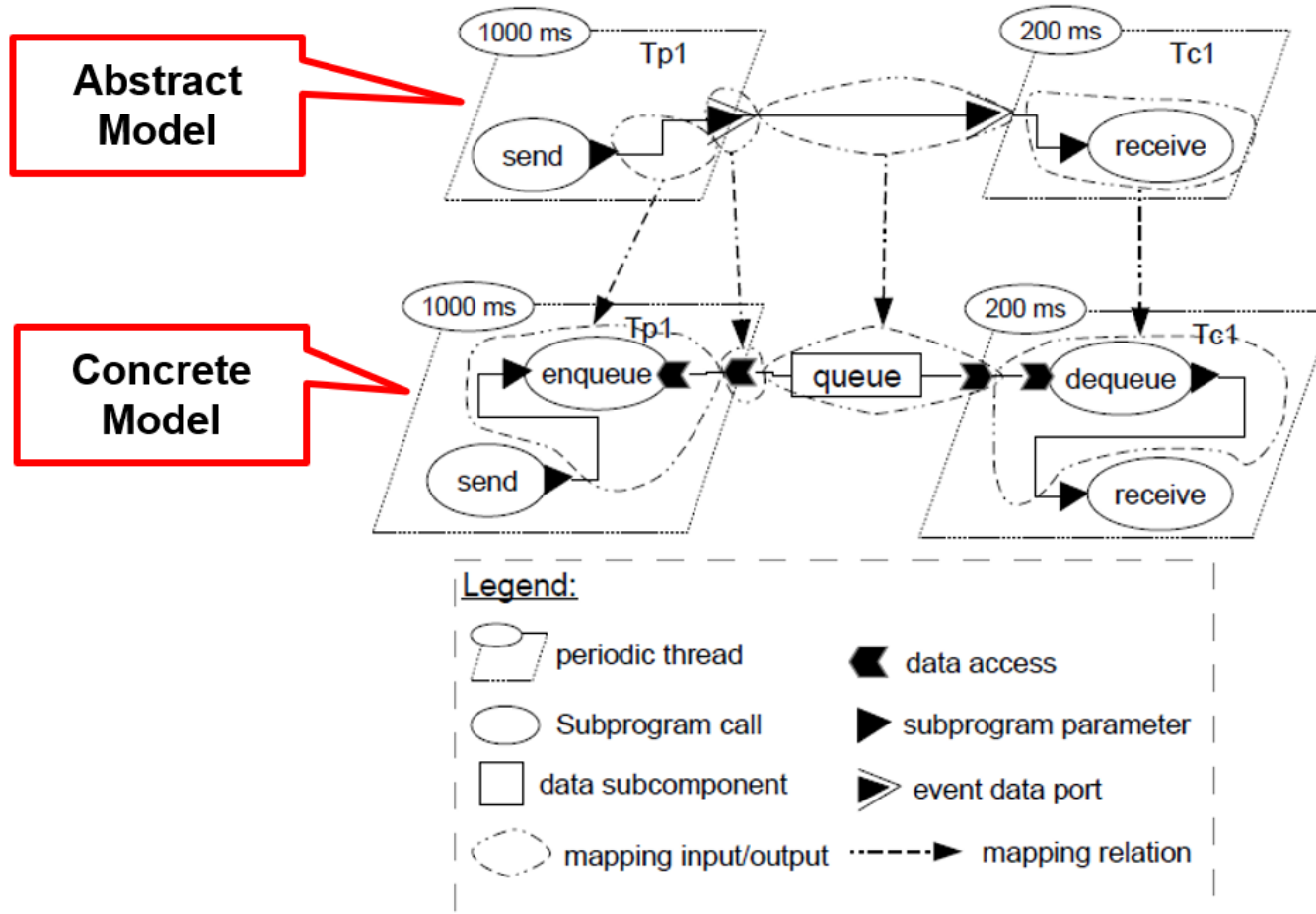
  thread receiver
  features
    p: in feature;
  end receiver;

end demo_models_refined;

```

Solution with preservation of information is very complicated with many extensions and refinements, even for a simple instance update!

# Real-life Scenario: RAMSES



Simplest refinement pattern in RAMSES:

Changing data port connection between two threads by replacing with a shared data component.

The data port features are changed to data access features.



Too many choices for de-instantiation in AADL!

+

Complications for information preservation due to many dependencies between elements and modularity.

=

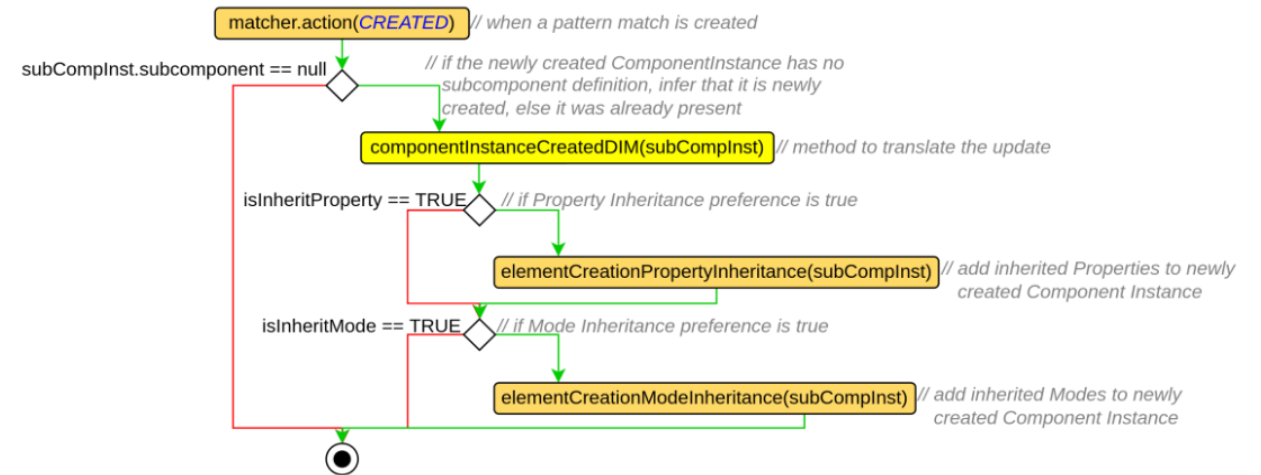
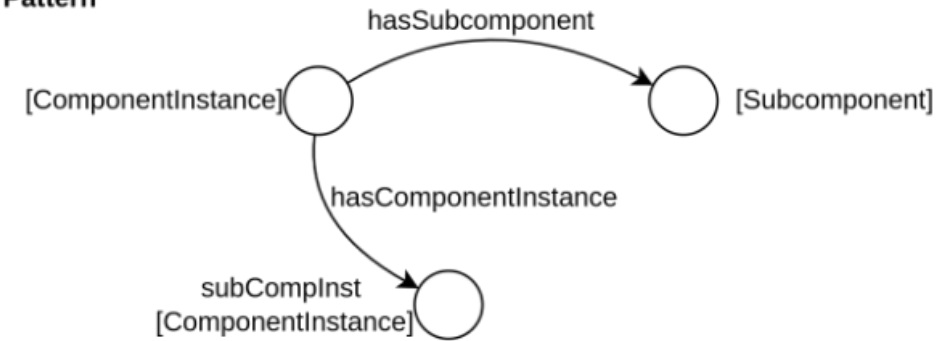
Makes de-instantiation of updates highly complex; requiring automation

# OSATE-DIM

- OSATE Declarative-Instance Mapping
- Eclipse/OSATE-based plugin
- Graph Transformations
  - VIATRA
    - Graphical Queries
    - Model Transformation Rules
    - Reactive nature (incremental)



Graphical Query Pattern

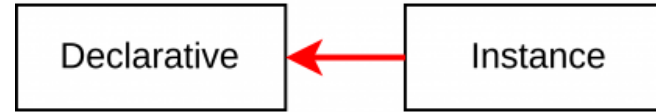


# OSATE-DIM Values/Aims

- Maximum Information Preservation
- Least/Minimal Change
- Very-well behaved lens (3 Lens laws)
  - No extraneous model updates.
  - Equality of updated-model state with updated view-state.
  - Composability of updates.
- Flexibility
  - Scenarios
  - User preferences

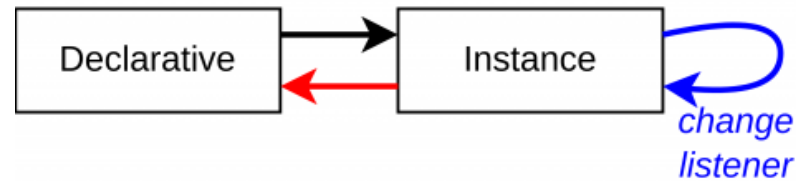
# Transformation Scenarios

## State-based



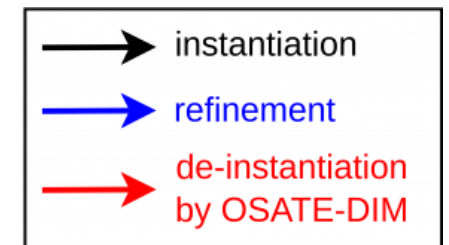
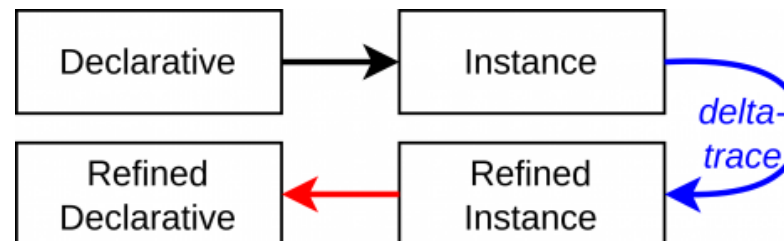
-- State-based Demo --

## Delta-based with In-place refinement

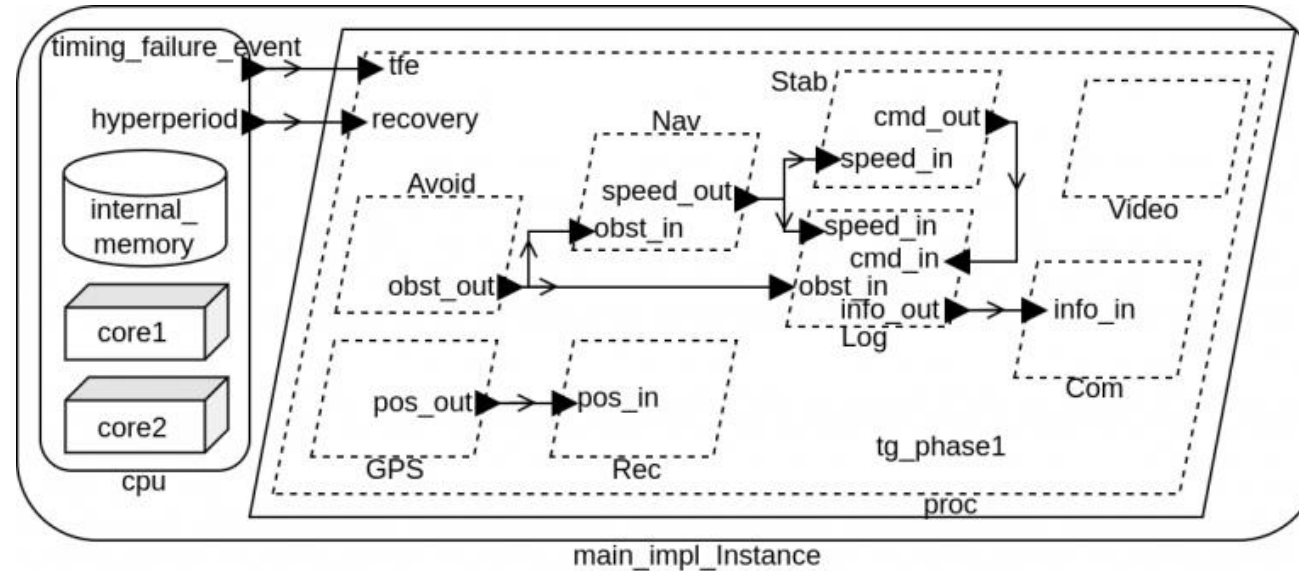


-- Delta in-place Demo --

## Delta-based with Out-of-place refinement

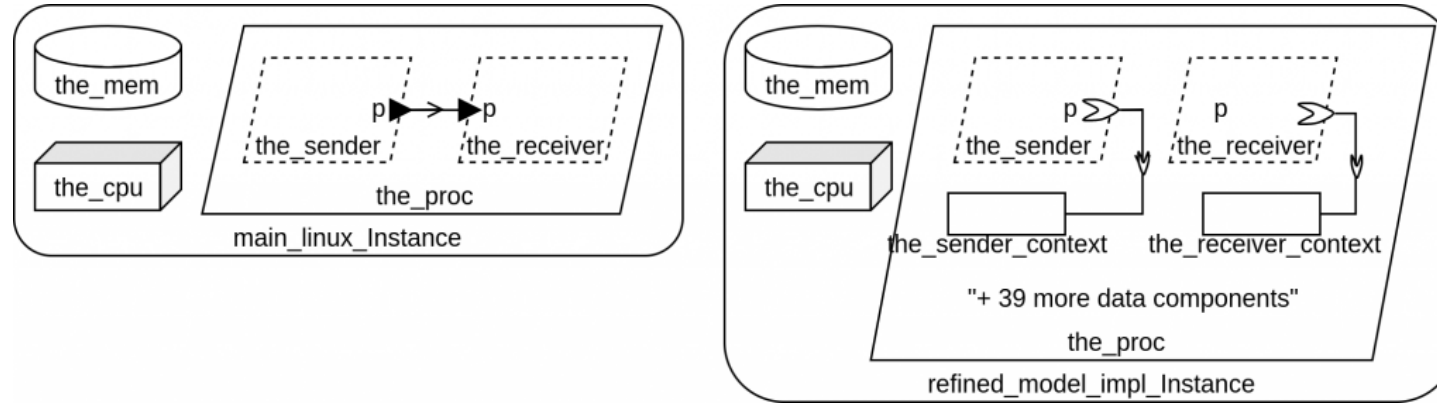


# Case Study: MC-DAG



- Addition of *Property Associations* (RAMSES::Execution\_Slots) for each *Thread*.
- Contain static scheduling tables for each *Thread* in different *Modes* LO and HI.
- Properties also reference the core and memory binding model elements, not just static data.

# Case Study: RAMSES



- Addition of 2 (+40) *Data Components* to a *Process Component*, which are shared by two threads.
- The *Port Features* interfacing the two threads with each other are changed to *Data Access* kinds.
- New *Data Access Connections* are also added between the shared *Data Components* and the *Threads*.
- The added *Data Components* have varying numbers of *Properties*, and the total number of newly added properties is 122.

# Conclusion

Introduced View-Update Problem in AADL-OSATE

OSATE-DIM is an automated solution for synchronizing Instance and Declarative models: 'de-instantiating' the Instance model

Three different scenarios (2 supported currently)

Wide range of view-updates supported

Tested on a preliminary test-bench

Simplifies the development of AADL model refinement tools

# Future Work

Complete Implementation of Delta Out-of-place scenario

Further validation

Integration with RAMSES

Concepts have potential to be used for "transpilation"



# Publications

Rakshit Mittal, Dominique Blouin, Anish Bhowe, Soumyadip Bandyopadhyay. 2022. Solving the instance model-view update problem in AADL. In Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems (MODELS '22). Association for Computing Machinery, New York, NY, USA, 55–65. <https://doi.org/10.1145/3550355.3552396>

Rakshit Mittal, Dominique Blouin. 2022. OSATE-DIM solves the instance model-view update problem in AADL. In Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings (MODELS '22). Association for Computing Machinery, New York, NY, USA, 1–6. <https://doi.org/10.1145/3550356.3559083>

Rakshit Mittal. 2022. The instance model-view update problem in AADL. In Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings (MODELS '22). Association for Computing Machinery, New York, NY, USA, 221–224. <https://doi.org/10.1145/3550356.3552373>

Thank you for your attention  
Questions? with Dominique



## Tool

Webpage: [mem4csd.telecom-paristech.fr](http://mem4csd.telecom-paristech.fr)  
Zenodo artifact DOI: 10.5281/zenodo.6971720



## Contact:

Dominique: [dominique.blouin@telecom-paris.fr](mailto:dominique.blouin@telecom-paris.fr)  
Rakshit: [rakshit.mittal@uantwerpen.be](mailto:rakshit.mittal@uantwerpen.be)