Dependability Modeling Based on AADL Description

(Architecture Analysis and Design Language)

Ana Rugina, Karama Kanoun and Mohamed Kaâniche

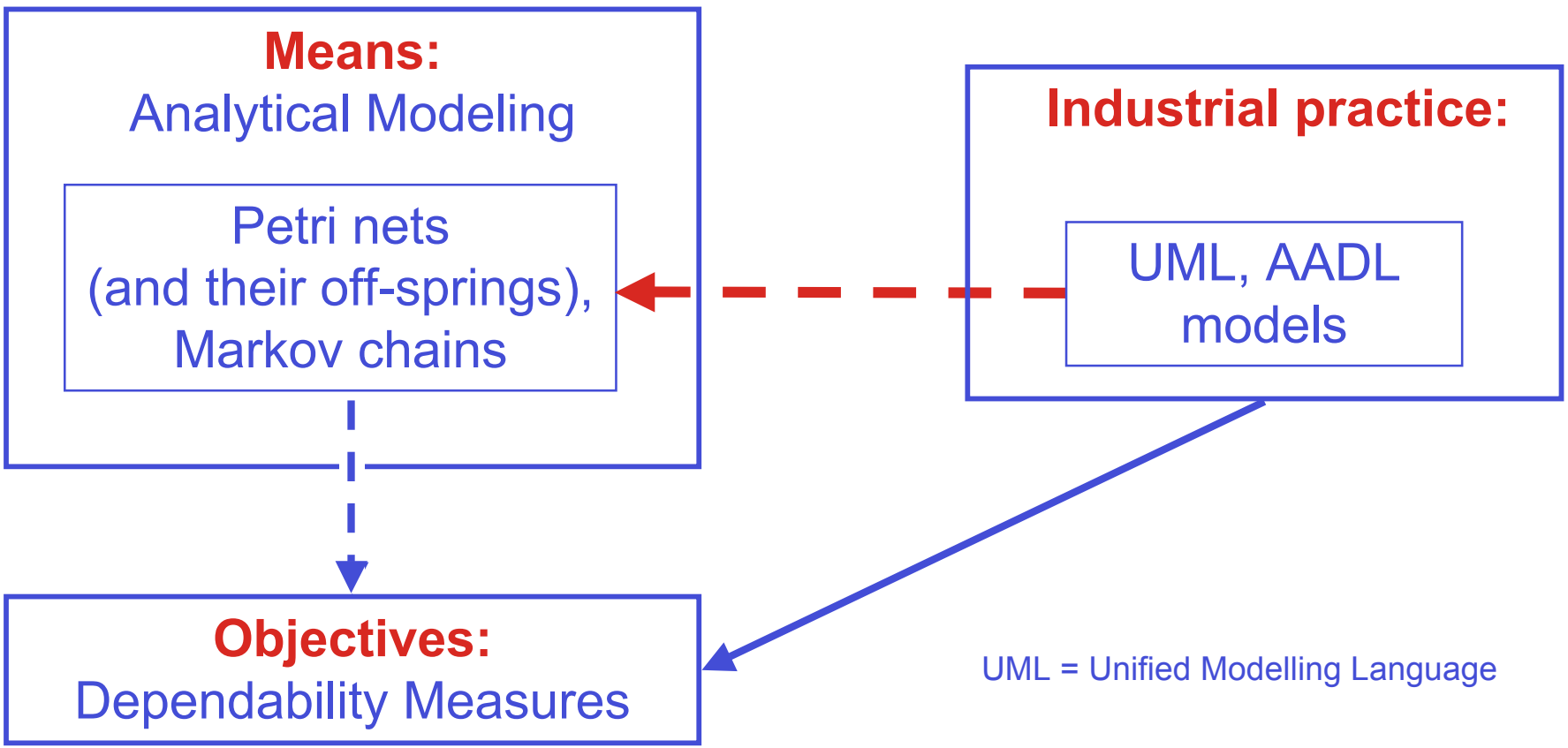
{rugina, kanoun, kaaniche}@laas.fr

European Integrated Project ASSERT

(Automated proof based System and Software Engineering for Real-Time Applications)

February 19, 2006

Context

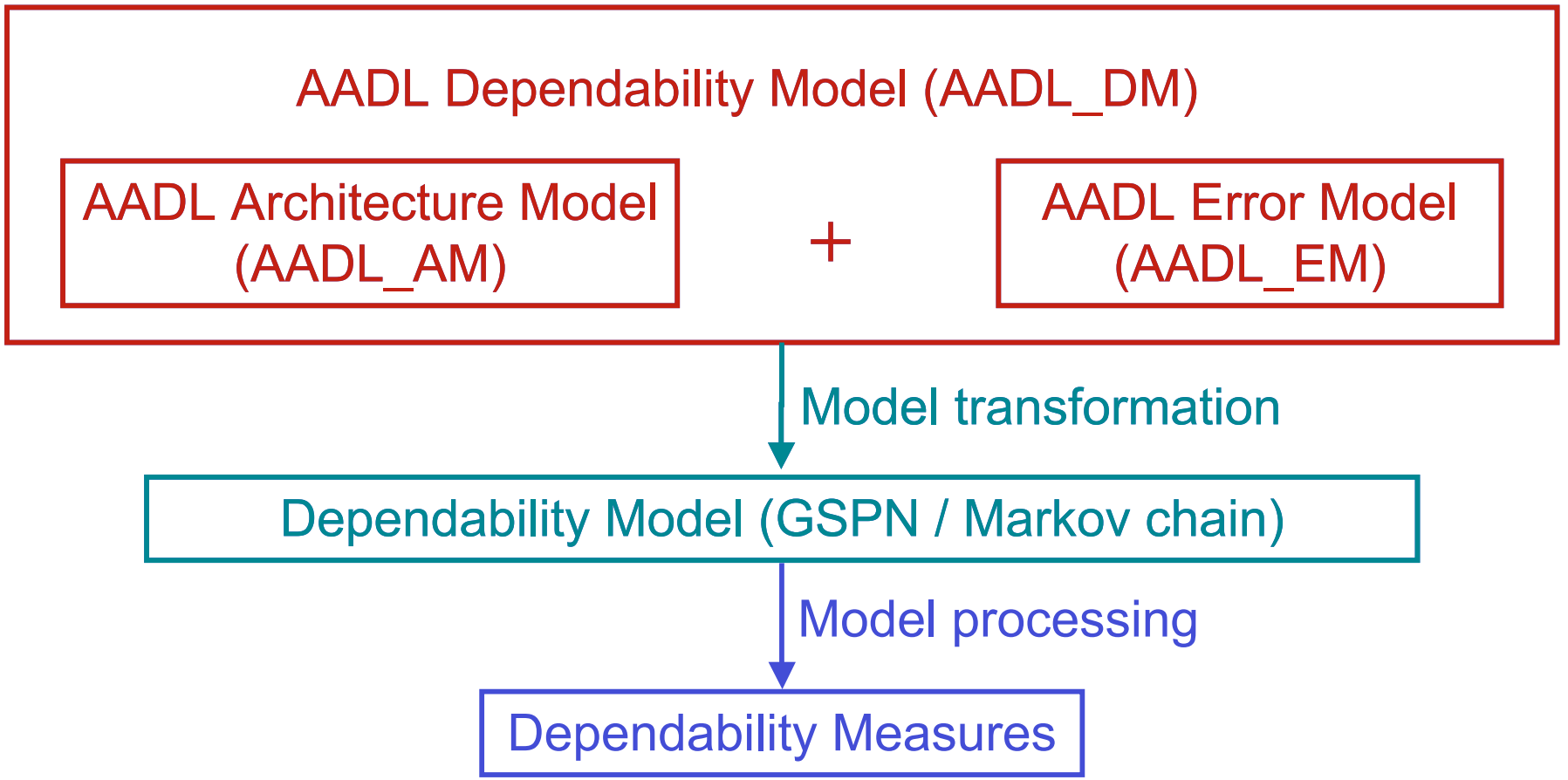


UML = Unified Modelling Language

AADL?

- AADL is an international standard for predictable model-based engineering of real-time and embedded computer systems
- AADL: a textual and graphical language with precise execution semantics for modeling the architecture of embedded software systems and their target platform
- What AADL can do?
  - Represent embedded systems as component-based system architecture
  - Model component interactions as flows, service calls, and shared access
  - Model task execution and communication with precise timing semantics
  - Model execution platform and specify application binding
  - Represent operational modes and fault tolerant configurations
  - Support component evolution and large-scale development
  - Accommodate analyses such as reliability & safety- criticality through extensions

Overview of the Modeling Approach



AADL Error Model (AADL_EM)

- The AADL_EM is:
  - Associated with an AADL_AM (Architecture Model) of a component
  - Built on the AADL_AM's skeleton
- Component AADL_EM characterizes the behavior in the presence of
  - Internal faults and repair events
  - External propagations from the component environment
- Only components that have associated AADL_EMs and all connections between them are part of the AADL_DM
- Not all the details of the AADL_AM are necessary for the AADL_DM
- AADL_EM =
  - A model type
  - one or more error model implementations

AADL Error Model (AADL_EM)

- Error model type declares features
  - Set of error states, events and incoming/outgoing propagation names
- Error model implementation
  - Transitions between states, triggered by events, in the error model type
  - Occurrence properties: arrival rates / probability of propagations and events
- Vote properties (Vote_in, Vote_out)
  - Control propagations by means of boolean expressions and predicates
  - Associated to ports, data components, client and server subprograms
  - The source error model sends the propagation out to all AADL component to which this error model is associated
    - This `out` propagation arrives to one or more receiver component's error models
    - Only receivers declaring `In` propagation with the same name (name matching) are influenced by this propagation
    - The state transitions and operational mode changes triggered by the `In` propagation are simultaneous

Error Model Example

```
error model Basic
features
    Error_Free: initial error state;
    Failed: error state;
    Fail, Repair: error event;
    No_Data: out error propagation;
end Basic;
```

```
error model implementation Basic.Nominal
transitions
    Error_Free-[Fail]->Failed;
    Failed-[Repair]->Error_Free;
    Failed-[out No_Data]->Failed;
properties
    Fail.Occurrence=>poisson λ;
    Repair.Occurrence=>poisson μ;
    No_Data.Occurrence => fixed 0.5;
end Basic.Nominal;
```

AADL Dependability Model (_DM) Construction

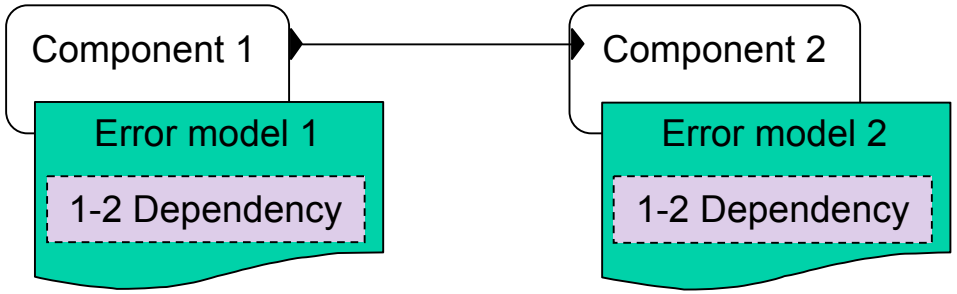
- Independent components:
  - The system AADL_DM is composed of the AADL_DMs of its components
- Dependencies
  - Architecture-based: functional, structural, system reconfiguration
  - Maintenance
  - Hierarchy

→ Needs for a structured and iterative approach for building progressively the AADL_EM and AADL_DM

  - First step: basic AADL_EM associated to components (isolated behavior)
  - Next steps: introduce incrementally dependencies between basic AADL_EMs

AADL Dependability Model

- Architecture-based dependency

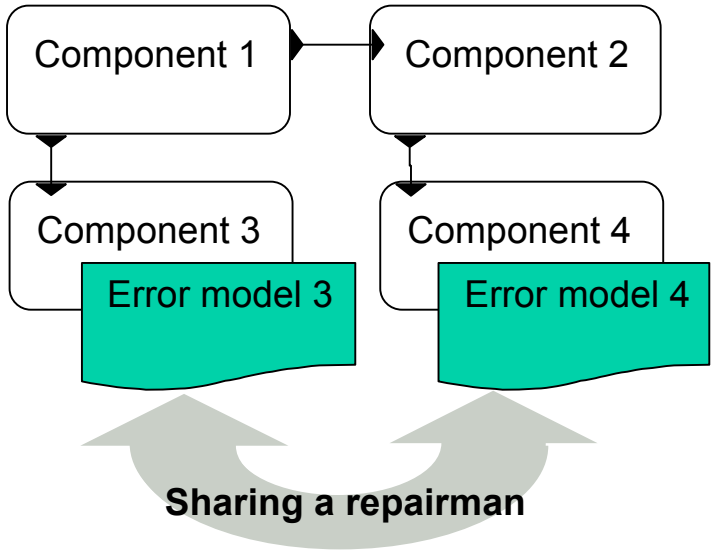


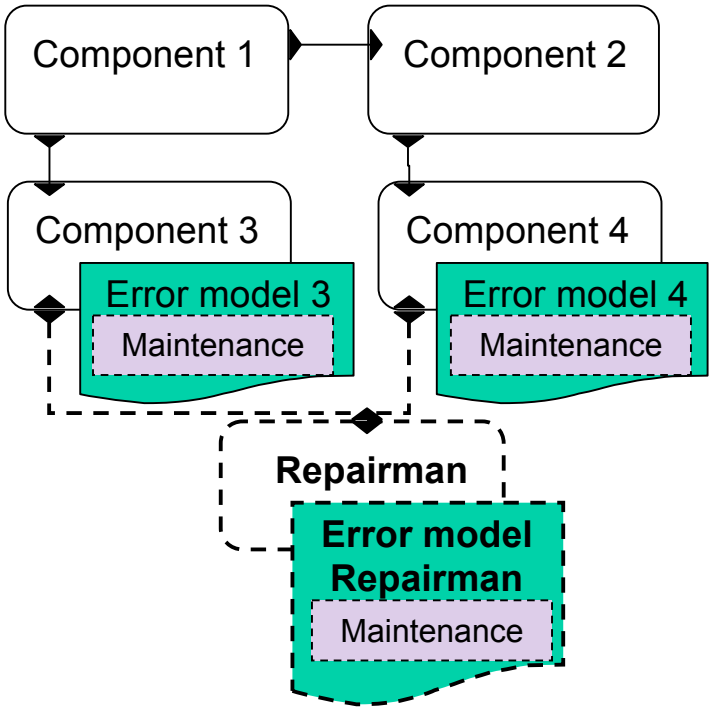

```
error model implementation for1.ex
transitions
    […]
    Failed-[out No_Data]->Failed;
properties
    […]
     No_Data.Occurrence => fixed 0.5;
end for1.ex;
```

```
error model implementation for2.ex
transitions
    […]
    Error_Free-[in No_Data]->Failed;




end for2.ex;
```

AADL Dependability Model

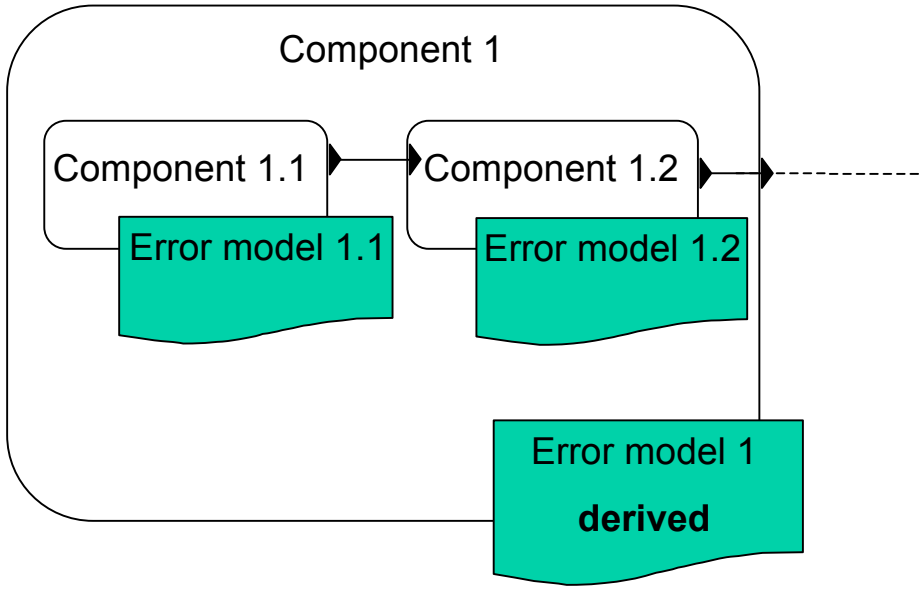
- Maintenance dependency



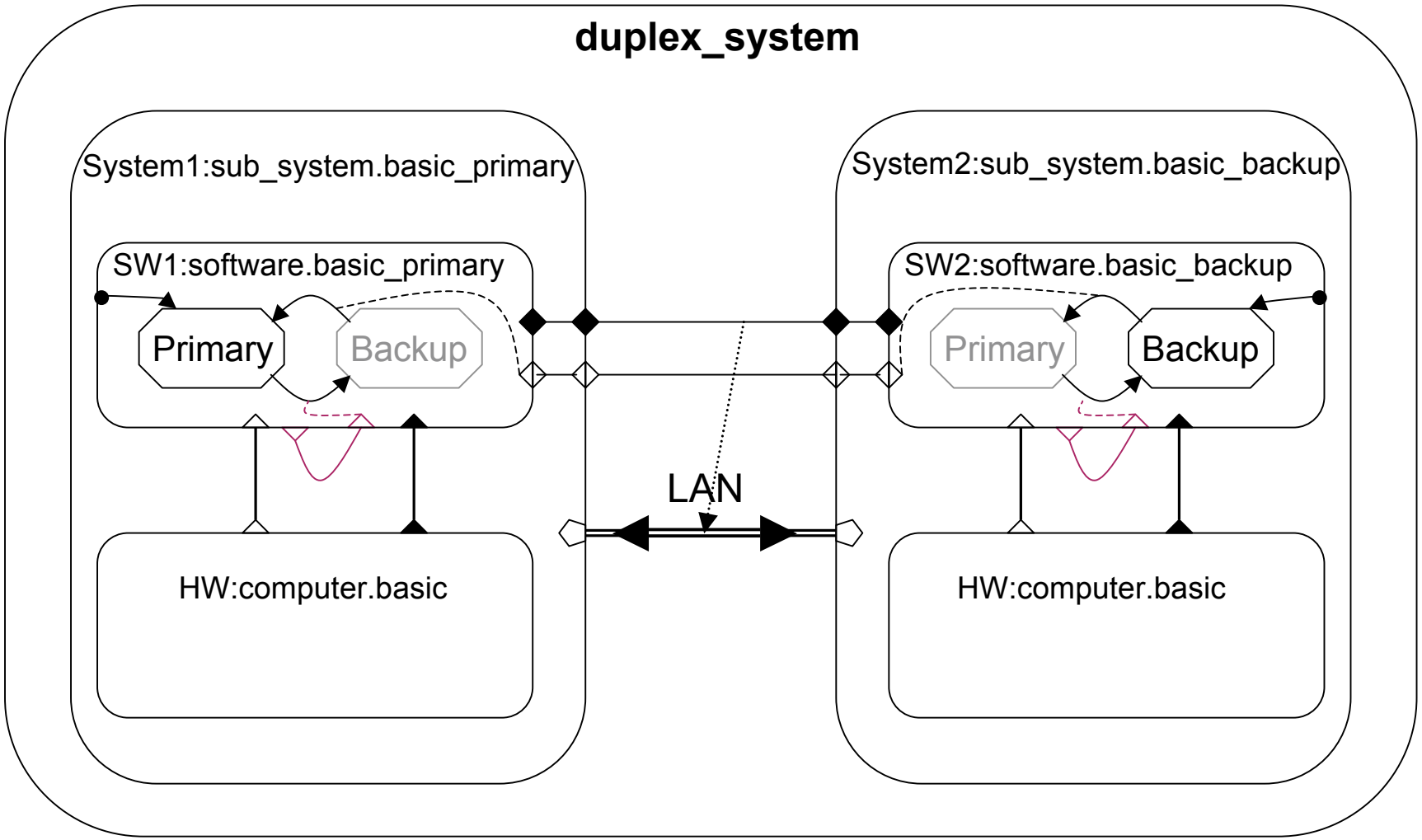


AADL Dependability Model

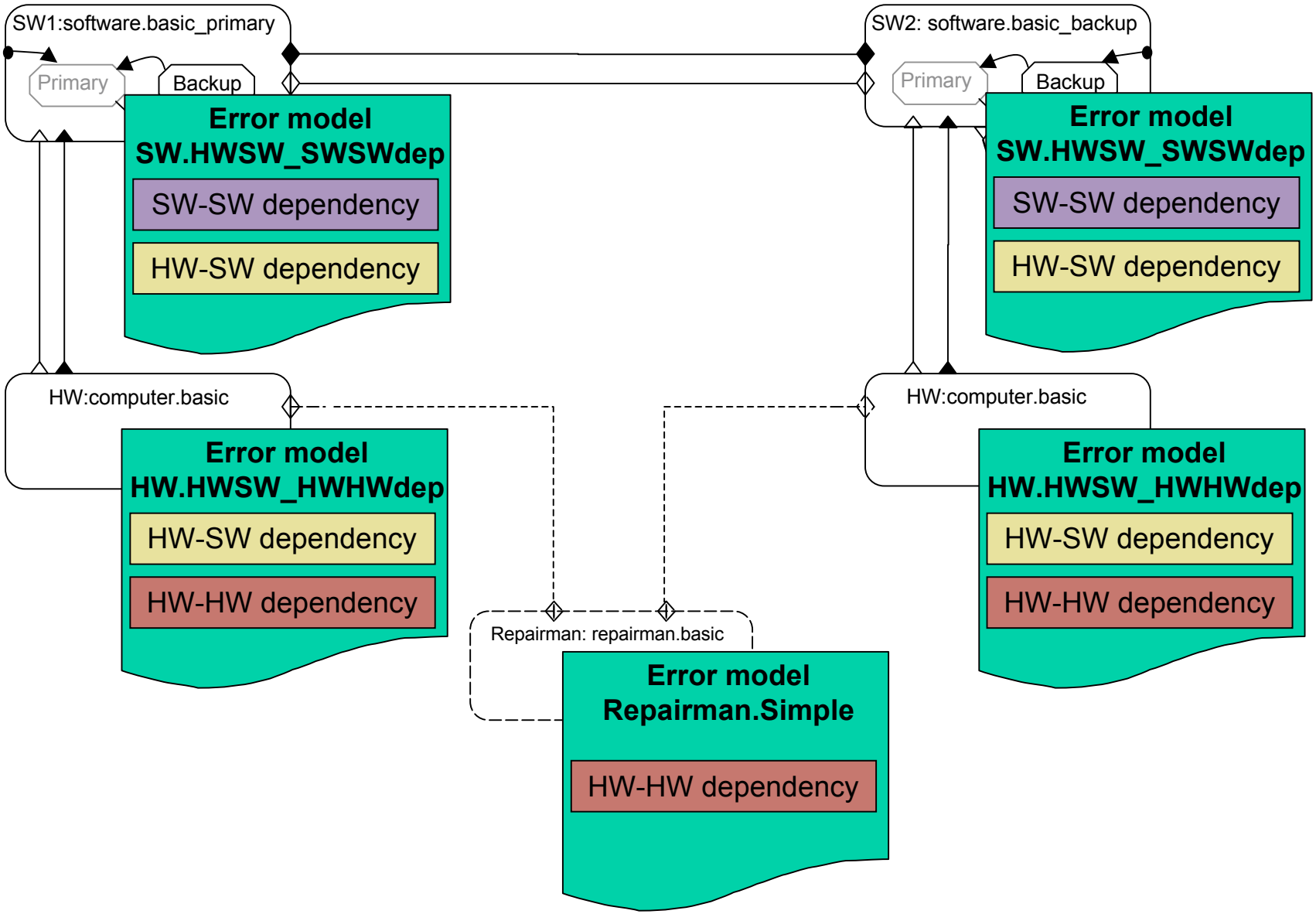
- Hierarchical systems
  - Derived dependency -> derived error model
  - Ignore containment details -> abstract error model



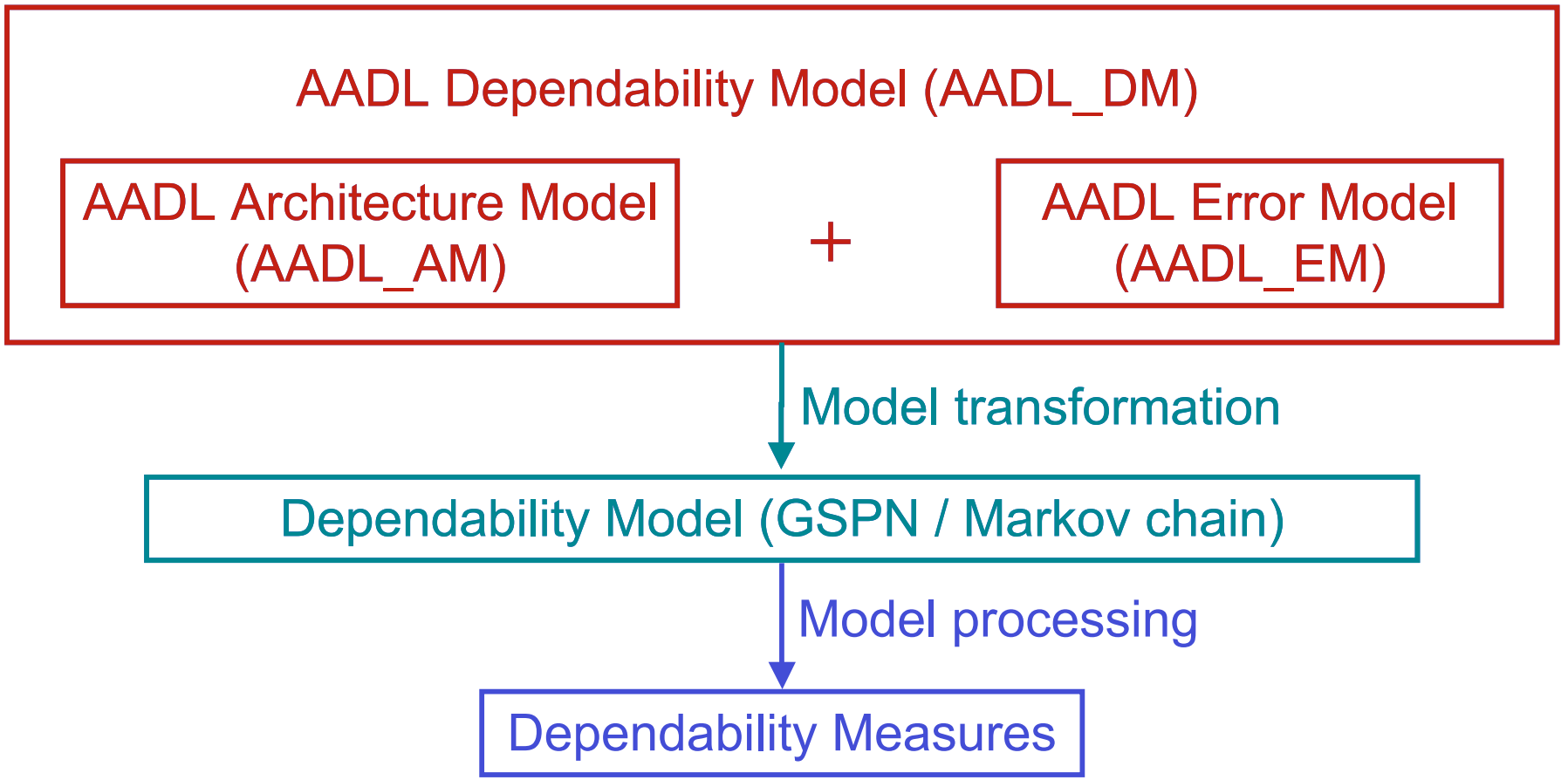
Case Study: Duplex System



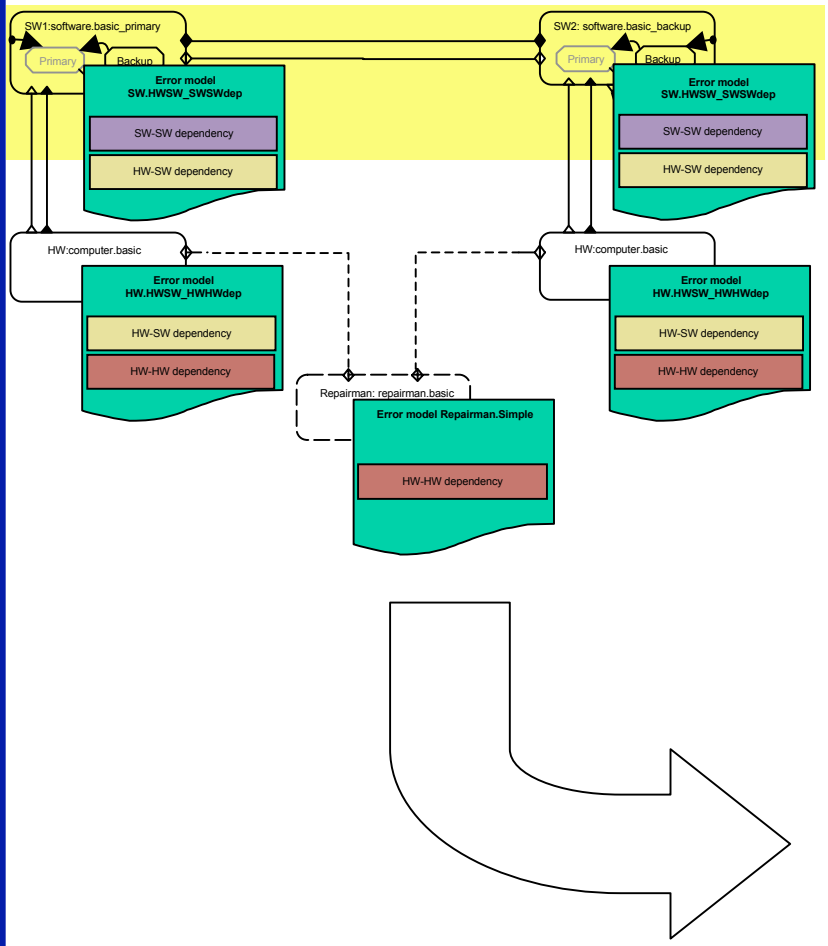
AADL Dependability Model

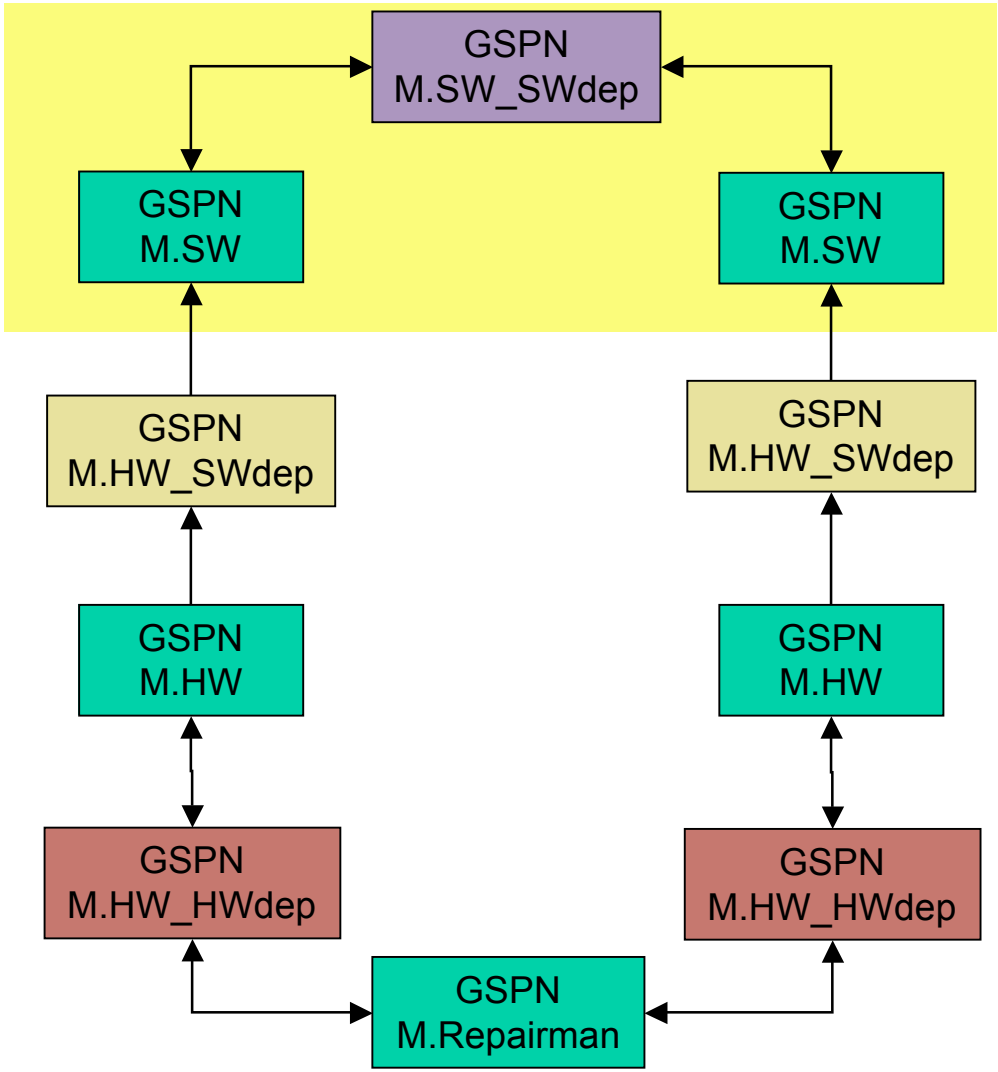


Overview of the Modeling Approach



GSPN Dependability Model





Conclusion

- AADL system error model
  - Stepwise construction
    - Building error models as if components were isolated
    - Adding dependencies progressively
  - Model transformation
- Error Model Annex assessment
  - Evolution proposals

Error Model Annex Evolution

- Occurrence properties

  Fixed values → unvaluated parameters

- Link between the mode model and the error model

  ⇒mode-dependent behaviour in presence of faults

- Vote_In and Vote_Out properties

  ⇒evaluate Boolean error expressions only when needed

- Inheritance and refinements

  ⇒similarly to the core standard mechanisms

- Ana is now a Visiting Scientist at the SEI in the Dynamic Systems Program, Performance-Critical Systems (PCS) Initiative (http://www.sei.cmu.edu/pcs/pcs.html)
- She works with Peter Feiler (the technical lead, author, and editor of the AADL standard, senior member of the technical staff) and his team (John Goodenough, Jorgen Hansson, Aaron Greenhouse, ...)
- She is participating in the design and writing of an AADL User's Dependability Modeling Guide
- Collaboration between SEI and LAAS after her return to France