Dependability modeling and analysis with the MARTE-DAM profile

PaCo Meeting, 25-26/06/09

UNITO Task: Development of a UML profile for dependability analysis

Simona Bernardi
UNITO

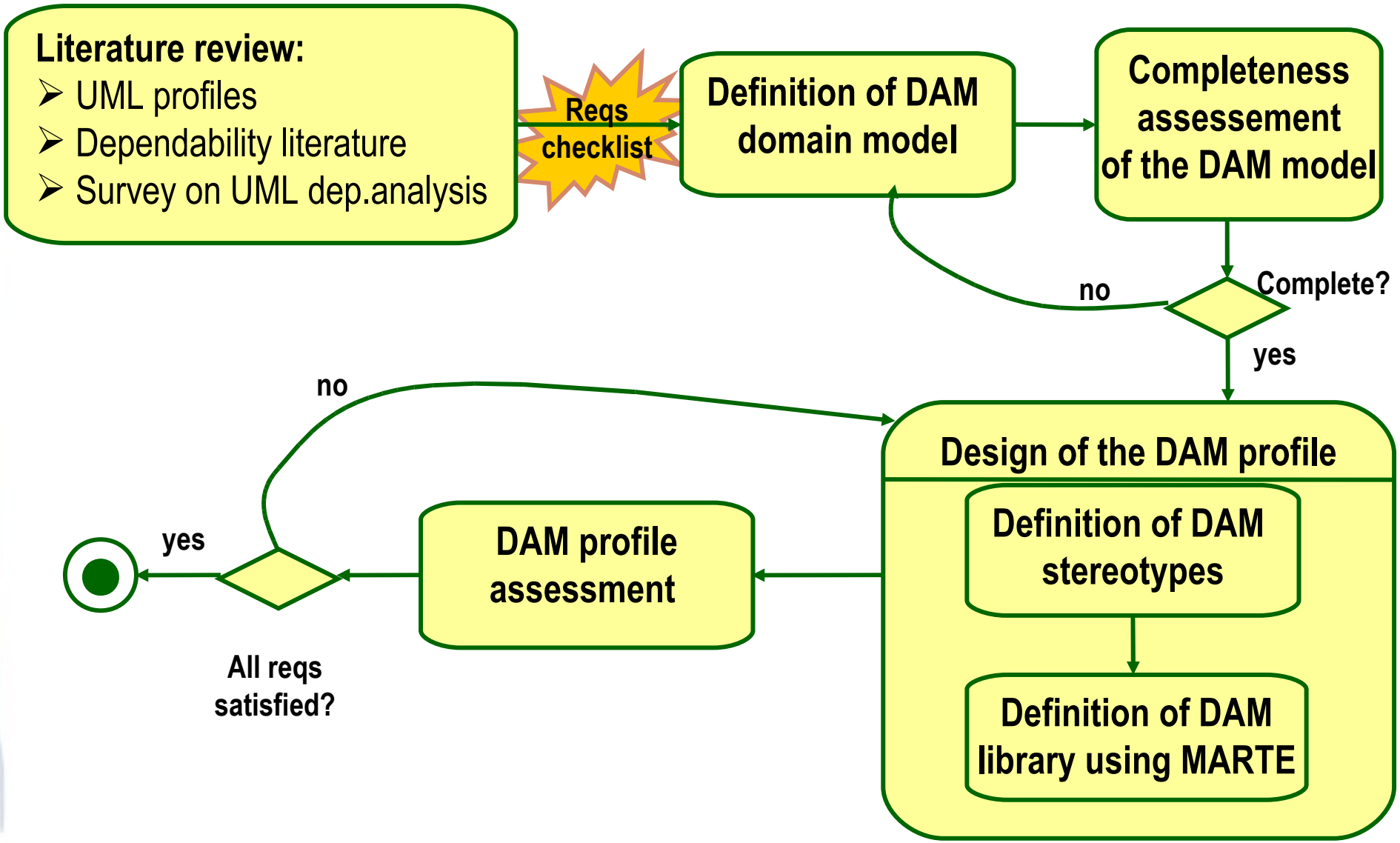
Recently completed works

- S.Bernardi, J. Merseguer, D.C. Petriu, A Dependability Profile within MARTE. Submitted to SOSYM journal, 2009.
- S.Bernardi, J. Merseguer, D.C. Petriu, Adding Dependability Analysis capabilities to the MARTE profile. MODELS08, October 2008.
- S. Bernardi, J. Merseguer, D.C. Petriu, An UML profile for dependability analysis and modeling of software systems, *Tech.Rep. no. RR-08-05, DIIS, Universidad de Zaragoza, Spain, May, 2008.*

Motivation and objectives

- The current standard UML profiles do not provide concrete capabilities for dependability analysis in a light-weight fashion
- Several proposals on deriving dependability models from UML-based models
- Propose a UML profile for ***quantitative*** dependability analysis of sw systems modeled with UML
- Focus on availability, reliability, maintainability and safety properties

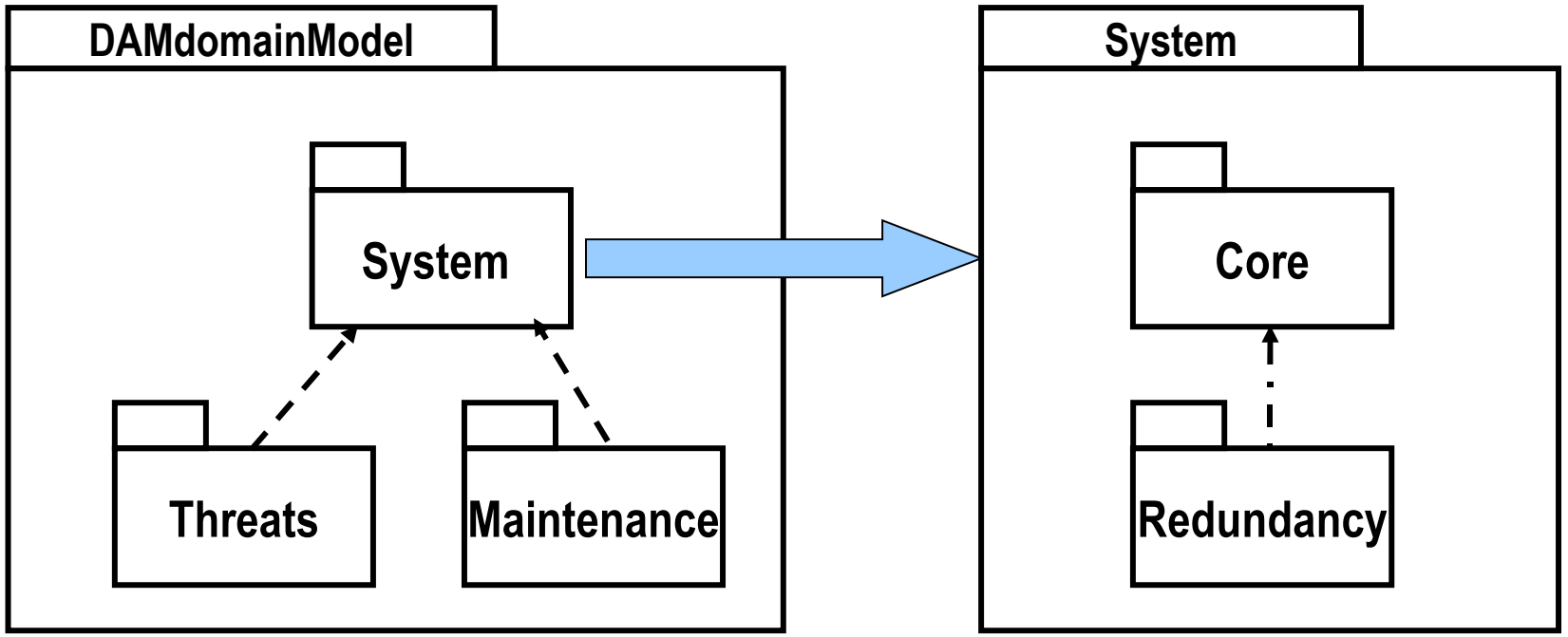
Methodological approach overview



Information requirement checklist

ID	Requirement Description
R1	Identification of the DAM context: reliability, availability, maintainability, safety
R2	Specification of dependability reqs in terms of upper/lower bounds
R3	Specification of dependability metrics to be estimated and properties to be verified (to assess R2)
R4	Threats characterization (faults, errors, failures, hazards, accidents) that may affect both hw/sw resources and their relationships (FEF chain, H-A, error propagation)
R5	(For repairable systems) Characterization of repair/recovery processes that remove basic/derived threats from the system
R6	Specification of incorrect behavior of the system affected by threats as well as the recovery actions that restore the system state
R7	(For fault tolerant systems) Specification of hw/sw redundant structures

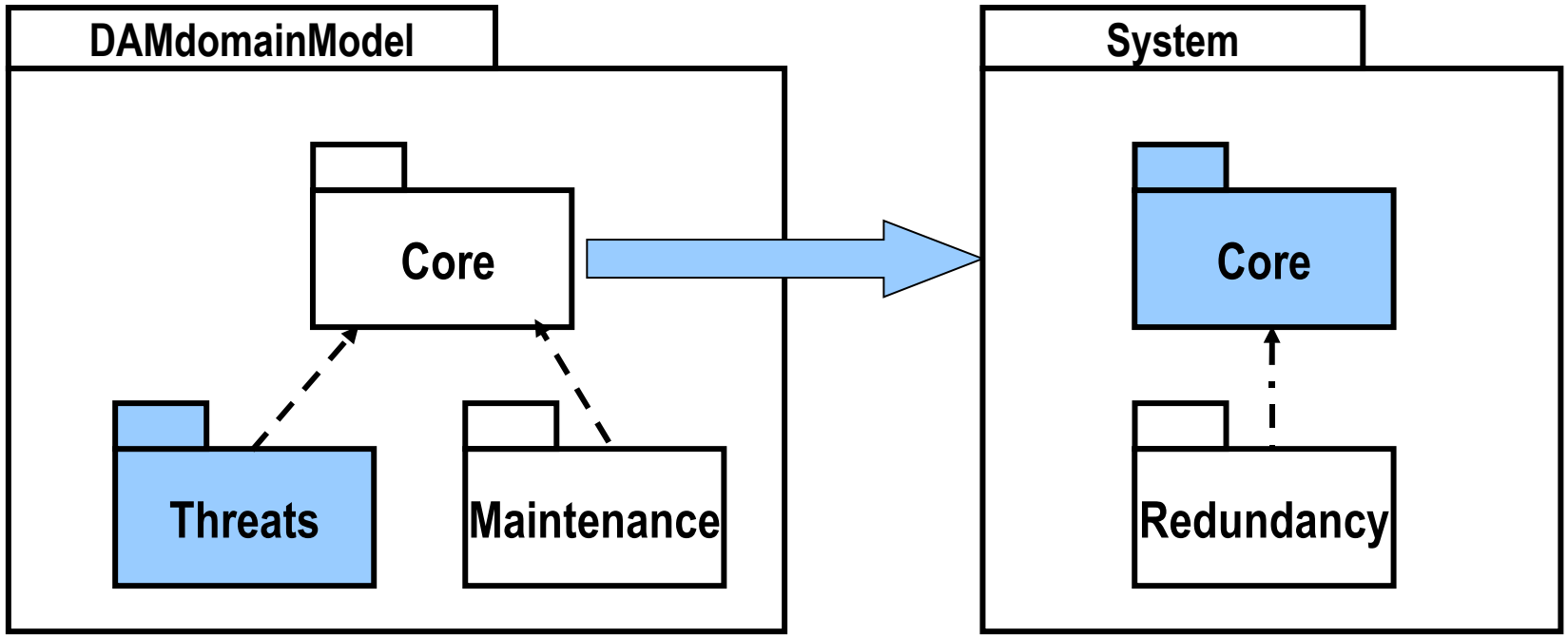
DAM domain model overview



Top-level package

System package

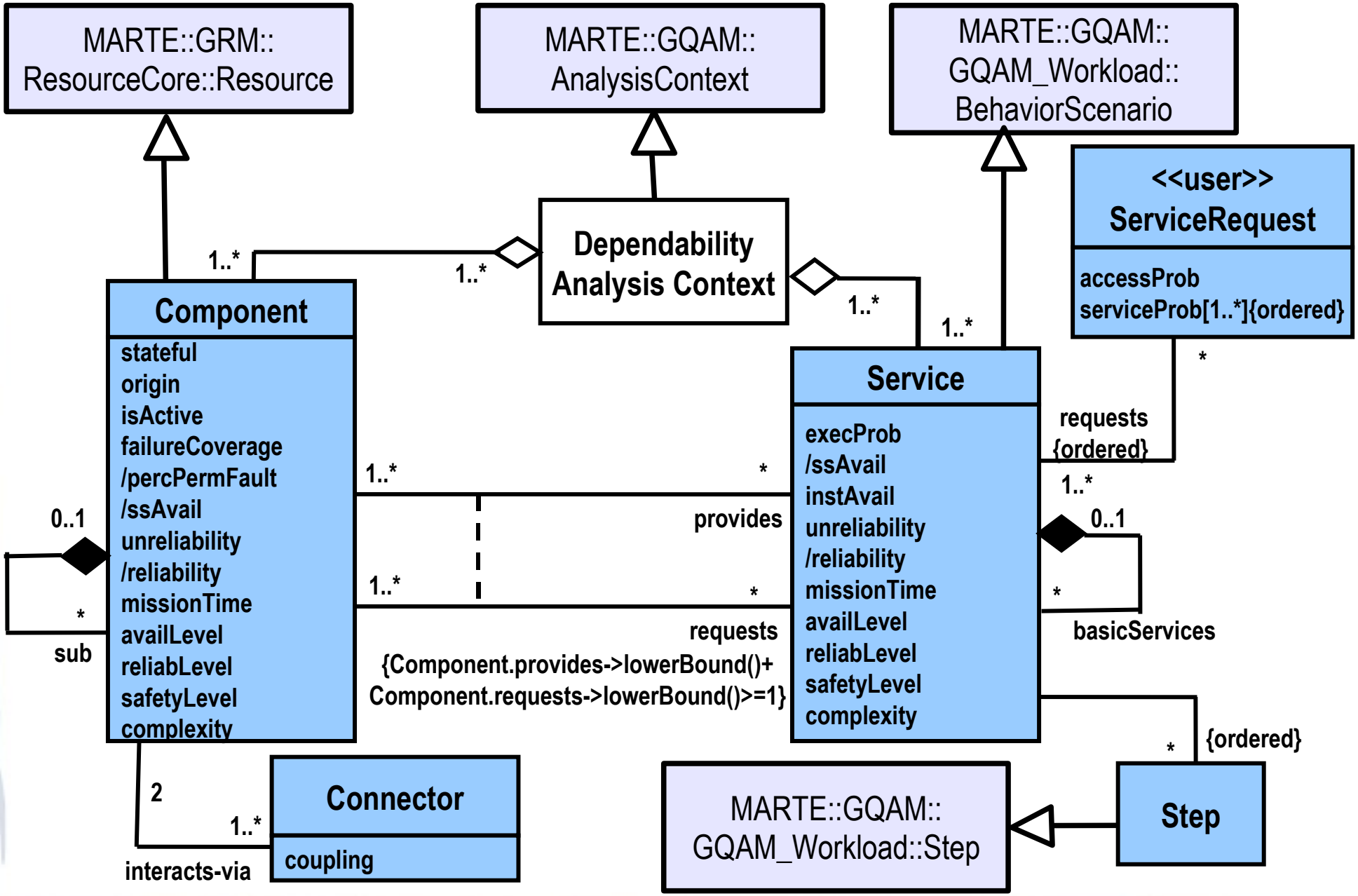
DAM domain model: Core & Threats



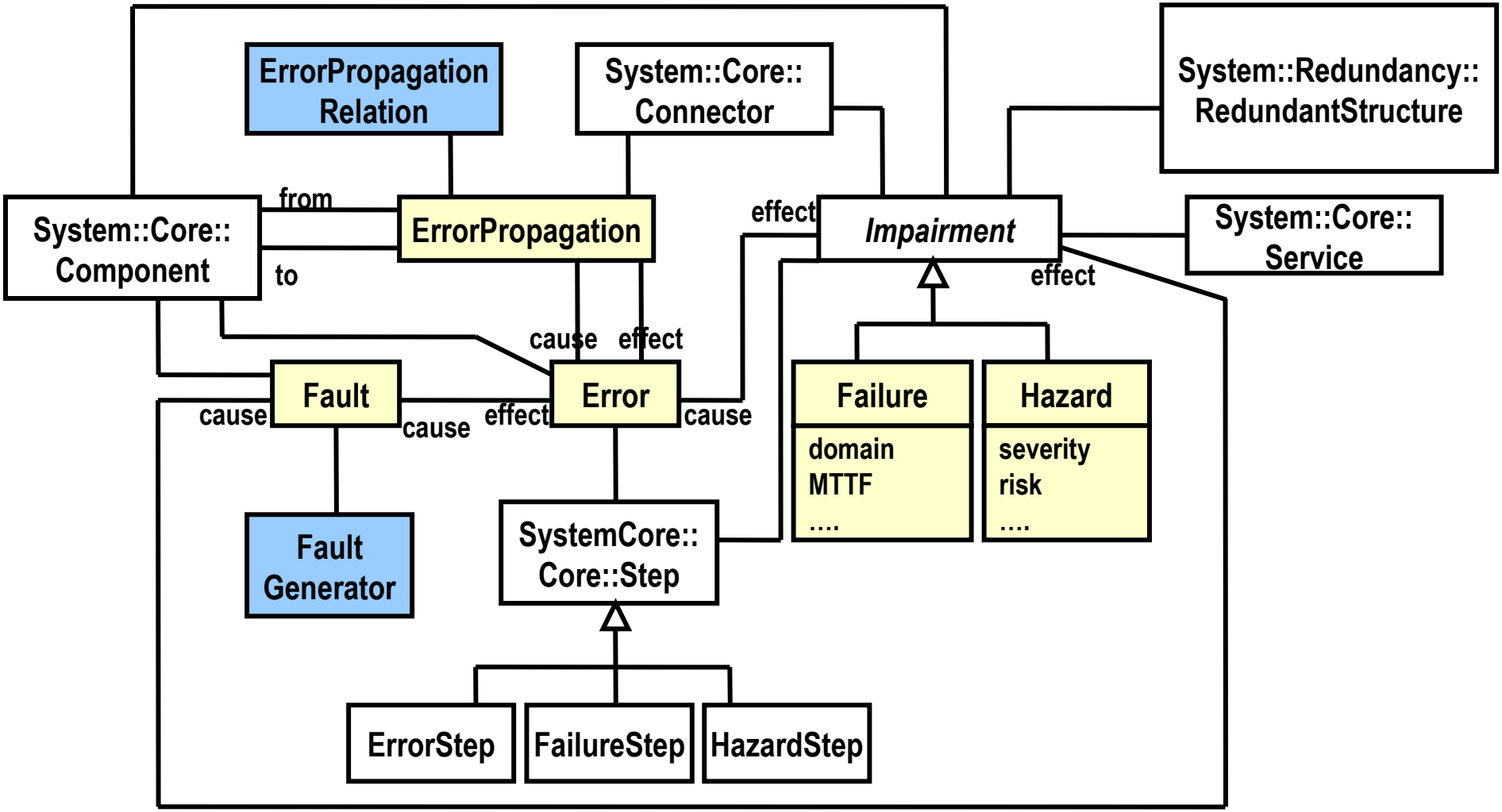
Top-level package

System package

DAM Core model



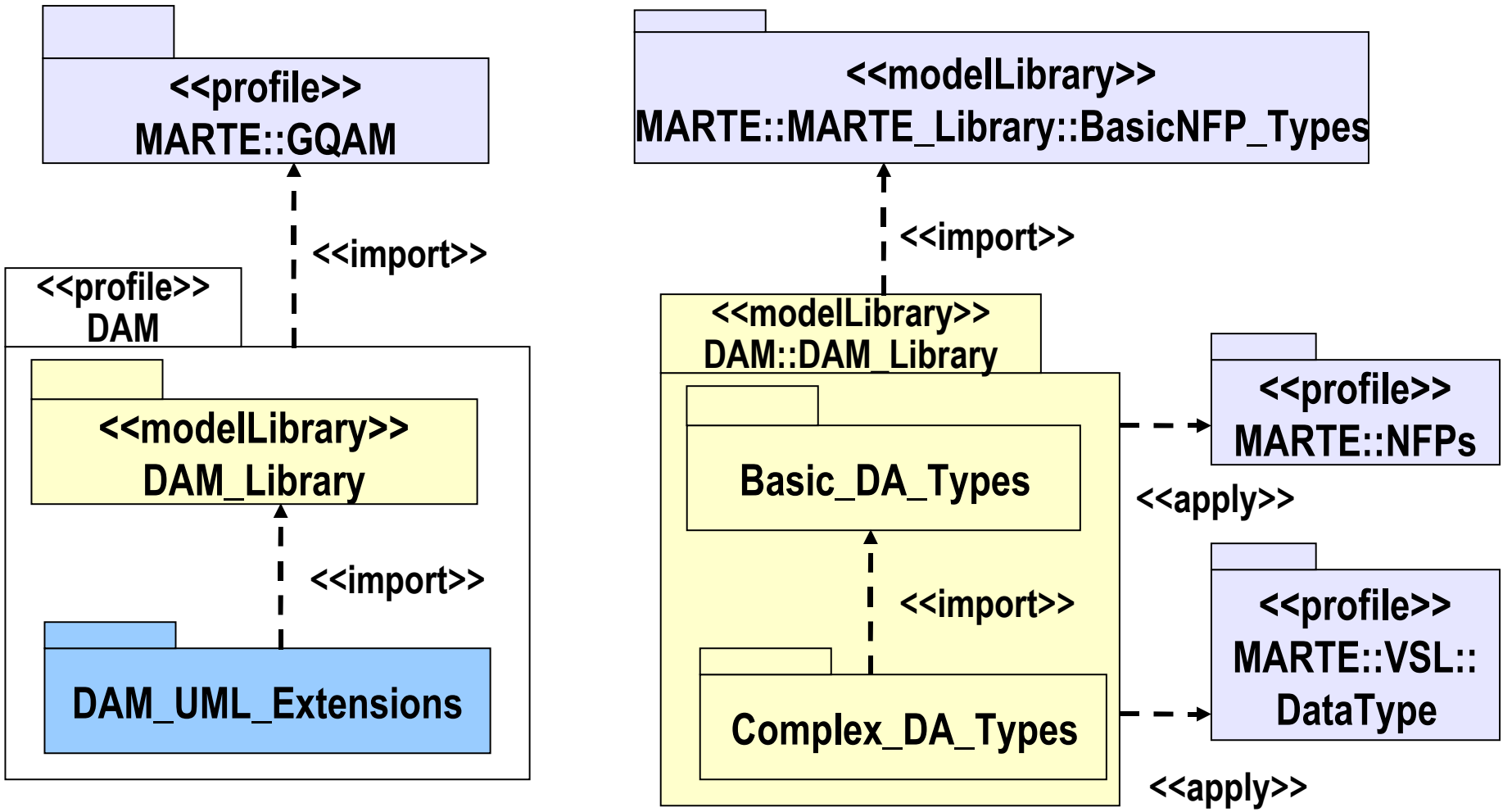
DAM Threats model



DAM profile definition

- The mapping process from the domain model elements to the DAM profile has been an iterative one
- We applied several guidelines (Selic) and patterns (Lagarde&al) to design a technically correct and consistent profile
- We used best practise of MARTE to trace the mapping
- We specialized MARTE to reuse already defined concepts

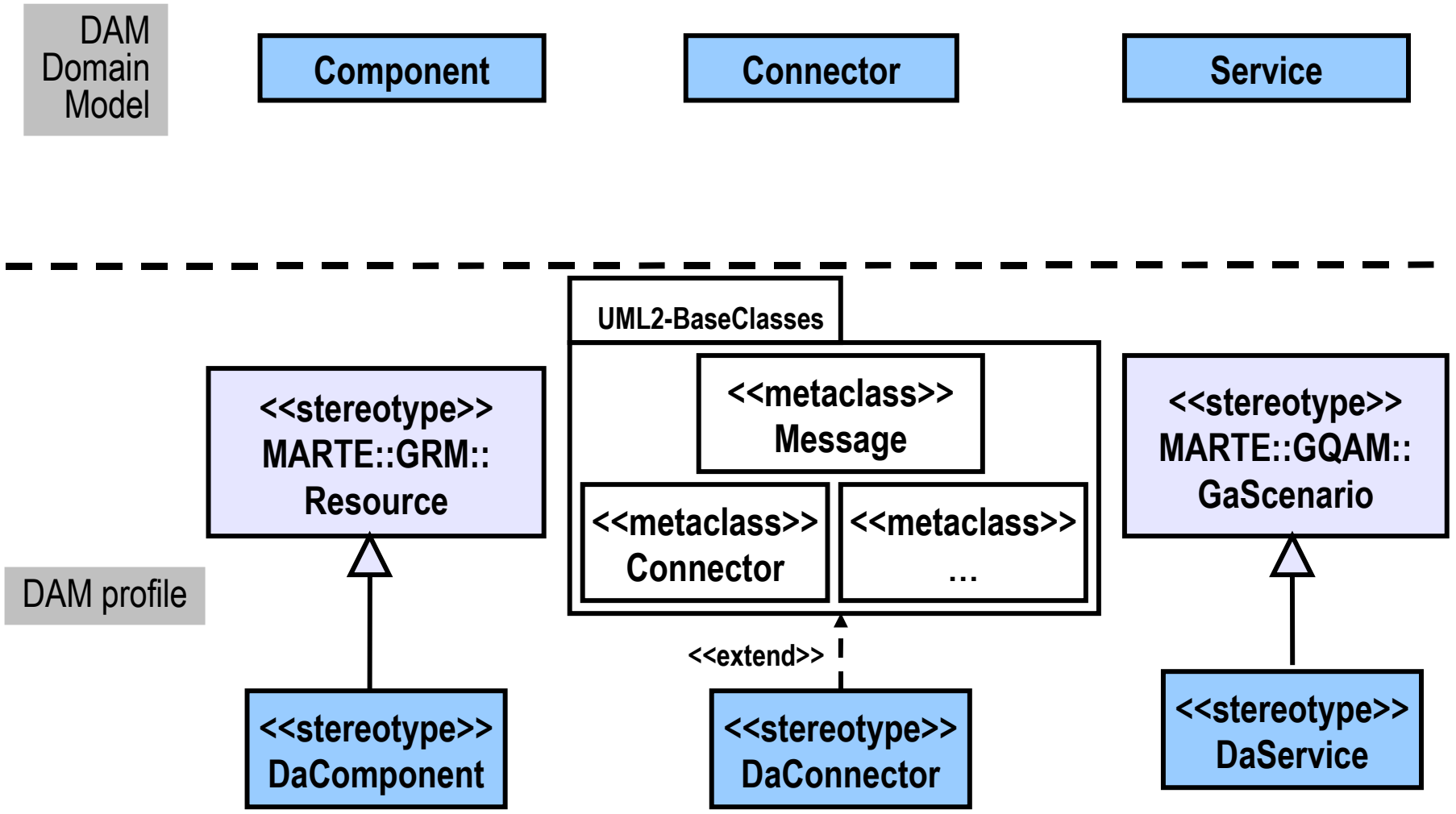
DAM profile overview



Mapping of domain classes

- Domain classes are good candidates to become stereotypes, but eventually only a subset of them have been mapped to a stereotype
- Objective: provide a “small” set of stereotypes
  - Abstract classes not considered
  - Threat/Maintenance concepts are complex dependability types of the DAM library
  - “Subsuming taxonomic concept pattern”: E/F/H steps classes become enumeration type values

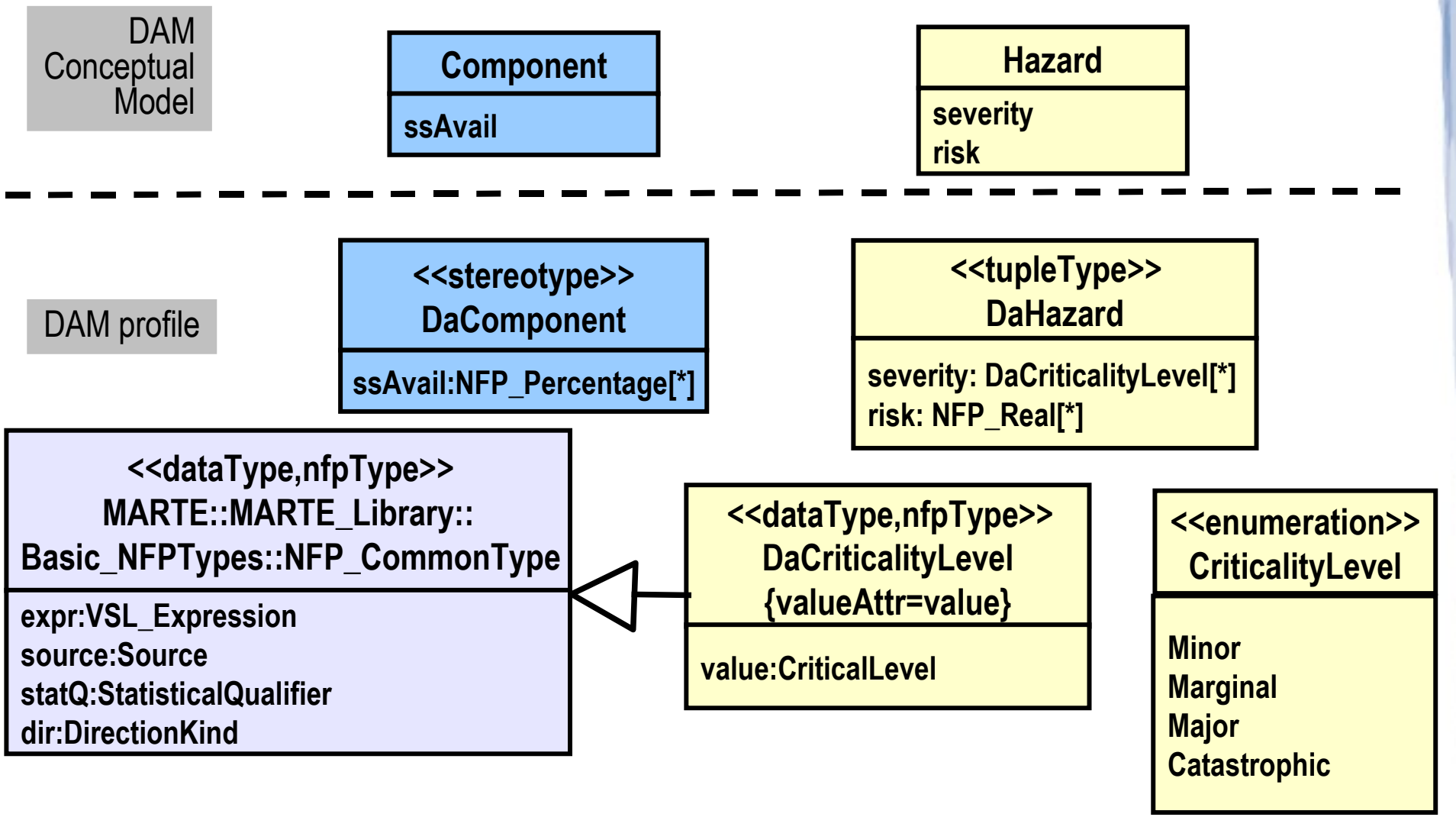
Stereotype definition



Mapping of domain attributes/associations

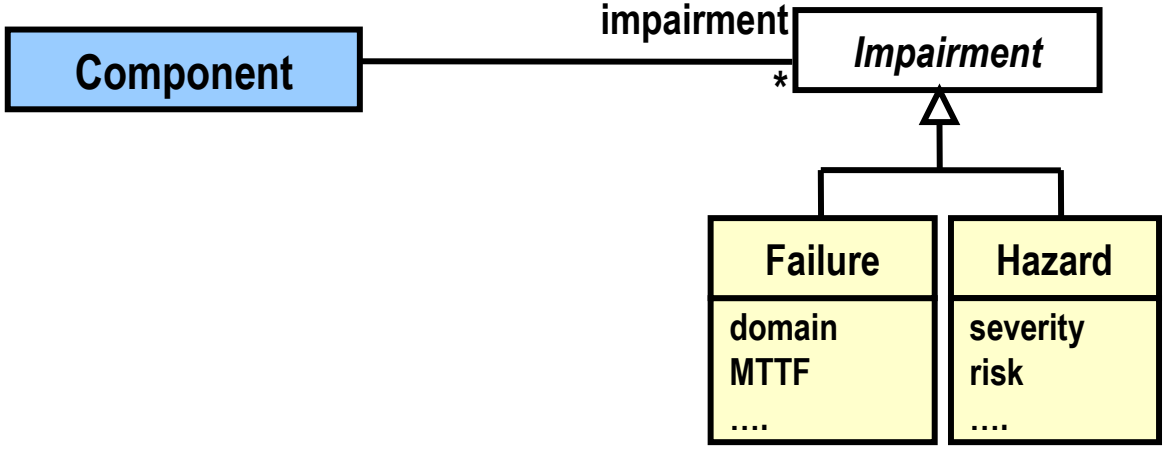
- Attributes have been mapped to either tags of stereotypes or to attributes of complex dependability types
  - For each attribute
    - A basic dependability type is associated/defined
    - A multiplicity is defined
- For associations, the “reference association pattern” is applied

Tag/attribute definition (I)

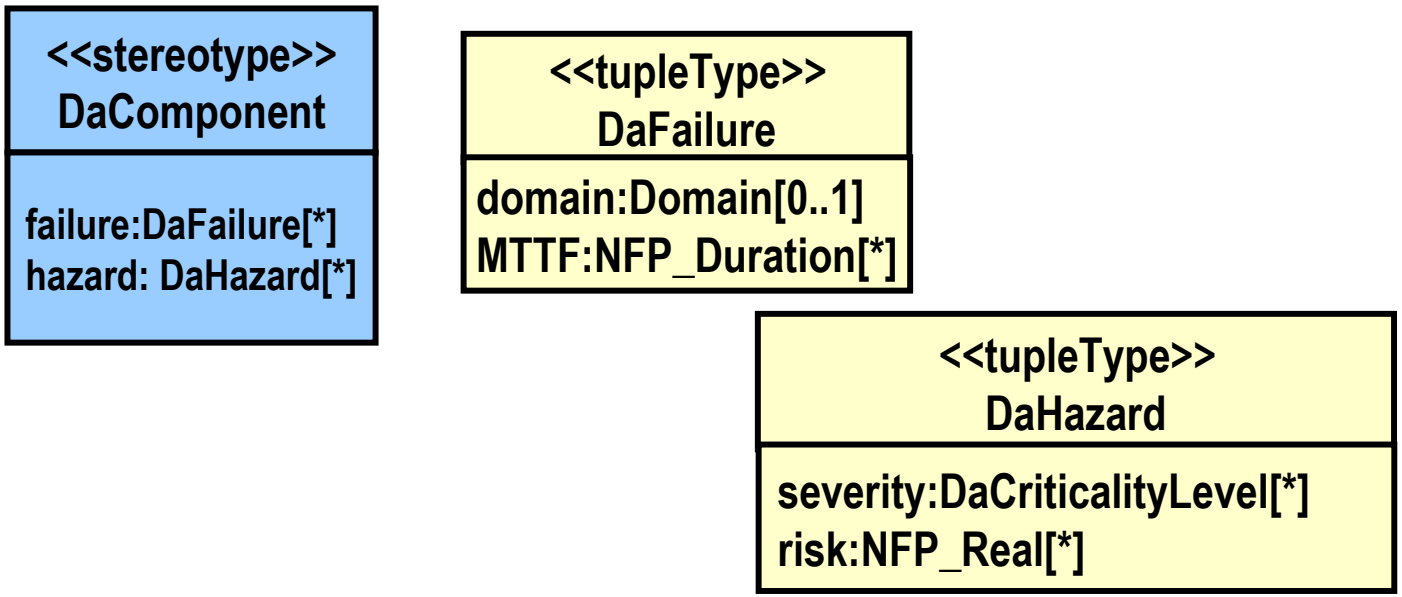


Tag/attribute definition (II)

DAM Conceptual Model



DAM profile



Usage of the DAM profile

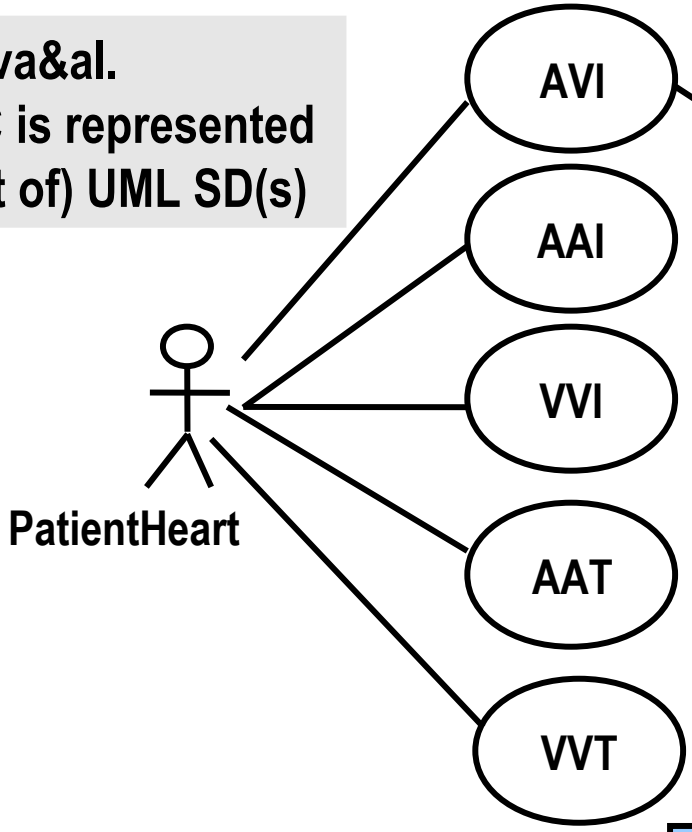
- Normal way of usage
  - At model spec level, the analyst may apply a DAM stereotype provided that the target model element belongs to a meta-class *extended* by that stereotype (e.g., *DaService* use case)
- Non trivial threat assumption specification
  - State-based failure conditions
  - Common-mode failures/hazards
  - Error propagation

Normal way of DAM usage

- Pacemaker example
  - From Goseva et al. “Architectural-Level Risk Analysis Using UML” TSE 29(10),2003
  - Where a methodology for safety risk assessment of UML based system models is presented
- No UML extensions were used by Goseva et al., NFP parameters were introduced in tabular form
- We use the DAM to annotate the UML model with NFPs

Use Case Diagram

In Goseva&al.
each UC is represented
by a (set of) UML SD(s)



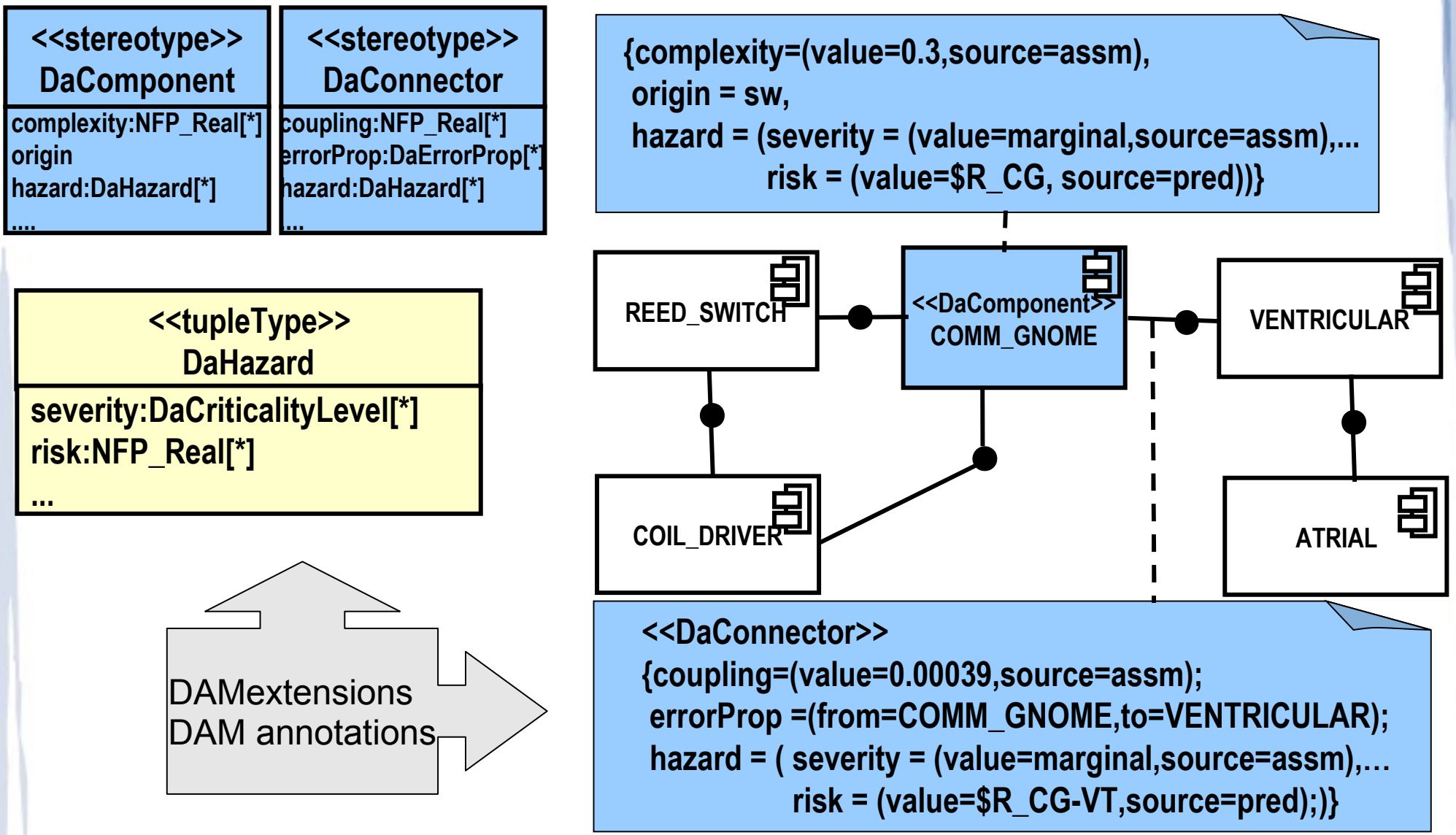
<<DaService>>
{execProb=(value=0.29,source=assm),
hazard = (risk=(value=$R_AVI, source=pred))}

DAM annotations
DAM extensions

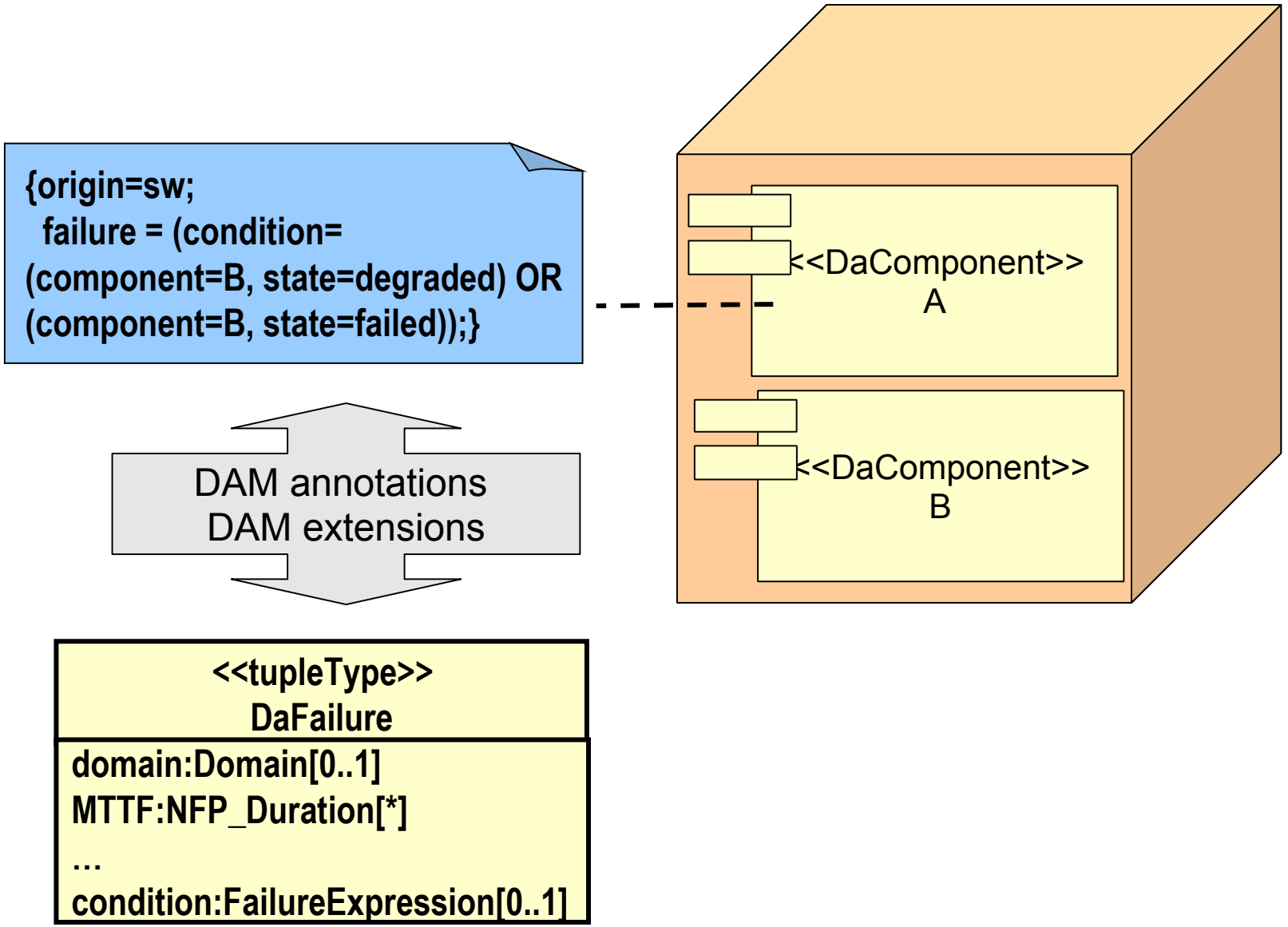
<<stereotype>> DaService
execProb:NFP_Real[*] hazard:DaHazard[*] ....

<<tupleType>> DaHazard
severity:DaCriticalityLevel[*] risk:NFP_Real[*] ...

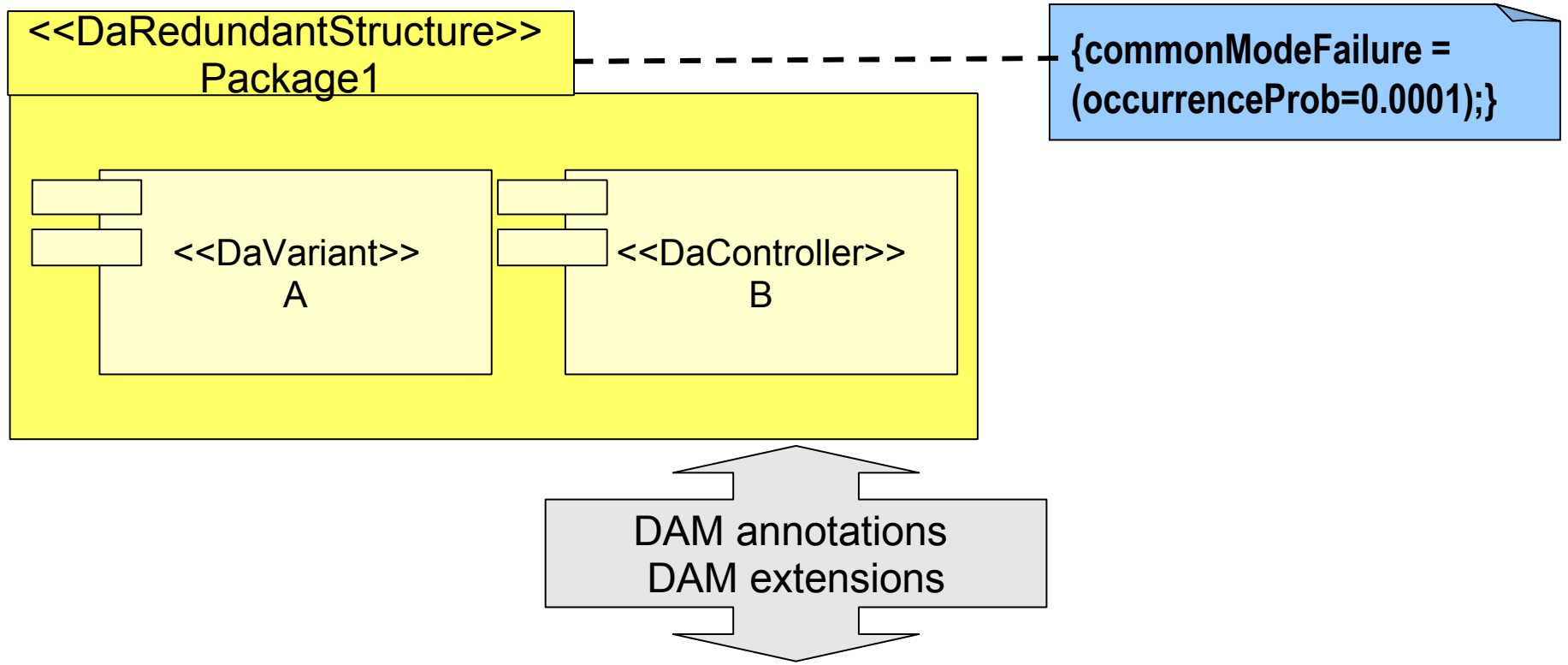
Pacemaker architecture



State-based failure conditions



Common-mode failure/hazard



DAM profile assessment

- Verification of the extensions w.r.t. the information requirement checklist (manual)
- Application of DAM to the examples in the literature and case studies
  - Production cell (Bondavalli et al.(1999)]
  - Mail system [D'Ambrogio et al.(2002)]
  - Pacemaker [Goseva et al. (2003)]
  - Elevator control system [Cortellessa et al.(2004)]
  - Message redundancy service [Bernardi et al.(2009)]
  - Intrusion tolerant firewall [Bernardi et al.(2009)]

On-going/future work

- Still assessing for completeness and consistency....
- Performability issues
- DAM within UP

Dependability requirement gathering in UP with the MARTE-DAM profile

PaCo Meeting, 25-26/06/09

UNITO Task: Development of a UML profile for dependability analysis

Simona Bernardi
UNITO

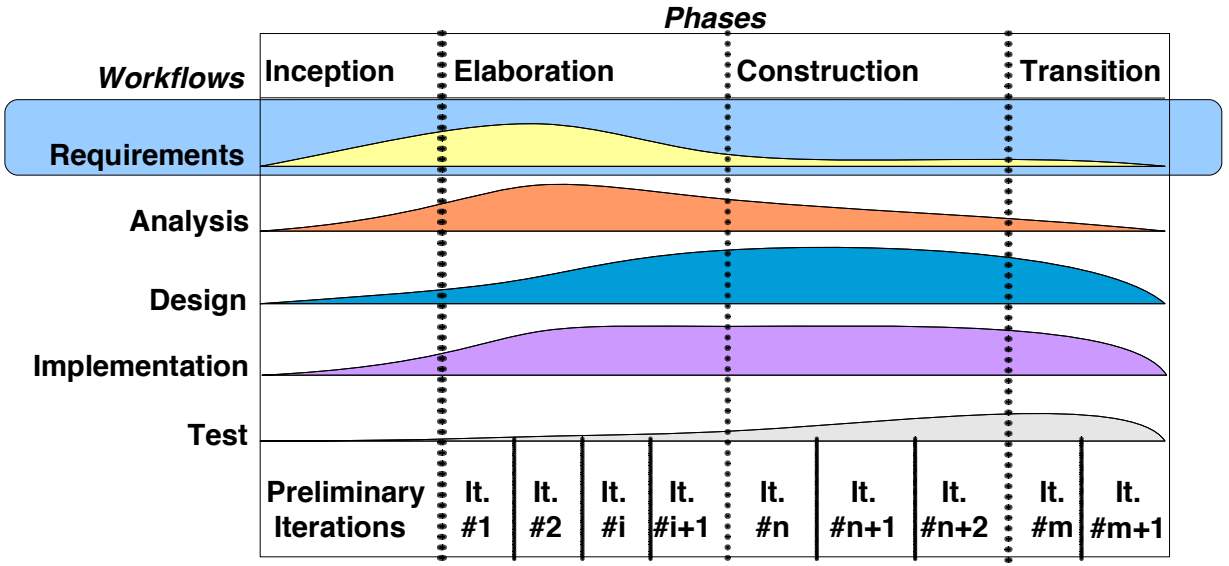
Recently completed works

- S.Bernardi, J. Merseguer, R.R.Lutz, Reliability and availability requirement engineering with UP and DAM profile. Submitted to ISSRE, 2009.

Outline

- Toward the definition of a methodology for the synergetic use of dependability techniques within the sw development process
- Why the Unified Process (UP) ?
  - Incremental & iterative: manages risks and handles changes in sw projects better than waterfall models
  - Uses UML as its specification language
  - Can be customized for different kind of sw systems/application domains
- UP pays little attention to non-functional reqs
- Several UML profiles exist that help to gather NFPs
  - DAM profile for dependability NFPs

Unified Process & req. workflow



System Analyst

Find actors & UCs

Structure UC model

Architect

Prioritize UCs

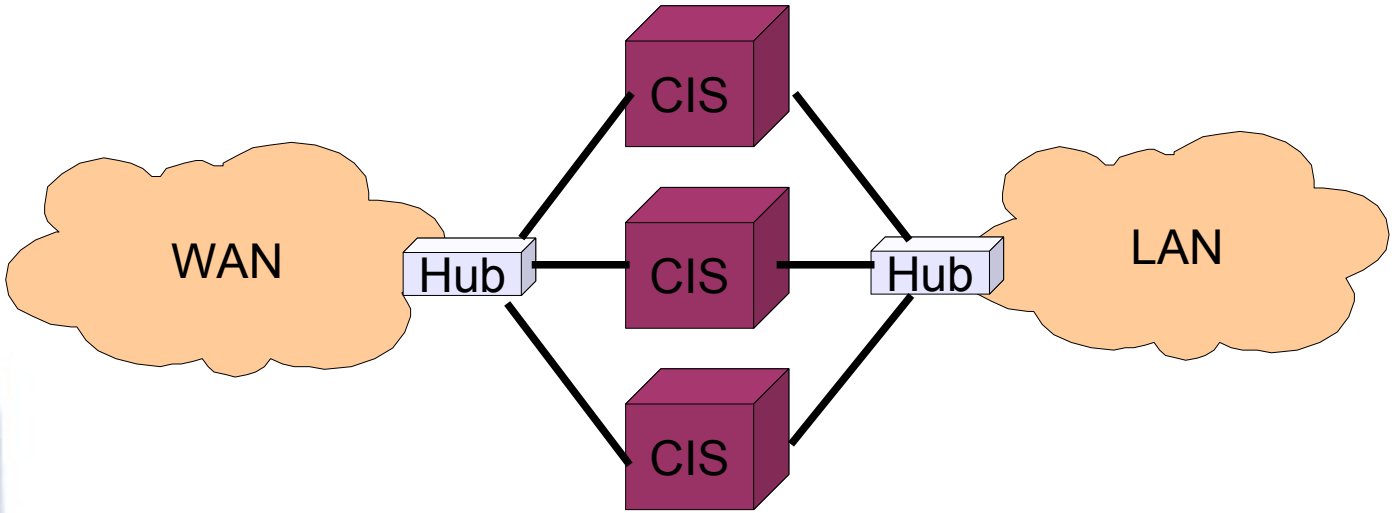
UC Specifier

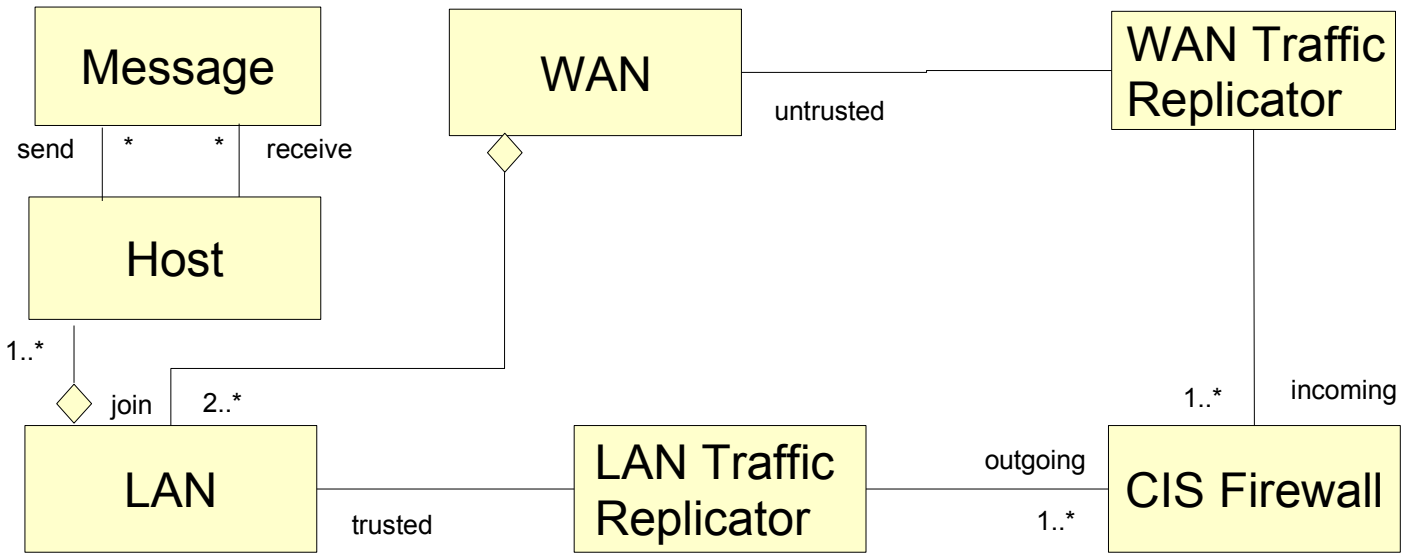
Detail UCs

UI Designer

Prototype UI

A running example from CRUTIAL project

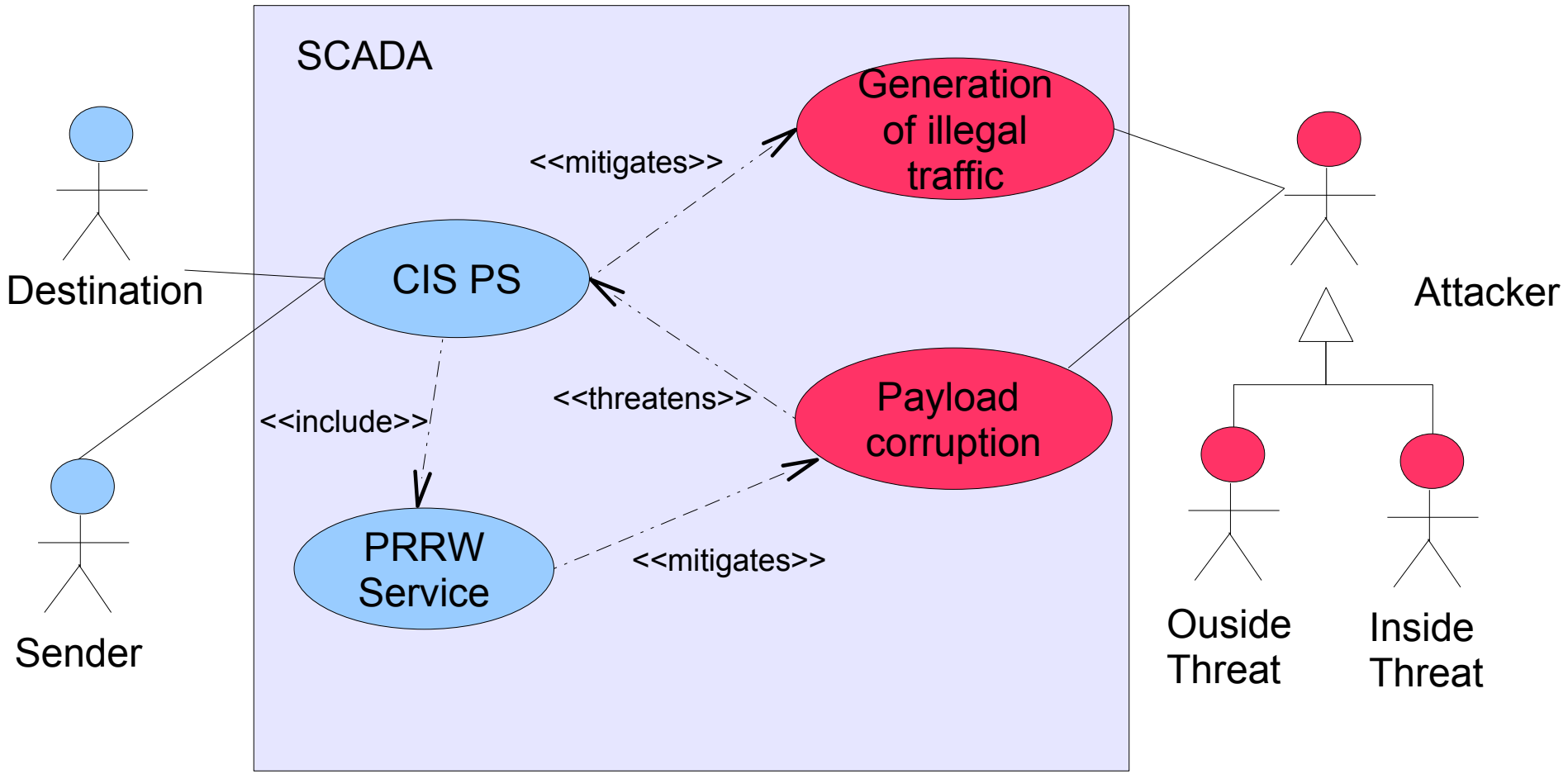




The set of dependability reqs specification techniques

- (Mis)Use cases
- IEEE Std. 830-1998
  - IEEE Recommended practise for sw requirements specification
- DAM profile
- Fault Trees

(Mis)Use Cases



- Use Cases are textual specifications
- Use of templates, like the Cockburn's one

IEEE 830-1998

- Recommends approaches for sw req specification and describes contents and qualities of a good SRS
- UP Supplementary Spec document inspired by IEEE 830-1998

Table of Contents

(a)

Template of "Specific requirements"

(b)

3.6 Other requirements:
(Fault Tolerance) There shall be at least 2f+1 CIS Firewalls to tolerate f concurrent faults

DAM profile

- DAM Profile has been devised to annotate the design, in this work we use it to specify dependability reqs.
- MARTE NFP types enable to describe relevant dependability aspect using properties:
  - Value: value/parameter name
  - Expr: VSL expression
  - Source: origin of the NFP (req,est,msr,assm)
  - StatQ: statistical qualifier (mean,min,max,..)

Fault Trees

- FTs are used to
  - Gather information about the potential contributing causes to threats
  - Trace the combination of faults/failures to use and misuse cases
  - Explore mitigating strategies for removing identified threats to dependability

Step-by-step process: ith iteration in the requirement workflow

Input: DMi-1,UCDi-1,SSi-1

Output: DMi,UCDi,SSi

1 Discover new UCs,MUCs and actors: UCDi ← UCDi-1 U UCnew U MUCnew U ACnew

2 Select UCs to be specified: selUCi ⊆ UDCi

3 Forall uc ∈ selUCi do

    1 Specify(uc)

4 Select MUCs related to selUCi: selMUCi ⊆ UDCi

5 Forall muc ∈ selMUCi do

    1 Specify(muc)

6 Discover new NFRs: SSi ← SSi-1 U NFRnew

7 Select a subset of requirements: selNFRi ⊆ SSi

8 Forall nfr ∈ selNFRi do

    1 Elaborate(nfr)

9 Restructure UCDi and DMi if necessary

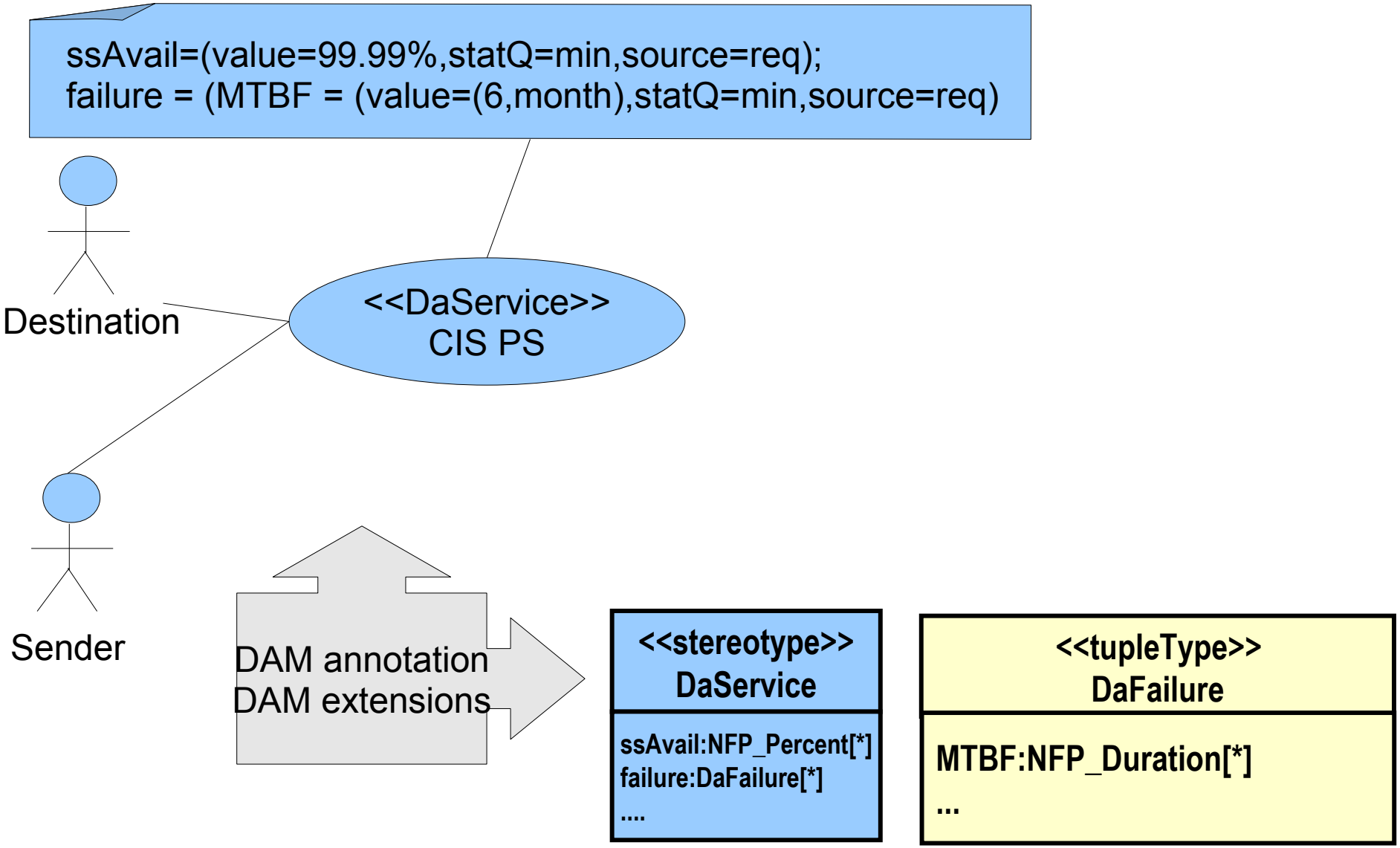
UC specify activity

- Textual description of the UC using Cockburn template
- Dependability reqs from the Special Requirement section
  - Application of DAM profile for rewriting them in a standard and disciplined form

CIS PS use case description

UC Name	CIS Protection Service
Scope	SCADA
Main Actors	Sender (computer from the WAN), Receiver (computer of the protected LAN)
Success guarantee	The correct message is eventually delivered The illegal message is not delivered
Main scenario	A message is sent by Sender to Receiver 1 It arrives to the CIS Firewall 2 Each CIS Firewall checks if it satisfies the security policy and votes 3 The CIS firewalls agree upon a final judgement (majority voting) 4 The message is correct and the CIS Firewall leader forwards it to the Receiver
Alternate scenarios	4.a The message is illegal, then it is not delivered
Special Reqs	A1. The CIS PS should be available 99.99% of the time R1. The MTBF shall be at least 6 months
Relationships	CIS includes PRRW Service, Payload Corruption threatens CIS PS, CIS PS mitigates Generation of illegal traffic

DAM annotation to CIS PS use case



MUC specify activity

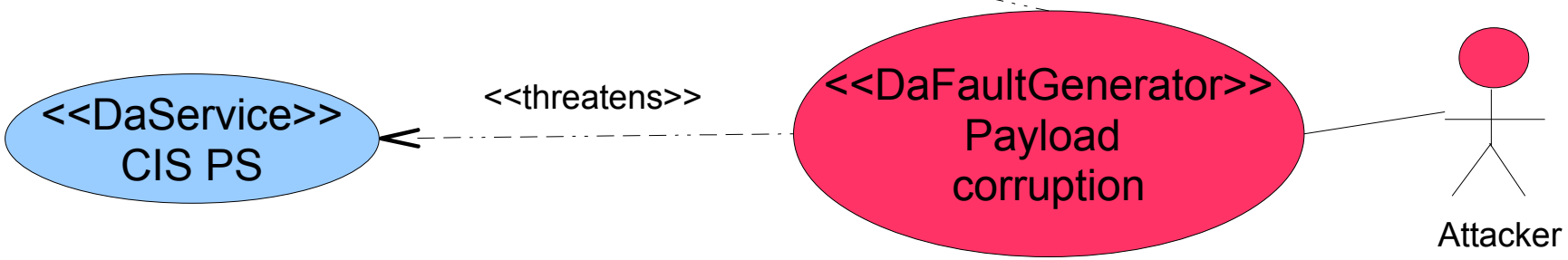
- Textual description of the MUC using Cockburn template
- Threats information from Success guarantee, Main/Alternate scenario and Other Reqs sections
  - Application of the DAM profile to characterize from both a qualitative/quantitative viewpoints faults/failures
- Faults Trees are used to formally specify UCD relationships
  - Among Negative Actor actions and Misuse Case success
  - Among Misuse Cases and related Use Case

Payload Corruption MUC description

MUC Name	Payload Corruption
Scope	CIS PS
Main Actors	Attacker: Outside and Inside Threats
Success guarantee	The Payload evaluates as “correct” an illegal message or it evaluate as “illegal” a correct message (FM1), or it is subject to a temporary omission (FM2)
Main Scenario (Outside Threat)	The Attacker identifies the WAN traffic replicator as potential target 1 The Attacker sniffs the network traffic 2 The Attacker gets an unauthorized access to an host in the LAN 3 The Attacker install a *malicious logics* in the accessed host 4 The hosted Payload behaves in an unpredicted manner.
Special Reqs	F1. At most f Payloads can be concurrently corrupted F2. f should be se according to the expected rate of fault occurrence
Relationships	Payload Corruption threatens CIS PS

DAM annotation to Payload Corruption MUC

```
numberOfFaults=(value=$f,statQ=max,source=est/msr);
fault = (type = (value=malicious-logic);
         occurrenceRate = (value=$fr1,statQ=mean,source=est/msr);
         effect = (domain = (value=invalid,omission)));
```



DAM annotation
DAM extensions

<<stereotype>>
DaFaultGenerator

numerOfFaults:NFP_Integer[*]
fault:DaFault

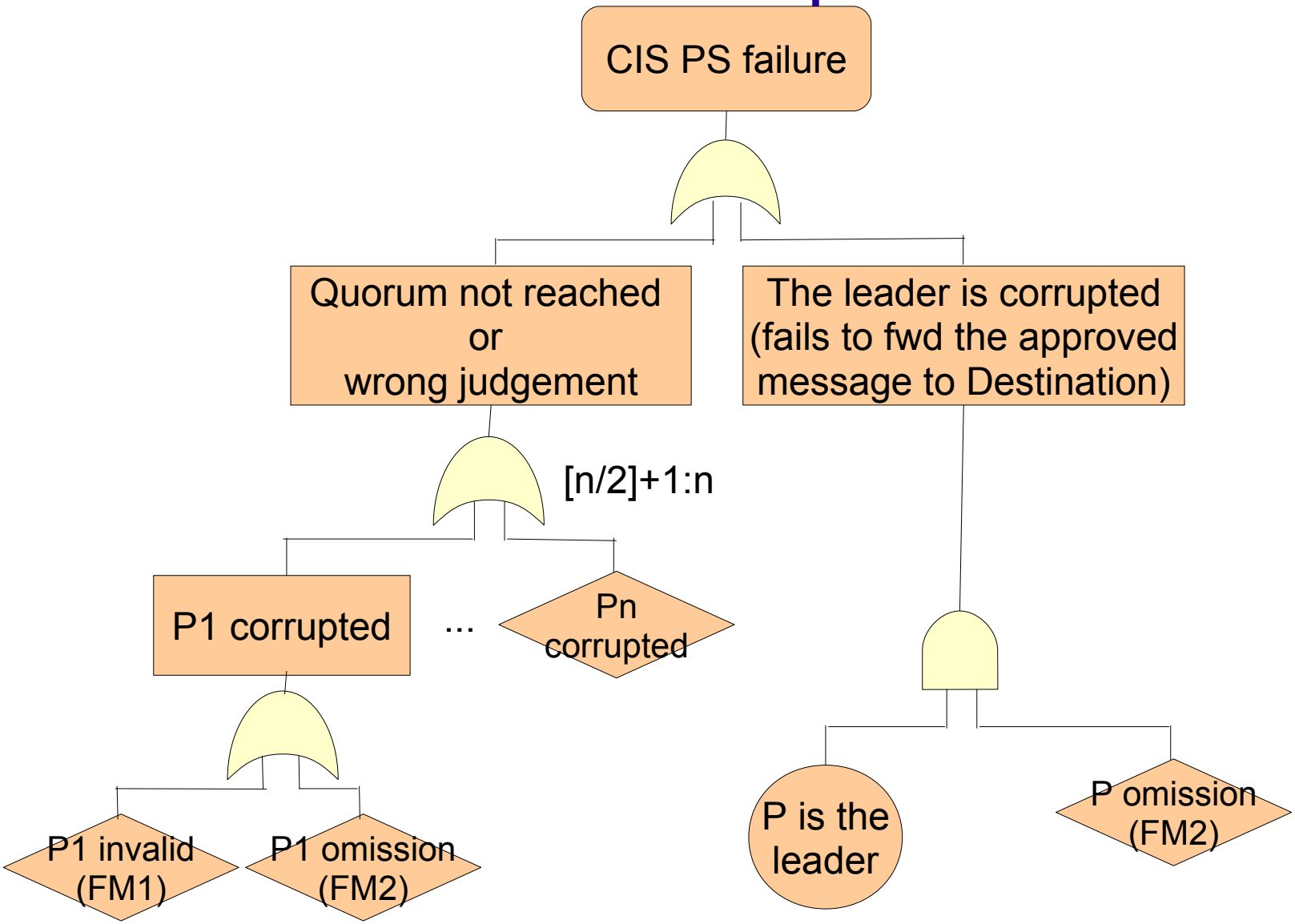
<<tupleType>>
DaFault

type:FaultType[*]
occurrenceRate:NFP_Frequency[*]
effect: DaFailure[*]

<<tupleType>>
DaFailure

domain:Domain[*]
...

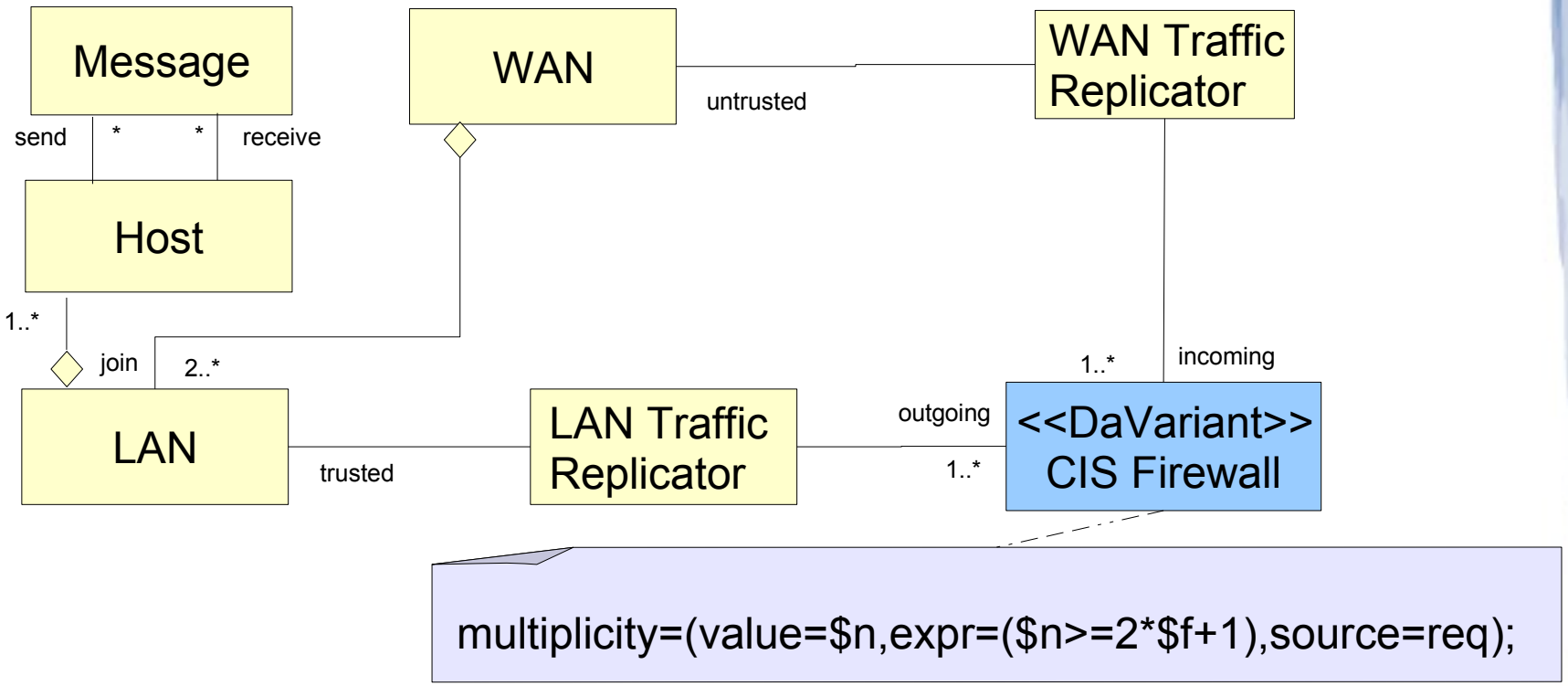
Use of FT to formalize MUC-UC relationships



NFR elaboration activity

- Rewriting of further NFR from the SS, related to dependability/fault-tolerance with the DAM profile
  - Annotation in the Domain Model/Use Case Diagrams

DAM annotation to the CIS Firewall Domain Model



3.6 Other requirements:
(Fault Tolerance) There shall be at least 2f+1 CIS Firewalls to tolerate f concurrent faults

Conclusions

- The DAM annotated UML artifacts (UCD,DM) provide input for the other UP workflows (design,test,..) as well as for V&V activities
- Next steps:
  - Study of the DAM applicability in the other UP workflows
  - V&V activities driven by DAM annotated M(UC)s