

Tool-Supported Dependability Evaluation of Redundant Architectures in Computer-Based Control Systems

István Majzik, Péter Domokos, Melinda Magyar
Budapest University of Technology and Economics
Dept. of Measurement and Information Systems

FORMS / FORMAT 2007
Braunschweig, January 25-26, 2007.

Introduction of the Speaker



István Majzik

Associate Professor

Dept. of Measurement and Information Systems,
Budapest University of Technology and Economics,
Budapest, Hungary

Fields of interest

Dependability modelling and evaluation

Verification and validation of software

Experience, projects

Development and testing of dependable and safety-critical systems

Software assessment (EN 50128)

SAFEDMI – Safe Driver Machine Interface for ERTMS
Automatic Train Control (FP6-2005-Transport-4)

Motivation

- **Architectural choices** have profound influence on system dependability (reliability, availability)
 - Degree and type of redundancy (fault tolerance)
- **Standards** require a thorough evaluation of possible failures and protection mechanisms
 - Quantitative evaluation: Computation of system level measures using component-level reliability parameters
- **Model-based dependability evaluation**
 - (Formal) dependability model is constructed
 - Component failure and repair (recovery) behaviour is modelled
 - Allows „what-if” kind of analysis in early design phases
 - Optimization of architectural choices (decisions)

Dependability modelling approach*

- Formalisms for dependability models
 - Combinatorial models (e.g. fault trees)
 - Stochastic state space models (CTMC, GSPN)
allow to capture dependencies between components
- Design models shall be supported
 - Construction of dependability models automatically
 - Assembling the state space of the model
taking into account failure states and repair processes
 - Integrating the expert knowledge in a tool
- UML: formalism of the design model,
GSPN: formalism of the dependability model

* I. Majzik, A. Pataricza and A. Bondavalli: Stochastic Dependability Analysis of System Architecture Based on UML Models. In „Architecting Dependable Systems”, LNCS 2677, Springer, 2003.

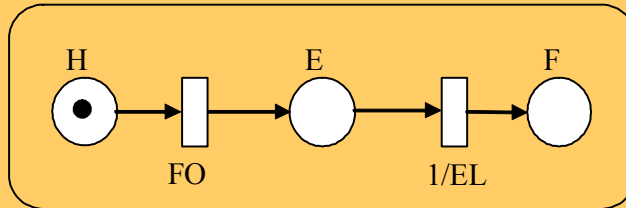
Dependability model construction

- Dependability model represents
 - Fault occurrences in components
 - Error propagations between components
 - Repair (maintenance) mechanisms
- **Component types** are assigned **GSPN** subnets that represent these processes
 - Hardware, software, stateful, stateless components are distinguished
 - Component types and related local dependability parameters can be identified in the design model (UML stereotypes and tagged values)

Dependability model construction

- **De**

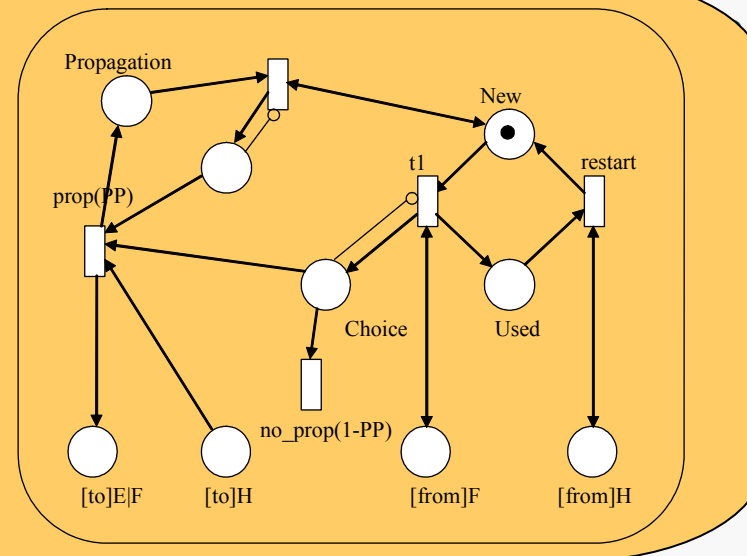
Failure subnet of a stateful hardware



- **Component types are assigned GSPN subnets that represent these processes**

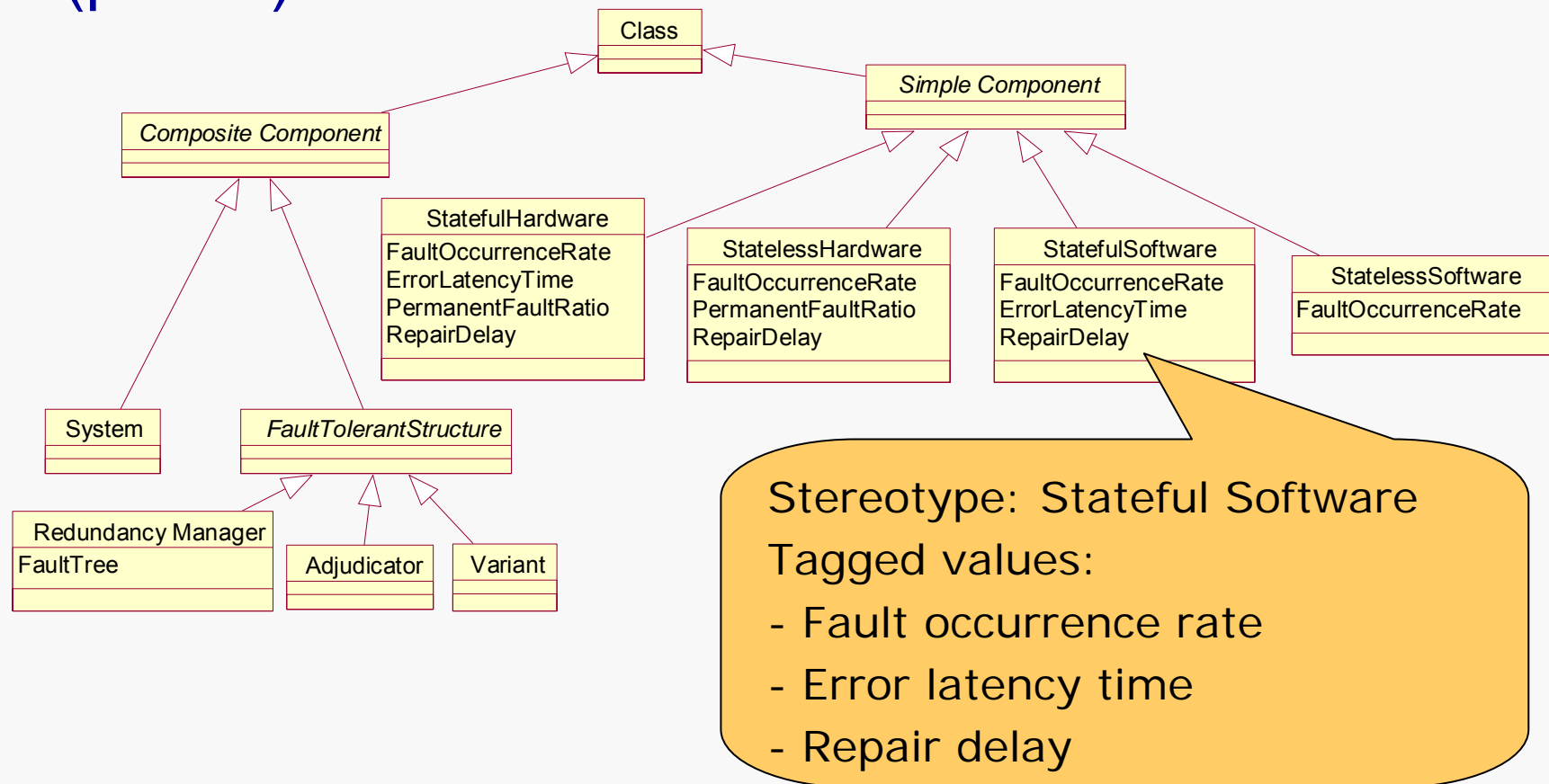
- Hardware are distinct
- Component parameters (UML ste

Generic error propagation subnet between components



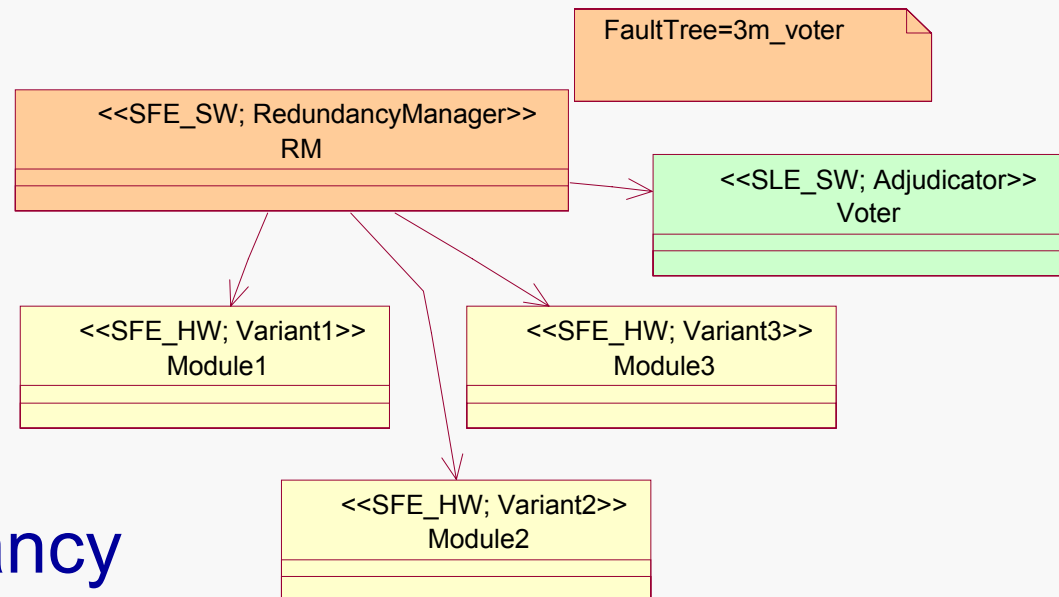
Extensions of the UML design model

- Identify component types and local parameters (profile)



Modelling redundancy

- Identification of roles
 - Redundancy manager
 - Variant
 - Adjudicator

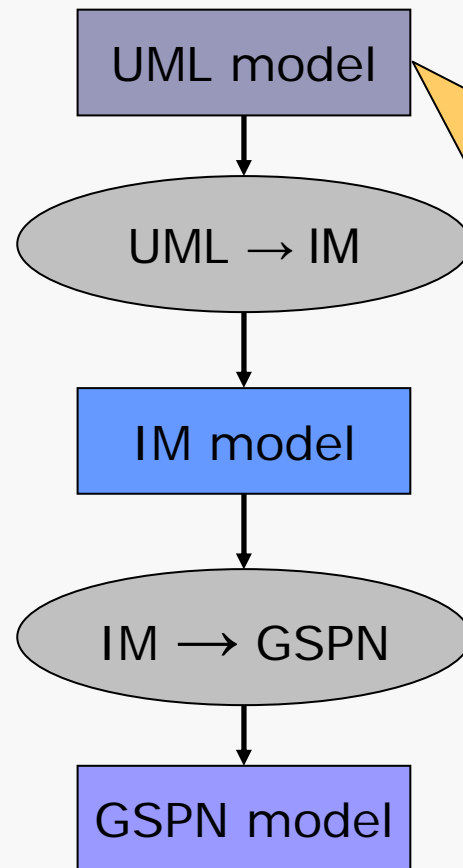


- „Logic” of redundancy (error propagation)
 - Fault tree
 - Specific GSPN subnet

Tool support for dependability modelling

- Integration of subnets assigned to components
 - Subnets are constructed by dependability experts
- Re-use of dependability subnets from a library
 - Assigned to common **redundancy management** (TMR, NMR, NVP, etc.)
 - Assigned to **architectural design patterns** (optionally handled as aspect models)
- Refinement of dependability subnets
 - Early phases of design: **Generic** subnets
 - Design refinements: **Refined** subnets
They can be **transformed from behavioural models**:
E.g. statechart of the redundancy manager → Fault tree

Tool architecture



System model

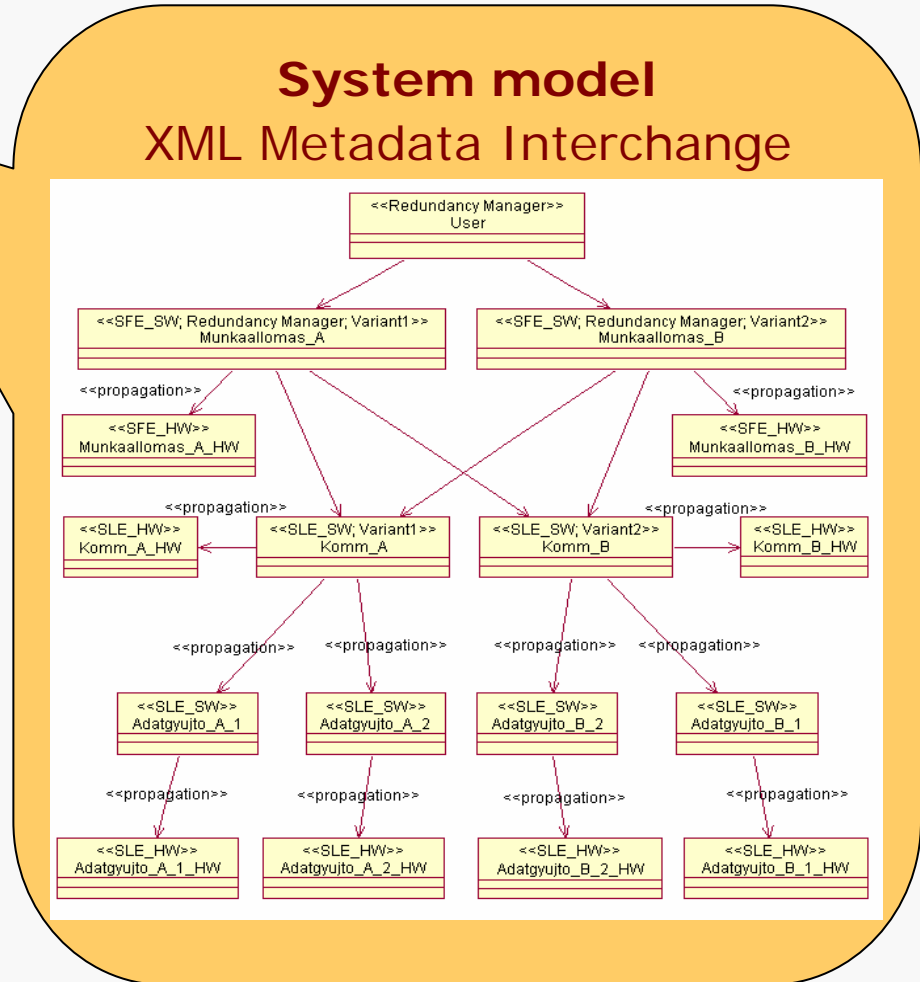
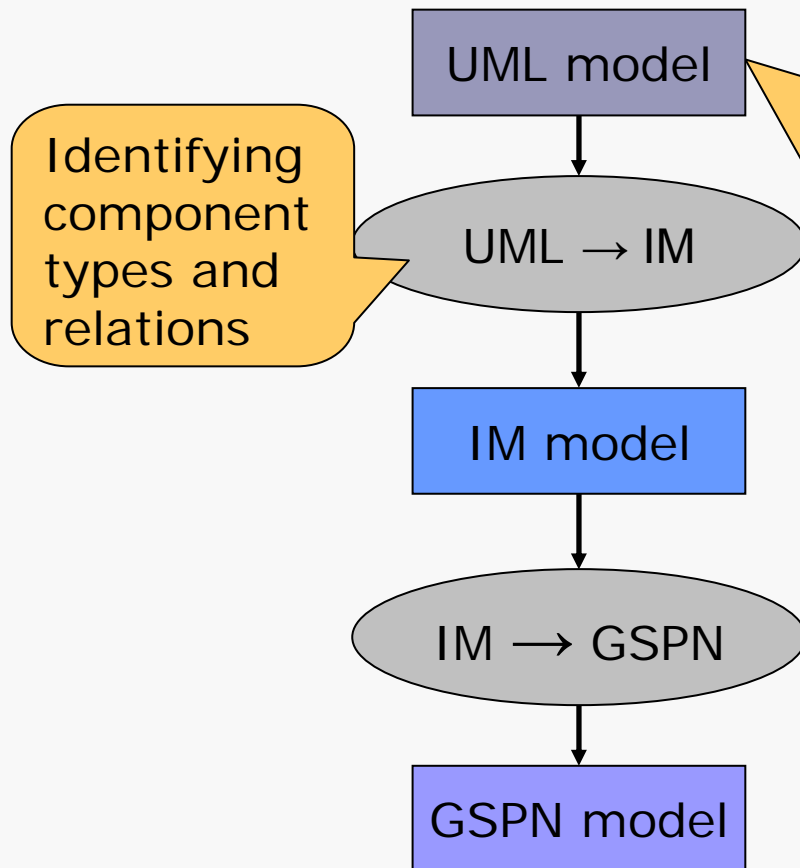
UML architecture diagrams:

- classes and objects
- associations / relations, deployment (sw on hw)
- packages (subsystems)

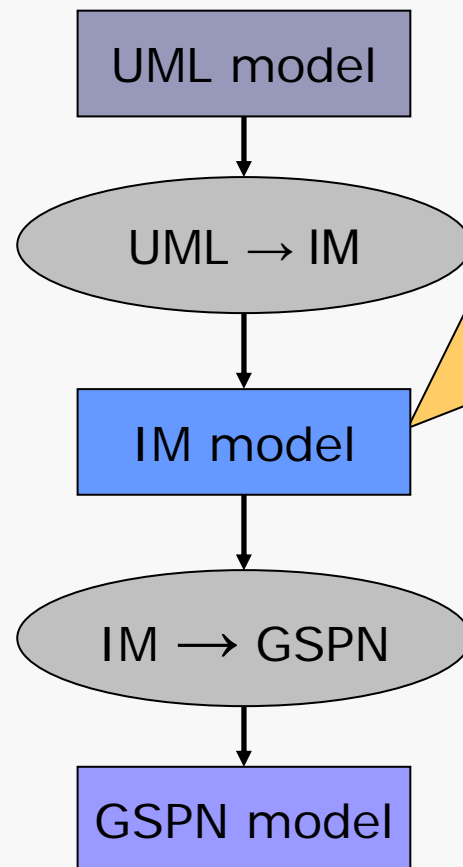
Extensions:

- Component **type** from dependability viewpoint (stateless, hardware etc.)
- Numerical **attributes** (local failure rate, repair rate, propagation probability etc.)

Tool architecture



Tool architecture



Dependability model

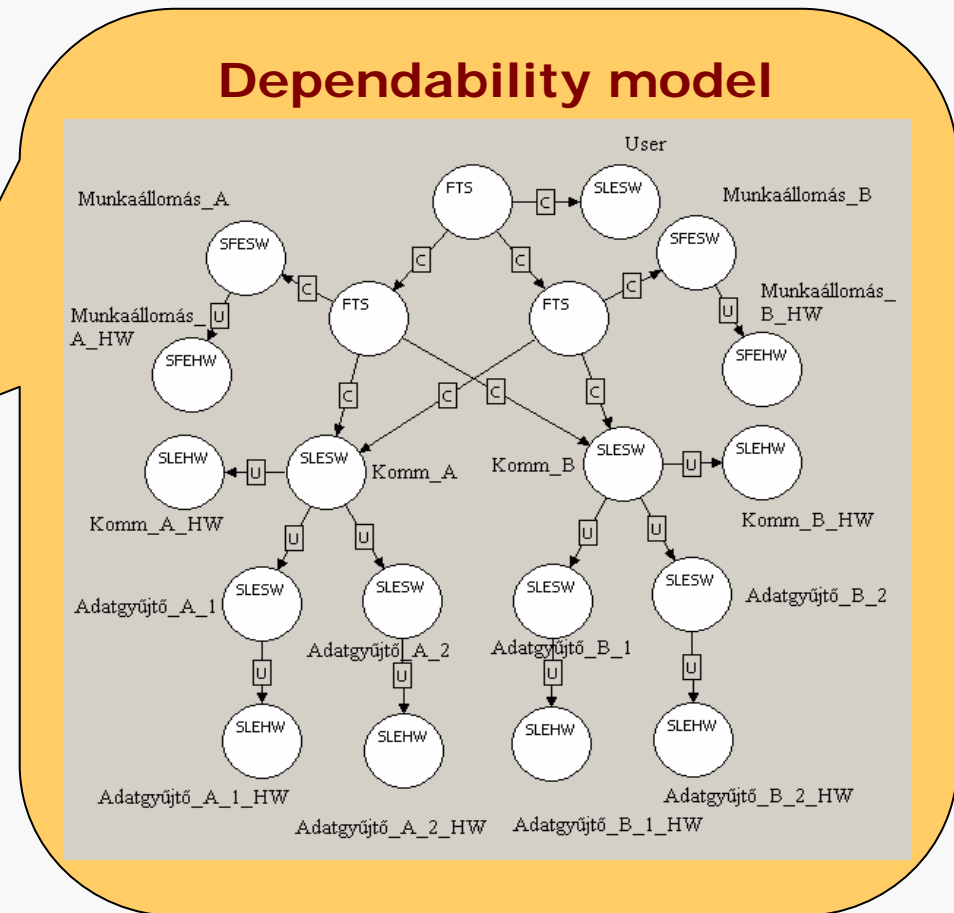
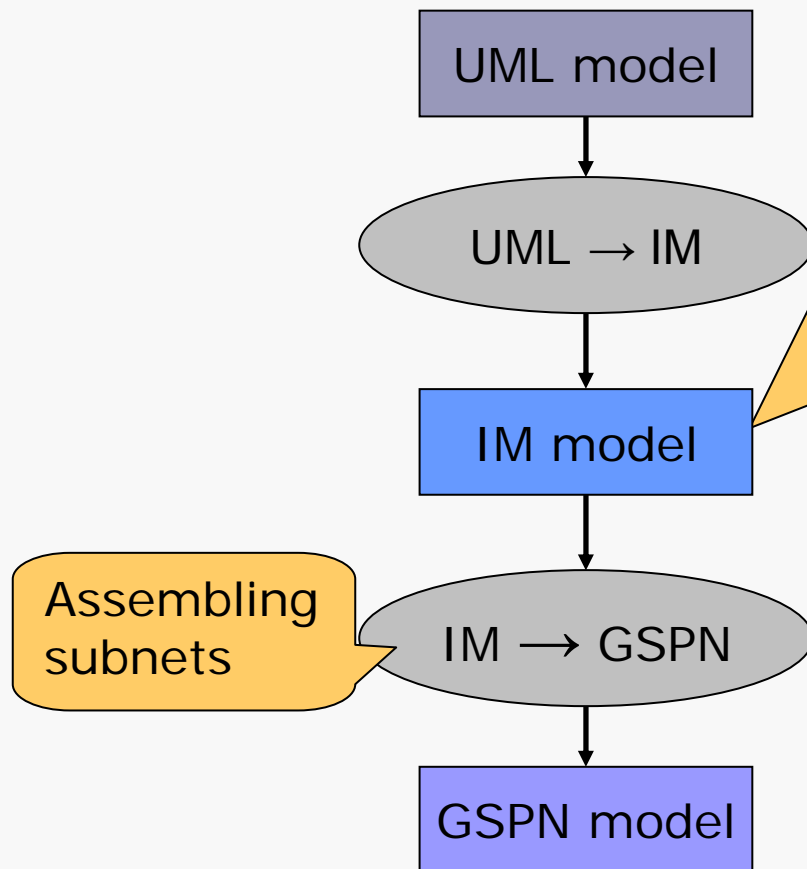
Elements:

- **components:** local failure / repair characteristics
- **subsystems:** measures

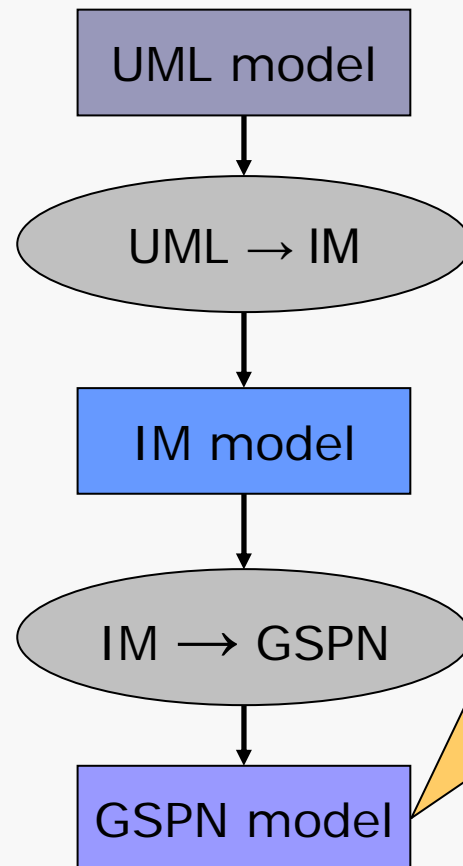
Relations:

- **component uses component:** error / repair propagation
- **subsystem is composed of (redundant) components:** (non-trivial) error propagation
- **system is composed of sub-systems:** error propagation

Tool architecture



Tool architecture



Analysis model

System-level GSPN:

composed of modular **subnets**

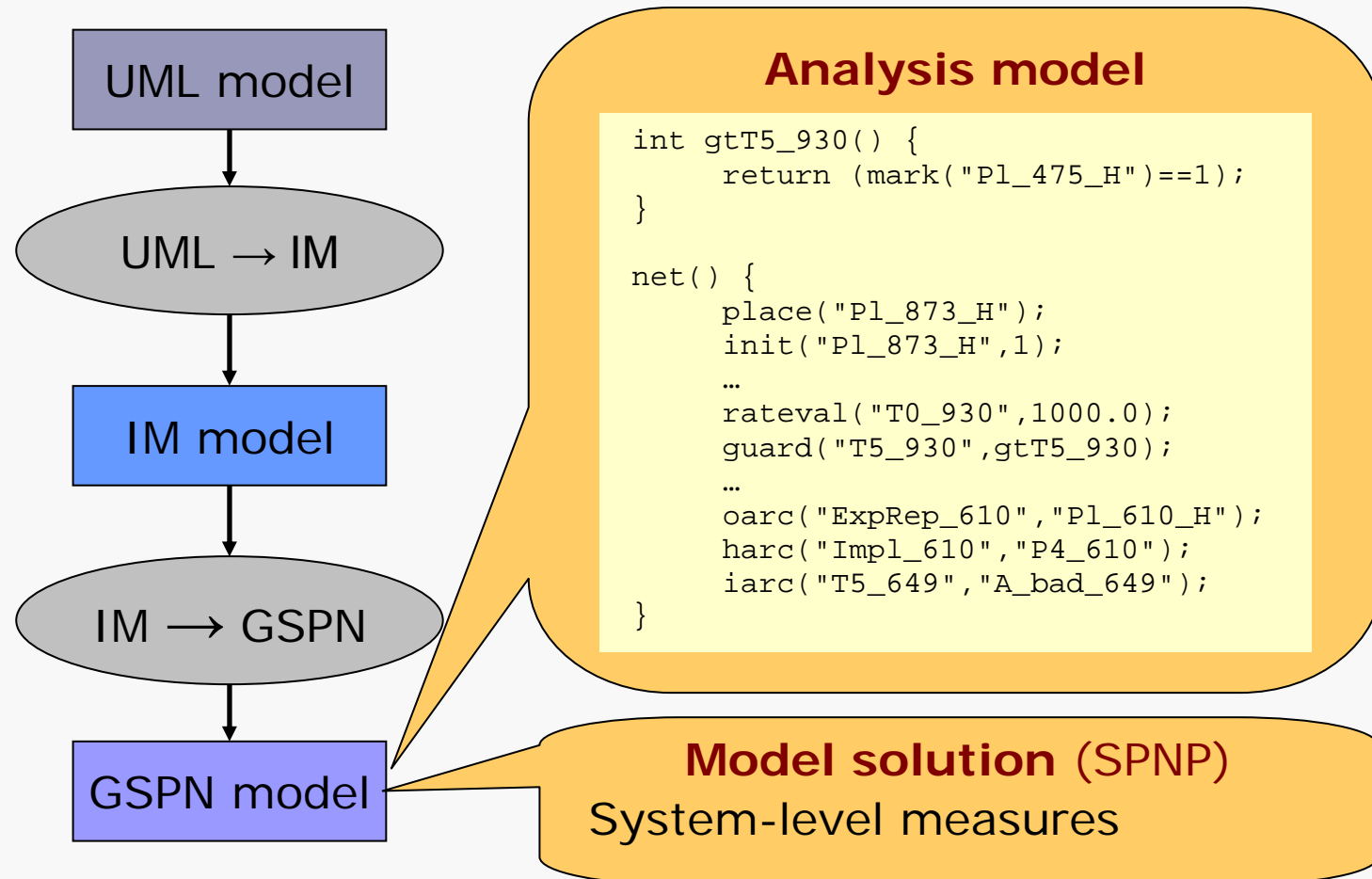
Component subnets:

- „local” failure/repair subnets

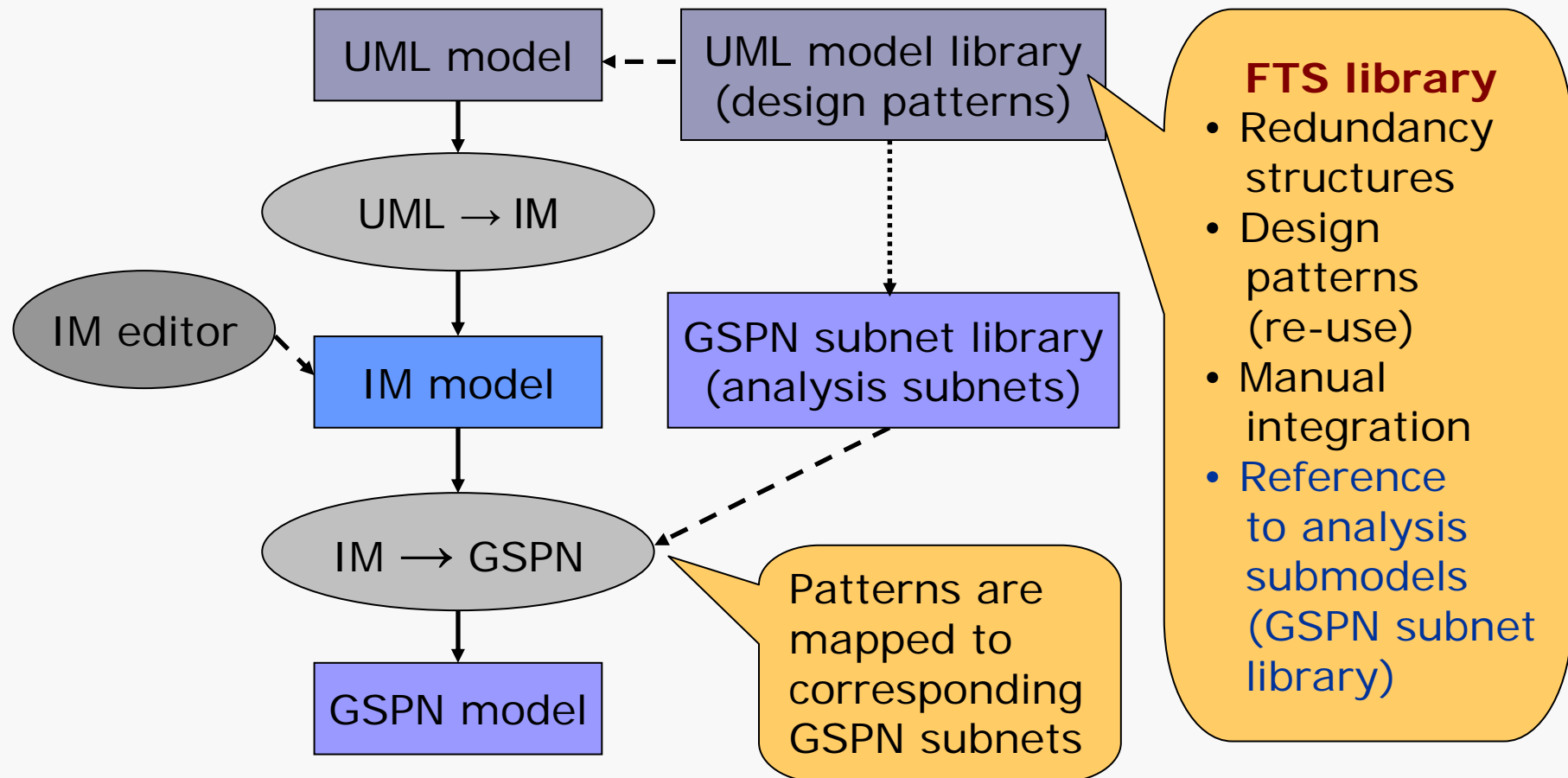
Relation subnets:

- **simple** error propagation (uses relation)
- **non-trivial** error propagation (composed-of relation, FT)
- **repair** propagation

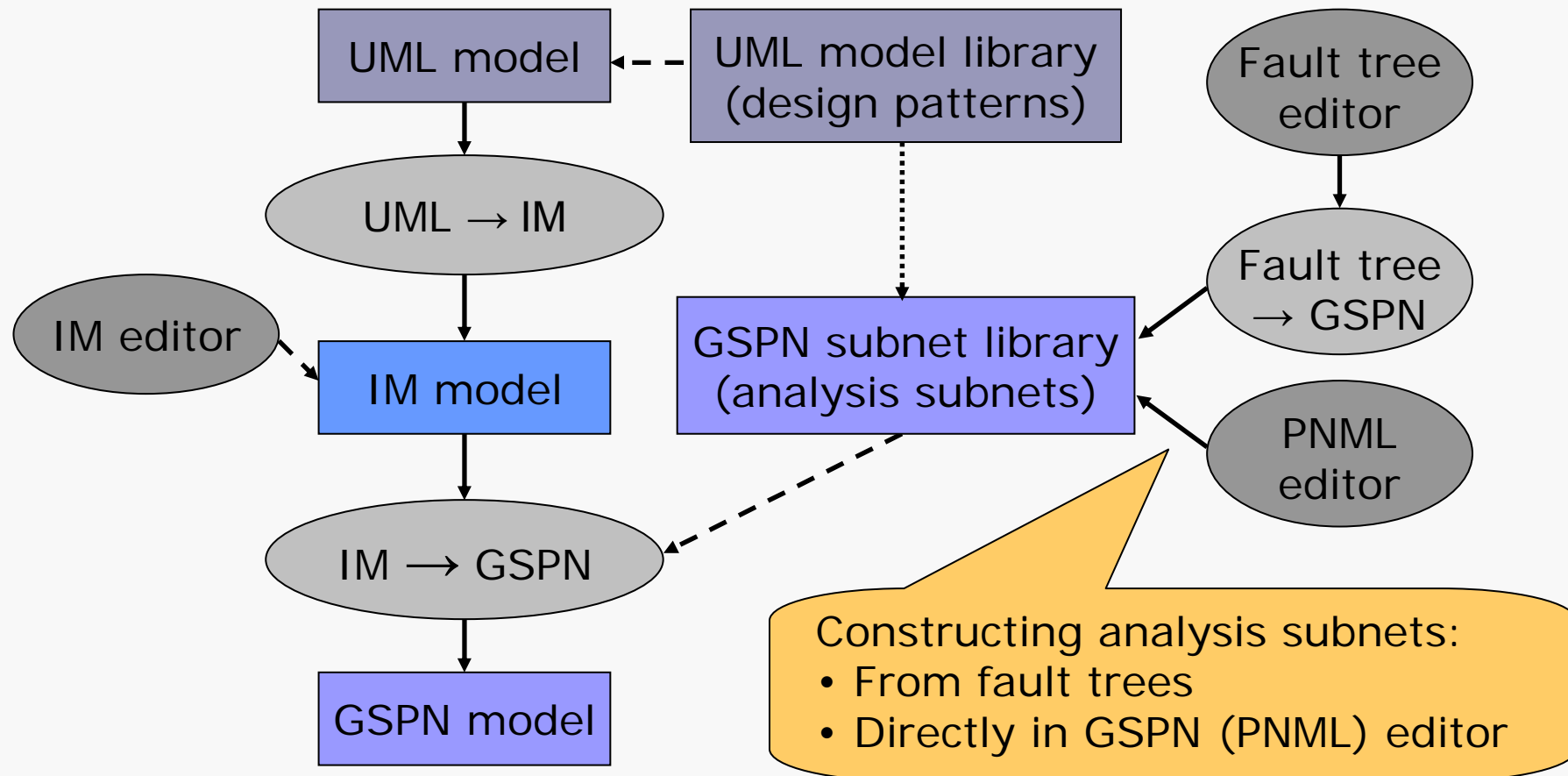
Tool architecture



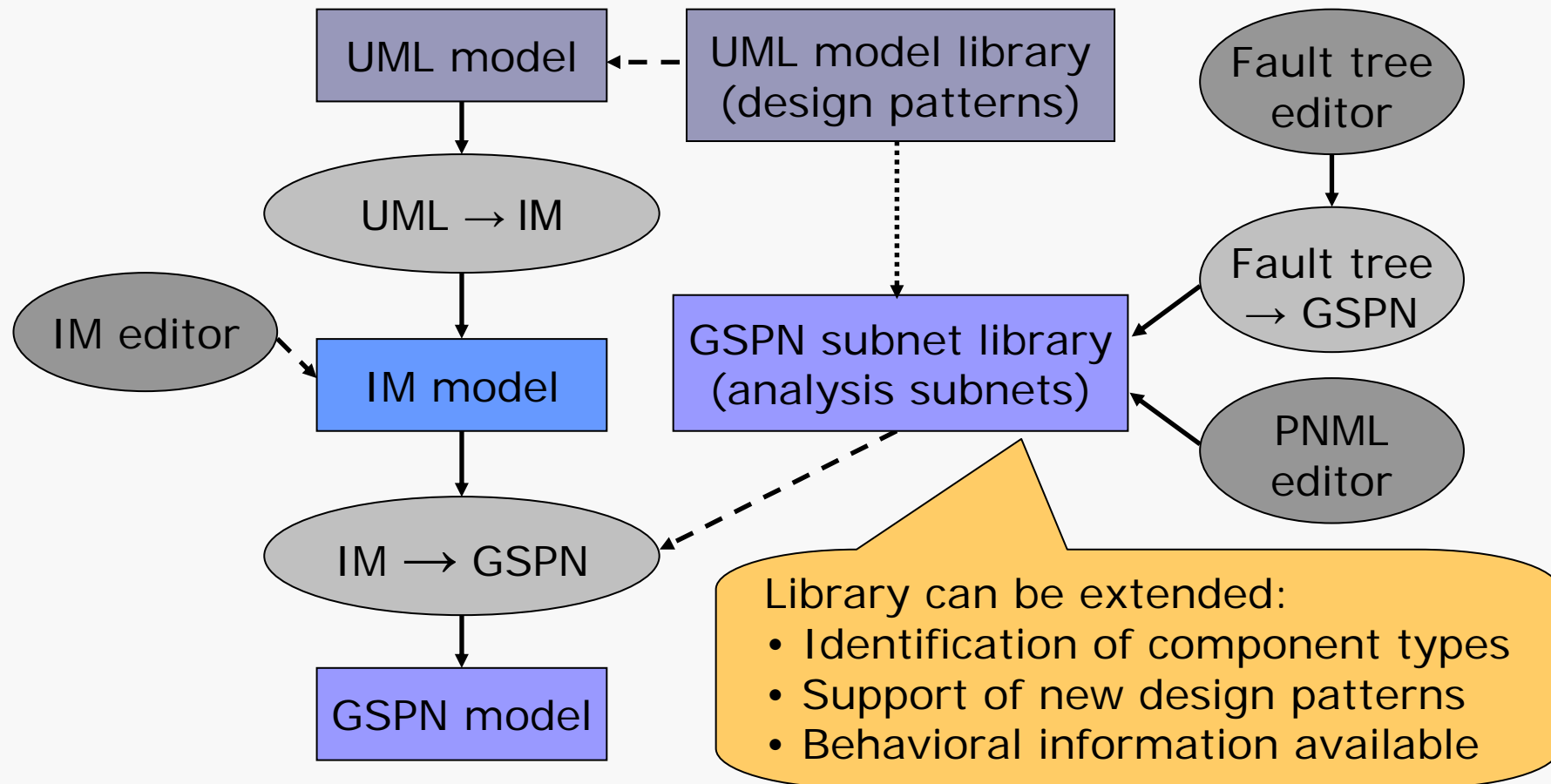
Tool architecture



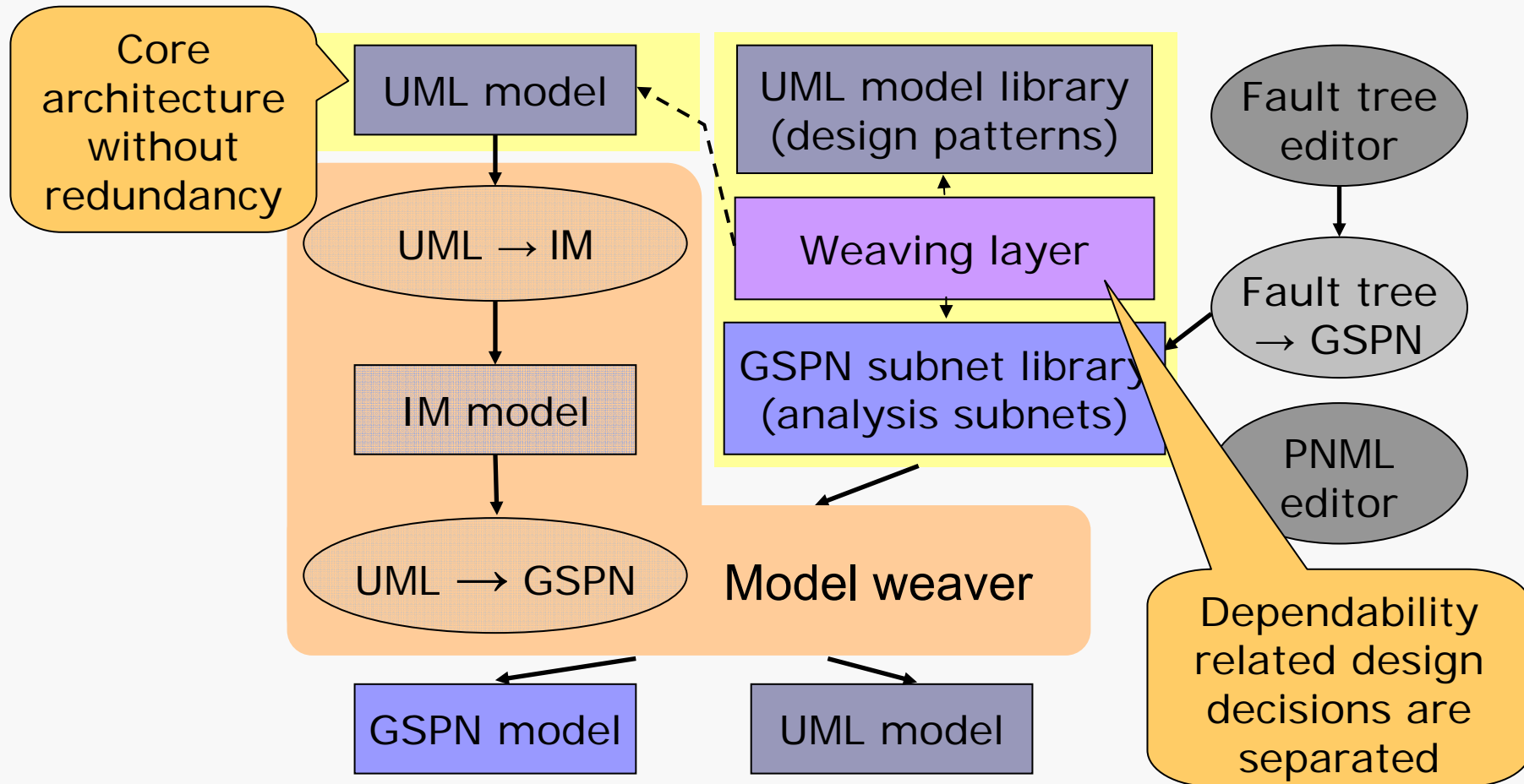
Tool architecture



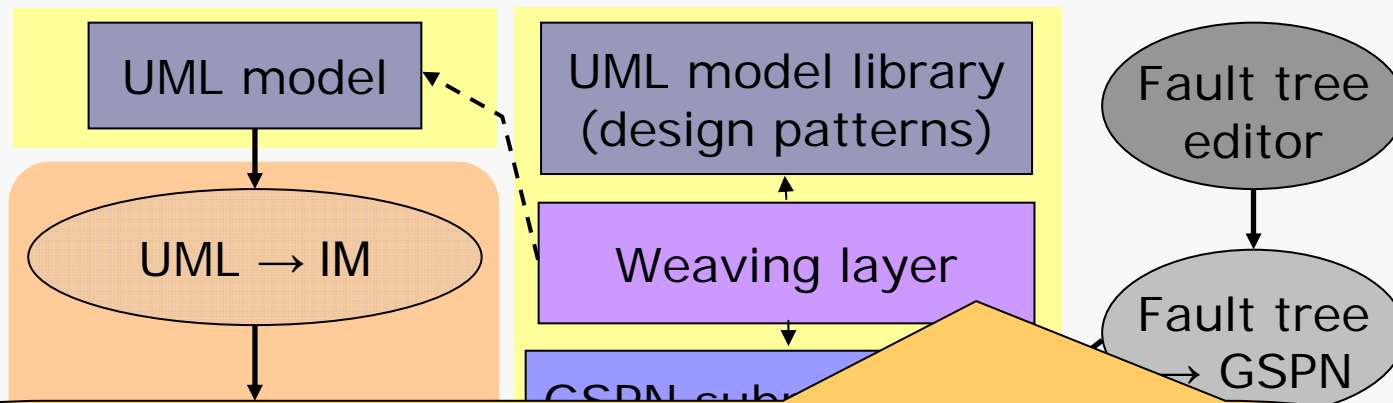
Tool architecture



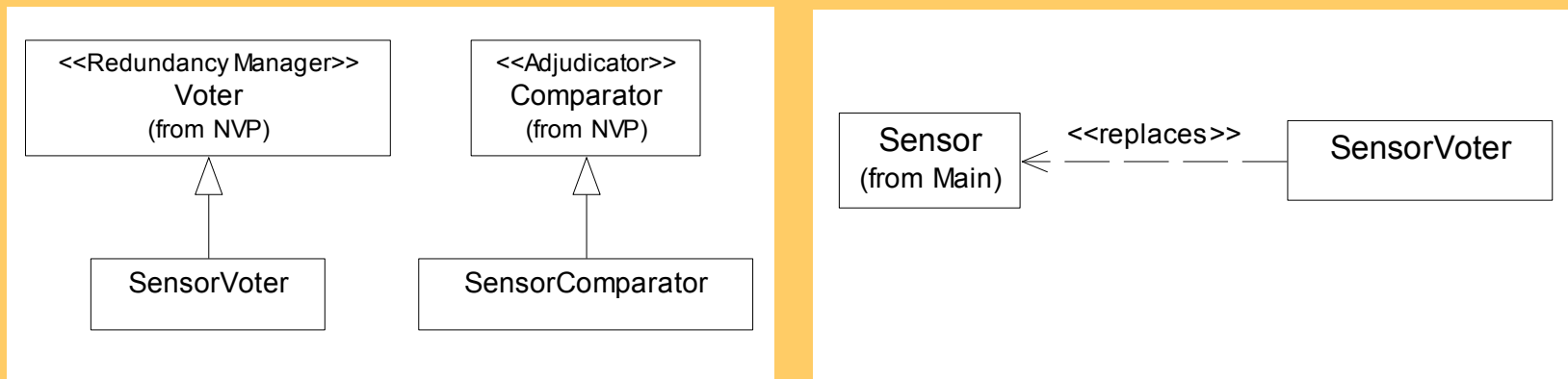
Aspect-oriented modelling approach



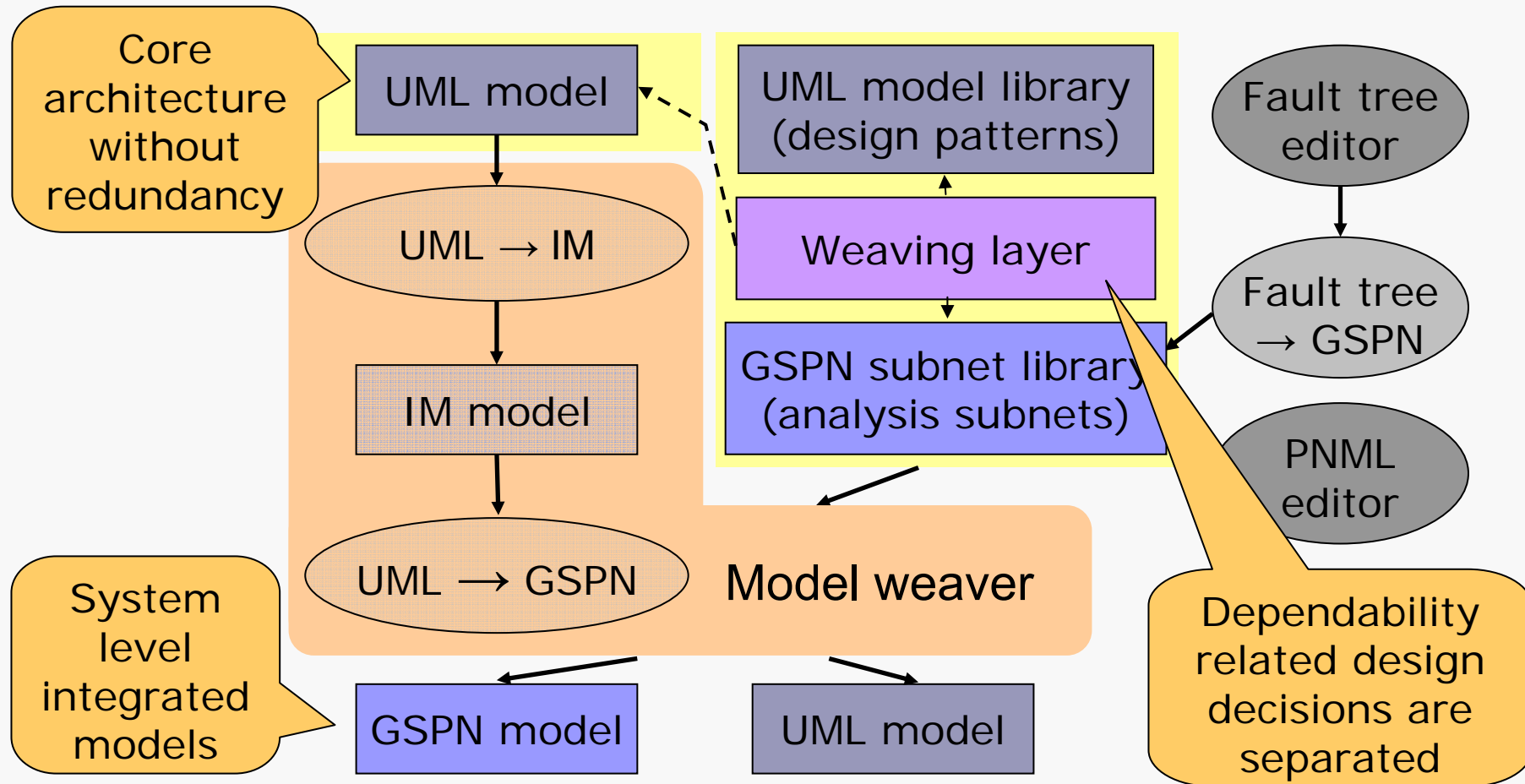
Aspect-oriented modelling approach



Where and how to apply redundancy: Instantiating components from design patterns and specifying links to the core UML model



Aspect-oriented modelling approach



Summary

- Method to construct GSPN dependability models
 - External GSPN solver → System level availability
- Adaptability to different input models
 - UML, AADL (in progress)
 - IM is the core mathematical formalism
- Extensibility: Subnet library for components
 - Specialisation, design refinement
- Aspect-oriented modelling of redundancy
 - Separation of design decisions related to fault tolerance
 - Weaving of design and analysis models