An observation metamodel for dependability tools

Laura Carnevali Stefania Cerboni Benedetta Picano
Leonardo Scommegna Enrico Vicario

Dept. of Information Engineering, University of Florence

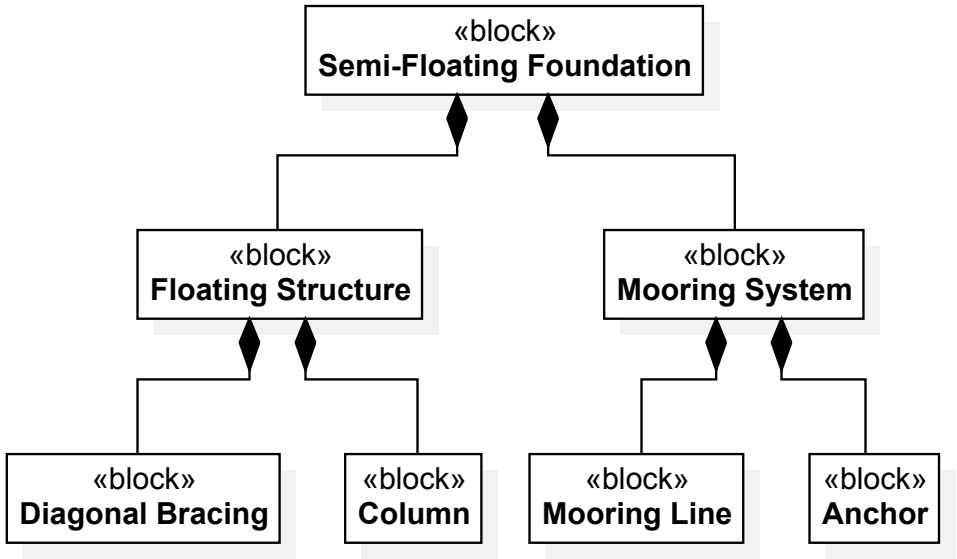
Software Technologies Lab - https://stlab.dinfo.unifi.it

{*laura.carnevali, stefania.cerboni, benedetta.picano, leonardo.scommegna, enrico.vicario*}*@unifi.it*

EDCC'24
Leuven, April 2024

- this is about:
  - Component-Based systems subject to failures over time
  - Propagation of faults among system components
  - Extention of FaultFlow Java Library to represent observations
  - . . . making it possible to test various dependability strategies

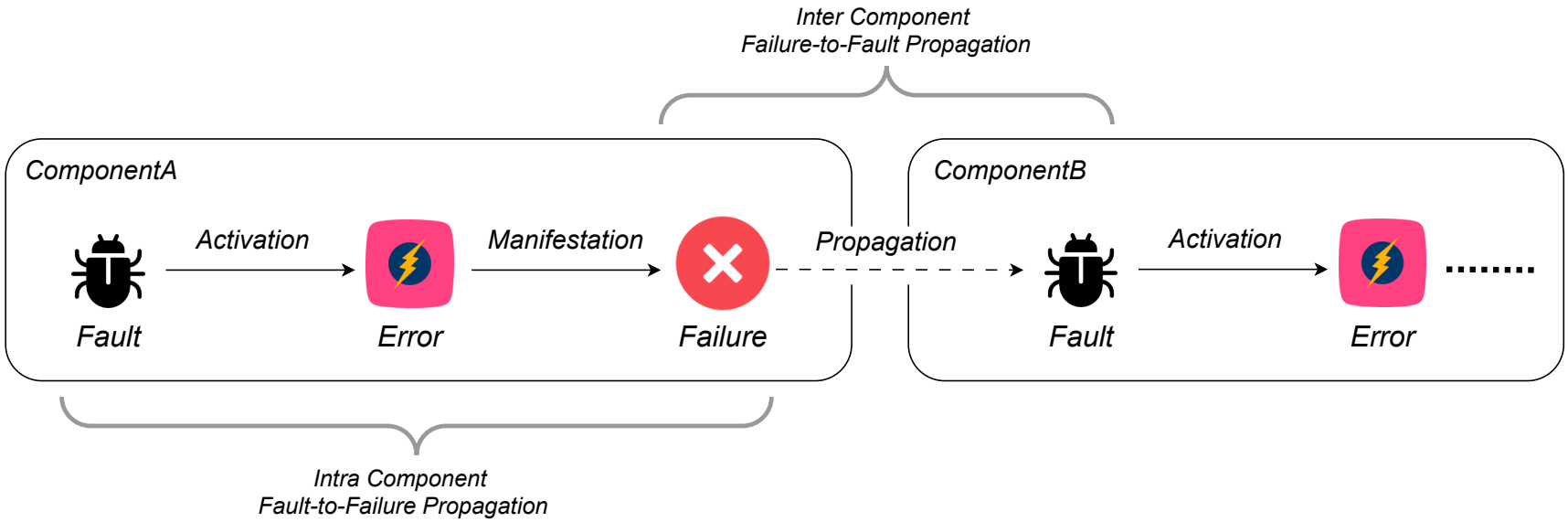
Context: Component-Based Systems



- Composition of multiple **loosely coupled and modular** components
- Components **interact** with each other through physical or communication interfaces
- **Hierarchical structure** frequently represented through SysML Block Defintion Diagrams

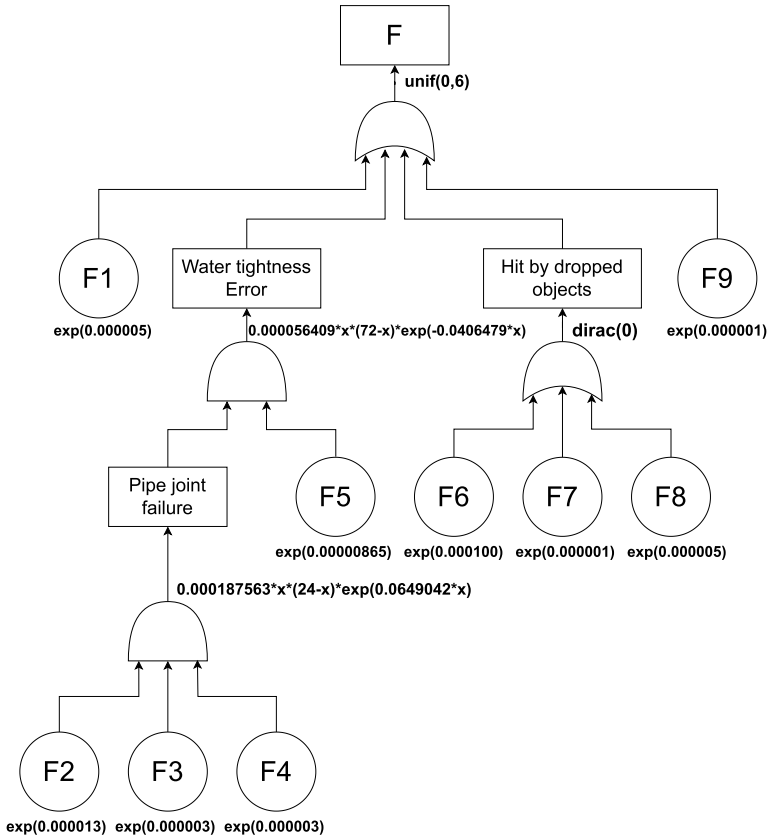
Context: Chain of Threats [1]

- Intra-component **Fault-to-Failure** propagation:
  - Internal component fault activates after a certain time
  - Component internal status becomes erroneous
  - Erroneous component may deliver incorrect services
- Inter-component **Failure-to-Fault** propagation:
  - Interaction with erroneous component propagates a fault
  - direct couplings
  - indirect couplings



[5] *Avizienis, Laprie, Randell, Landwehr. Basic concepts and taxonomy of dependable and secure computing. IEEE TDSC, 2004.*

Failure Logic through Stochastic Fault Tree (SFT)

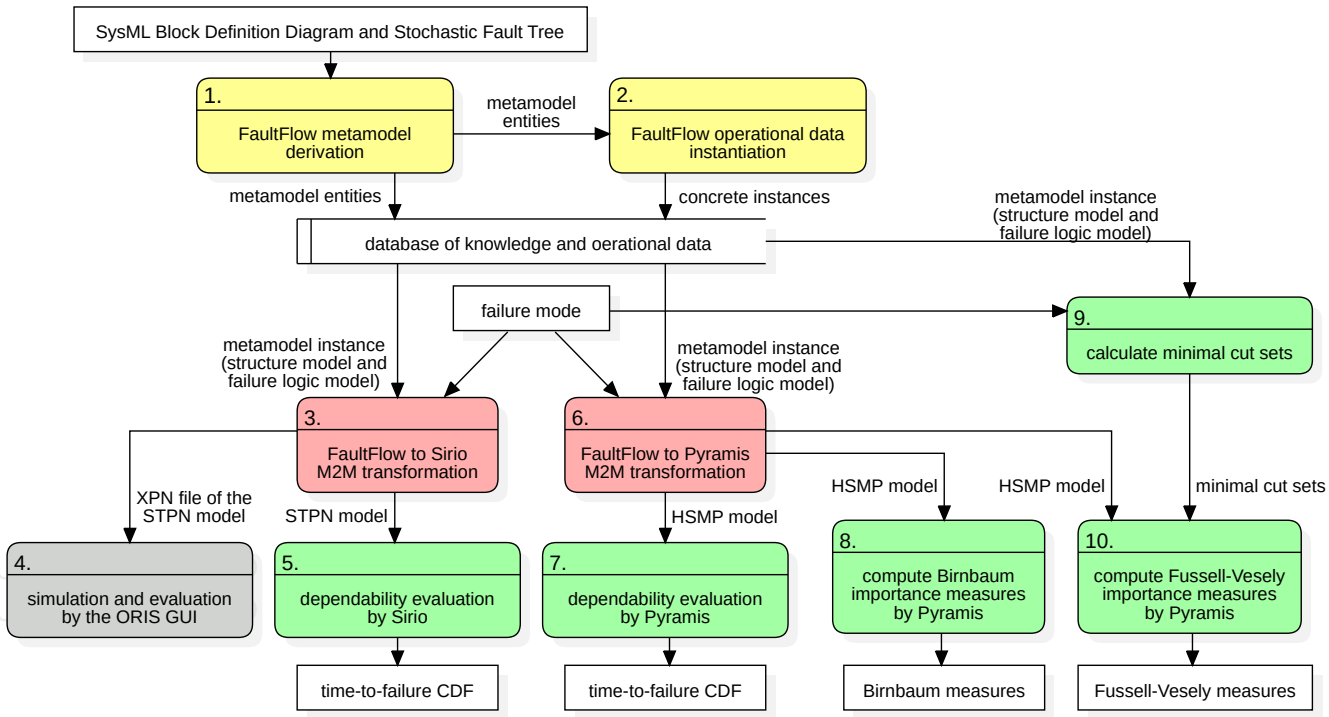


- Leaf nodes as **internal faults**
- Non-leaf nodes as external faults i.e., **Failure-to-Fault** propagations
- Logical gates as **Fault-to-Failure** propagations
- Node and ports associated with a delay having a non-Markovian distribution

The Fault Flow Java Library [2,3,4]

[2] *Carnevali, Cerboni, Montecchi, Vicario. FaultFlow: an MDE Java Library for Dependability Evaluation of SoS. Submitted.*
[3] *Parri, Sampietro, Vicario. Faultflow: a tool supporting an mde approach for timed failure logic analysis. EDCC, 2021.*
[4] *Vicario et al. Automated generation and efficient analysis of the timed failure logic of component-based systems. IWES, 2021.*

- **Metamodel** allowing representation of complex systems and their related failure logic
- **Non-Markovian distributions** supported for activation and propagation of faults
- **Model-to-Model** (M2M) transformations:
  - Automated derivation of metamodel instances from semi-formal artifacts
  - Automated derivation of failure process duration distribution (by Sirio library)
  - Automated derivation of fault importance measures (by Pyramis library)

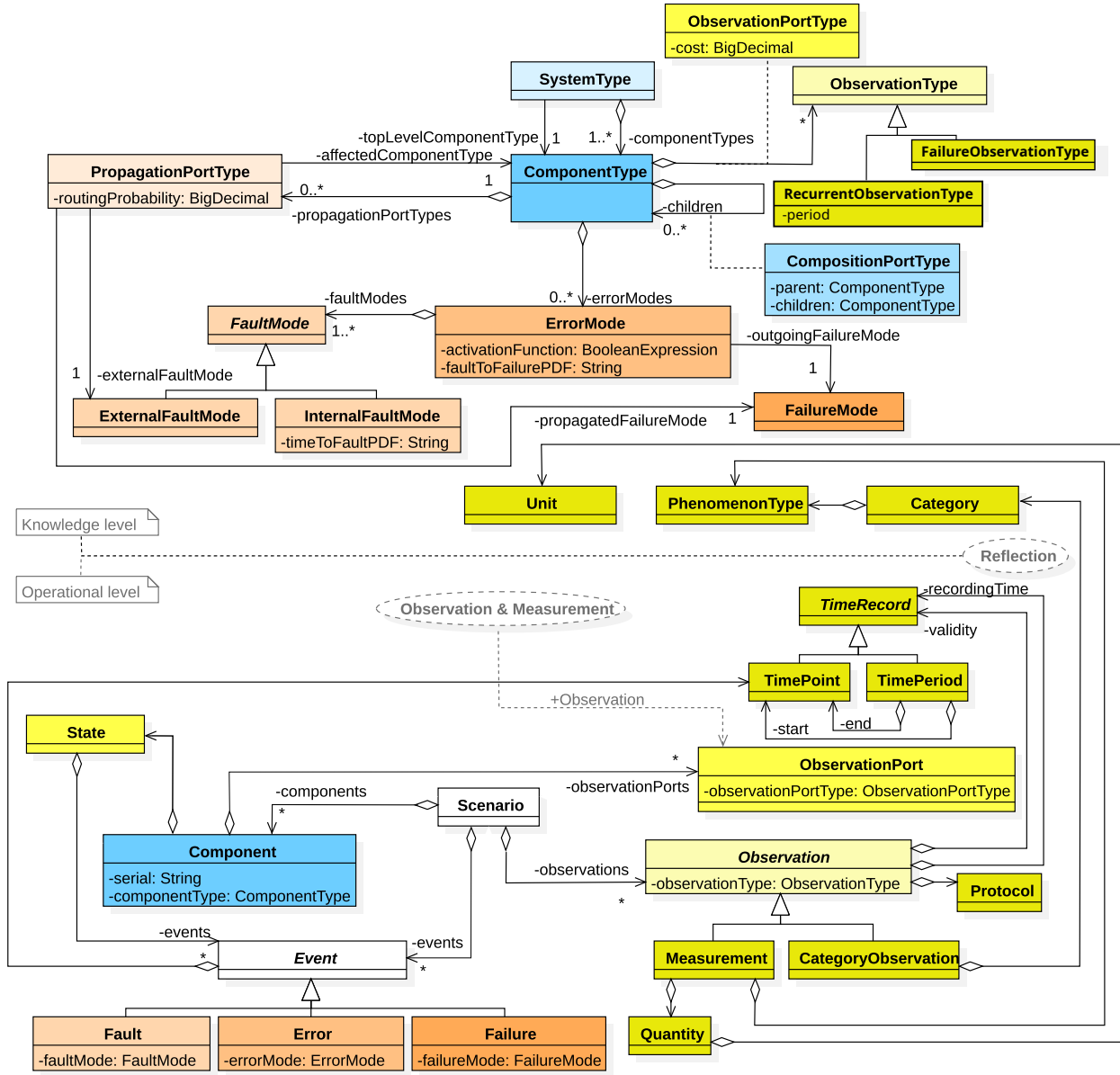


Contribution of the work

- Provide a software environment to test **dependability evaluation** methods
  - Evaluate different **monitoring** policies, **rejuvenation** policies etc . . .
  - Support **online failure prediction** methods
  - **Synthetic data** generation
- Extension of the Fault Flow Java Library
  - Extension of the Fault Flow **Metamodel**
  - Extension of the FaultFlow to Sirio **M2M transformation rules**

Extending FaultFlow with an Observation Metamodel

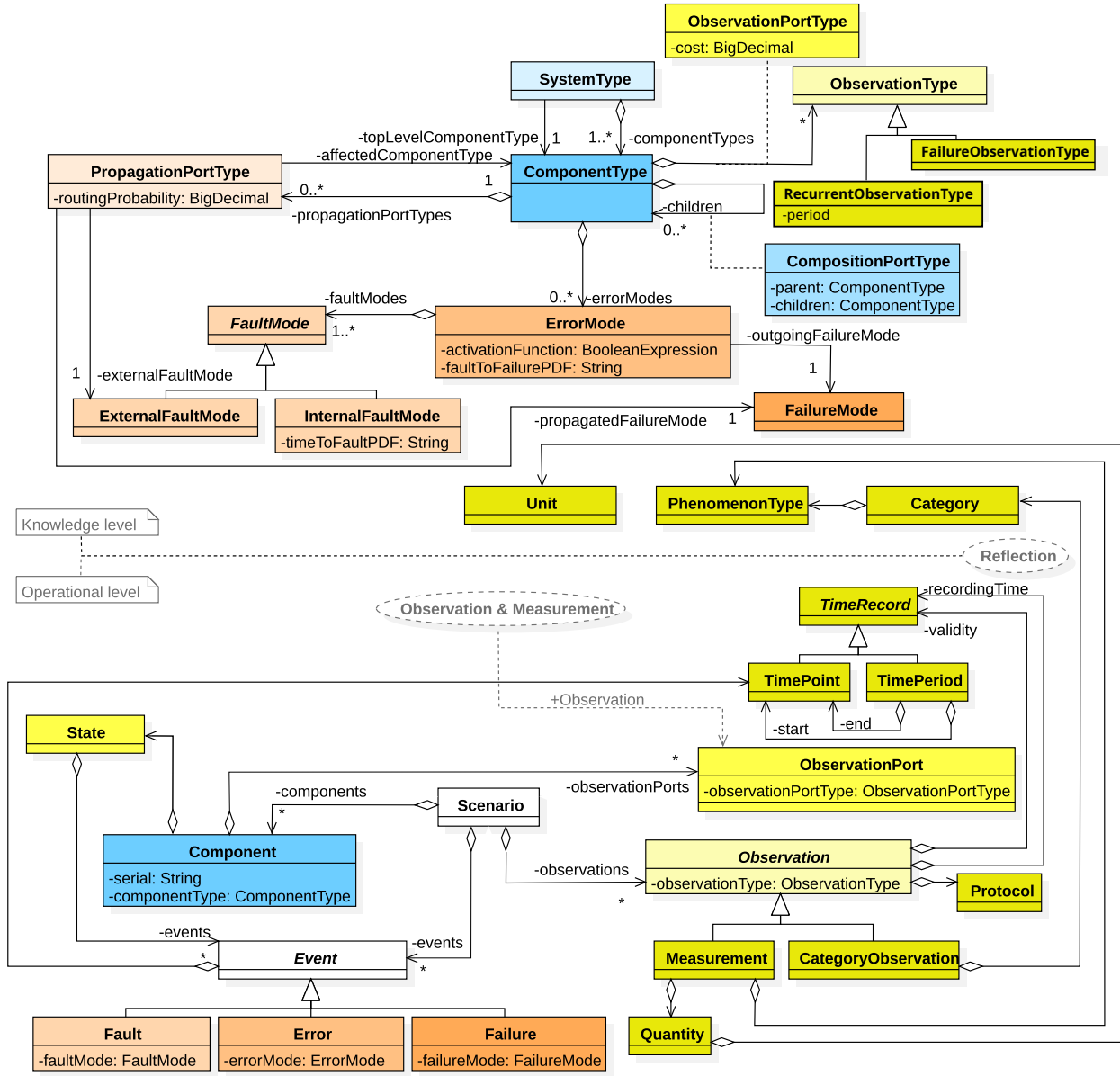
- Observation & Measurement Analysis Pattern
- Reflection Pattern [5]
- **Application domain-unaware** metamodel



[5] *Schmidt, Stal, Rohnert, Buschmann. Pattern-oriented software architecture, volume 1: a system of patterns. John Wiley & Sons, 1996.*

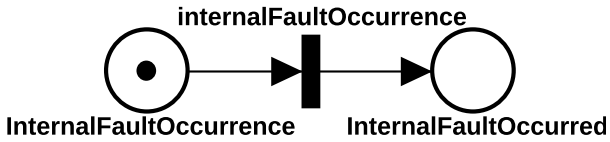
Extending FaultFlow with an Observation Metamodel

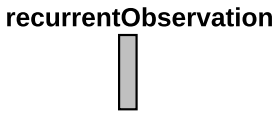
- Recurrent or one-shot
- Qualitative or quantitative
- Referred to an instant or a time period

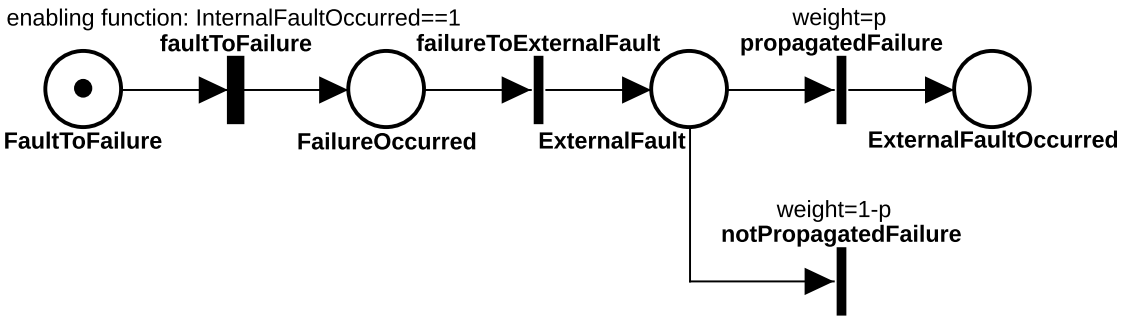


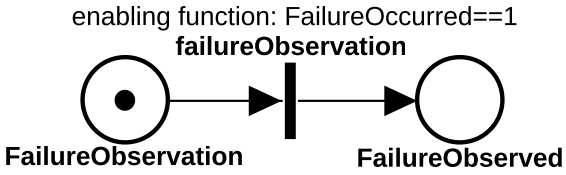
Extending the FaultFlow-to-Sirio M2M transformations

- Rules to transform observation-related metamodel instances into Petri Net **places** and **transitions**
- STPN includes transitions modeling occurrence of observations
  - IMM transitions for observations taken at event occurrence (one-shot)
  - DET/GEN transitions for observations taken at specific times (recurrent)
- Extension of **STPN simulator** to get observation values at transition firings
  - Entry-point methods returning observation value based on component state



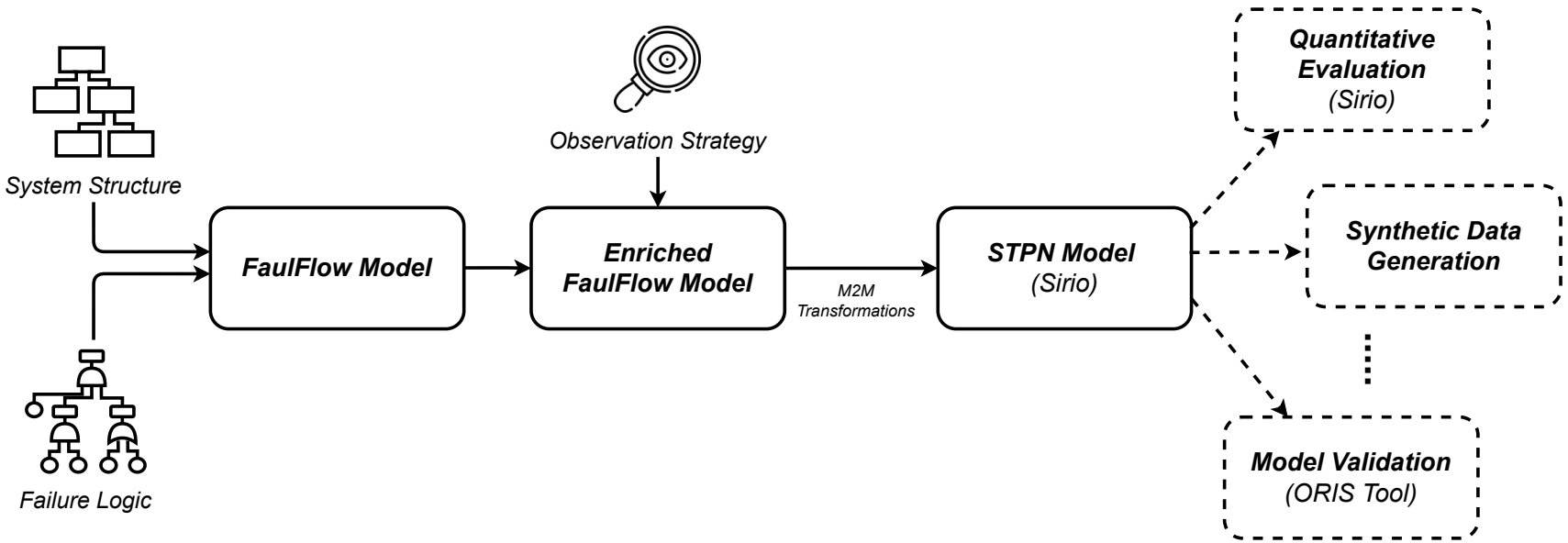






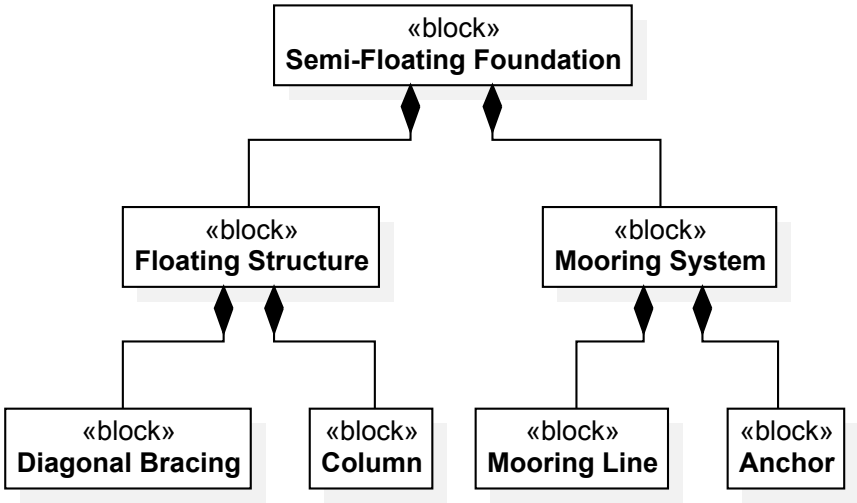
Extended FaultFlow Java Library Workflow

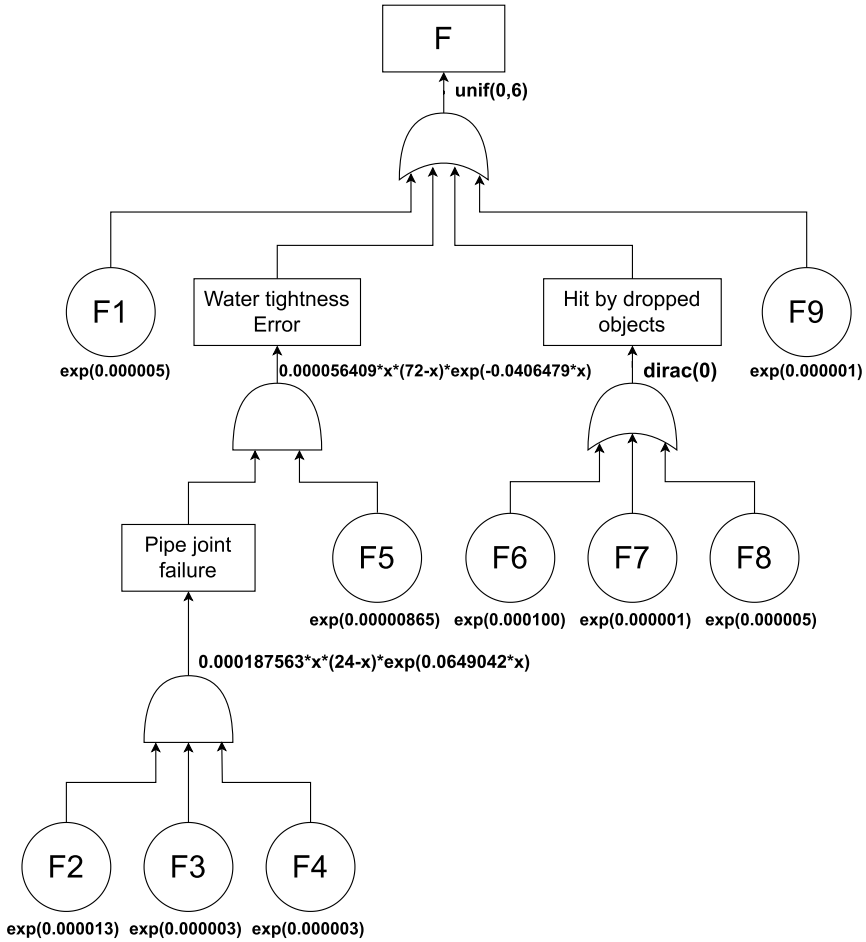
- **Quantitative Evaluation** of monitoring setup
- **Validation** of Models
- Generation of **realistic synthetic datasets** of typed and time-stamped observations



[6] *https://www.oris-tool.org*
[7] *https://www.oris-tool.org/sirio*
[8] *Paolieri, Biagi, Carnevali, Vicario. The ORIS Tool: Quantitative Evaluation of Non-Markovian Systems. IEEE TSE, 2021.*

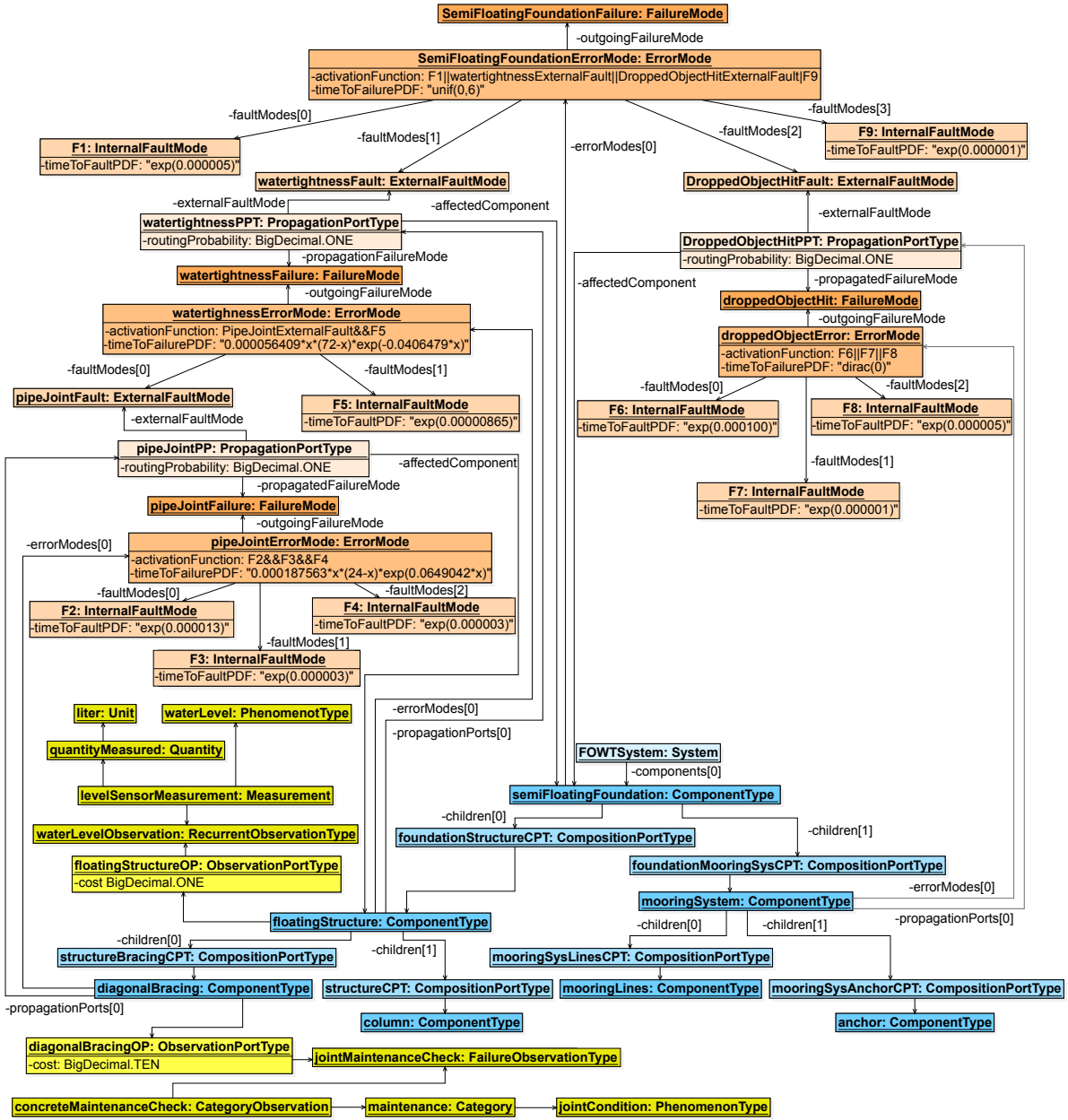
Floating Offshore Wind Turbine Case Study [9]: Structure and Failure Logic



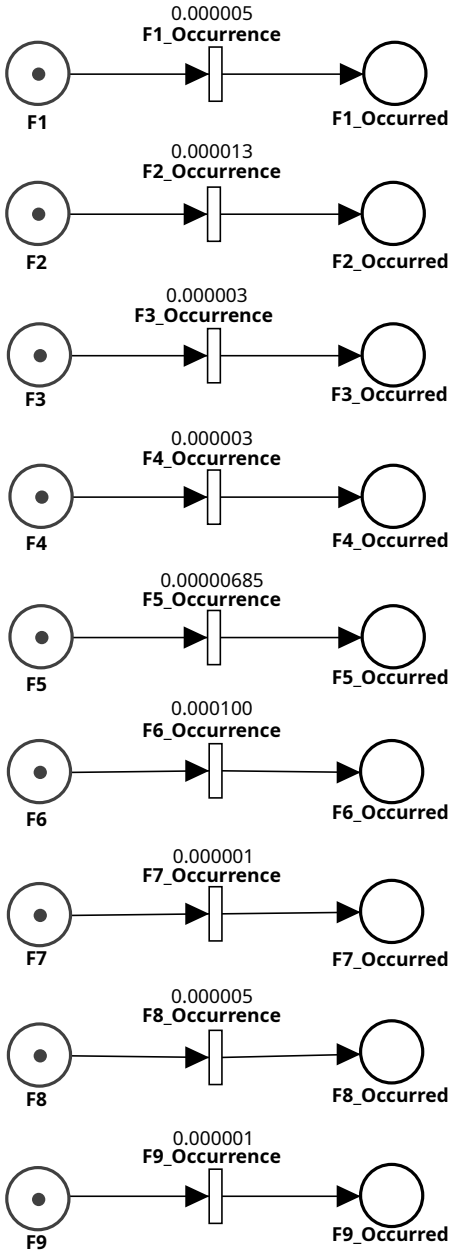


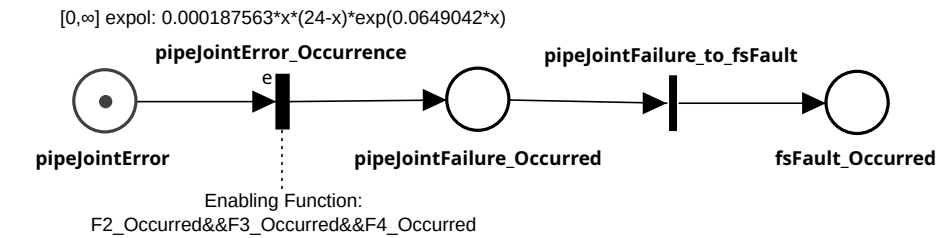
[9] *Zhang, Sun, Sun, Guo, Bai. Floating offshore wind turbine reliability analysis based on system grading and dynamic FTA. JWEIA, 2016*

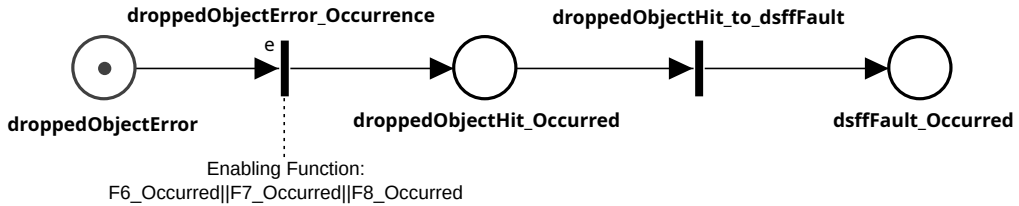
Floating Offshore Wind Turbine Case Study: Fault Flow Representation

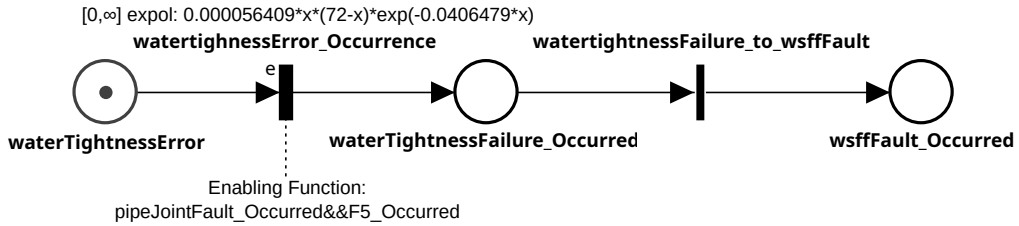


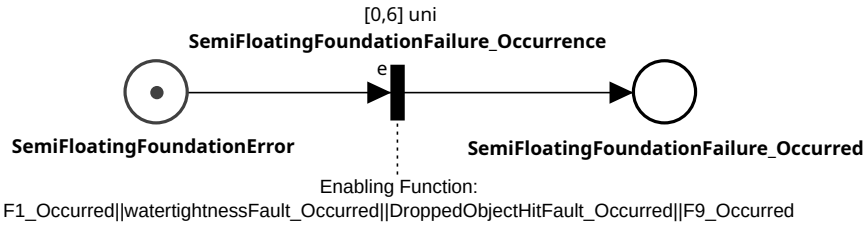
Floating Offshore Wind Turbine Case Study: STPN Representation

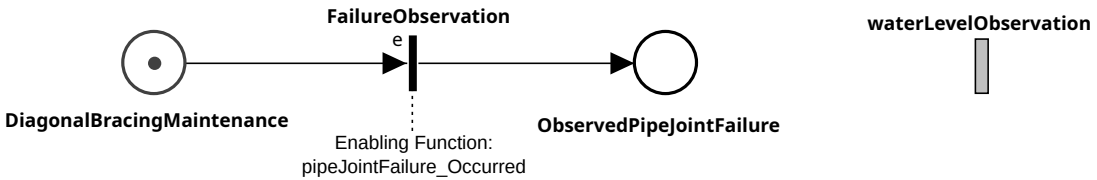












Discussion and Future Directions

- **Aim of the Work:** Provide a flexible and extensible metamodel to facilitate customization of monitoring strategies
- **Ongoing Direction:**
  - Learning failure logic models through model-based and data-driven methods
  - Markov Arrival Process parametrization for online failure prediction