NTNU

Kunnskap for en bedre verden

Modelling of a HIPPS with AltaRica

RAMS seminar 03.03.2017

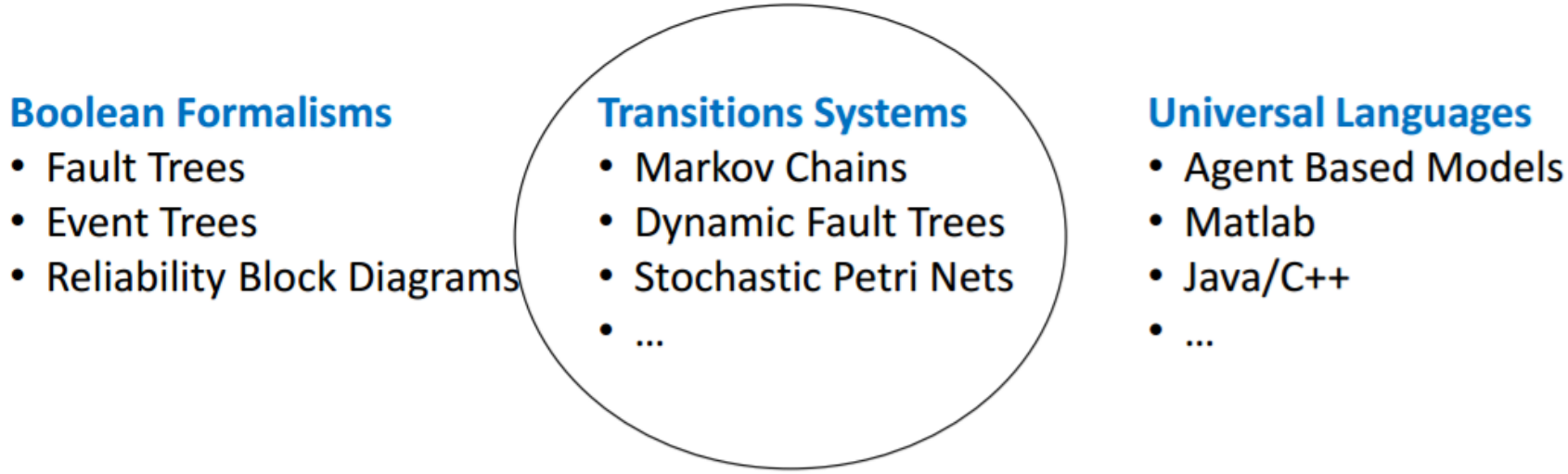
Yun Zhang

Antoine Rauzy

Issues

- Reliability assessment of complex system raises a number of challenging issues:
  - How to design model efficiently?
  - How to reuse parts/components of models?
  - How to validate models?
  - How to have a quality assurance on the outputs of the modeling process?
  - How to maintain models throughout the life-cycle of systems?
  - How to better integrate reliability engineering with other disciplines? …

Categories of modelling languages



Is there a generic/unified modeling framework to assess reliability of complex systems?

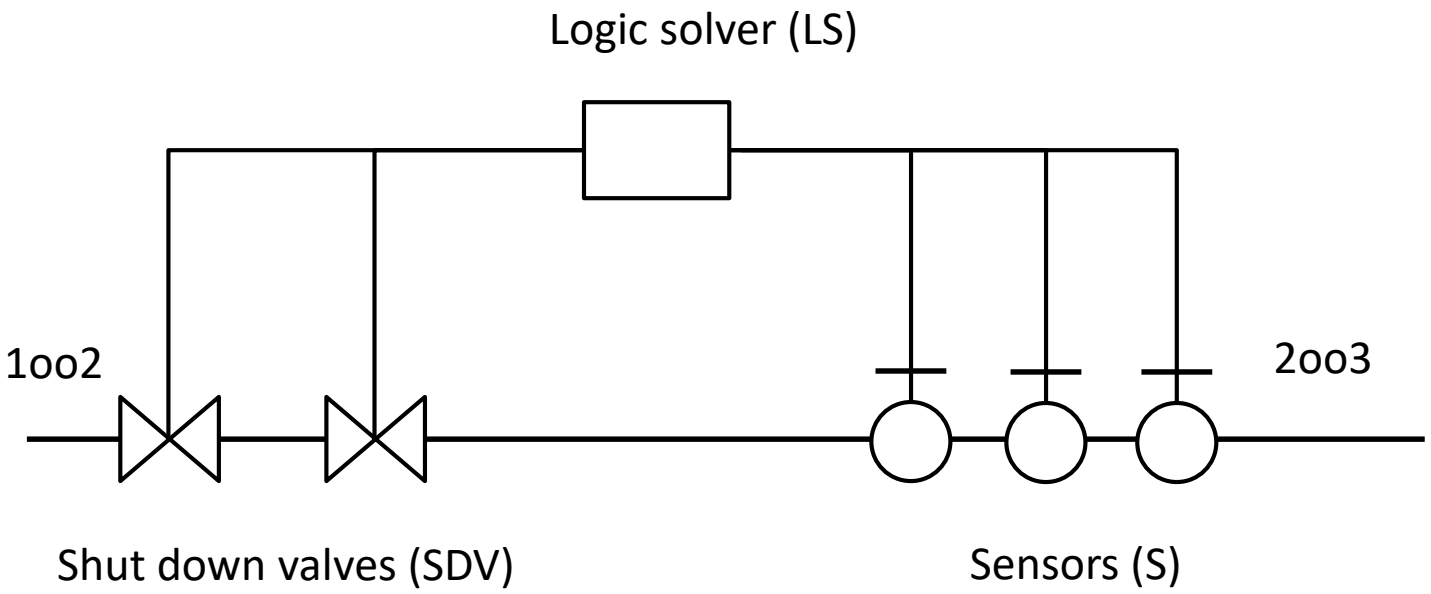
Guarded Transitions Systems as implemented in AltaRica

Problem statement

- We focus on modelling of a HIPPS with AltaRica
- The problem is mainly about
  - how to design a maintenance model to help assessing condition monitoring and optimization of maintenance policies given the description of constraints

- But it could also be relevant for problems like
  - how to analyze production of the system
  - how to do risk analysis
  - …

System description

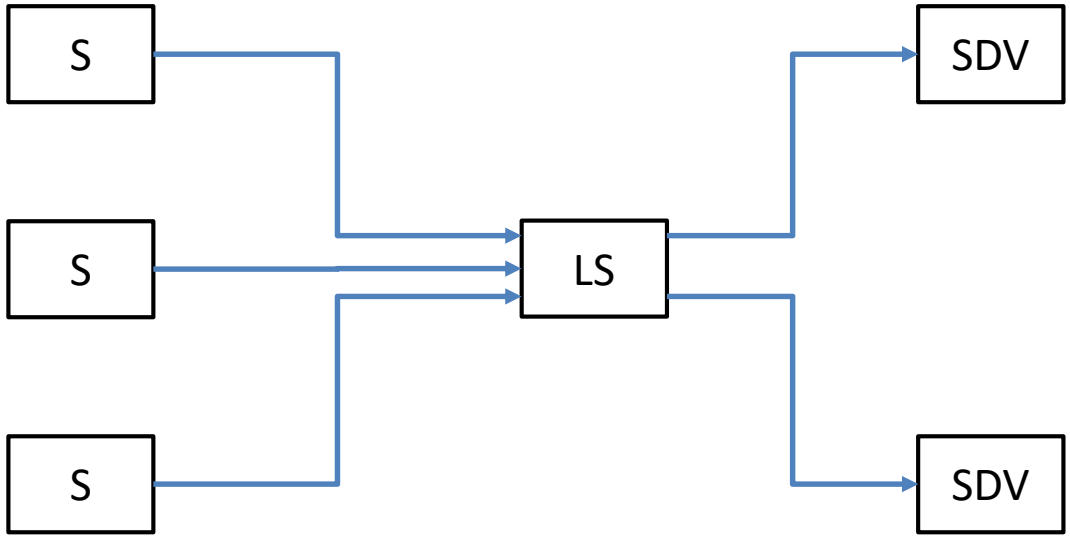


Features of the HIPPS

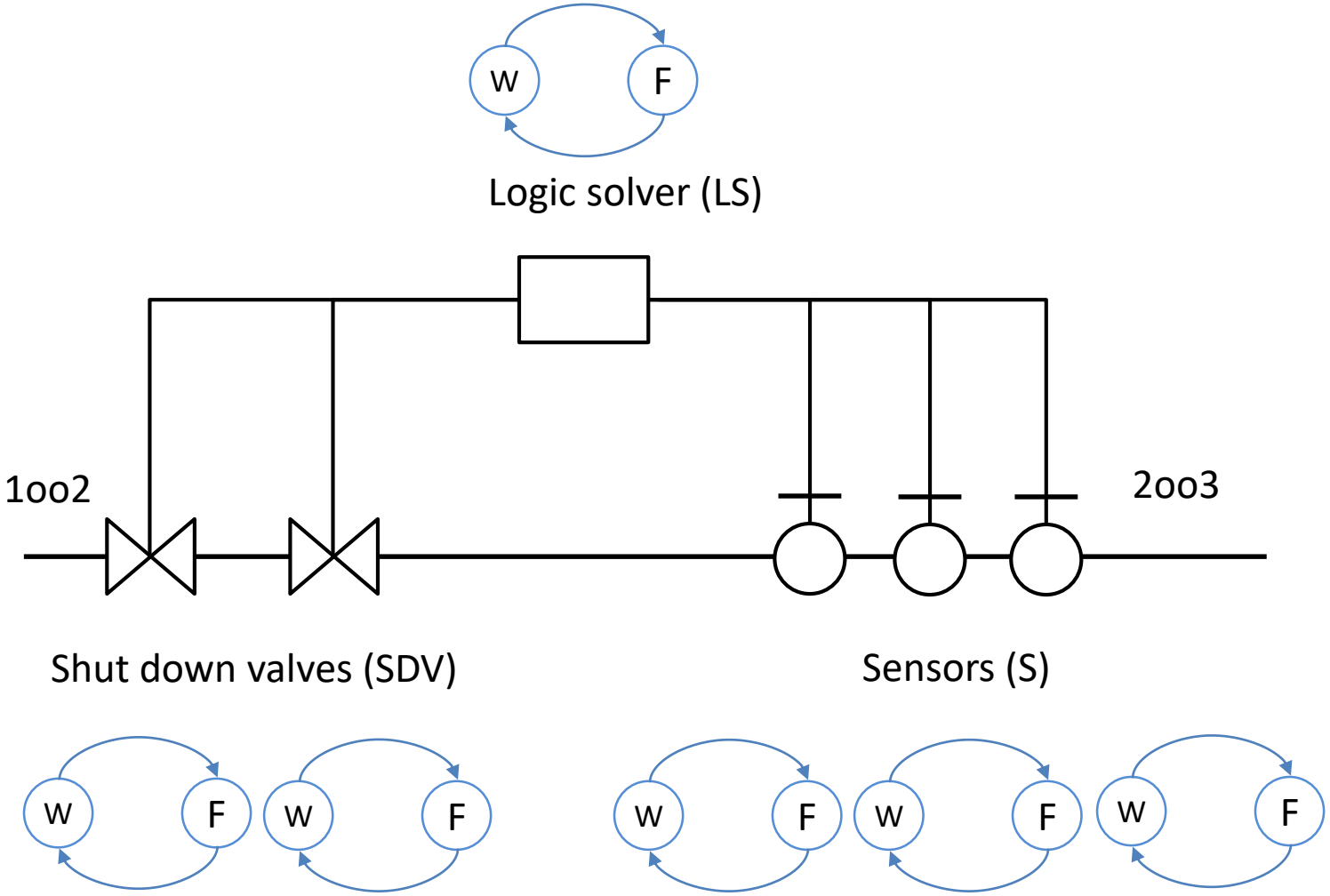
- **Components:**
  - Continuously monitored components (logic solver and sensors)
  - Periodically tested components (shutdown valves)

- **Objectives:**
  - Reduce number of maintenance interventions
  - Minimize system downtime

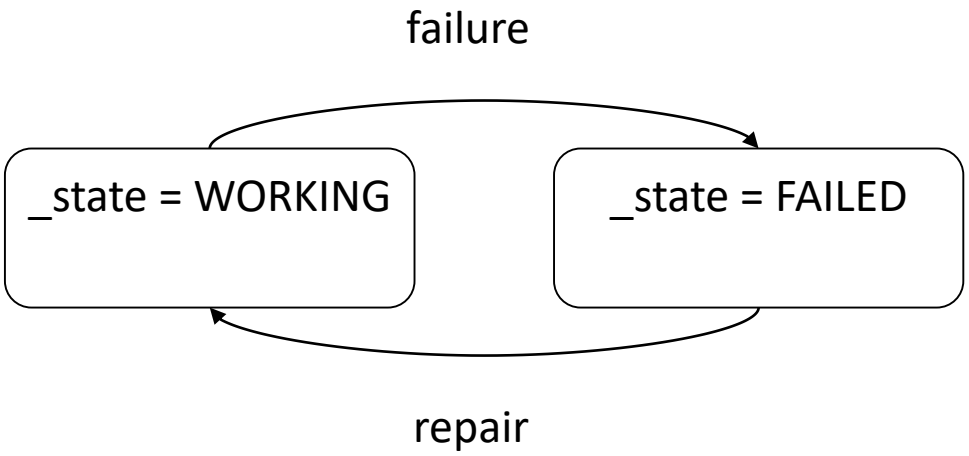
Step 0. Block diagram



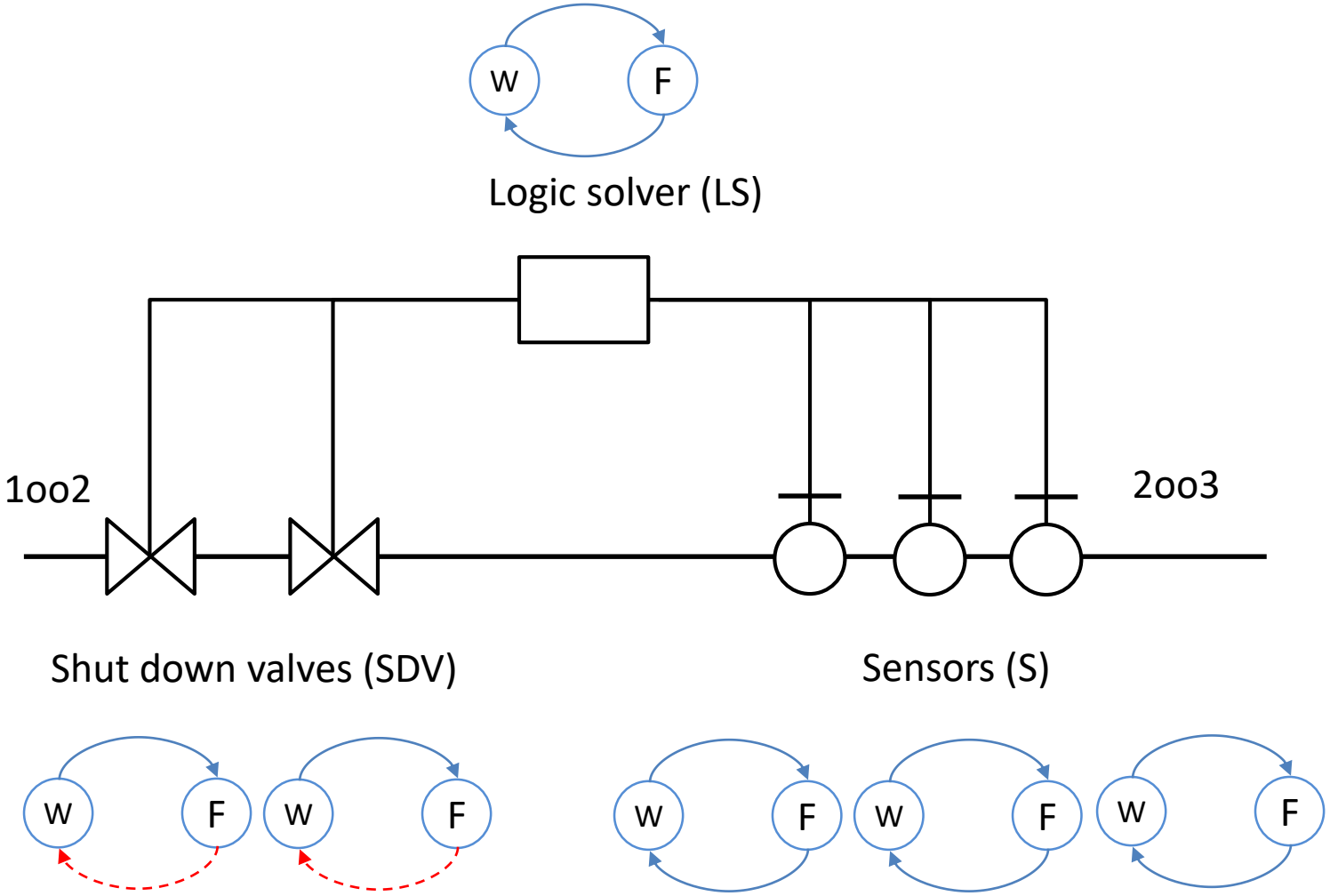
Step 1. Repairable component



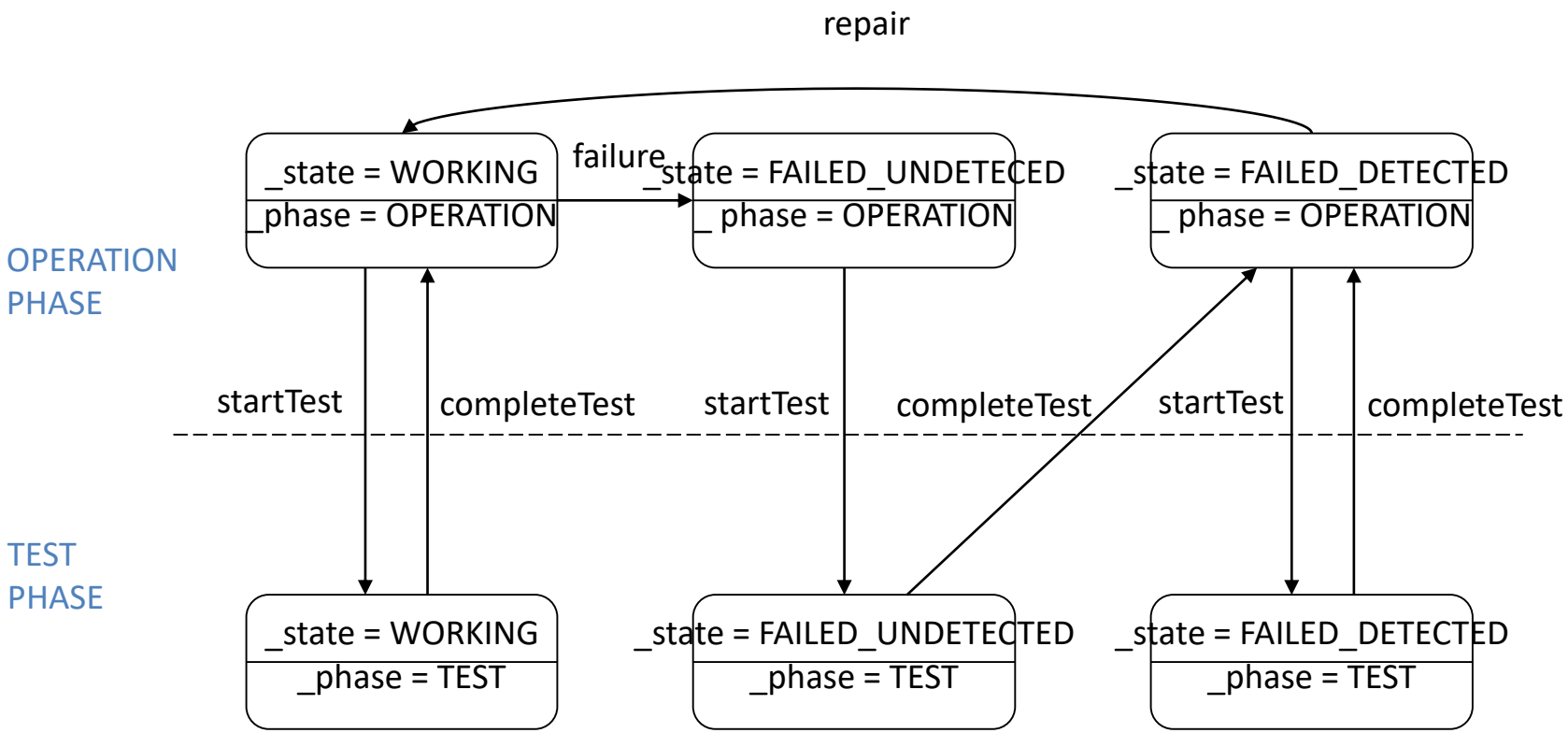
Finite state automaton for repairable component



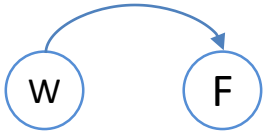
Step 2. Tested component

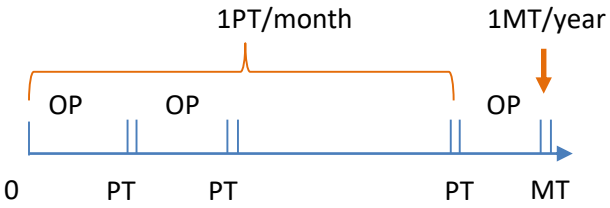


Finite state automaton for a tested component

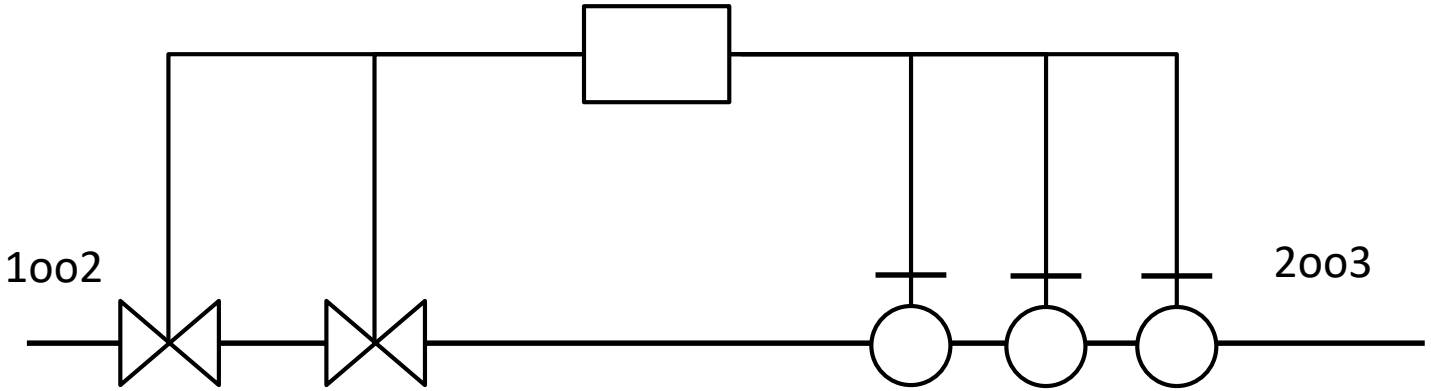


Step 3. Periodic maintenance

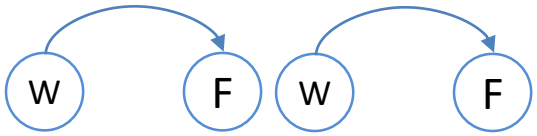




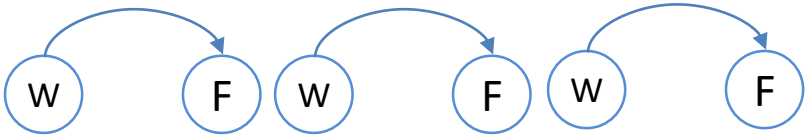
Logic solver (LS)



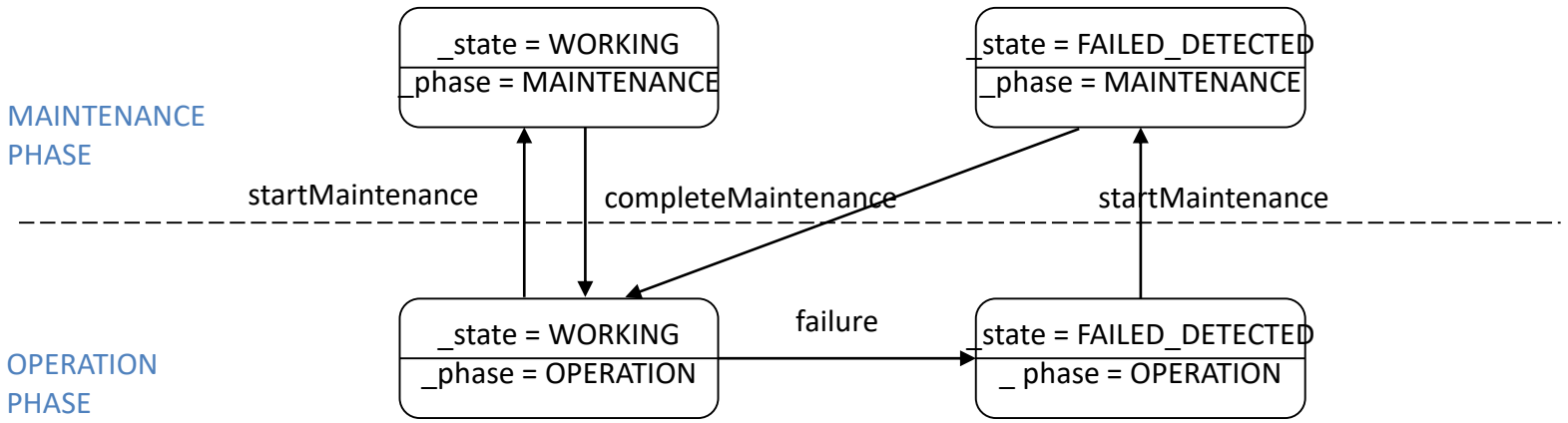
Shut down valves (SDV)



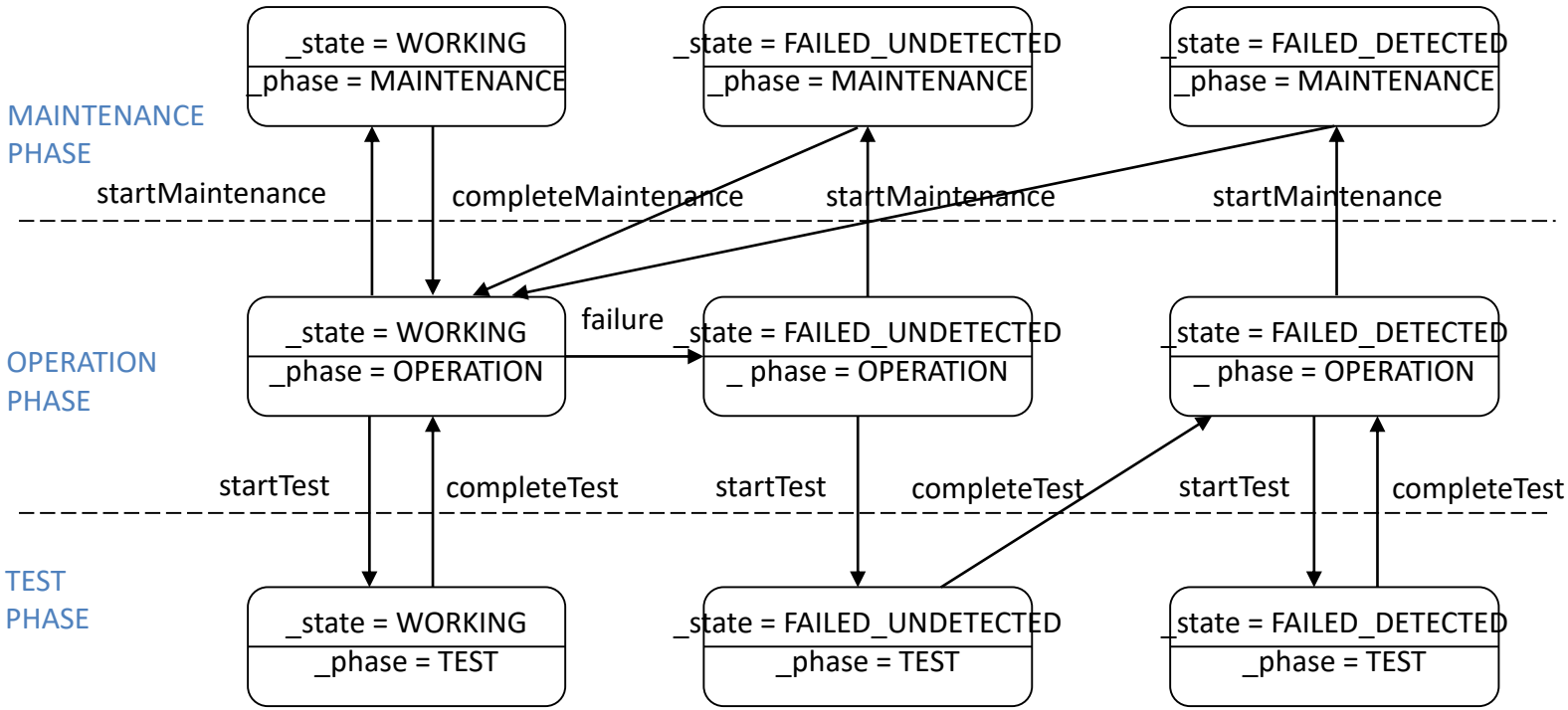
Sensors (S)



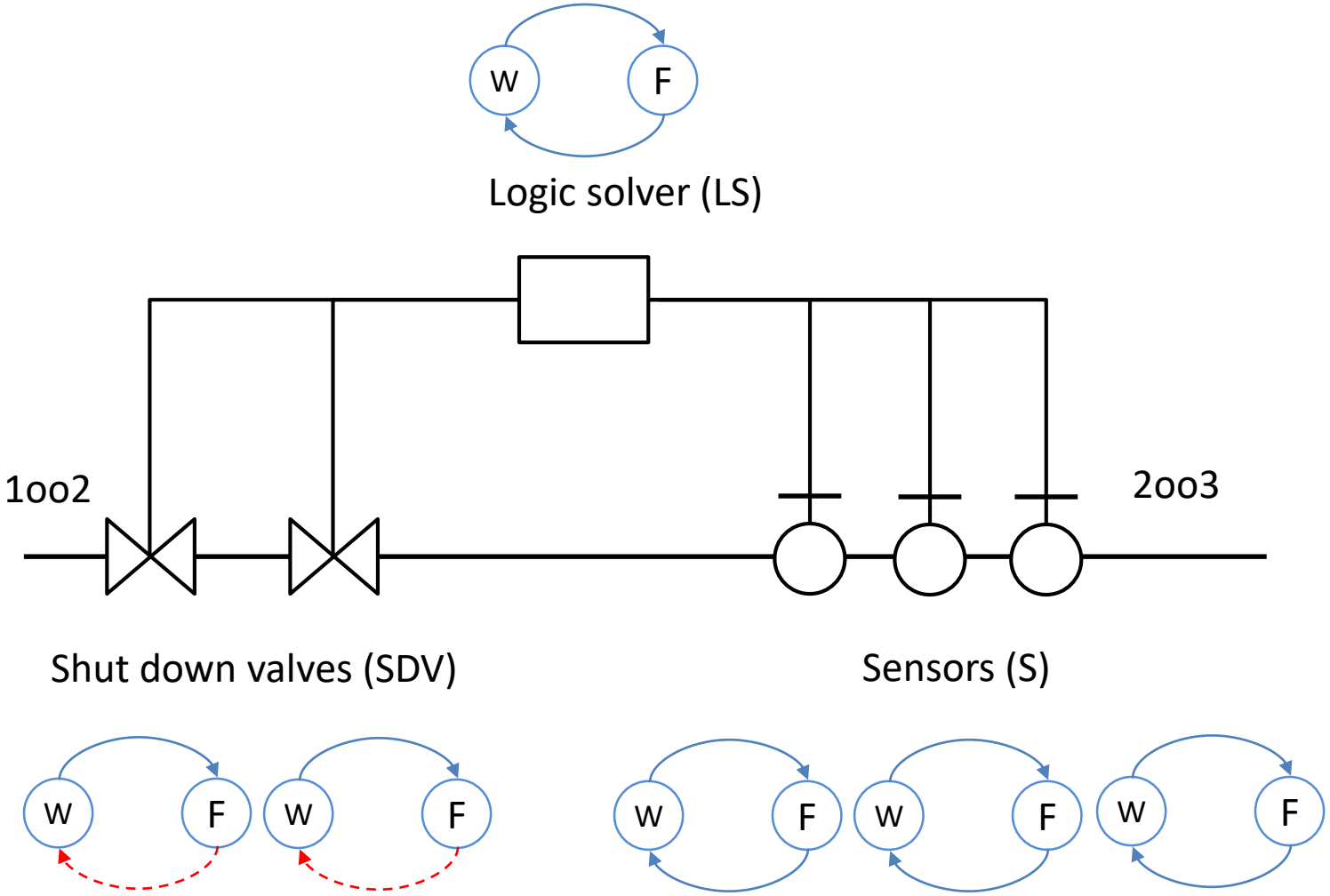
Finite state automaton for a monitored component



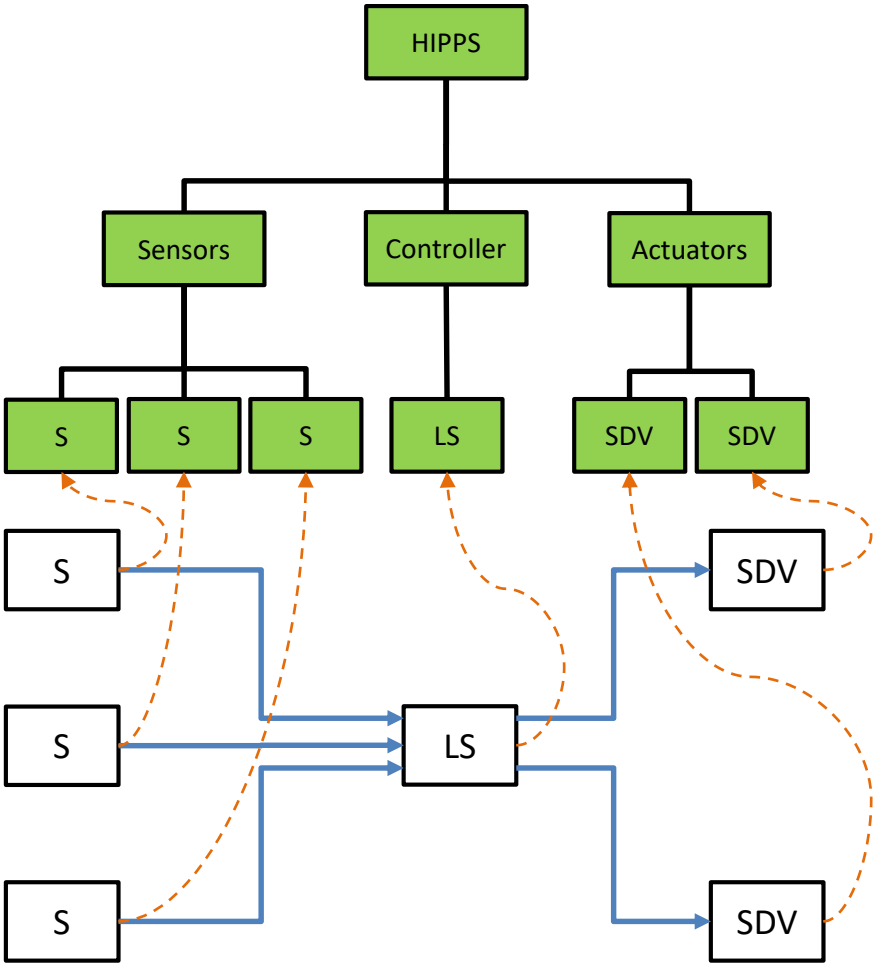
Finite state automaton for a tested component



Step 4. Condition based maintenance

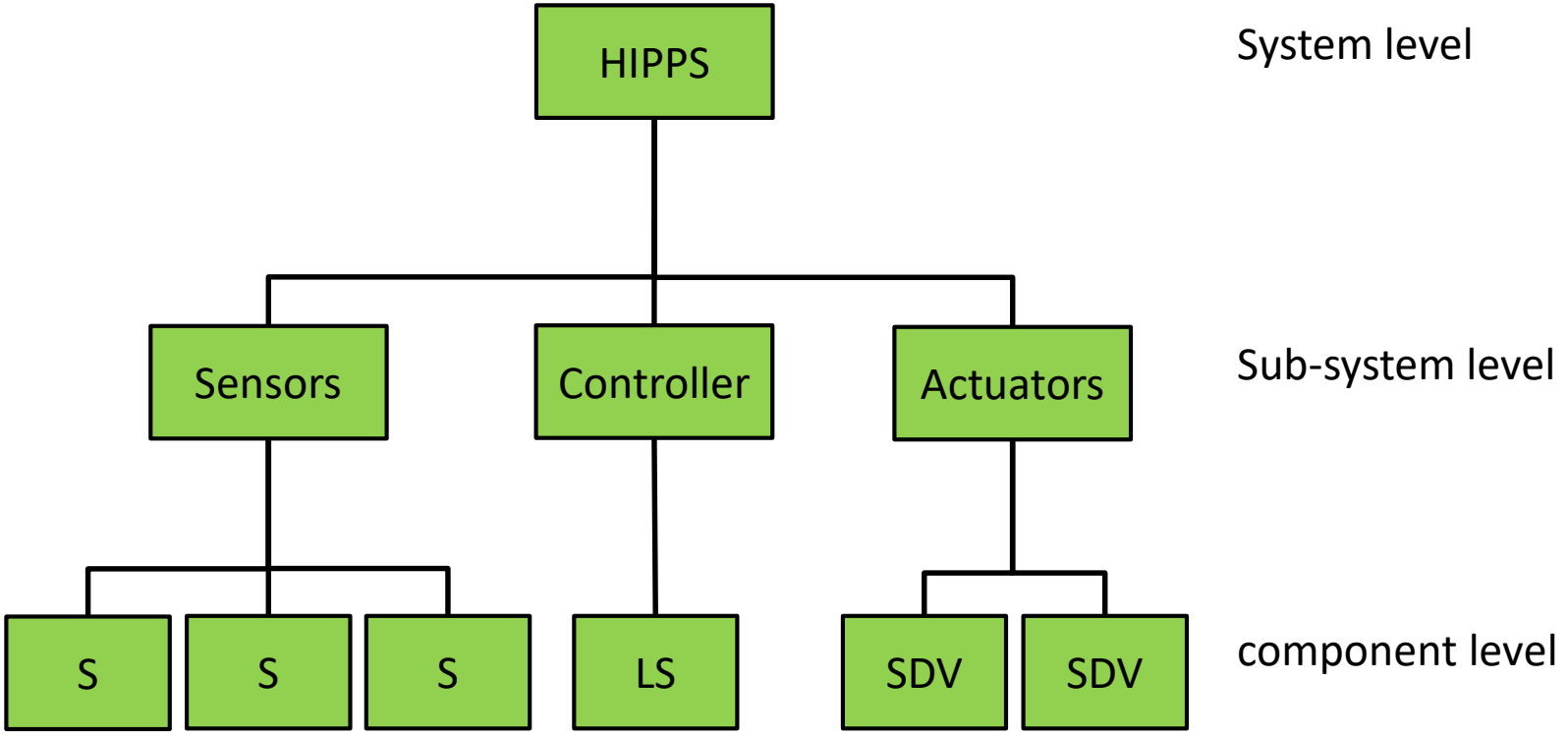


Our knowledge about the state of the system maybe different from actual state of the system…



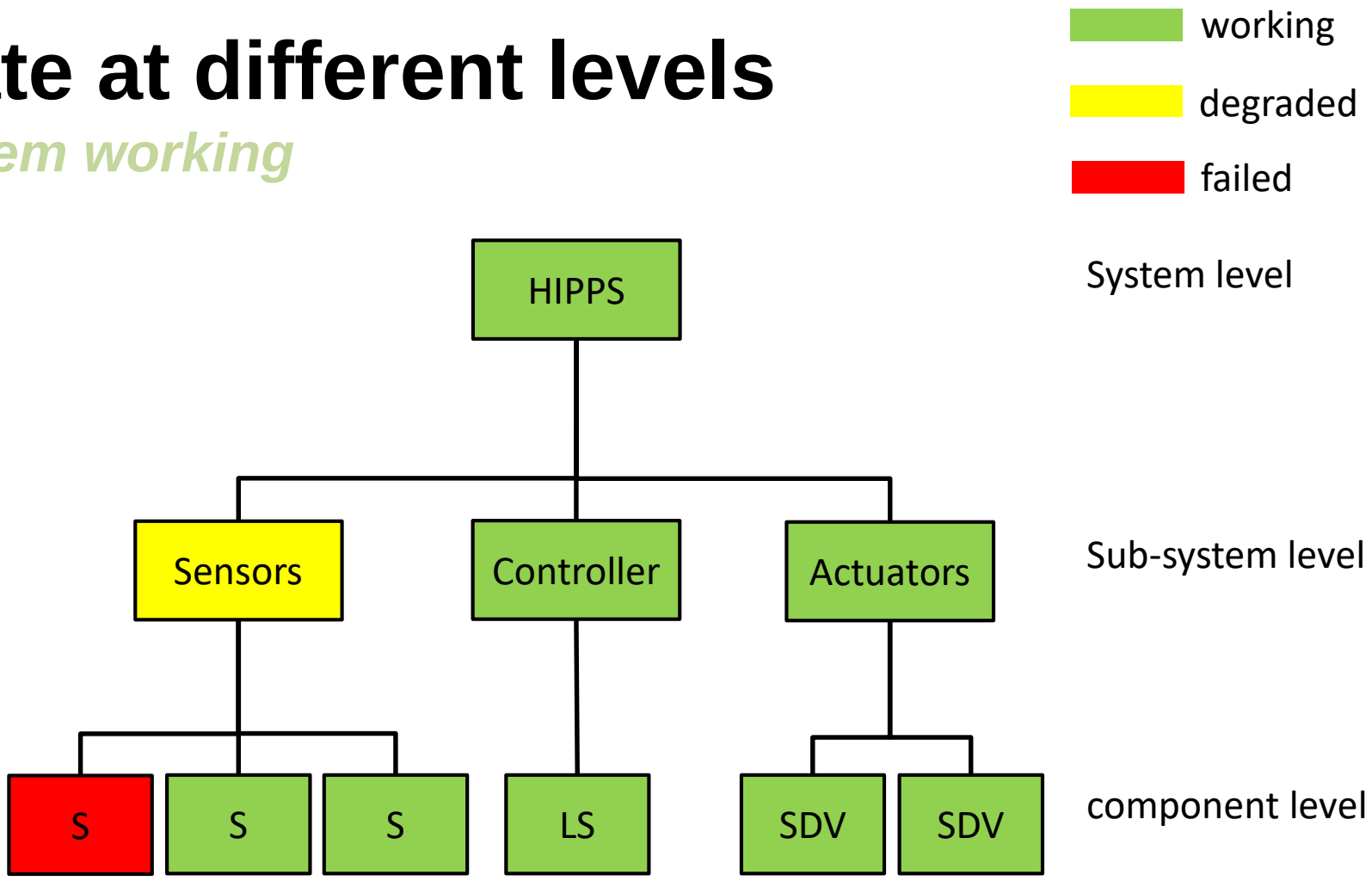
State at different levels

System working



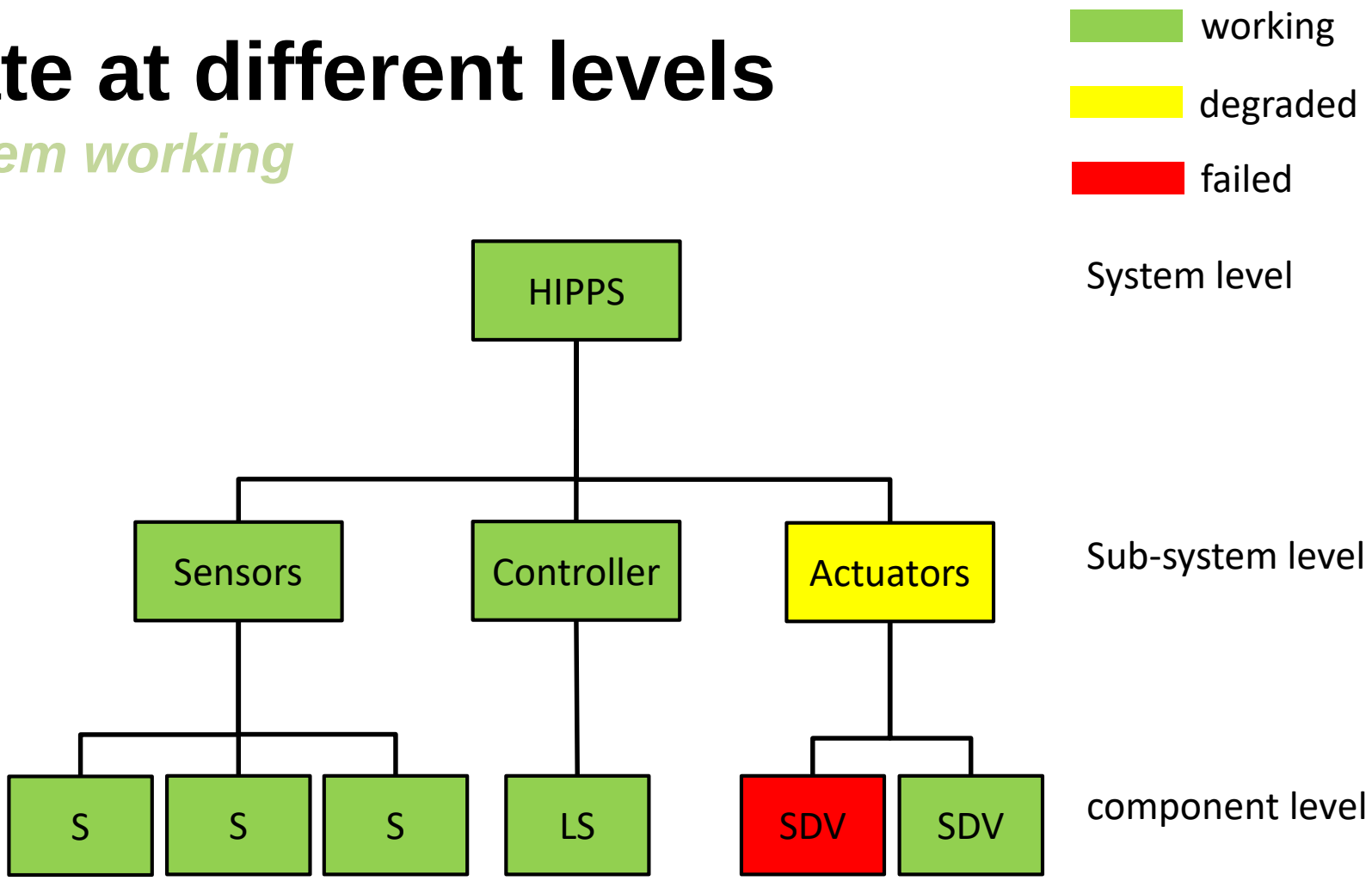
State at different levels

System working



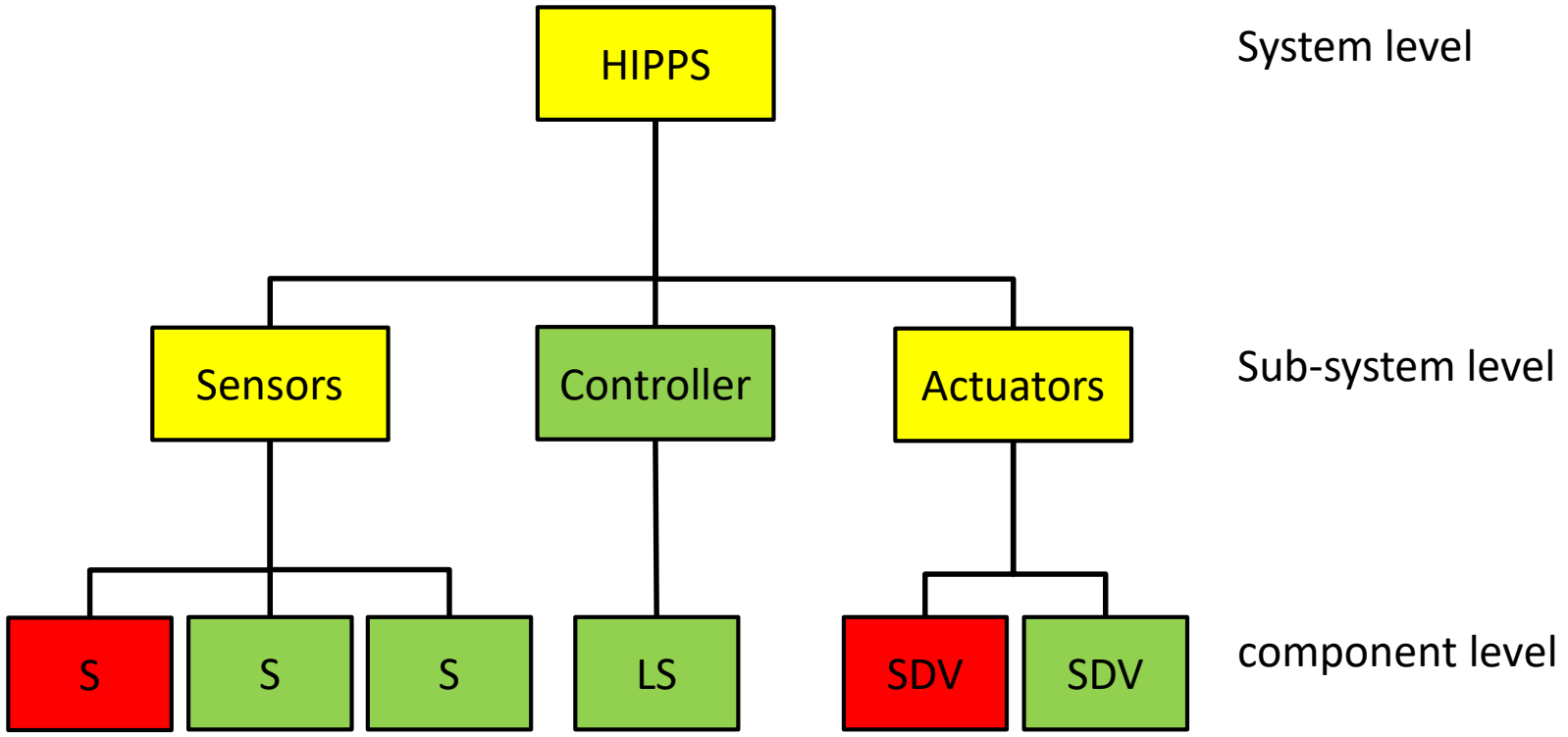
State at different levels

System working



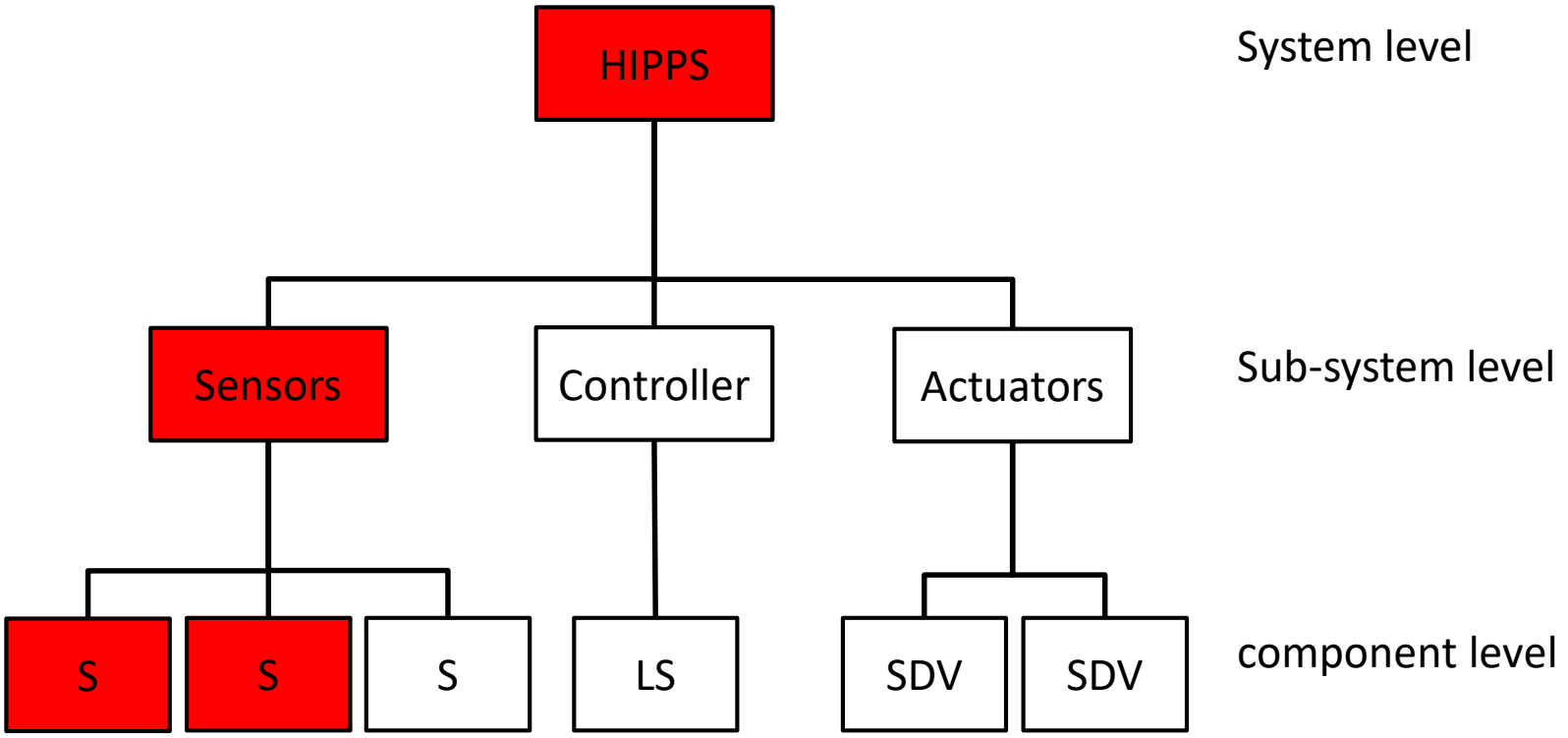
State at different levels

System degraded



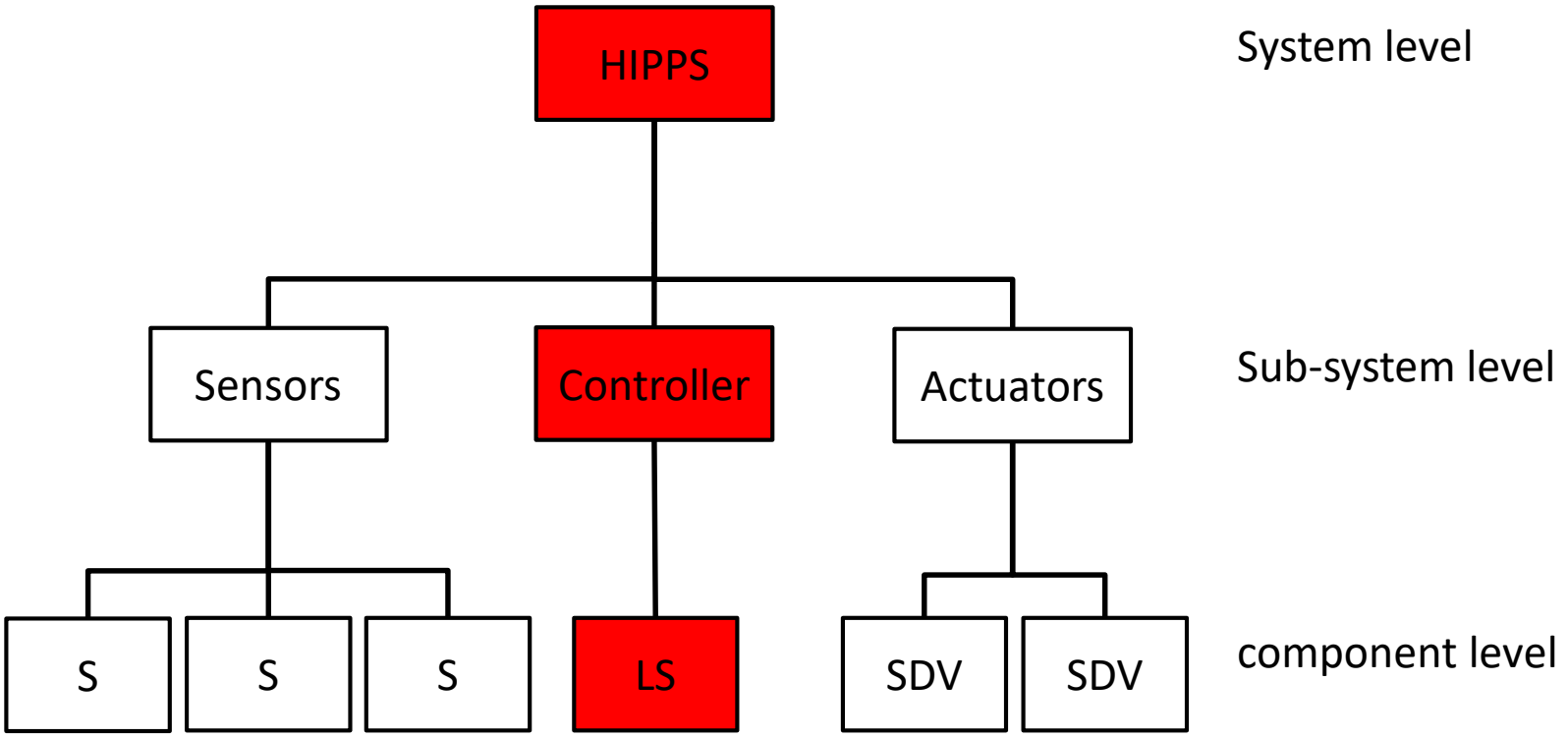
State at different levels

System failed



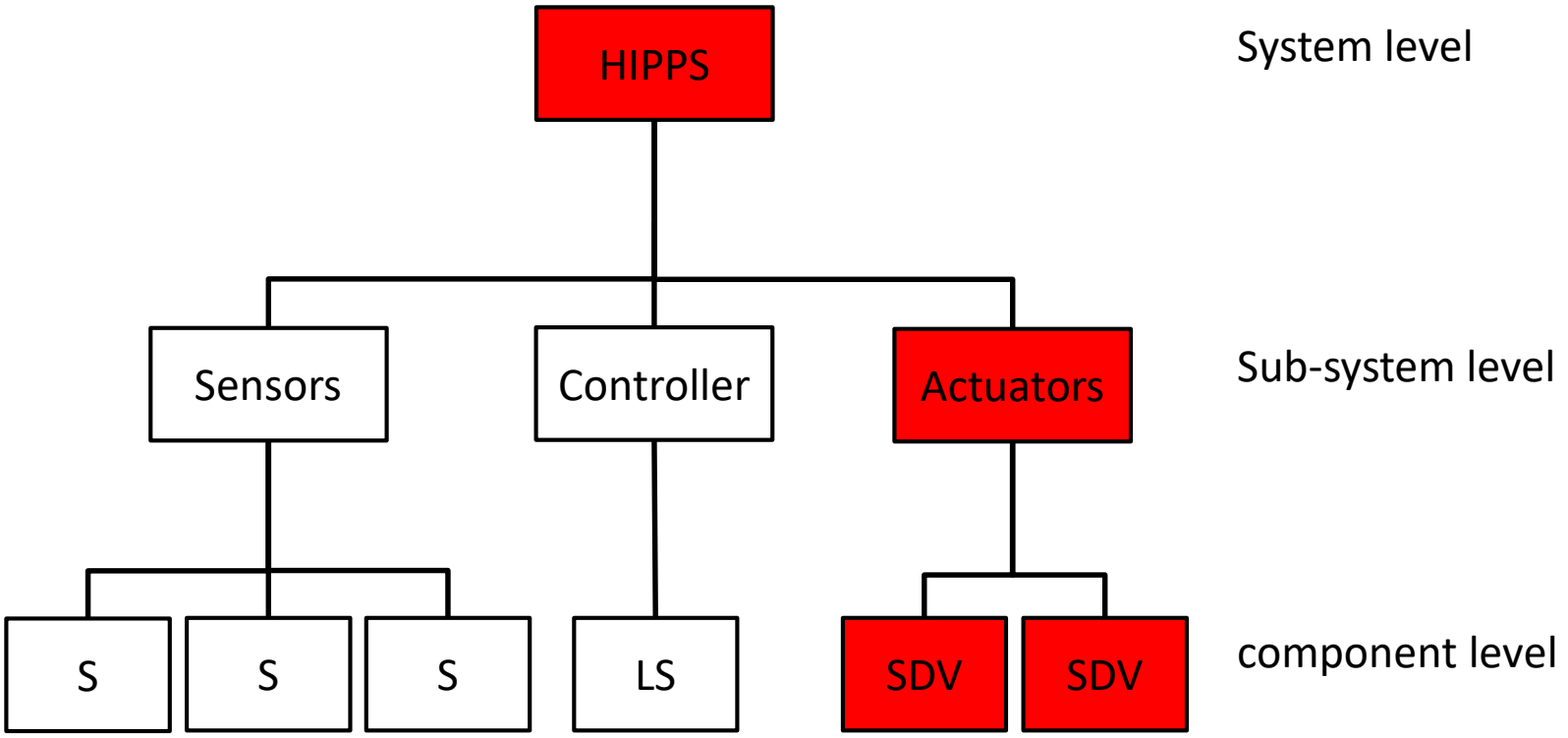
State at different levels

System failed



State at different levels

System failed



Component states

Components			
S1	working		failed detected
S2	working		failed detected
S3	working		failed detected
LS	working		failed detected
SDV1	working	failed undetected	failed detected
SDV2	working	failed undetected	failed detected

Sub-system states

Sub-system	working	degraded	failed
Sensors	3 working	2 working	1 or 0 working
Logic solver	1 working		0 working
Shutdown valves	2 working	1 working	0 working

System states

System	working	degraded	failed
Sensors	(and) working or degraded	(and) degraded	(or) failed
Logic solver	(and) working	(and) working	(or) failed
Shutdown valves	(and) working or degraded	(and) degraded	(or) failed

Finite state automaton for decision rules