Exercise: Boiler Water Contents

- A device is needed to monitor the depth of water in a boiler vessel
- Two sensors are provided
  - “water low” and “water high”
- When the water is low a "fill valve" is to be opened.
- When the water is high a "drain valve" is to be opened.
- When neither high nor low signal is present both valves are closed.
- To prevent the valves *chattering* some delay of operation is required with the valves only operating after 10 successive, consistent signals have been received from the associated sensor.

System boundary

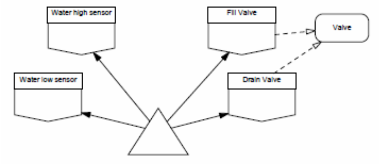
- It encompasses
  - the water vessel itself
  - the valves
  - the sensors
- Presumably other things are going on such as
  - heating the water and taking off steam
  - however these are outside the system boundary
- The physical inputs are
  - “water high” and “water low” signals
- The physical outputs are
  - the “fill valve” and “drain valve"

SPARK boundary

- abstract view: we are concerned only with
  - “sensors” and “actuators”
  - would leave us unable to reason about whether the fill valve opened in response to a low or high water level
- simple system: we can take the physical inputs and outputs identified above directly:
  - “water high sensor”, “water low sensor”
  - “fill valve” and “drain valve”
- Outside the SPARK boundary
  - the implementations of those boundary variables
  - e.g. the precise way that the logical signal “water high” is obtained from the physical environment.

SPARK boundary (2)

- Signals “water high” and “water low” can be considered Boolean
- valves are normally Open or Shut.
  - Since we have two specific valves (Fill and Drain) that share this characteristic
  - a type package to represent this shared characteristic of valves
- Quindi:



Lo stato del sistema

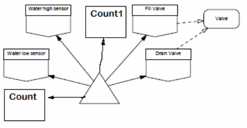
- Requirement: to only act when 10 successive (high or low) hits have been recorded
- We need to store counts of the number of times the sensors have recorded the water being high or low
- Dove memorizzarlo?
  - package globali usate dal main
  - nei sensori stessi
  - con una locale nel main

Package globali usate dal main

- The main program would poll the raw sensor values and, if an event was registered, update the appropriate variable package.
- A function exported by the variable package would indicate when sufficient events had been recorded for

an actuator to be used.

Cons: Stato di Count globale nel main
flusso non necessario
Inoltre due variabili uguali (ne potrebbe bastare una?)

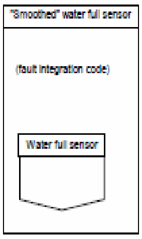


Nelle variabili di confine

- Better solution: the information flow of the main program would be solely in terms of “smoothed” sensor stream values. No unnecessary information flow would be taking place.
- Cons: duplicated code
- (senza cambiare figura) Such duplicated code might also be outside the SPARK boundary (because it would be in the body of the boundary variable packages which might perhaps not be implemented in SPARK)

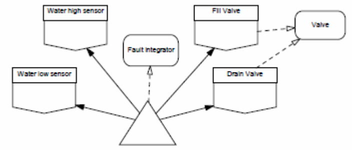
Astraendo le variabili di confine

- Analizzabile in SPARK
- Rimane la duplicazione
- Tecnica utile



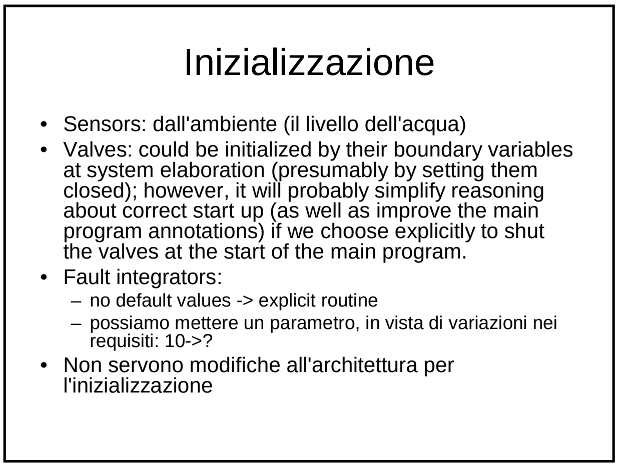
Come locale del main

- main program’s annotations in terms of the sensors and actuators only
- We may want to reason that the requirement for delay in operating the actuators is implemented correctly:
  - better making the fault counting visible within the main program rather than hiding it in the boundary



Nessuna duplicazione
Possibile verifica

Grafo aciclico



Inizializzazione

- Sensors: dall'ambiente (il livello dell'acqua)
- Valves: could be initialized by their boundary variables at system elaboration (presumably by setting them closed); however, it will probably simplify reasoning about correct start up (as well as improve the main program annotations) if we choose explicitly to shut the valves at the start of the main program.
- Fault integrators:
  - no default values -> explicit routine
  - possiamo mettere un parametro, in vista di variazioni nei requisiti: 10->?
- Non servono modifiche all'architettura per l'inizializzazione