Interreg North Sea
Co-funded by the European Union

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

A Complete Formal Specification and Verification of the BesW software control system of the Maeslant Storm Surge Barrier

Adrian Beers, Jore Booy, Jan Friso Groote, Johan van den Bogaard, Mark Bouwman

The Maeslant Barrier is a storm surge barrier that protects Rotterdam and its harbour from storm surges in the North Sea. The BesW software control system is responsible for all the movements of the barrier except for pushing and pulling it (Figure 3).

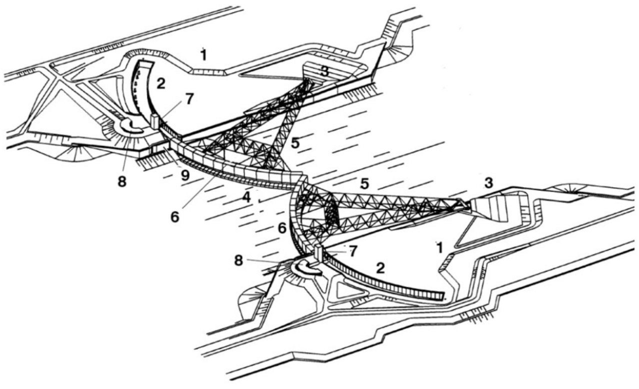

Figure 1: The Maeslant Barrier.

The model is around 5.000 lines of mCRL2 code (see Figure 4) and has a state space size of $4.98 * 10^{14}$. It contains the full behaviour of the software control system, including failing hardware, and rest and testing modes (ITO). The model is based on the current specification document of BesW, which includes 15.000 lines of PLC code. We specified 40 properties in the modal μ-calculus (see Figure 5). We were able to verify each property within forty minutes.
The process of verification is described in Figure 2. We verified the properties against the model the mCRL2 specification into a Linear Process Specification (LPS), which LPS is combined with a modal formula into a Parameterized Boolean Equation System (PBES). Then, we solve it.

$$\begin{tikzcd} \text{mcrl2} \arrow[d] & \text{mcf} \arrow[dr] & & \\ \text{lps} \arrow[rr] & & \text{pbes} \arrow[r] & \text{T/F} \end{tikzcd}$$

Figure 2: Verifying properties against the model.

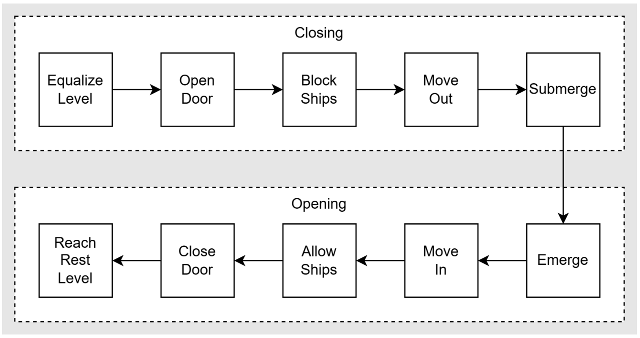

Figure 3: The operational processes.

```
proc Process_HI_OpenDoor(processMode: ProcessMode,
     system: MainSystem) =
  (processMode == active) -> (
    (system == dock) -> (
      Dock_Door_ChooseWinchOpen_InOrder(
           requireCatchTimer=false).
      Dock_Door_Open(requireDevicesReady=true).
      Dock_Door_CheckOpened
    ) +
    (system == besl) -> ( skip ) +
    (system == joint) -> ( Joint_Jack_Control ) +
    (system == ballast) -> (
      Ballast_Comps_DrainWithPumps_General.
      Ballast_Comps_DrainWithRestPumps
    )
  ) + ...
```

Figure 4: A fragment of mCRL2 specification.

```
[ true*.
  internal_controlStart(operational, processOpenDoor,
       active).
  (!internal_controlEnd)*
]
(forall doorOpenedSensors: List(List(Bool)).
    val(#doorOpenedSensors == 3 && (forall i: Nat.
    i < 3 => #(doorOpenedSensors.i) == 2)) =>
  [
    input_dockDoorOpened(doorOpenedSensors) .
    (!internal_controlEnd)* .
    internal_controlEnd . (!internal_controlEnd)*
  ]
  (forall processMode: ProcessMode.
    [internal_controlStart(operational, processOpenDoor,
          processMode)]
    val((processMode == finished)
      ==
      (exists i,j,i’,j’: Nat. (i < 3 && j < 2 && i’ < 3
           && j’ < 2 && i != i’ && doorOpenedSensors.i.j
           && doorOpenedSensors.i’.j’)
) ) ) )
```

Figure 5: The process Open Door will finish if and only if at least two limit switches indicate the door is opened.

Formal System Analysis, Mathematics and Computer Science