Supervisor Synthesis Models for the Oisterwijksebaan Bridge

March 30, 2020

1 System overview

An overview of the Oisterwijksebaan bridge (OBB) is given in Fig. 1.

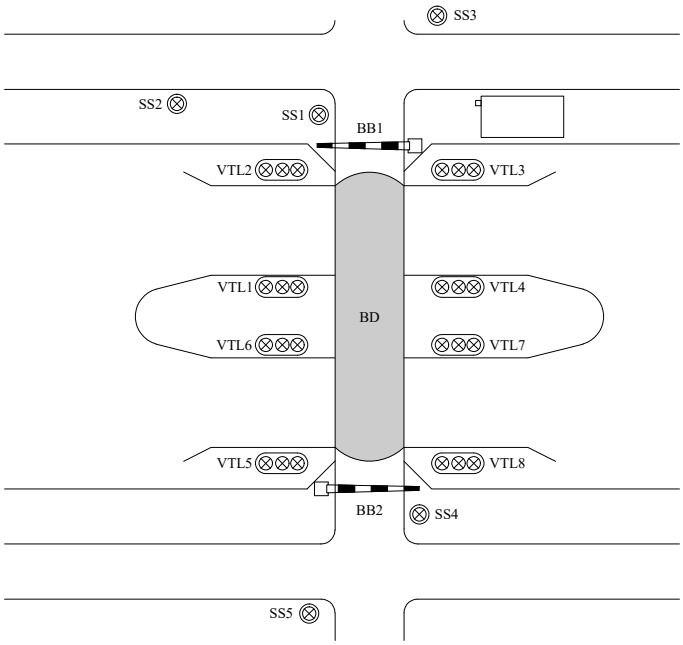

Figure 1: Overview of the OBB.

The physical system can be divided into the following subsystems subsystems:

- Stop signs - denoted by SS1 - SS5
- Boom barriers - denoted by BB1 and BB2
- Vessel traffic lights - denoted by VTL1 - VTL8
- Locking Mechanism - denoted by LM
- Brake - denoted by BR
- Bridge deck - denoted by BD

The SSs, BBs, VTLs, and BD are also shown in Fig. 1. Aside from the physical subsystems, a graphical user-interface is used to operate the bridge.

2 Plant model

A plant model has been developed for the OBB. For the plant model, each subsystem is further divided into components, for which component models are made. In the following subsections, for each subsystem the sensors and actuators are listed, and the component models are provided.

2.1 Stop signs

The OBB contains five stop signs, three are positioned at the east side of the bridge and two are positioned at the west side of the bridge. The PLC has one output to enable all five stop signs simultaneously. Additionally, each stop sign has a sensor to measure if the lamp is enabled, which are inputs for the PLC. The I/O list for the stop-signs subsystem is provided in Table 1.

Table 1: I/O of the stop sign.

S / A	**Description**	**1**	**0**	**EFA name**
Actuator	Stop signs	Off	On	`SS.A`
Sensor	Stop sign 1	On	Off	`SS.S1`
Sensor	Stop sign 2	On	Off	`SS.S2`
Sensor	Stop sign 3	On	Off	`SS.S3`
Sensor	Stop sign 4	On	Off	`SS.S4`
Sensor	Stop sign 5	On	Off	`SS.S5`

The component models for the actuator and the component model for a sensor are given in the left-hand side and right-hand side of Fig. 2, respectively. The location names correspond with the signals values given in Table 1.

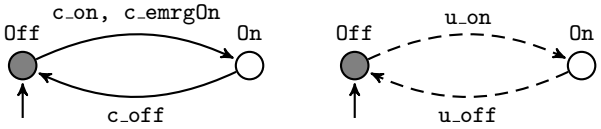

Figure 2: Model of the stop-sign actuator (left) and the stop-sign sensor (right).

2.2 Boom barrier

The OBB consists of two boom barriers, one at the east side of the bridge and one at the west side of the bridge. Each boom barrier is modeled as a separate (identical) subsystem.

A boom barrier can rotate upwards (towards the open position) and downwards (towards the closed position), which are the two outputs of the PLC. Additionally, it is equipped with four sensors, two sensors measure the fully open position and two sensors measure the fully closed position. One sensor is used as a safety sensor to measure if a boom barrier reached its end position, the other sensor (the not-open sensor or not-closed sensor) is used to measure when the motor can be turned off. For safety reasons, two different sensors are used.

The boom barriers share an actuator that enables the warning lights on the barriers. A boom barrier contains three lights, one at the rotation point, one at the middle, and one at the end. Either lamp 1 and 3 (mode 1), or lamp 1 and 2 (mode 2) are enabled, this is an additional PLC output.

The I/O list of the boom barriers is provided in Table 2. In the 'EFA name' column, X equals 1 or 2, for boom barrier 1 or boom barrier 2, respectively.

Table 2: I/O of the boom barriers.

S / A	**Description**	**1**	**0**	**EFA name**
Actuator	Open boom barrier	Open	Idle	BBX.A
Actuator	Close boom barrier	Close	Idle	
Actuator	Lights	Off	On	BB.L
Actuator	Light alternation	Mode 1	Mode 2	BB.Alternate
Sensor	Open	Open	Not open	BBX.Open
Sensor	Closed	Closed	Not closed	BBX.Closed
Sensor	Not open	Not open	Open	BBX.NotOpen
Sensor	Not closed	Not closed	Closed	BBX.NotClosed

The component models for the open and close actuator are combined. The combined component model is shown in the top left-hand side of Fig. 3. The boom barrier can never open and close simultaneously. There is a differentiation between a regular stop and an emergency stop, this allows to specify different requirements for them. The models of the sensors are as depicted on the top right-hand side and on the middle of Fig. 3. The safe situation is where the boom barrier is in the upward position, and the motor is idle, the marked states are chosen accordingly. The models for the light actuator and the mode alternator are as depicted on the bottom of Fig. 2.

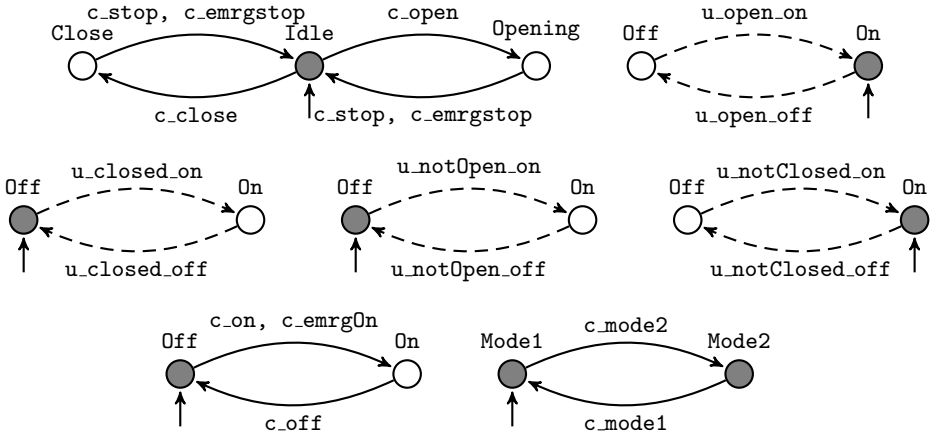

Figure 3: Model of the boom-barrier actuator (top left), the open sensor (top right), the closed sensor (middle left), the not-open sensor (middle middle), the not-closed sensor (middle right), the barrier lights actuator (bottom left) and the alternation (bottom right).

2.3 Vessel traffic light

The OBB consists of eight vessel traffic lights (VTLs). Each VTL is modeled as a separate (identical) subsystem. Each VTL consists of three lamps, red, green, and red. The bottom red lamp is referred to as the red2 lamp. Additionally, each lamp is equipped with a sensor that measures if the lamp is on.

The I/O list of a VTLs is provided in Table 3. In the 'EFA name' column, X can be $1, \ldots, 8$, for a specific VTL.

Table 3: I/O of the vessel traffic lights.

S / A	**Description**	**1**	**0**	**EFA name**
Actuator	Red	Off	On	`VTL`X`.A` `VTL`X`.Activated`
Actuator	Green	On	Off	
Actuator	Red2	Off	On	
Sensor	Red	On	Off	`VTL`X`.Red`
Sensor	Green	On	Off	`VTL`X`.Green`
Sensor	Red2	On	Off	`VTL`X`.Red2`

The component model for the VTL actuator is given in the top left-hand side of Fig. 4. For the actuator model, all three lamps are combined such that each location represents a specific sign aspect. Moreover, another EFA is used for activating and deactivating the traffic light, in case of a fault, given on the top right-hand side in Fig. 4. The model for each sensors is as depicted on the bottom of Fig. 4.

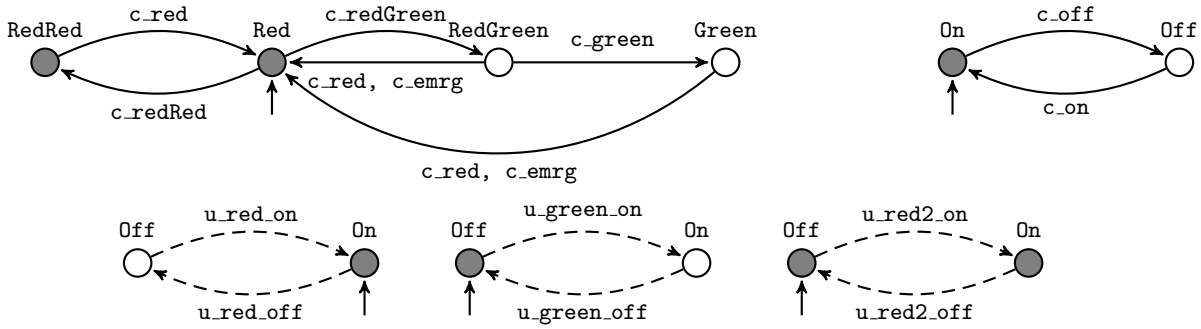

Figure 4: Model of the VTL actuator (top left), the VTL activation (top right), the red sensor (bottom left), the green sensor (bottom middle), and the red2 sensor (bottom right).

2.4 Bridge braking mechanism

The brake of the bridge is used to keep the bridge in its rest position. The brake consists of a single actuator and a single sensor that measures if the brake is released. Furthermore, it the bridge motor sends a signal when the brake can be released. The I/O list of the brake subsystem is provided in Table 4.

Table 4: I/O of the brake subsystem.

S / A	**Description**	**1**	**0**	**EFA name**
Actuator	Brake	Release	Apply	`BR.A`
Sensor	Brake	Released	Applied	`BR.S`
Sensor	Release brake	Release brake	Apply brake	`BR.Release`

The component model of the brake actuator, the brake sensor, and the brake-release sensor are depicted on the left-hand side, middle, and right-hand side of Fig. 5, respectively.

Figure 5: Model of the brake actuator (left), the brake sensor (middle), and the brake release sensor (right).

2.5 Bridge locking mechanism

The locking mechanism is responsible for lowering the bridge before rotation (unlocking) and raising the bridge after rotation (locking). To this end, a pump and two hydraulic valves are available. Two end-position sensors are available to measure the locked position and the unlocked position.

The I/O list of the locking mechanism is provided in Table 5.

Table 5: I/O of the locking mechanism subsystem.

S / A	**Description**	**1**	**0**	**EFA name**
Actuator	Pump	On	Off	`LM.Pump`
Actuator	Valve locking	Locking	Idle	`LM.Valve`
Actuator	Valve unlocking	Unlocking	Idle	
Sensor	Locked	Not locked	Locked	`LM.Locked`
Sensor	Unlocked	Not unlocked	Unlocked	`LM.Unlocked`

The component model of the the hydraulic valve actuator and the pump actuator are depicted on the top left-hand side and top right-hand side of Fig. 6, respectively. Both component models differentiate between a regular stop and an emergency stop. The locking mechanism sensors are as shown on the bottom of Fig. 6.

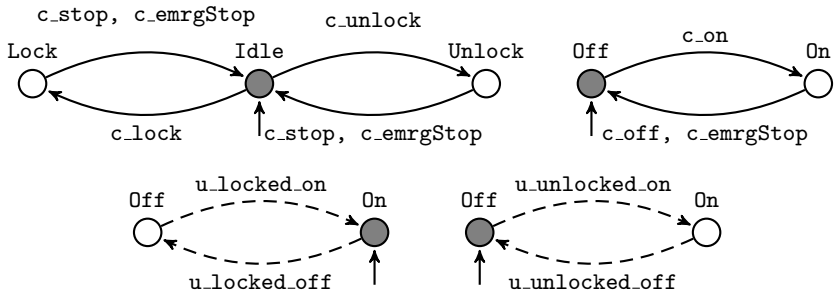

Figure 6: Model of the valve actuators (top left), the pump actuator (top right), the locked sensor (bottom left), and the unlocked sensor (bottom right).

2.6 Bridge rotating mechanism

The bridge deck of the OBB can be rotated to let vessels pas the bridge. In order to do so, an electric motor is present. The PLC can control three things, 1) the power of the motor (either on or off), 2) the rotation direction of the motor (either open or close), and 3) the speed of the motor. The speed is an analog signal that ranges from 0 to 100%. The bridge

can be at stand still (0%), slow speed (19%) or full speed (100%). The speeds of the bridge should be lowered when the bridge is almost open or almost closed, such that it does not collide at full speed with the side.

To this end, six position sensors measure the rotation of the bridge (closed, before closed, before before closed, before before open, before open, and open). Based on these sensors, the speed is controlled. A second closed sensor is installed for safety, to be sure that the bridge is closed.

The full I/O list is provided in Table 6.

Table 6: I/O of the bridge deck subsystem.

S / A	**Description**	**1**	**0**	**EFA name**
Actuator	Motor	On	Off	`BD.Motor`
Actuator	Direction open	Open	Idle	`BD.Direction`
Actuator	Direction close	Close	Idle	
Actuator	Speed	Analog		`BD.Speed`
Sensor	Closed	Closed	Not Closed	`BD.Closed2`
Sensor	Closed	Not closed	Closed	`BD.Closed`
Sensor	Before closed	Not bclosed	Bclosed	`BD.BClosed`
Sensor	Before before closed	Not bbclosed	Bbclosed	`BD.BBClosed`
Sensor	Before before open	Not bbopen	Bbopen	`BD.BBOpen`
Sensor	Before open	Not bopen	Bopen	`BD.BOpen`
Sensor	Open	Not open	Open	`BD.Open`

The actuators are modeled as the automata depicted in Fig. 7. The sensors are modeled as the sensor model shown in Fig. 2.

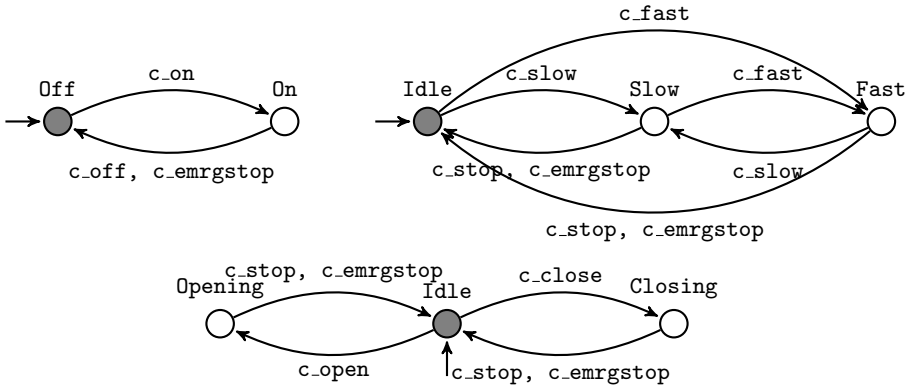

Figure 7: Model of the motor (top left), the speed actuator (top right), and the direction actuator (bottom).

2.7 Bridge commands

A GUI is available to provide commands from the operator to the supervisory controller. It is only possible for one command to be given at the same time. The available commands are listed in Table 7. The location name of the EFA `GUI` where this command is active, is given as well. Not all commands can be given at any time, for this a flow diagram is included in appendix A, see Fig. 15.

Table 7: Commands available for the bridge.

Command	**Desired reaction**	`GUI` location
Close land traffic	Switch on stop signs and turn on barrier lights	`CloseLT`
Close barriers	Lower barriers	`CloseBB`
Stop barriers	Stop barrier movement	`StopBB`
Open land traffic	Raise barriers and turn off stop signs and barrier lights	`OpenLT`
Open bridge	Unlock bridge, release brake, open bridge, apply brake	`OpenB`
Close bridge	Release brake, close bridge, apply brake, lock bridge	`CloseB`
Stop bridge	Stop locking, unlocking, or rotating, and apply brake	`StopB`

An emergency stop is also present. The component model is shown in Fig. 8.

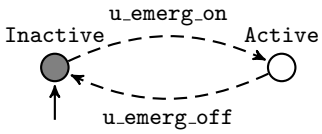

Figure 8: Model of the emergency stop.

2.8 VTL commands

From the GUI, commands to the VTLs can be given. For this, an icon is present on which the operator can click to switch between sign aspects. One icon is used to represent and give commands to two VTLs. The VTLs are grouped based on their position (see Fig. 1), VTL 1 and 2, VTL 3 and 4, VTL 5 and 6, and VTL 7 and 8 are grouped. These VTLs are grouped such that an operator cannot request two adjacent VTLs to display different aspects. The available commands are listed in Table 8.

Table 8: Commandos available for the VTLs.

Commando	**Desired reaction**	`VTLGUI` location
Red aspect	Top red light on, green light off, and bottom red lamp off	`Red`
Red-green aspect	Top red light on, green light on, and bottom red lamp off	`RedGreen`
Green aspect	Top red light off, green light on, and bottom red lamp off	`Green`
Double red aspect	Top red light on, green light off, and bottom red lamp on	`RedRed`

An override switch is available. When this switch is activated, the bridge can be closed even if the signs do not display a red aspect (in case of a failure). The component model of this switch is shown in Fig. 9.

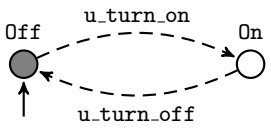

Figure 9: Model of the override switch.

2.9 Timers

Some processes in the bridge depend on timing information, for this, timers are used. A timer tracks if a certain condition c (e.g., the barriers are closed) has been satisfied for a specific time. In the discrete-event plant model this time is not included, but is included for simulation and implementation. The component model of a timer is depicted in Fig. 10.

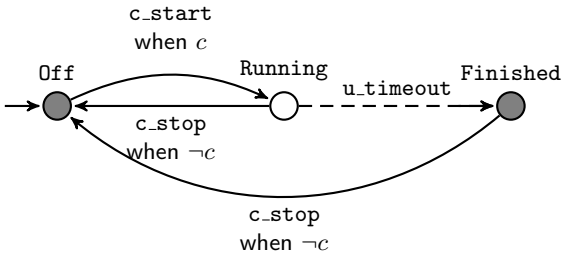

Figure 10: Model of the timer that is enabled when c is satisfied.

Processes that require timing are, e.g., closing the barriers after the stop signs have been turned on on for at least 16.4 seconds, applying extra pressure for a short amount of time after an actuator has reached its end position, and waiting for a short amount of time after the bridge has been unlocked and can start rotating. In Table 9 all timers, their condition, their duration, and their purpose are given.

Table 9: Timers in the OBB model.

Condition	**Duration** [s]	**Purpose**	**EFA name**
`S1.On` $\wedge$ `S4.On`	16.4	Safety	`SSsOnTimer`
`BB1.Open.On` $\wedge$ `BB2.Open.On`	3.0	Safety	`BBsOpenTimer`
`BB1.Open.On`	0.6	Motor stop	`BB1OpenTimer`
`BB1.Closed.On`	0.3	Motor stop	`BB1ClosedTimer`
`BB2.Open.On`	0.6	Motor stop	`BB2OpenTimer`
`BB2.Closed.On`	0.3	Motor stop	`BB2ClosedTimer`
`LM.Locked.On`	0.5	Pump stop	`LMLockedTimer`
`LM.Unlocked.On`	0.1	Pump stop	`LMUnlockedTimer100`
`LM.Unlocked.On`	3.0	Bridge opening	`LMLockedTimer3`
`BD.Closed.On`	3.0	FC stop	`BDClosedTimer`

3 Diagnoser model

In this section, for each subsystem, the diagnoser models are shown. A diagnoser is used to determine if a fault has occurred in the system. If a fault has occurred, this is signaled to the supervisory controller, and thereafter, to the operator.

3.1 Stop signs

In a stop sign, two faults can occur, 1) a lamp can stay off when it should turn on, and 2) a lamp can stay on when it should turn off. For the first fault, it can be that the actuator is broken (i.e., it does not correctly turn on the lamp), or the sensor is broken (i.e., the lamp is on, but it is measured incorrectly). Differentiation between these two faults is not possible. A similar situation exists for the second fault.

The diagnoser uses two signals. Signal Q indicates whether the actuator is on and signal I indicates whether the sensor is on. t_f is the fault time, if the signals of Q and I do not match for more than t_f seconds, a fault has occurred. If Q is enabled and there is a fault, this indicates that the lamp stays off. Opposite, when Q is disabled and there is a fault, this indicates that the lamp stays on. The locations where a fault has been diagnosed are colored red. The diagnoser is shown in Fig. 11.

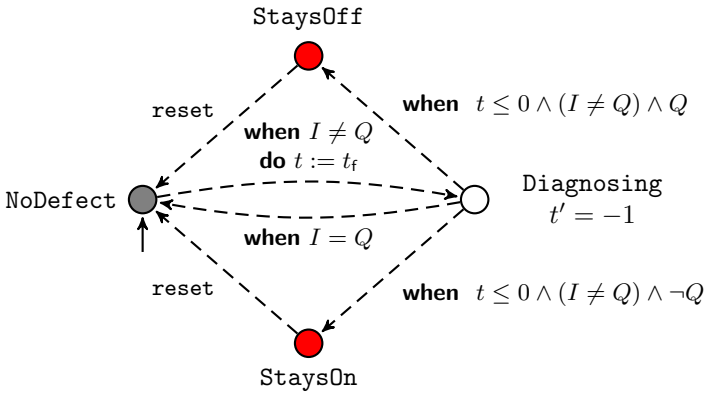

Figure 11: Model of the diagnoser for a stop sign fault.

3.2 Boom barrier

In a boom barrier, three faults can occur, 1) the barrier can get stuck, 2) the barrier can undesirably leave the open position, and 3) the barrier can undesirably leave the closed position.

The diagnosers use four signals. Signal $Q_{\mathtt{o}}$ and $Q_{\mathtt{c}}$ indicate whether the actuator is moving in the open or moving in the closed direction, respectively. Signals $I_{\mathtt{o}}$ and $I_{\mathtt{c}}$ indicate that the barrier is in the fully open position or fully closed position, respectively.

To determine whether the barrier is stuck, it is measured if the movement takes more than t_{o} or t_{c} seconds if moving from the closed to open position or vice versa, respectively. The diagnoser for the stuck barrier fault is shown in Fig. 12.

To determine whether the barrier undesirably leaves the open position, it is measured whether the open position sensor switches off, without the closed actuator being activated. If this is the case for at least $t_{\mathtt{f}}$ seconds, a fault has occurred. Such a minimum time is used to reduce the number of false positive fault diagnoses. The diagnoser for undesirably leaving the open position is shown in Fig 13.

The diagnoser for undesirably leaving the closed position is similar to the diagnoser shown in Fig. 13. Differently, the signals $I_{\mathtt{o}}$ and $Q_{\mathtt{c}}$ are substituted by signals $I_{\mathtt{c}}$ and $Q_{\mathtt{o}}$, respectively.

3.3 Vessel traffic light

In a vessel traffic light, six faults can occur, 1) the top red lamp can stay off when it should turn on, 2) the top red lamp can stay on when it should turn off, 3) the green lamp can stay off when it should turn on, 4) the green lamp can stay on when it should turn off, 5) the bottom red lamp can stay off when it should turn on, and 6) the bottom red lamp can stay on when it should turn off.

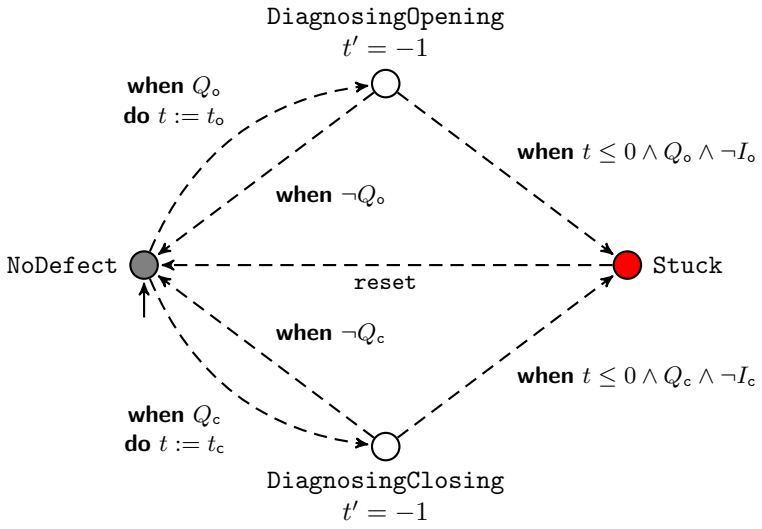

Figure 12: Model of the diagnoser for a stuck barrier fault.

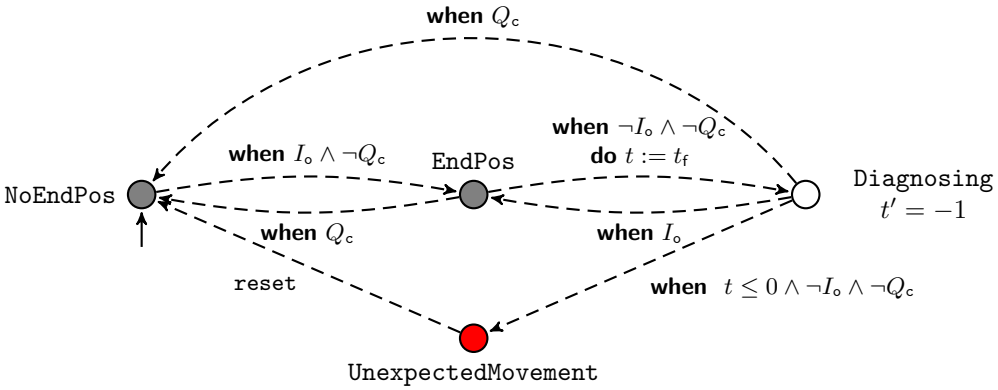

Figure 13: Model of the diagnoser for a undesirably leaving the open position fault.

The diagnosers use six signals, three signals for the lamp actuations and three signals for the measurements. The diagnosers are similar to the diagnoser shown in Fig. 11.

3.4 Bridge braking mechanism

The bridge braking mechanism does not have any diagnosers.

3.5 Bridge locking mechanism

In the locking mechanism, two faults can occur 1) the mechanism gets stuck and 2) the locking mechanism undesirably unlocks.

For the first fault, a similar diagnoser as the one shown in Fig. 12 is used. For the second fault, a similar diagnoser as the one shown in Fig. 13 is used.

3.6 Bridge rotating mechanism

In the bridge rotating mechanism, four faults can occur 1) the bridge undesirable leaves the open position, 2) the bridge undesirable leaves the closed position, 3) the bridge does not

decelerate while opening, and 4) the bridge does not decelerate while closing.

For the first and second faults, a similar diagnoser as the one show in Fig. 13 is used.

For measuring whether the bridge correctly decelerates while opening, the time it between the before-before-opening sensor (denoted by $I_{\texttt{bbo}}$) switches on and the before-opening sensor (denoted by $I_{\texttt{bo}}$) switches on is measured. If this time is less than $t_{\texttt{f}}$ seconds, a fault has been diagnosed. A timer is used to measure if the before-before-open sensor is switched on long enough, denoted by signal `TimerBBO.Finished` (shown in Fig. 10). When $I_{\texttt{bo}}$ switches on, when the timer is not finished, a fault has been diagnosed.

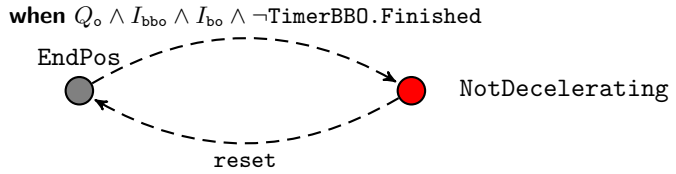

Figure 14: Model of a diagnoser for a not decelerating fault.

For measuring whether the bridge correctly decelerates while close, a similar diagnoser as in Fig. 14 is used.

4 Requirements model

A requirements model has been developed for the OBB. For the model, requirements are divided into categories, safety requirements, functional requirements, and operator commands.

4.1 Safety requirements

In this section all safety requirements for the OBB are listed.

4.1.1 Stop signs and boom barriers

The boom barriers are only allowed to close when the stop signs have been on for 16.4 seconds or longer. If SS2, 3, or 5 stay of, because they are broken, the barriers can still be closed safely. Whenever SS1 or SS4 is broken, it is unsafe to close barriers.

```
BB1.A.c_close needs SS1.On                    and
                    (SS2.On or SS2.StaysOff)  and
                    (SS3.On or SS3.StaysOff)  and
                     SS4.On                   and
                    (SS5.On or SS5.StaysOff)  and
                     SSsOnTimer.Finished;
BB2.A.c_close needs SS1.On                    and
                    (SS2.On or SS2.StaysOff)  and
                    (SS3.On or SS3.StaysOff)  and
                     SS4.On                   and
                    (SS5.On or SS5.StaysOff)  and
                     SSsOnTimer.Finished;
```

4.1.2 Boom barriers and bridge

The bridge is only allowed to start moving, i.e., unlocking, rotating, and releasing the brake, when the boom barriers are closed.

```
BR.A.c_release       needs BB1.Closed.On and BB2.Closed.On;
LM.Pump.c_on         needs BB1.Closed.On and BB2.Closed.On;
LM.Valve.c_lock      needs BB1.Closed.On and BB2.Closed.On;
LM.Valve.c_unlock    needs BB1.Closed.On and BB2.Closed.On;
BD.Motor.c_on        needs BB1.Closed.On and BB2.Closed.On;
BD.Direction.c_open  needs BB1.Closed.On and BB2.Closed.On;
BD.Direction.c_close needs BB1.Closed.On and BB2.Closed.On;
```

4.1.3 Bridge and vessel traffic lights

The VTLs are only allowed to show a green aspect, when the bridge is fully open and the brake is applied. The following model is for VTL1, and is similar for all other VTLs.

```
VTL1.A.c_green needs BD.Open.On and BR.S.Applied and BR.A.Apply;
```

4.1.4 Vessel traffic lights and other vessel traffic lights

The VTLs are only allowed to show a green aspect, when the two opposite signs show a red or double red aspect. The only exception is when one of the two opposite signs is broken, then only one of the two has to show a red aspect. The following model is for VTL1, and is similar for all other VTLs.

```
VTL1.A.c_green needs
   ((VTL3.Red.On and VTL3.Green.Off) or (VTL4.Red.On and VTL4.Green.Off)) and
   (not VTL3.Red.StaysOff            or  not VTL4.Red.StaysOff);
```

The VTLs are only allowed to show a red-green aspect when the opposite VTL does not show a red green aspect. The following model is for VTL1, and is similar for all other VTLs

```
VTL1.A.c_redGreen needs not (VTL3.S_Red.On and VTL3.S_Green.On) and
                         not (VTL4.S_Red.On and VTL3.4_Green.On);
```

4.1.5 Vessel traffic lights and bridge

The bridge is allowed to start closing when the VTLs show a red or double red aspect. The only exception is when one of the two signs in a waterway is broken, then only one of the two has to show a red aspect. In case both VTLs are broken the override switch must be used. For brevity, a variable is introduced that indicates whether enough lamps display a red aspect.

```
redAspectsShown =
   ((VTL1.Red.On and VTL1.Green.Off) or (VTL2.Red.On and VTL2.Green.Off)) and
   (not VTL1.Red.StaysOff            or not VTL2.Red.StaysOff)            and
   ((VTL3.Red.On and VTL3.Green.Off) or (VTL4.Red.On and VTL4.Green.Off)) and
   (not VTL3.Red.StaysOff            or not VTL4.Red.StaysOff)            and
   ((VTL5.Red.On and VTL5.Green.Off) or (VTL6.Red.On and VTL6.Green.Off)) and
   (not VTL5.Red.StaysOff            or not VTL6.Red.StaysOff)            and
```

```
   ((VTL7.Red.On and VTL7.Green.Off) or (VTL8.Red.On and VTL8.Green.Off)) and
   (not VTL7.Red.StaysOff)           or not VTL8.Red.StaysOff))

BR.A.c_release       needs redAspectShown or OverrideSwitch.On;
LM.Pump.c_on         needs redAspectShown or OverrideSwitch.On;
LM.Valve.c_lock      needs redAspectShown or OverrideSwitch.On;
LM.Valve.c_unlock    needs redAspectShown or OverrideSwitch.On;
BD.Motor.c_on        needs redAspectShown or OverrideSwitch.On;
BD.Direction.c_open  needs redAspectShown or OverrideSwitch.On;
BD.Direction.c_close needs redAspectShown or OverrideSwitch.On;
```

4.1.6 Bridge and boom barriers

The boom barriers are only allowed to open when the bridge is fully closed and the locking mechanism is locked.

```
BB1.A.c_open needs BD.Closed.On and LM.Locked.On;
BB2.A.c_open needs BD.Closed.On and LM.Locked.On;
```

4.1.7 Boom barriers and stop signs

The stop signs are only allowed to turn off, when the boom barriers are fully open for 3 seconds.

```
SS.A.c_off needs BB1.Open.On and BB2.Open.On and BBsOpenTimer.Finished;
BB.L.c_off needs BB1.Open.On and BB2.Open.On and BBsOpenTimer.Finished;
```

4.1.8 Unexpected movements and stop signs

When the barriers or the bridge is unexpectedly moving, the stop signs should enable to warn land traffic.

```
SS.A.c_emrgOn needs BB1.UnexpectedClose or BB2.UnexpectedClose or
                    BD.UnexpectedOpen   or LM.UnexpectedUnlock;
BB.L.c_emrgOn needs BB1.UnexpectedClose or BB2.UnexpectedClose or
                    BD.UnexpectedOpen   or LM.UnexpectedUnlock;

SS.A.c_off    needs not BB1.UnexpectedClose and not BB2.UnexpectedClose and
                    not BD.UnexpectedOpen   and not LM.UnexpectedUnlock;
BB.L.c_off    needs not BB1.UnexpectedClose and not BB2.UnexpectedClose and
                    not BD.UnexpectedOpen   and not LM.UnexpectedUnlock;
```

4.1.9 Dangerous vessel traffic light aspects

If the top red light of a VTL fails, the red-green aspect will appear as a green aspect, this is dangerous. Therefore, a requirement is added that states that a VTL can turn off when its red lamp stays off. The following model is for VTL1, and is similar for all other VTLs

```
VTL1.Activated.c_off needs VTL1.Red.StaysOff
VTL1.Activated.c_on  needs not VTL1.Red.StaysOff
```

4.1.10 Emergency stop

In case of an emergency, all movements have to be stopped and all VTLs should display a red or double red aspect. An emergency stop has to happen in the emergency stop is active, a bridge declaration fault, or an unexpected bridge close fault. Additionally, if the bridge operator gives the command to stop, the movements have to stop and the aspects should turn red.

```
emergency = EmergencyStop.Active or BD.NotDecelerating or BD.UnexpectedClose;

BB1.A.c_emrgStop        needs emergency or GUI.StopBB;
BB2.A.c_emrgStop        needs emergency or GUI.StopBB;
BR.A.c_emrgStop         needs emergency or GUI.StopB;
LM.Valve.c_emrgStop     needs emergency or GUI.StopB;
LM.Pump.c_emrgStop      needs emergency or GUI.StopB;
BD.Motor.c_emrgStop     needs emergency or GUI.StopB;
BD.Direction.c_emrgStop needs emergency or GUI.StopB;
BD.Speed.emrgStop       needs emergency or GUI.StopB;
VTL1.A.emrgStop         needs emergency or GUI.StopB;
```

Also, as long as there is an emergency, no movement may start and no red-green aspect may be shown:

```
BB1.A.c_open         needs not emergency;
BB1.A.c_close        needs not emergency;
BB2.A.c_open         needs not emergency;
BB2.A.c_close        needs not emergency;
BR.A.c_release       needs not emergency;
LM.Valve.c_unlock    needs not emergency;
LM.Valve.c_lock      needs not emergency;
LM.Pump.c_on         needs not emergency;
BD.Motor.c_on        needs not emergency;
BD.Direction.c_open  needs not emergency;
BD.Direction.c_close needs not emergency;
BD.Speed.c_fast      needs not emergency;
BD.Speed.c_slow      needs not emergency;
VTL1.A.c_redGreen    needs not emergency;
VTL1.A.c_green       needs not emergency;
```

Additionally, some actuators are not directly activated via a GUI command, and have to be blocked when the GUI gives the stop command (all other are automatically stopped, because they need a GUI start command to begin moving):

```
BR.A.c_release    needs not GUI.StopB;
LM.Pump.c_on      needs not GUI.StopB;
BD.Motor.c_on     needs not GUI.StopB;
BD.Speed.c_fast   needs not GUI.StopB;
BD.Speed.c_slow   needs not GUI.StopB;
VTL1.A.c_redgreen needs not GUI.StopB;
VTL1.A.c_green    needs not GUI.StopB;
```

4.2 Functional requirements

4.2.1 Boom barriers

A boom barrier can only close if it is not already closed and not stuck.

```
BB1.A.c_close needs BB1.NotClosed.On and not BB1.Stuck;
BB2.A.c_close needs BB2.NotClosed.On and not BB2.Stuck;
```

A boom barrier can only open if it is not already open and not stuck.

```
BB1.A.c_open needs BB1.NotOpen.On and not BB1.Stuck;
BB2.A.c_open needs BB2.NotOpen.On and not BB2.Stuck;
```

A boom barrier can stop regularly when it is closed for 300ms, open for 600ms, or when it is stuck.

```
BB1.A.c_stop needs
   (BB1.Closing and BB1.ClosedTimer.Finished and BB1.NotClosed.Off) or
   (BB1.Opening and BB1.OpenTimer.Finished   and BB1.NotOpen.Off)   or
   (BB1.Stuck);
BB2.A.c_stop needs
   (BB2.Closing and BB2.ClosedTimer.Finished and BB2.NotClosed.Off) or
   (BB2.Opening and BB2.OpenTimer.Finished   and BB2.NotOpen.Off)   or
   (BB2.Stuck);
```

4.2.2 Bridge braking mechanism

The release brake sensor sends a signal when the breaks should be applied and released, as follows:

```
BR.A.c_release needs BR.Release.On;
BR.A.c.apply   needs BD.Release.Off;
```

4.2.3 Bridge locking mechanism

The locking mechanism can only unlock when it is not already unlocked and the bridge motor is not rotating.

```
LM.Valve.c_unlock needs LM.Unlocked.Off and BD.Direction.Idle;
```

The locking mechanism can only lock when it is not already locked, both bridge position sensors measure the closed position, the brake is applied, and the bridge motor is not rotating.

```
LM.Valve.c_lock needs LM.Locked.Off and BD.Closed.On      and
                      BD.Closed2.On and BD.Direction.Idle and BR.S.Applied;
```

The locking mechanism can only stop regularly when it is unlocked for 100ms or locked for 500ms.

```
LM.Valve.c_stop needs
   (LM.Valve.Unlock and LMUnlockedTimer100.Finished and LM.Unlocked.On) or
   (LM.Valve.Lock   and LMLockedTimer.Finished      and LM.Locked.On);
```

The pomp follows the hydraulic valves and automatically starts and stops depending on the valve's position.

```
LM.Pump.c_on  needs not LM.Valve.Idle;
LM.Pump.c_off needs     LM.Valve.Idle;
```

4.2.4 Bridge rotating mechanism

The motor can start whenever it is allowed by the safety requirements. The motor should stop when it is no longer allowed, i.e., when the barriers open or not enough VTL shows the red aspect (when the override switch is not switched on).

```
BD.Motor.c_off needs not(BB1.Closed.On and BB2.Closed.On) or
                     not(redAspectsShown or OverrideSwitch.On);
```

The bridge deck may start rotating in the open direction when the bridge is not already open and the locking mechanism has been unlocked for 3s.

```
BD.Direction.c_open needs not BD.Open.On and LMUnlockedTimer3.Finished and
                          LM.Unlocked.On;
```

The bridge deck may start rotating in the close direction when, the bridge is not already closed and the locking mechanism has been unlocked.

```
BD.Direction.c_close needs not BD.Closed.On and LM.Unlocked.On;
```

The bridge deck may stop rotating regularly when it is open or when it is closed for 3s.

```
BD.Direction.c_stop needs
   (BD.Direction.Opening and BD.Open.On) or
   (BD.Direction.Closing and BD.Closed.On and BDClosedTimer.Finished)
```

The speeds of the motor can be set to fast when the bride is opening and the bridge is not in the positions open, before open, or before before open. Additionally, the motor can be set to fast when the bridge is closing and the bridge is not in the positions closed, before closed, or before before closed.

```
BD.Speed.c_fast needs (BD.Direction.Opening and BD.Open.Off   and
                       BD.BOpen.Off         and BD.BBOpen.Off) or
                      (BD.Direction.Closing and BD.Closed.Off and
                       BDSBClosed.Off       and BD.BBClosed.Off);
```

The speed of the motor can be set to slow when fast speed is not allowed and the bridge is opening and the bridge is not in the position open. Additionally, the motor can be set to slow when the bridge is closing and the bridge is not in the position closed.

```
BD.Speed.c_slow needs not ((BD.Direction.Opening and BD.Open.Off   and
                            BD.BOpen.Off         and BD.BBOpen.Off) or
                           (BD.Direction.Closing and BD.Closed.Off and
                            BD.BClosed.Off       and BD.BBClosed.Off)) and
                      ((BD.Direction.Opening and BD.Open.Off) or
                       (BD.Direction.Closing and BD.Closed.Off));
```

The speed should be set to zero when the direction actuator is idle.

```
BD.Speed.c_off needs BD.Direction.Idle;
```

4.3 Operator requirements

An operator is responsible for activating a process, via the GUI. The following requirements list when a specific actuation may start, based on the commands given via the GUI.

Stopping land traffic:

```
SS.A.c_on            needs GUI.CloseLT;
BB.L.c_on            needs GUI.CloseLR;
```

Closing boom barriers:

```
BB1.A.c_close        needs GUI.CloseBB;
BB2.A.c_close        needs GUI.CloseBB;
```

Opening the bridge:

```
BD.Direction.c_open  needs GUI.OpenB;
LM.Valve.c_unlock    needs GUI.OpenB;
```

Closing the bridge:

```
BD.Direction.c_close needs GUI.CloseB;
LM.Valve.c_lock      needs GUI.CloseB;
```

Opening boom barriers:

```
BB1.A.c_open         needs GUI.OpenBB;
BB2.A.c_open         needs GUI.OpenBB;
```

Opening land traffic:

```
SS.A.c_off           needs GUI.OpenLT;
BB.L.c_off           needs GUI.OpenLT;
```

The VTLs are controlled as follows. The following model is for VTL1, and is similar for all other VTLs

```
VTL1.A.c_red         needs VTL12GUI.Red;
VTL1.A.c_redGreen    needs VTL12GUI.RedGreen;
VTL1.A.c_green       needs VTL12GUI.Green;
VTL1.A.c_redRed      needs VTL12GUI.RedRed;
```

A Bridge control window

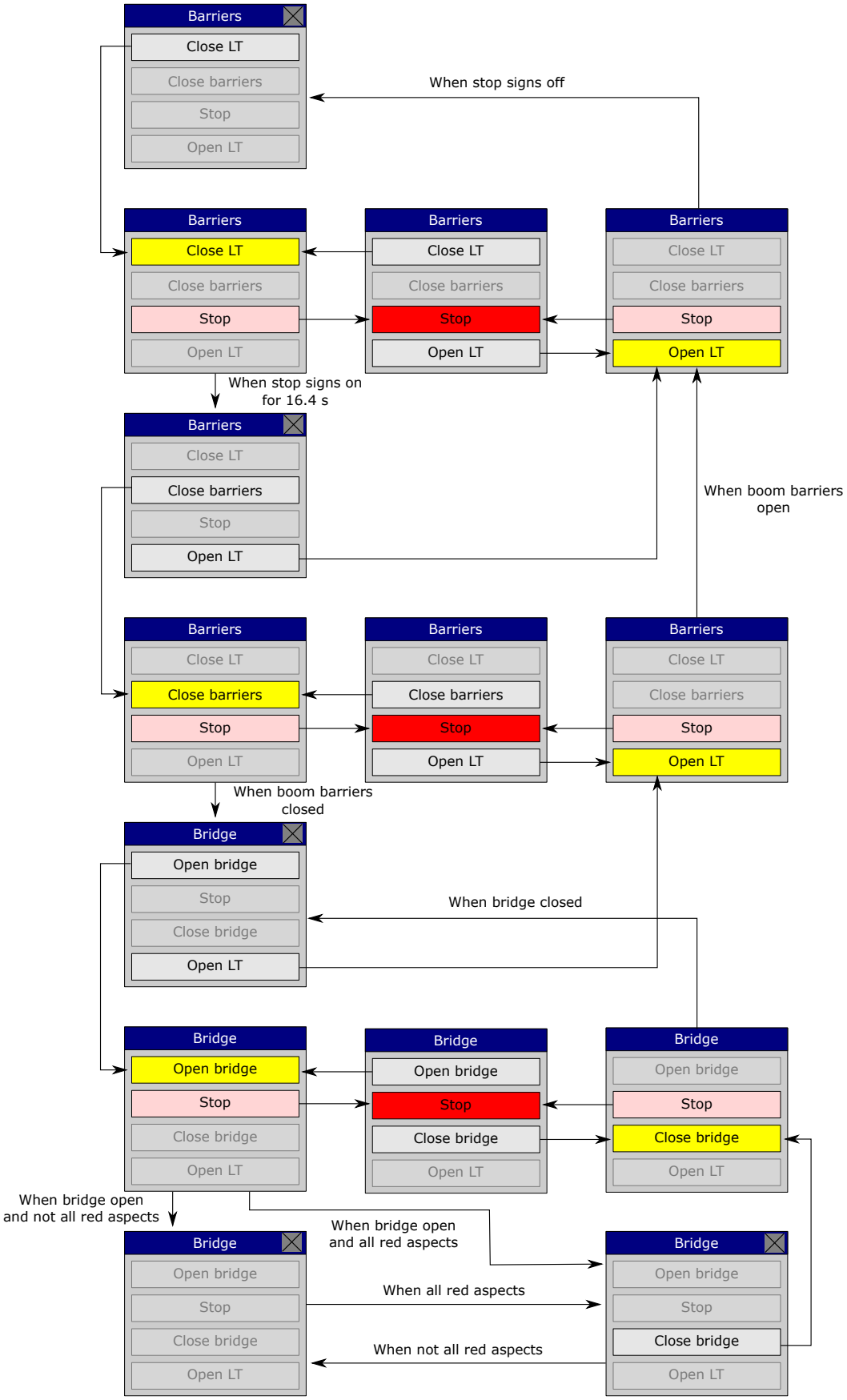

Figure 15: Control flow for bridge GUI.