Protocol Validation Homework Assignment
Movable Patient Support for an MRI Scanner

Mullai[1][2701556] and Neeraj Sathyan[1][2660192]

Vrije Universiteit Amsterdam `https://www.cs.vu.nl/`

1 Global Requirements

In this section, we formally state the global requirements of the Move-able Patient Support System (MPSP) which have been derived by the problem statement of the assignment.

- **P01** If a motor is on, the corresponding brake must not be applied.
- **P02:** The horizontal motor should be on only if the MPSP is docked.
- **P03:** The horizontal brake should not be applied if the MPSP is docked.
- **P04:** The bed must be in the rightmost position if the MPSP is undocked.
- **P05:** The horizontal brake must be applied if the MPSP is undocked.
- **P06:** The vertical brake must always be applied if the vertical motor is off.
- **P07:** The vertical brake must always be applied if the MPSP is docked.
- **P08:** In emergency mode, the horizontal brake must be released.
- **P09:** If the MPSP is undocked and uncalibrated, the bed should only be moved up and down.
- **P10:** The MPSP should not move above the uppermost position, below the lowermost position.
- **P11:** When the MPSP is docked and at the rightmost position, the `reset` button will set the standard height.
- **P12:** When the MPSP is undocked and the `reset` button is pressed, the standard height is forgotten.
- **P13:** If the MPSP is docked and calibrated, the vertical brakes are applied and the vertical motor turned off.

- **P14:** If the `stop` button is pressed, the MPSP goes into emergency mode.

- **P15:** When the MPSP goes into emergency mode, the horizontal brake is released so that the patient can be taken out of the scanner.

- **P16:** When the MPSP is in emergency mode, the `resume` button puts it back into the normal functioning mode.

- **P17:** When the MPSP is docked and in the rightmost position, `undock` button sends a message to the scanner to disconnect.

- **P18:** When the MPSP is docked and not at the rightmost position, the `down` button moves the bed to the left (towards the out of the scanner).

- **P19:** When the MPSP is docked and calibrated, the `up` button moves the bed into the scanner.

- **P20:** When the MPSP is docked, uncalibrated and lower than the uppermost position, the `up` button moves the bed upwards.

- **P21:** When the MPSP is docked, uncalibrated and above the lowermost position, the `down` button moves the bed downwards.

2 Interactions

Component	*Name*	*Action*
`Motors and Brakes`	`m1UpA`	Vertical motor M1 is turned on and moves the bed upward
	`m1DownA`	Vertical motor M1 is turned on and moves the bed downwards
	`m1OffA`	Vertical motor M1 is turned off
	`m2LeftA`	Horizontal motor M2 is turned on and moves the bed into the scanner
	`m2RightA`	Horizontal motor M2 is turned on and moves the bed out of the scanner
	`m2OffA`	Horizontal motor M2 is turned off
	`vBrakeApply`	Vertical brake B1 is applied
	`vBrakeRelease`	Vertical brake B1 is released
	`hBrakeApply`	Horizontal brake B2 is applied
	`hBrakeRelease`	Horizontal brake B2 is released
`Sensor`	`isAtLeftMostPos`	Checks if the bed is fully inside the scanner
	`isAtRightMostPos`	Checks if the bed is fully outside the scanner
	`isAtStandardHeight`	Checks if the bed is at standard height
	`isAtMaxHeight`	Checks if the bed is at the maximum height
	`isAtMinHeight`	Checks if the bed is at the minimum height
	`dock`	Checks if the bed is docked
`Button actions`	`pressUP`	The UP button pressed
	`releaseUP`	The UP button released
	`pressDOWN`	The DOWN button pressed
	`releaseDOWN`	The DOWN button released
	`clickUNDOCK`	The UNDOCK button is clicked
	`clickRESET`	The RESET button is clicked
	`clickRESUME`	The RESUME button is clicked
	`clickSTOP`	The STOP button is clicked
`Mode`	`Normal Mode`	Checks if the MPSP is in normal mode.
	`Emergency Mode`	Checks if the MPSP is in emergency mode

3 Interactions of Control Systems

The requirements are set up into the model specifications using mu-calculus properties and are checked for validity. These requirement properties form an integral interaction systems behaviour of the MPSP model. The interactions are mentioned here:

- **P01** If a motor is on, the corresponding brake must not be applied.
  If m1OffA or m2OffA is false, then correspondingly vBrakeApply or hBrakeApply is set to true.
- **P02:** The horizontal motor should be on only if the MPSP is docked.
  If dock is true then m1UpA and m1DownA can activate.
- **P03:** The horizontal brake should not be applied if the MPSP is docked.
  if dock is true then hBrakeRelease should be set to true and hBrakeApply to false.
- **P04:** The bed must be in the rightmost position if the MPSP is undocked.
  if dock is false then isAtLeftMostPos is true.
- **P05:** The horizontal brake must be applied if the MPSP is undocked.
  if dock is false then hBrakeRelease should be set to false and hBrakeApply to true.
- **P06:** The vertical brake must always be applied if the vertical motor is off.
  if m1OffA is set then vBrakeApply is carried out.
- **P07:** The vertical brake must always be applied if the MPSP is docked.

- **P08:** In emergency mode, the horizontal brake must be released.

- **P09:** If the MPSP is undocked and uncalibrated, the bed should only be moved up and down.

- **P10:** The MPSP should not move above the uppermost position, below the lowermost position.

- **P11:** When the MPSP is docked and at the rightmost position, the `reset` button will set the standard height.

- **P12:** When the MPSP is undocked and the `reset` button is pressed, the standard height is forgotten.

- **P13:** If the MPSP is docked and calibrated, the vertical brakes are applied and the vertical motor turned off.

- **P14:** If the `stop` button is pressed, the MPSP goes into emergency mode.

- **P15:** When the MPSP goes into emergency mode, the horizontal brake is released so that the patient can be taken out of the scanner.

- **P16:** When the MPSP is in emergency mode, the `resume` button puts it back into the normal functioning mode.

- **P17:** When the MPSP is docked and in the rightmost position, `undock` button sends a message to the scanner to disconnect.

- **P18:** When the MPSP is docked and not at the rightmost position, the `down` button moves the bed to the left (towards the out of the scanner).

- **P19:** When the MPSP is docked and calibrated, the `up` button moves the bed into the scanner.

- **P20:** When the MPSP is docked, uncalibrated and lower than the uppermost position, the `up` button moves the bed upwards.

- **P21:** When the MPSP is docked, uncalibrated and above the lowermost position, the `down` button moves the bed downwards.

4 Architecture

The architecture for the control system is designed with the set of controllers mentioned in the problem statement. The control systems are segregated into input controller, sensory controller and output controller.

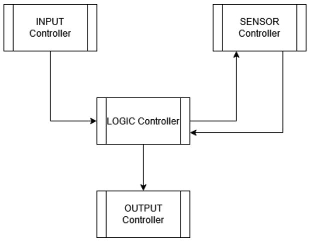

Fig. 1. Architecture Overview

4.1 Console Input Controller

Input controller consists of the console inputs to MPSP which are Up, Down, Stop, Resume, Undock and Reset. Since the action requirements can be set according to preference, we are assuming a simplistic approach of button press and release. The button can be continuously pressed for a respective continuous action and then released.

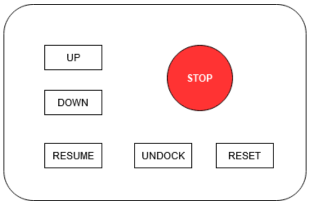

Fig. 2. Input Controller

4.2 Sensor Input Controller

Sensor controller consists of sensors to identify the location of the bed and docking activity of the MPSP unit. This controller senses the horizontal location, vertical location and docking status (docked or undocked). Furthermore, it also senses the calibrated standard height, uncalibrated maximum height, uncalibrated minimum height, horizontal leftmost position, horizontal rightmost position. To further simplify the model, it is assumed that this controller senses the manual motion in the case of an emergency situation as well.

4.3 Output Controller

The third controller is the output controller, which activates the responses for the input controller based on certain requirement logics. This controller hence controls the motors and brakes including the docking and undocking mechanism. The overall systems included in this are horizontal motor, horizontal brake, vertical motor, vertical brake and docking spring.

4.4 Logic Controller

To simplify and add the requirement logics to the input-output actions, a further logic controller is added which receives live inputs from the input controller and sensory controller and based on the input requirements the corresponding action is sent to the output controller. It also helps in keeping track of the previous state and current state. This helps in case of recovering from an emergency stop action if the current state is invalid to bring it back to a correct state.

So whenever an input action is initiated, the logic controller gets input from the input controller as well as gets live sensory info from the sensory controller and based on the requirements logic provides the suitable action that is sent to the output controller. All the channels are assumed to have zero latency. The channel is unidirectional from input to logic, bidirectional from logic to sensory and unidirectional from logic to output controllers.

5 Reflections

This course, Protocol Validation, gave us an insight on the modelling of distributed systems, how to write specifications and validate protocols. We learnt how to model in mCRL2 and use the toolset available. This course was given in a self study manner this academic year (2021-2022), due to which we struggle a bit. It was advised to take the course next year but due to some constraints we continued with the course and tried to finish it by giving our best shot. With many ups and downs in doing this assignment, we learnt quite a lot and for that we would like to thank course coordinator, Dr. WJ Fokkink, and teaching assistant Wolf bij 't Vuur. We tried our best to finish the mCRL specification but could only model the Console input controller, the sensor input controller and the output controller. We couldn't finish the logic controller as it showed too many errors. We have tried till the last minute to finish it and make it work. We are hence submitting till what we have done. We respect the firm deadline and submitting the report. We will continue working on the assignment and try to do it perfectly. If re-submission is allowed and the assignment is finished before the re-submission deadline, we will submit the more perfect model. Even otherwise, we would work till we have learnt to model this assignment problem.