



SOFTWARE VALIDATION

Painkiller System

Group 3

Author(V) : XXXX

Table of Contents

T1: Unit Test

T1.1: Controller Unit Test

T1.1.1: Test auto_inject()

T1.1.1.1: Test auto_inject() for different states

T1.1.1.2: Test auto_inject() for different Left_amounts

T1.1.1.3: Test auto_inject() for extreme cases(exceed limits)

T1.1.2: Test limit_control()

T1.1.2.1: Test limit_control() 's Initialization

T1.1.2.2: Test limit_control() for hour_limit exceeding

T1.1.2.3: Test limit_control() for day_limit exceeding

T1.1.2.4: Test limit_control() for auto_inject_on/off

T1.1.2.5: Test limit_control() for hour_injected calculation

T1.1.2.6: Test limit_control() for day_injected calculation

T1.2: UI Unit Test (test single button)

T1.2.1: PatientUI Button Test

T1.2.1.1: Test CallPhysicianButtonPushed()

T1.2.1.2: Test InjectBolusButtonPushed()

T1.2.2: PhysicianUI Button Test

T1.2.2.1: Test SwitchValueChanged()

T1.2.2.2: Test ResolvePatientCallButtonPushed()

T1.2.2.3: Test FillInjectorButtonPushed()

T1.2.2.4: Test FuctionOfThisHourButtonPushed()

T1.2.2.5: Test FuctionOfTodayButtonPushed()

T2: Functional Test

T2.1: Use Case "(Patient's) Basic operation"

T2.1.1: Test 'Call physician'

T2.2: Use Case "(Physician's) Basic operations"

T2.2.1: Test 'Resolve patient's call'

T2.2.2: Test 'Fill injector'

T2.2.3: Test 'Get information by statistics'

T2.2.4: Test 'Get hour_injected information by image'

T2.2.5: Test 'Get day_injected information by image'

T2.3: Use Case “(Patient) Inject bolus”

T2.3.1: Test ‘InjectBolusValid’

T2.3.2: Test ‘InjectBolusInvalid’

T2.4: Use Case “(Physician) Set parameters”

T2.4.1: Test ‘SetBolusValid()’

T2.4.2: Test ‘SetBaselineValid()’

T2.4.3: Test ‘SetBolusInvalid()’

T2.4.4: Test ‘SetBaselineInvalid()’

T3: Acceptance Test (Controller+PatientUI+PhysicianUI & complex cases)

T3.1: Test normal cases

T3.2: Test extreme cases

T4: Risk Management

T5: Model Checking

Controller

Physician

Patient

Justify the properties

T1: Unit Test

T1.1: Controller Unit Test

T1.1.1: Test auto_injector()

T1.1.1.1: Test auto_inject() for different states

Coverage Criteria: Branch coverage

Test case

| | |
|-----------------|---|
| | Test Case T1.1.1.1.1 |
| Coverage Item | Tcover1.1.1.1.1 |
| Input | auto_inject() |
| State | Injector system is off. |
| Expected Output | There's no auto injection. All the properties of the system have their initialized value. |

| | |
|-----------------|--|
| | Test Case T1.1.1.1.2 |
| Coverage Item | Tcover1.1.1.1.2 |
| Input | auto_inject() |
| State | Injector system is on. Left_amount is 10. There's no limit-exceeding. All the other properties of the system have their initialized value. |
| Expected Output | Injector system begins auto injection. Left_amount changes from 10 to 9.999. |

Test coverage: $2/2=100\%$

Test result: all passed

T1.1.1.2: Test auto_inject() for different Left_amounts

Coverage Criteria: Branch coverage

Test case

| | |
|-----------------|---|
| | Test Case T1.1.1.2.1 |
| Coverage Item | Tcover1.1.1.2.1 |
| Input | auto_inject() |
| State | Injector system is on. Left_amount is -5(invalid). All the other properties of the system have their initialized value. |
| Expected Output | There's no auto injection. Left_amount becomes zero, other properties of the system have their initialized value. |

| | |
|---------------|----------------------|
| | Test Case T1.1.1.2.2 |
| Coverage Item | Tcover1.1.1.2.2 |
| Input | auto_inject() |

| | |
|-----------------|---|
| State | Injector system is on. Left_amount is 5. There's no limit-exceeding. All the other properties of the system have their initialized value. |
| Expected Output | Injector system begins auto injection. Left_amount changes from 5 to 4.999. |

Test coverage: 2/2=100%

Test result: all passed

T1.1.1.3: Test auto_inject() for extreme cases(exceed limits)

Coverage Criteria: Branch coverage

Test case

| | |
|-----------------|---|
| | Test Case T1.1.1.3.1 |
| Coverage Item | Tcover1.1.1.3.1 |
| Input | auto_inject() |
| State | Injector system is on. The volume of injection exceeds day_limit. Left_amount is 10. All the other properties of the system have their initialized value. |
| Expected Output | There's no auto injection. |

| | |
|-----------------|---|
| | Test Case T1.1.1.3.2 |
| Coverage Item | Tcover1.1.1.3.2 |
| Input | auto_inject() |
| State | Injector system is on. The volume of injection exceeds day_limit. The volume of injection exceeds hour_limit. Left_amount is 10. All the other properties of the system have their initialized value. |
| Expected Output | There's no auto injection. |

| | |
|-----------------|--|
| | Test Case T1.1.1.3.3 |
| Coverage Item | Tcover1.1.1.3.3 |
| Input | auto_inject() |
| State | Injector system is on. There's no limit-exceeding. Left_amount is 10. All the other properties of the system have their initialized value. |
| Expected Output | Injector system begins auto injection. Left_amount changes from 10 to 9.999. |

Test coverage: 3/3=100%

Test result: all passed

T1.1.2: Test limit_control()

T1.1.2.1: Test limit_control() 's Initialization

Coverage Criteria: Statement coverage

Test case

| | |
|-----------------|--|
| | Test Case T1.1.2. 1 |
| Coverage Item | Tcover1.1.2.1 |
| Input | limit_control() |
| State | Injector system is on. The volume of injection exceeds day_limit. The volume of injection exceeds hour_limit. The auto_inject is off. All the other properties of the system have their initialized value. |
| Expected Output | controller.exceed_limit_hour == 0; controller.exceed_limit_day == 0; controller.auto_inject_on == 1; |

Test coverage: 1/1=100%

Test result: 1 passed

T1.1.2.2: Test limit_control() for hour_limit exceeding

Coverage Criteria: Branch coverage

Test case

| | |
|-----------------|---|
| | Test Case T1.1.2.2 |
| Coverage Item | Tcover1.1.2.2 |
| Input | limit_control() |
| State | Injector system is on. There's no limit-exceeding. The hour_injected (the volume injected this hour) is 1. All the other properties of the system have their initialized value. |
| Expected Output | controller.exceed_limit_hour == 1; |

Test coverage: 1/1=100%

Test result: 1 passed

T1.1.2.3: Test limit_control() for day_limit exceeding

Coverage Criteria: Branch coverage

Test case

| | |
|---------------|--|
| | Test Case T1.1.2.3 |
| Coverage Item | Tcover1.1.2.3 |
| Input | limit_control() |
| State | Injector system is on. There's no limit-exceeding. The day_injected (the volume injected this day) is 3. |

| | |
|-----------------|--|
| | All the other properties of the system have their initialized value. |
| Expected Output | controller.exceed_limit_day = 1; |

Test coverage: 1/1=100%

Test result: 1 passed

T1.1.2.4: Test limit_control() for auto_inject_on/off

Coverage Criteria: Branch coverage

Test case

| | |
|-----------------|--|
| | Test Case T1.1.2.4.1 |
| Coverage Item | Tcover1.1.2.4.1 |
| Input | limit_control() |
| State | Injector system is on. There's no limit-exceeding. All the other properties of the system have their initialized value. |
| Expected Output | controller.auto_inject_on = 1; |

| | |
|-----------------|--|
| | Test Case T1.1.2.4.2 |
| Coverage Item | Tcover1.1.2.4.2 |
| Input | limit_control() |
| State | Injector system is on. The day_injected (the volume injected this day) is 3. All the other properties of the system have their initialized value. |
| Expected Output | controller.auto_inject_on = 0; |

| | |
|-----------------|---|
| | Test Case T1.1.2.4.3 |
| Coverage Item | Tcover1.1.2.4.3 |
| Input | limit_control() |
| State | Injector system is on. The day_injected (the volume injected this day) is 3. The hour_injected (the volume injected this hour) is 1. All the other properties of the system have their initialized value. |
| Expected Output | controller.auto_inject_on = 0; |

| | |
|-----------------|--------------------------------|
| | Test Case T1.1.2.4.4 |
| Coverage Item | Tcover1.1.2.4.4 |
| Input | limit_control() |
| State | Injector system is off. |
| Expected Output | controller.auto_inject_on = 0; |

| | |
|-----------------|---|
| | Test Case T1.1.2.4.5 |
| Coverage Item | Tcover1.1.2.4.5 |
| Input | limit_control() |
| State | Injector system is on. The Left_amount is 0. All the other properties of the system have their initialized value. |
| Expected Output | controller.auto_inject_on = =0; |

Test coverage: 5/5=100%

Test result: all passed

T1.1.2.5: Test limit_control() for hour_injected calculation

Coverage Criteria: Branch coverage

Test case

| | |
|-----------------|---|
| | Test Case T1.1.2.5.1 |
| Coverage Item | Tcover1.1.2.5.1 |
| Input | Call limit_control() 1000 times |
| State | Injector system is on. All the other properties of the system have their initialized value. |
| Expected Output | controller.hour_injected = =0.60; |

| | |
|-----------------|---|
| | Test Case T1.1.2.5.2 |
| Coverage Item | Tcover1.1.2.5.2 |
| Input | Call limit_control() 600 times |
| State | Last output state + set controller.baseline = 0.02. |
| Expected Output | controller.hour_injected = =1.00; |

| | |
|-----------------|---------------------------------------|
| | Test Case T1.1.2.5.3 |
| Coverage Item | Tcover1.1.2.5.3 |
| Input | Call limit_control() 100 times |
| State | Last output state + the system is off |
| Expected Output | controller.hour_injected = =0.80; |

| | |
|-----------------|-----------------------------------|
| | Test Case T1.1.2.5.4 |
| Coverage Item | Tcover1.1.2.5.4 |
| Input | Call limit_control() 490 times |
| State | Last output state |
| Expected Output | controller.hour_injected = =0.01; |

Test coverage: 4/4=100%

Test result: all passed

T1.1.2.5: Test limit_control() for day_injected calculation

Coverage Criteria: Branch coverage

Test case

| | |
|-----------------|--|
| | Test Case T1.1.2.6.1 |
| Coverage Item | Tcover1.1.2.6.1 |
| Input | Call limit_control() 600 times |
| State | Injector system is on. All the other properties of the system have their initialized value. |
| Expected Output | controller.day_injected = =0.60; |

| | |
|-----------------|----------------------------------|
| | Test Case T1.1.2.6.2 |
| Coverage Item | Tcover1.1.2.6.2 |
| Input | Call limit_control() 600 times |
| State | Last output state |
| Expected Output | controller.day_injected = =1.20; |

| | |
|-----------------|---------------------------------------|
| | Test Case T1.1.2.6.3 |
| Coverage Item | Tcover1.1.2.6.3 |
| Input | Call limit_control() 100 times |
| State | Last output state + the system is off |
| Expected Output | Controller.day_injected = =1.20; |

| | |
|-----------------|--------------------------------------|
| | Test Case T1.1.2.6.4 |
| Coverage Item | Tcover1.1.2.6.4 |
| Input | Call limit_control() 600 times |
| State | Last output state + the system is on |
| Expected Output | controller.day_injected = =1.80; |

| | |
|-----------------|--|
| | Test Case T1.1.2.6.5 |
| Coverage Item | Tcover1.1.2.6.5 |
| Input | Call limit_control() 600 times |
| State | Last output state + set controller.baseline = 0.02 |
| Expected Output | controller.day_injected = =2.80; |

| | |
|-----------------|----------------------------------|
| | Test Case T1.1.2.6.6 |
| Coverage Item | Tcover1.1.2.6.6 |
| Input | Call limit_control() 200 times |
| State | Last output state |
| Expected Output | controller.day_injected = =3.00; |

Test coverage: 6/6=100%

Test result: all passed

T1.2: UI Unit Test (test single button)

T1.2.1: PatientUI Button Test

T1.2.1.1: Test CallPhysicianButtonPushed()

Coverage Criteria: Statement coverage

Test case

| | |
|-----------------|---|
| | Test Case T1.2.1.1 |
| Coverage Item | Tcover1.2.1.1 |
| Input | Press CallPhysicianButton on PatientUI. |
| State | PatientUI's initial state + PhysicianUI's initial state |
| Expected Output | The PatientCallingLamp on PhysicianUI becomes red. |

Test coverage: 1/1=100%

Test result: 1 passed

T1.2.1.2: Test InjectBolusButtonPushed()

Coverage Criteria: Statement coverage

Test case

| | |
|-----------------|--|
| | Test Case T1.2.1.2 |
| Coverage Item | Tcover1.2.1.2 |
| Input | Press switch on PhysicianUI. Press InjectBolusButton on PatientUI. |
| State | PatientUI's initial state + PhysicianUI's initial state |
| Expected Output | Both hour_inject and day_injected increase 0.2. |

Test coverage: 1/1=100%

Test result: 1 passed

T1.2.2: PhysicianUI Button Test

T1.2.2.1: Test SwitchValueChanged()

Coverage Criteria: Statement coverage

Test case

| | |
|---------------|------------------------------|
| | Test Case T1.2.2.1.1 |
| Coverage Item | Tcover1.2.2.1.1 |
| Input | Press switch on PhysicianUI. |

| | |
|-----------------|---|
| State | PatientUI's initial state + PhysicianUI's initial state |
| Expected Output | The PowerLamp on PhysicianUI becomes green. |

| | |
|-----------------|--|
| | Test Case T1.2.2.1.2 |
| Coverage Item | Tcover1.2.2.1.2 |
| Input | Press switch on PhysicianUI. |
| State | Last state |
| Expected Output | The PowerLamp on PhysicianUI becomes grey. |

Test coverage: 2/2=100%

Test result: 1 passed

T1.2.2.2: Test ResolvePatientCallButtonPushed()

Coverage Criteria: Statement coverage

Test case

| | |
|-----------------|--|
| | Test Case T1.2.2.2 |
| Coverage Item | Tcover1.2.2.2 |
| Input | Press ResolvePatientCallButton on PatientUI. |
| State | PatientUI's PatientCallingLamp is red. |
| Expected Output | PatientUI's PatientCallingLamp becomes grey. |

Test coverage: 1/1=100%

Test result: 1 passed

T1.2.2.3: Test FillInjectorButtonPushed()

Coverage Criteria: Statement coverage

Test case

| | |
|-----------------|---|
| | Test Case T1.2.2.3 |
| Coverage Item | Tcover1.2.2.3 |
| Input | Press FillInjectorButton on PatientUI. |
| State | The injector system has been on for 2 minutes. |
| Expected Output | The system's Left_amount becomes 10. PatientUI's Left_amountText.Value is 10. |

Test coverage: 1/1=100%

Test result: 1 passed

T1.2.2.4: Test FuctionOfThisHourButtonPushed()

Coverage Criteria: Statement coverage

Test case

| | |
|-----------------|---|
| | Test Case T1.2.2.4 |
| Coverage Item | Tcover1.2.2.4 |
| Input | Press FuctionOfThisHourButton on PatientUI. |
| State | The injector system has been on for 5 minutes. |
| Expected Output | Physician can get the graph of the volume in recent 1 hour. |

Test coverage: 1/1=100%

Test result: 1 passed

T1.2.2.5: Test FuctionOfTodayButtonPushed()

Coverage Criteria: Statement coverage

Test case

| | |
|-----------------|--|
| | Test Case T1.2.2.5 |
| Coverage Item | Tcover1.2.2.5 |
| Input | Press FuctionOfTodayButton on PatientUI. |
| State | The injector system has been on for 5 minutes. |
| Expected Output | Physician can get the graph of the volume in this day. |

Test coverage: 1/1=100%

Test result: 1 passed

T2: Functional Test

The implementation only contains 2 UIs and one “Controller”, it is simple to do the Integration Test. Therefore, we make Integration Test part of Functional Test. When we do Functional Test, we can cover all the Integration Tests.

T2.1: Use Case “(Patient’s) Basic operation”

T2.1.1: Test ‘Call physician ’

It is a simple case. Please refer to T1.2.1.1 .

T2.2: Use Case “(Physician’s) Basic operations”

T2.2.1: Test ‘Resolve patient’s call’

It is a simple case. Please refer to T1.2.2.2.

T2.2.2: Test ‘Fill injector ’

It is a simple case. Please refer to T1.2.2.3 .

T2.2.3: Test ‘Get information by statistics ’

Test case

| | |
|-------------------|---|
| | Test Case T2.2.3 |
| Coverage Item | Tcover2.2.3, 1.1.1.1, 1.1.1.2, 1.1.2.1, 1.1.2.4, 1.1.2.5, 1.1.2.6, 1.2.1.2, 1.2.2.1 |
| State | Initial state |
| Operation | Switch on. Pause some seconds. Check hour_injected, day_injected, baseline, left_amount. Set baseline = 0.02. Pause some seconds. Check hour_injected, day_injected, baseline, left_amount. Set baseline = 0.01. Check hour_injected, day_injected, baseline, left_amount. Press InjectBolusButton. Check hour_injected, day_injected, baseline, left_amount. |
| Expected Behavior | All the statistics shown on the physicianUI is correct. Physician can get patient’s basic injection information by these statistics on the UI. |

Test coverage: 9/9=100%

Test result: all passed

T2.2.4: Test ‘Get hour_injected information by image ’

Test case

| | |
|---------------|---|
| | Test Case T2.2.4 |
| Coverage Item | Tcover2.2.4, 1.1.1.1, 1.1.1.2, 1.1.2.1, 1.1.2.4, 1.1.2.5, 1.2.1.2, 1.2.2.1, 1.2.2.4 |
| State | Initial state |
| Operation | Switch on. Pause some seconds. Press |

| | |
|-------------------|--|
| | FunctionofthishourButton. Set baseline = 0.03. Pause some seconds. Press FunctionofthishourButton. Set baseline = 0.02. Press InjectBolusButton. Press FunctionofthishourButton. |
| Expected Behavior | After pressing FunctionofthishourButton, we can get the graph of the volume injected in recent 1 hour. After we set the baseline to 0.03, we can see the change of the slope on the graph. After we set the baseline to 0.02, we can also see the change of the slope on the graph. After we press InjectBolusButton, we can see a sharp increase of 0.2 on the graph. |

Test coverage: 9/9=100%

Test result: all passed

T2.2.5: Test 'Get day_injected information by image '

Test case

| | |
|-------------------|--|
| | Test Case T2.2.5 |
| Coverage Item | Tcover2.2.5, 1.1.1.1, 1.1.1.2, 1.1.2.1, 1.1.2.4, 1.1.2.6, 1.2.1.2, 1.2.2.1,1.2.2.5 |
| State | Initial state |
| Operation | Switch on. Pause some seconds. Press FunctionofTodayButton. Set baseline = 0.03. Pause some seconds. Press FunctionofTodayButton. Set baseline = 0.02. Press InjectBolusButton. Press FunctionofTodayButton. |
| Expected Behavior | After pressing FunctionofTodayButton, we can get the graph of the volume injected in this day. After we set the baseline to 0.03, we can see the change of the slope on the graph. After we set the baseline to 0.02, we can also see the change of the slope on the graph. After we press InjectBolusButton, we can see a sharp increase of 0.2 on the graph. |

Test coverage: 9/9=100%

Test result: all passed

T2.3: Use Case "(Patient) Inject bolus"

T2.3.1: Test 'InjectBolusValid'

Test case

| | |
|-------------------|--|
| | Test Case T2.3.1 |
| Coverage Item | Tcover2.3.1, 1.1.1.1, 1.1.1.2, 1.1.2.1, 1.1.2.4, 1.1.2.5, 1.1.2.6, 1.2.1.2, 1.2.2.1, 1.2.2.4, 2.2.3, 2.2.4 |
| State | Initial state |
| Operation | Switch on. Pause some seconds. Press InjectBolusButton. Check hour_injected, day_injected. Press FunctionofthishourButton. Press InjectBolusButton. Check hour_injected, day_injected. Press FunctionofthishourButton. |
| Expected Behavior | After pressing InjectBolusButton, we can find the statistics on the pyhsicianUI have a sharp increase. After we press FunctionofthishourButton, we can see a sharp increase of 0.2 on the graph. After pressing InjectBolusButton, we can find the statistics on the pyhsicianUI have a sharp increase again. After we press FunctionofthishourButton, we can see a sharp increase of 0.2 on the graph again. |

Test coverage: 12/12=100%

Test result: all passed

T2.3.2: Test 'InjectBolusInvalid'

Test case

| | |
|-------------------|--|
| | Test Case T2.3.2 |
| Coverage Item | Tcover2.3.2, 1.1.1.1, 1.1.1.2, 1.1.1.3, 1.1.2.1, 1.1.2.2, 1.1.2.4, 1.1.2.5, 1.1.2.6, 1.2.1.2, 1.2.2.1, 1.2.2.4, 2.2.3, 2.2.4, 2.3.1 |
| State | Initial state |
| Operation | Switch on. Pause some seconds. Press InjectBolusButton. Press InjectBolusButton. Press InjectBolusButton. Press InjectBolusButton. Press FunctionofthishourButton. Check hour_injected, day_injected. Press InjectBolusButton. Press FunctionofthishourButton. |
| Expected Behavior | After pressing InjectBolusButton 4 times, we can find the statistics on the pyhsicianUI have 4 sharp increases. |

| | |
|--|---|
| | <p>After we press FunctionofthishourButton, we can see 4 sharp increases of 0.2 on the graph.</p> <p>After pressing InjectBolusButton again, we can get error 4 msgbox which tells us the volume injected this hour will exceed . It means this bolus injection is invalid.</p> <p>After we press FunctionofthishourButton, we cannot see a sharp increase of 0.2 on the graph again.</p> |
|--|---|

Test coverage: 15/15=100%

Test result: all passed

T2.4: Use Case “(Physician) Set parameters”

T2.4.1: Test ‘SetBolusValid()’

Test case

| | |
|-------------------|--|
| | Test Case T2.4.1 |
| Coverage Item | Tcover2.4.1, 1.1.1.1, 1.1.1.2, 1.1.2.1, 1.1.2.4, 1.1.2.5, 1.1.2.6, 1.2.1.2, 1.2.2.1,1.2.2.4, 2.2.3, 2.2.4,2.3.1 |
| State | Initial state |
| Operation | Switch on. Press InjectBolusButton. Set bolus = 0.4. Press InjectBolusButton. Check hour_injected, day_injected. Press FunctionofthishourButton. |
| Expected Behavior | <p>After pressing InjectBolusButton, we can find the statistics on the pyhsicianUI have a sharp increase of 0.2.</p> <p>After setting bolus = 0.4 and pressing InjectBolusButton, we can find the statistics on the pyhsicianUI have a sharp increase of 0.4.</p> <p>After pressing FunctionofthishourButton, we can see a sharp increase of 0.2 and a sharp increase of 0.4 on the graph.</p> |

Test coverage: 13/13=100%

Test result: all passed

T2.4.2: Test ‘SetBaselineValid()’

Test case

| | |
|--|------------------|
| | Test Case T2.4.2 |
|--|------------------|

| | |
|-------------------|--|
| Coverage Item | Tcover2.4.2, 1.1.1.1, 1.1.1.2, 1.1.2.1, 1.1.2.4, 1.1.2.5, 1.1.2.6, 1.2.2.1,1.2.2.4, 2.2.3, 2.2.4 |
| State | Initial state |
| Operation | Switch on. Set baseline = 0.05. Check hour_injected, day_injected. Press FunctionofthishourButton. |
| Expected Behavior | After setting baseline = 0.05, we can find the increasing speed of statistics on the pyhsicianUI becomes much faster. After pressing FunctionofthishourButton, we can see the change of slope on the graph. |

Test coverage: 11/11=100%

Test result: all passed

T2.4.3: Test 'SetBolusInvalid()'

Test case

| | |
|-------------------|--|
| | Test Case T2.4.3 |
| Coverage Item | Tcover2.4.3, 1.1.1.1, 1.1.1.2, 1.1.1.3, 1.1.2.1, 1.1.2.4, 1.1.2.5, 1.1.2.6, 1.2.1.2, 1.2.2.1,1.2.2.4, 2.2.3, 2.2.4,2.3.1 |
| State | Initial state |
| Operation | Switch on. Press InjectBolusButton. Set bolus = 0.6. Press InjectBolusButton. Check hour_injected, day_injected. Press FunctionofthishourButton. |
| Expected Behavior | After pressing InjectBolusButton, we can find the statistics on the pyhsicianUI have a sharp increase of 0.2. After we set bolus = 0.6 and press InjectBolusButton, we can get error 1 msgbox which tells us the volume setted is out of range . It means our operation is invalid. After we press FunctionofthishourButton, we cannot see a sharp increase of 0.6 on the graph but see a sharp increase of 0.2. |

Test coverage: 14/14=100%

Test result: all passed

T2.4.4: Test 'SetBaselineInvalid()'

Test case

| | |
|-------------------|--|
| | Test Case T2.4.4 |
| Coverage Item | Tcover2.4.4, 1.1.1.1, 1.1.1.2, 1.1.2.1, 1.1.2.4, 1.1.2.5, 1.1.2.6, 1.2.2.1,1.2.2.4, 2.2.3, 2.2.4, 2.4.2 |
| State | Initial state |
| Operation | Switch on. Set baseline = 0.05. Set baseline = 0.15. Check hour_injected, day_injected. Press FunctionofthishourButton. |
| Expected Behavior | After setting baseline = 0.05, we can find the increasing speed of statistics on the pyhsicianUI becomes much faster. After setting baseline = 0.15, we can get error 2 msgbox which tells us the volume setted is out of range . It means our operation is invalid. We cannot find the increasing speed of statistics on the pyhsicianUI becomes much faster. After pressing FunctionofthishourButton, we cannot see the change of slope on the graph. |

Test coverage: 12/12=100%

Test result: all passed

T3: Acceptance Test (Controller+PatientUI+PhysicianUI & complex cases)

Acceptance Test is shown to the customers. We will show our products' functionality and robustness by Acceptance Test.

T3.1: Test normal cases

Test case

| | |
|-----------|---|
| | Test Case T3.1 |
| State | Initial state |
| Operation | Switch on. Check Left_amount, baseline, hour_injected, day_injected. Press CallPhysicianButton. Press ResolvePatientCallButton. Press FillInjectorButton. Set Baseline = 0.04. Check Left_amount, baseline, hour_injected, day_injected. Press FunctionofthishourButton. Press FunctionofTodayButton. Set Baseline = 0.01. Press InjectBolusButton. |

| | |
|-------------------|---|
| | <p>Check Left_amount, baseline, hour_injected, day_injected. Press FunctionofthishourButton. Press FunctionofTodayButton. Press InjectBolusButton. Check Left_amount, baseline, hour_injected, day_injected. Press FunctionofthishourButton. Pause 50 seconds. Set bolus = 0.3. Press InjectBolusButton. Check Left_amount, baseline, hour_injected, day_injected. Press FunctionofthishourButton.</p> |
| Expected Behavior | <p>After pressing CallPhysicianButton, the PatientCallingLamp on PhysicianUI becomes red. After pressing ResolvePatientCallButton, the PatientCallingLamp on PhysicianUI becomes grey. After setting baseline = 0.05, we can find the increasing speed of statistics on the pyhsicianUI becomes much faster. After pressing FunctionofthishourButton and FunctionofTodayButton, we can see the change of slope on the graphs. After pressing InjectBolusButton, we can find the statistics on the pyhsicianUI have a sharp increase of 0.2. After pressing FunctionofthishourButton and FunctionofTodayButton, we can see a sharp increase of 0.2 on the graphs. After pressing InjectBolusButton again, we can get error 4 msgbox which tells us the volume injected this hour will exceed . It means this bolus injection is invalid. After pressing FunctionofthishourButton and FunctionofTodayButton, we cannot see a sharp increase of 0.2 on the graphs. Because we have reached the hour_limit, we can find the auto injection stops. 50 seconds pass. After setting bolus = 0.3 and pressing InjectBolusButton, we can find the statistics on the pyhsicianUI have a sharp increase of 0.3. After pressing FunctionofthishourButton, we can see a sharp increase of 0.3 on the graph.</p> |

Test result: passed

T3.2: Test extreme cases

Test case

| | |
|-------------------|--|
| | Test Case T3.2 |
| State | Initial state |
| Operation | <p>Switch on. Press InjectBolusButton. Set bolus = 0.6. Press InjectBolusButton. Press FunctionofthishourButton. Set baseline = 0.05. Check Left_amount, baseline, hour_injected, day_injected. Set baseline = 0.15. Check Left_amount, baseline, hour_injected, day_injected. Press FunctionofthishourButton. Press InjectBolusButton. Check Left_amount, baseline, hour_injected, day_injected. Press FunctionofthishourButton.</p> |
| Expected Behavior | <p>After pressing InjectBolusButton, we can find the statistics on the pyhsicianUI have a sharp increase of 0.2. After we set bolus = 0.6 and press InjectBolusButton, we can get error 1 msgbox which tells us the volume setted is out of range . It means our operation is invalid. After we press FunctionofthishourButton, we cannot see a sharp increase of 0.6 on the graph but see a sharp increase of 0.2. After setting baseline = 0.05, we can find the increasing speed of statistics on the pyhsicianUI becomes much faster. After pressing FunctionofthishourButton and FunctionofTodayButton, we can see the change of slope on the graphs. After setting baseline = 0.15, we can get error 2 msgbox which tells us the volume setted is out of range . It means our operation is invalid. We cannot find the increasing speed of statistics on the pyhsicianUI becomes much faster. After pressing FunctionofthishourButton, we cannot see the change of slope on the graph. After pressing InjectBolusButton, we can get error 4 msgbox which tells us the volume injected this hour will exceed . It means this bolus injection is invalid. After pressing FunctionofthishourButton and FunctionofTodayButton, we cannot see a sharp increase of 0.2 on the graphs.</p> |

Test result: passed

T4: Risk Management

Risk: Patients get too much injection.

Corresponding solution: No more than 1ml/h, No more than 3ml/day.

Corresponding test: T1.1.1.3, T1.1.2.2, T1.1.2.3, T2.3.2, T3.1, T3.2

Risk: Physician sets wrong baseline.

Corresponding solution: Check everything which physician enters.

Corresponding test: T2.4.4, T3.2

Risk: Physician sets wrong bolus.

Corresponding solution: Check everything which physician enters.

Corresponding test: T2.4.3, T3.2

T5: Model Checking

We use UPPAAL to do model checking for the PainKiller system.

Since UPPAAL is weak to check 'double' type, so we do these changes below:

baseline 0.01->1;

bolus 0.2->20;

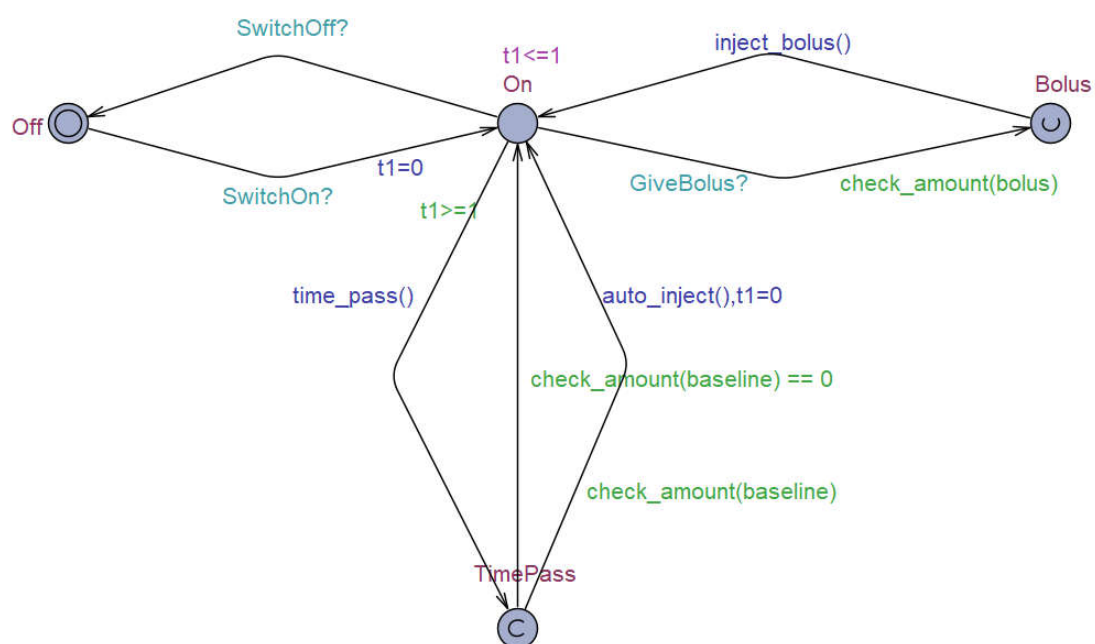
day_limit 3->300;

hour_limit 1->100;

A 24*60 time array is used for recording the injection history of every minute.

24*60 minutes (1 day)

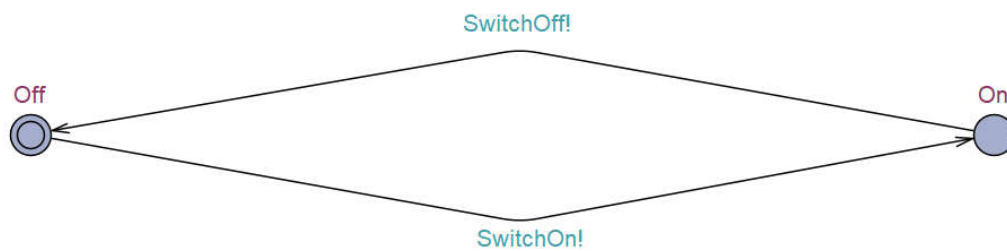
Controller model



The On state is updated every 1 second (which represents 1min in reality), which is implemented in the function 'time_pass()'. The function 'auto_inject()' will do auto injection(add baseline) and update the injection history in the array. And it is only called if the added baseline is valid under current situation.

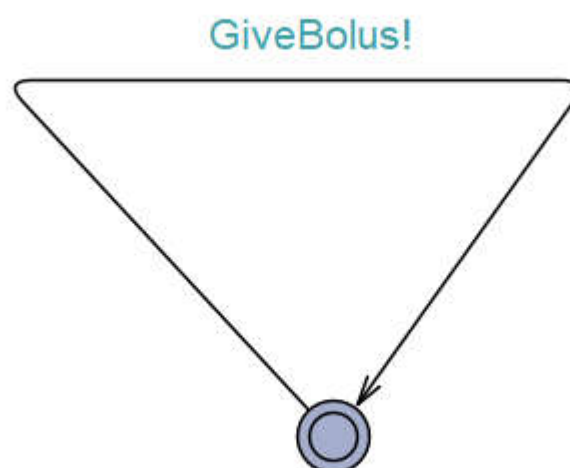
When controller receives 'GetBolus' signal, it will check if this bolus is valid under current situation. If it is valid, bolus injection will be done and the injection history array will be updated at the same time.

Physician model



Physician can switch on/off the system.

Patient model



Patient can choose to inject bolus to reduce pain.

Justify the properties

finalmodel.xml - UPPAAL

文件 编辑 视图 工具 选项 帮助

编辑器 模拟器 模拟器 2 验证器

性质列表

- A◇ baseline>=1 and baseline<=10
- A◇ bolus>=20 and bolus<=50
- A◇ hour_injected<=100
- A◇ day_injected<=300

开始验证

Get Trace

添加

Insert Below

删除

备注

待验证性质

A◇ day_injected<=300

备注

验证进度与结果

验证时间/kernei时间/总时间: 0s / 0s / 0.003s.

常驻内存/虚拟内存的使用峰值: 11,692KB / 50,532KB.

满足该性质.