

Model Report

Version •



Date/Time Generated: 5/23/2022 12:53:07 AM
Author: chioma

EA Repository : Y:\School\iti8520\lab1\TLC_Model.eapx

CREATED WITH  ENTERPRISE ARCHITECT

Traffic Light System Model

Package in package "

Traffic Light System Model
Version 1.0 Phase 1.0 Proposed
chioma created on 5/23/2022. Last modified 5/23/2022

Index

Package in package 'Traffic Light System Model'

Index
Version 1.0 Phase 1.0 Proposed
chioma created on 2/19/2022. Last modified 2/19/2022

Index diagram

Custom Diagram Style diagram in package 'Index'

Index
Version 1.0
chioma created on 2/19/2022. Last modified 5/22/2022



Figure 1: Index

Architecture

Boundary in package 'Index'

Architecture
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022
Extends

Requirements

Boundary in package 'Index'

Requirements
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 4/16/2022
Extends

Controller : Controller

Text in package 'Index'

Controller : Controller

Controller : Controller
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022
Extends

Design : Relationship Map

Text in package 'Index'

Design : Relationship Map

Design : Relationship Map
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022
Extends

External : External

Text in package 'Index'

External : External

External : External
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022
Extends

HLR : Functional Requirement Hierarchy

Text in package 'Index'

HLR : Functional Requirement Hierarchy

HLR : Functional Requirement Hierarchy
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 2/20/2022
Extends

HLR : Non-Functional Requirement Hierarchy

Text in package 'Index'

HLR : Non-Functional Requirement Hierarchy

HLR : Non-Functional Requirement Hierarchy
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 2/20/2022
Extends

Low Level Requirements : Call Button Event Sequence

Text in package 'Index'

Architecture : Call Button Event Sequence Diagram

Low Level Requirements : Call Button Event Sequence
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022
Extends

Low Level Requirements : Low Level Requirements

Text in package 'Index'

LLR : Low Level Requirements

Low Level Requirements : Low Level Requirements
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022
Extends

System Architecture : LightSequence

Text in package 'Index'

Architecture : Controller State Machine

System Architecture : LightSequence
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022
Extends

Design

Package in package 'Traffic Light System Model'

Design
Version 1.0 Phase 1.0 Proposed
chioma created on 5/5/2022. Last modified 5/5/2022

Relationship Map diagram

Custom diagram in package 'Design'

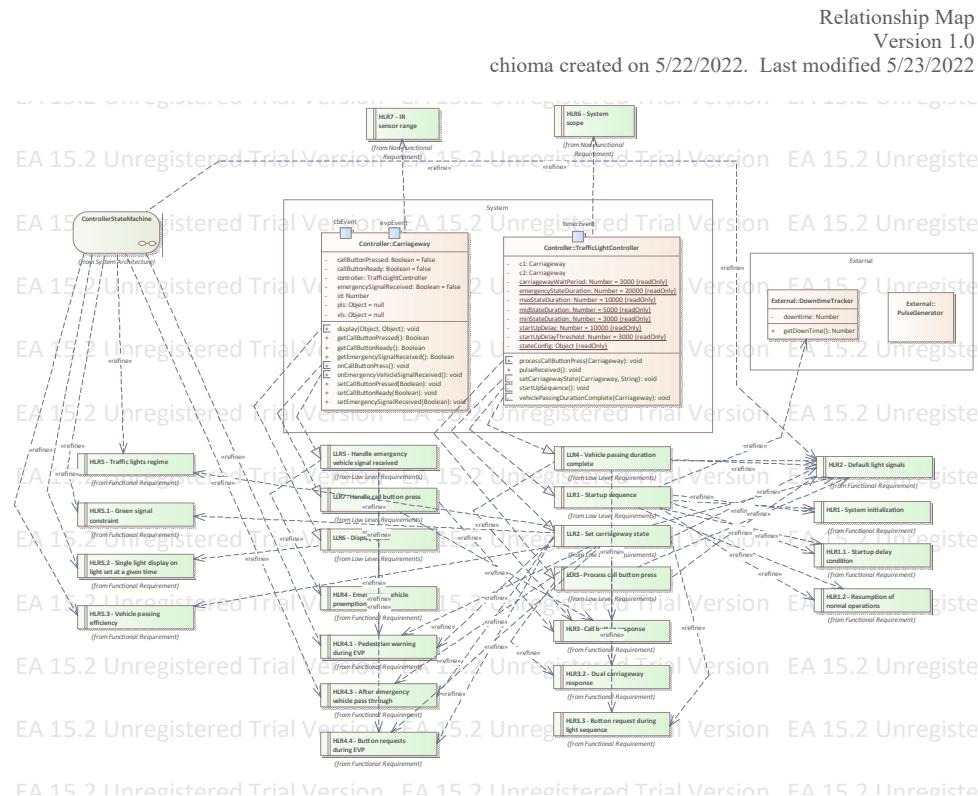


Figure 2: Relationship Map

HLR1 - System initialization

Requirement «Functional» in package 'Functional Requirement'

Upon resumption of power to the controller, no signals SHALL be displayed for a time not less than x seconds nor greater than y seconds before normal operations begin.

HLR1 - System initialization

ELEMENTS OWNED BY HLR1 - System initialization
<p>■ HLR1.1 - Startup delay condition : Requirement «Functional»</p> <p>If the system has already been turned off for more than x seconds, the system shall skip the start up delay. Otherwise, the system shall start operations only after a fixed delay.</p>
<p>■ HLR1.2 - Resumption of normal operations : Requirement «Functional»</p> <p>The controller shall resume operation at "Vehicles Passing Ready" phase. That is, a yellow light signal for the vehicle light set and a red light signal for the pedestrian light set.</p>

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Aggregation from «Functional» HLR1.1 - Startup delay condition to «Functional» HLR1 - System initialization [Direction is 'Source -> Destination'.]</p>
<p>⇒ Aggregation from «Functional» HLR1.2 - Resumption of normal operations to «Functional» HLR1 - System initialization [Direction is 'Source -> Destination'.]</p>

CONNECTORS
<p>↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1 - System initialization : Requirement, Public</p>
<p>↘ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1 - System initialization : Requirement, Public</p>

HLR1.1 - Startup delay condition

Requirement «Functional» owned by 'HLR1 - System initialization', in package 'Functional Requirement'

If the system has already been turned off for more than x seconds, the system shall skip the start up delay.
Otherwise, the system shall start operations only after a fixed delay.

HLR1.1 - Startup delay condition
Version 1.0 Phase 1.0 Proposed
chioma created on 2/19/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from «Functional» HLR1.1 - Startup delay condition to «Functional» HLR1 - System initialization	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.1 - Startup delay condition : Requirement, Public	
↗ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.1 - Startup delay condition : Requirement, Public	

HLR1.2 - Resumption of normal operations

Requirement «Functional» owned by 'HLR1 - System initialization', in package 'Functional Requirement'

The controller shall resume operation at "Vehicles Passing Ready" phase. That is, a yellow light signal for the vehicle light set and a red light signal for the pedestrian light set.

HLR1.2 - Resumption of normal operations
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from «Functional» HLR1.2 - Resumption of normal operations to «Functional» HLR1 - System initialization	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.2 - Resumption of normal operations : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.2 - Resumption of normal operations : Requirement, Public	

HLR2 - Default light signals

Requirement «Functional» in package 'Functional Requirement'

The traffic lights for both carriageways shall remain in one default configuration until a call button on either side of the carriageway is pressed. This configuration is green for the vehicle light set and red for the pedestrian light set.

HLR2 - Default light signals
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

CONNECTORS	
 Refine «refine»	Source -> Destination
From:	LLR3 - Process call button press : Requirement, Public
To:	HLR2 - Default light signals : Requirement, Public
 Refine «refine»	Source -> Destination
From:	LLR4 - Vehicle passing duration complete : Requirement, Public
To:	HLR2 - Default light signals : Requirement, Public
 Abstraction «Refine»	Source -> Destination
From:	LLR4 - Vehicle passing duration complete : Requirement, Public
To:	HLR2 - Default light signals : Requirement, Public
 Abstraction «Refine»	Source -> Destination
From:	LLR3 - Process call button press : Requirement, Public
To:	HLR2 - Default light signals : Requirement, Public
 Abstraction «Refine»	Source -> Destination
From:	ControllerStateMachine : StateMachine, Public
To:	HLR2 - Default light signals : Requirement, Public
 Abstraction «Refine»	Source -> Destination
From:	LLR2 - Set carriageway state : Requirement, Public
To:	HLR2 - Default light signals : Requirement, Public
 Refine «refine»	Source -> Destination
From:	LLR2 - Set carriageway state : Requirement, Public
To:	HLR2 - Default light signals : Requirement, Public

HLR3 - Call button response

Requirement «Functional» in package 'Functional Requirement'

When the call button on either carriageway is pressed, the system shall initiate the sequence for light transition to green for pedestrians and red for motorists.

HLR3 - Call button response
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

ELEMENTS OWNED BY HLR3 - Call button response

ELEMENTS OWNED BY HLR3 - Call button response	
HLR3.1 - Pedestrian crossing wait time : Requirement «Functional»	When the call button on either carriageway is pressed, the system shall signal the pedestrian to cross in no longer than X seconds.
HLR3.2 - Dual carriageway response : Requirement «Functional»	The system shall automatically trigger a button press on the other carriageway after X seconds when the call button on either carriageway is pressed
HLR3.3 - Button request during light sequence : Requirement «Functional»	When the call button on either carriageway is pressed and the light sequence has initiated, the system shall not service any new requests.

INCOMING STRUCTURAL RELATIONSHIPS	
→ Aggregation from «Functional» HLR3.1 - Pedestrian crossing wait time to «Functional» HLR3 - Call button response	[Direction is 'Source -> Destination'.]
→ Aggregation from «Functional» HLR3.2 - Dual carriageway response to «Functional» HLR3 - Call button response	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Refine «refine» Source -> Destination	
From: LLR4 - Vehicle passing duration complete : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	
↗ Refine «refine» Source -> Destination	
From: LLR7 - Handle call button press : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	
↗ Dependency Source -> Destination	
From: HLR3.3 - Button request during light sequence : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination	
From: LLR4 - Vehicle passing duration complete : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination	
From: LLR7 - Handle call button press : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	

HLR3.1 - Pedestrian crossing wait time

Requirement «Functional» owned by 'HLR3 - Call button response', in package 'Functional Requirement'

When the call button on either carriageway is pressed, the system shall signal the pedestrian to cross in no longer than X seconds.

HLR3.1 - Pedestrian crossing wait time
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR3.1 - Pedestrian crossing wait time to «Functional» HLR3 - Call button response
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination
From: LLR7 - Handle call button press : Requirement, Public
To: HLR3.1 - Pedestrian crossing wait time : Requirement, Public

HLR3.2 - Dual carriageway response

Requirement «Functional» owned by 'HLR3 - Call button response', in package 'Functional Requirement'

The system shall automatically trigger a button press on the other carriageway after X seconds when the call button on either carriageway is pressed

HLR3.2 - Dual carriageway response
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR3.2 - Dual carriageway response to «Functional» HLR3 - Call button response
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination
From: LLR3 - Process call button press : Requirement, Public
To: HLR3.2 - Dual carriageway response : Requirement, Public

↗ Abstraction «Refine» Source -> Destination
From: LLR3 - Process call button press : Requirement, Public
To: HLR3.2 - Dual carriageway response : Requirement, Public

HLR3.3 - Button request during light sequence

Requirement «Functional» owned by 'HLR3 - Call button response', in package 'Functional Requirement'

When the call button on either carriageway is pressed and the light sequence has initiated, the system shall not service any new requests.

HLR3.3 - Button request during light sequence
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

CONNECTORS	
 Dependency	Source -> Destination From: HLR3.3 - Button request during light sequence : Requirement, Public To: HLR3 - Call button response : Requirement, Public
 Refine «refine»	Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public
 Abstraction «Refine»	Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public

HLR4 - Emergency vehicle preemption (EVP)

Requirement «Functional» in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, the system shall initiate the preemption process

HLR4 - Emergency vehicle preemption (EVP)
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

ELEMENTS OWNED BY HLR4 - Emergency vehicle preemption (EVP)	
 HLR4.1 - Pedestrian warning during EVP : Requirement «Functional»	When the system receives an approach signal from an emergency vehicle on one carriageway, the pedestrian traffic light shall be a red colour with flashing effects for that carriageway.
 HLR4.2 - EVP state transition : Requirement «Functional»	When the system receives an approach signal from an emergency vehicle on one carriageway, the controller shall transition from its current state at the end of its duration to the emergency vehicle approaching state. The exception to this is when the pedestrian traffic light is green.

ELEMENTS OWNED BY HLR4 - Emergency vehicle preemption (EVP)	
■ HLR4.3 - After emergency vehicle pass through : Requirement «Functional»	After the emergency vehicle has passed through, the traffic light state shall transition from the emergency vehicle approaching state to the vehicle slow down state.
■ HLR4.4 - Button requests during EVP : Requirement «Functional»	When the system receives an approach signal from an emergency vehicle on one carriageway, all button presses on that carriageway are ignored until after the emergency vehicle has passed through.
INCOMING STRUCTURAL RELATIONSHIPS	
→ Aggregation from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]
→ Aggregation from «Functional» HLR4.4 - Button requests during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]
→ Aggregation from «Functional» HLR4.2 - EVP state transition to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]
→ Aggregation from «Functional» HLR4.1 - Pedestrian warning during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Refine «refine» Source -> Destination	
From: LLR5 - Handle emergency vehicle signal received : Requirement, Public	
To: HLR4 - Emergency vehicle preemption (EVP) : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination	
From: LLR5 - Handle emergency vehicle signal received : Requirement, Public	
To: HLR4 - Emergency vehicle preemption (EVP) : Requirement, Public	

HLR4.1 - Pedestrian warning during EVP

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, the pedestrian traffic light shall be a red colour with flashing effects for that carriageway.

HLR4.1 - Pedestrian warning during EVP

OUTGOING STRUCTURAL RELATIONSHIPS
↳ Aggregation from «Functional» HLR4.1 - Pedestrian warning during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP) [Direction is 'Source -> Destination'.]

CONNECTORS
↗ Refine «refine» Source -> Destination From: LLR6 - Display lights : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public
↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public
↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public
↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public

HLR4.2 - EVP state transition

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, the controller shall transition from its current state at the end of its duration to the emergency vehicle approaching state. The exception to this is when the pedestrian traffic light is green.

OUTGOING STRUCTURAL RELATIONSHIPS
↳ Aggregation from «Functional» HLR4.2 - EVP state transition to «Functional» HLR4 - Emergency vehicle preemption (EVP) [Direction is 'Source -> Destination'.]

CONNECTORS

CONNECTORS

 **Refine** «refine» Source -> Destination
 From: LLR5 - Handle emergency vehicle signal received : Requirement, Public
 To: HLR4.2 - EVP state transition : Requirement, Public

 **Refine** «refine» Source -> Destination
 From: LLR2 - Set carriageway state : Requirement, Public
 To: HLR4.2 - EVP state transition : Requirement, Public

HLR4.3 - After emergency vehicle pass through

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

After the emergency vehicle has passed through, the traffic light state shall transition from the emergency vehicle approaching state to the vehicle slow down state.

HLR4.3 - After emergency vehicle pass through
 Version 1.0 Phase 1.0 Proposed
 chioma created on 2/20/2022. Last modified 5/22/2022

EXTERNAL REQUIREMENTS

Requirement. HLR4.4 - Button requests during EVP

When the system receives an approach signal from an emergency vehicle on one carriageway, all button presses on that carriageway are ignored until after the emergency vehicle has passed through.
 [Stereotype is «Functional».]

OUTGOING STRUCTURAL RELATIONSHIPS

 Realization from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4.4 - Button requests during EVP
 [Direction is 'Source -> Destination'.]

 Aggregation from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4 - Emergency vehicle preemption (EVP)
 [Direction is 'Source -> Destination'.]

CONNECTORS

 **Abstraction** «Refine» Source -> Destination
 From: LLR2 - Set carriageway state : Requirement, Public
 To: HLR4.3 - After emergency vehicle pass through : Requirement, Public

 **Abstraction** «Refine» Source -> Destination
 From: ControllerStateMachine : StateMachine, Public
 To: HLR4.3 - After emergency vehicle pass through : Requirement, Public

CONNECTORS

↗ **Refine** «refine» Source -> Destination
 From: LLR2 - Set carriageway state : Requirement, Public
 To: HLR4.3 - After emergency vehicle pass through : Requirement, Public

HLR4.4 - Button requests during EVP

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, all button presses on that carriageway are ignored until after the emergency vehicle has passed through.

HLR4.4 - Button requests during EVP
 Version 1.0 Phase 1.0 Proposed
 chioma created on 2/20/2022. Last modified 5/22/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↖ Aggregation from «Functional» HLR4.4 - Button requests during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP)
 [Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

↗ Realization from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4.4 - Button requests during EVP
 [Direction is 'Source -> Destination'.]

CONNECTORS

↗ **Refine** «refine» Source -> Destination
 From: LLR5 - Handle emergency vehicle signal received : Requirement, Public
 To: HLR4.4 - Button requests during EVP : Requirement, Public

↗ **Refine** «refine» Source -> Destination
 From: LLR2 - Set carriageway state : Requirement, Public
 To: HLR4.4 - Button requests during EVP : Requirement, Public

↗ **Abstraction** «Refine» Source -> Destination
 From: LLR2 - Set carriageway state : Requirement, Public
 To: HLR4.4 - Button requests during EVP : Requirement, Public

HLR5 - Traffic lights regime

Requirement «Functional» in package 'Functional Requirement'

The traffic lights shall be switched based on the standard lights regime: Green - Yellow - Red and vice versa.
The yellow lights must always be between the green and red lights transitions.

HLR5 - Traffic lights regime
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

ELEMENTS OWNED BY HLR5 - Traffic lights regime	
 HLR5.1 - Green signal constraint : Requirement «Functional»	The vehicle and pedestrian light set on one carriageway must never be green at the same time.
 HLR5.2 - Single light display on light set at a given time : Requirement «Functional»	When the light switching occurs, each signal head must display no more than one light signal at a time.
 HLR5.3 - Vehicle passing efficiency : Requirement «Functional»	x seconds after the pedestrian light set has been switched to red, vehicle light set shall be switched to green.

INCOMING STRUCTURAL RELATIONSHIPS	
 Aggregation from «Functional» HLR5.2 - Single light display on light set at a given time to «Functional» HLR5 - Traffic lights regime	[Direction is 'Source -> Destination'.]
 Aggregation from «Functional» HLR5.3 - Vehicle passing efficiency to «Functional» HLR5 - Traffic lights regime	[Direction is 'Source -> Destination'.]
 Aggregation from «Functional» HLR5.1 - Green signal constraint to «Functional» HLR5 - Traffic lights regime	[Direction is 'Source -> Destination'.]

CONNECTORS	
 Refine «refine» Source -> Destination	
From: LLR2 - Set carriageway state : Requirement, Public	
To: HLR5 - Traffic lights regime : Requirement, Public	
 Abstraction «Refine» Source -> Destination	
From: ControllerStateMachine : StateMachine, Public	
To: HLR5 - Traffic lights regime : Requirement, Public	

HLR5.1 - Green signal constraint

Requirement «Functional» owned by 'HLR5 - Traffic lights regime', in package 'Functional Requirement'

The vehicle and pedestrian light set on one carriageway must never be green at the same time.

HLR5.1 - Green signal constraint
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR5.1 - Green signal constraint to «Functional» HLR5 - Traffic lights regime
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Abstraction «Refine» Source -> Destination
From: ControllerStateMachine : StateMachine, Public
To: HLR5.1 - Green signal constraint : Requirement, Public

↗ Refine «refine» Source -> Destination
From: LLR2 - Set carriageway state : Requirement, Public
To: HLR5.1 - Green signal constraint : Requirement, Public

HLR5.2 - Single light display on light set at a given time

Requirement «Functional» owned by 'HLR5 - Traffic lights regime', in package 'Functional Requirement'

When the light switching occurs, each signal head must display no more than one light signal at a time.

HLR5.2 - Single light display on light set at a given time
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR5.2 - Single light display on light set at a given time to «Functional» HLR5 - Traffic lights regime
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination
From: LLR6 - Display lights : Requirement, Public
To: HLR5.2 - Single light display on light set at a given time : Requirement, Public

↗ Abstraction «Refine» Source -> Destination
From: ControllerStateMachine : StateMachine, Public
To: HLR5.2 - Single light display on light set at a given time : Requirement, Public

HLR5.3 - Vehicle passing efficiency

Requirement «Functional» owned by 'HLR5 - Traffic lights regime', in package 'Functional Requirement'

x seconds after the pedestrian light set has been switched to red, vehicle light set shall be switched to green.

HLR5.3 - Vehicle passing efficiency
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR5.3 - Vehicle passing efficiency to «Functional» HLR5 - Traffic lights regime

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Abstraction «Refine» Source -> Destination
From: ControllerStateMachine : StateMachine, Public
To: HLR5.3 - Vehicle passing efficiency : Requirement, Public

↗ Refine «refine» Source -> Destination
From: LLR2 - Set carriageway state : Requirement, Public
To: HLR5.3 - Vehicle passing efficiency : Requirement, Public

HLR1.1 - Startup delay condition

Requirement «Functional» owned by 'HLR1 - System initialization', in package 'Functional Requirement'

If the system has already been turned off for more than x seconds, the system shall skip the start up delay.
Otherwise, the system shall start operations only after a fixed delay.

HLR1.1 - Startup delay condition
Version 1.0 Phase 1.0 Proposed
chioma created on 2/19/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR1.1 - Startup delay condition to «Functional» HLR1 - System initialization

[Direction is 'Source -> Destination'.]

CONNECTORS

CONNECTORS
<p>↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.1 - Startup delay condition : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.1 - Startup delay condition : Requirement, Public</p>

HLR1.2 - Resumption of normal operations

Requirement «Functional» owned by 'HLR1 - System initialization', in package 'Functional Requirement'

The controller shall resume operation at "Vehicles Passing Ready" phase. That is, a yellow light signal for the vehicle light set and a red light signal for the pedestrian light set.

HLR1.2 - Resumption of normal operations

Version 1.0 Phase 1.0 Proposed

chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS
<p>↳ Aggregation from «Functional» HLR1.2 - Resumption of normal operations to «Functional» HLR1 - System initialization [Direction is 'Source -> Destination'.]</p>

CONNECTORS
<p>↗ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.2 - Resumption of normal operations : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.2 - Resumption of normal operations : Requirement, Public</p>

HLR3.2 - Dual carriageway response

Requirement «Functional» owned by 'HLR3 - Call button response', in package 'Functional Requirement'

The system shall automatically trigger a button press on the other carriageway after X seconds when the call button on either carriageway is pressed

HLR3.2 - Dual carriageway response

Version 1.0 Phase 1.0 Proposed

chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from «Functional» HLR3.2 - Dual carriageway response to «Functional» HLR3 - Call button response [Direction is 'Source -> Destination'.]	

CONNECTORS	
↗ Refine «refine» Source -> Destination From: LLR3 - Process call button press : Requirement, Public To: HLR3.2 - Dual carriageway response : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination From: LLR3 - Process call button press : Requirement, Public To: HLR3.2 - Dual carriageway response : Requirement, Public	

HLR3.3 - Button request during light sequence

Requirement «Functional» owned by 'HLR3 - Call button response', in package 'Functional Requirement'

When the call button on either carriageway is pressed and the light sequence has initiated, the system shall not service any new requests.

HLR3.3 - Button request during light sequence
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

CONNECTORS	
↗ Dependency Source -> Destination From: HLR3.3 - Button request during light sequence : Requirement, Public To: HLR3 - Call button response : Requirement, Public	
↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public	

HLR4.1 - Pedestrian warning during EVP

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, the pedestrian traffic light shall be a red colour with flashing effects for that carriageway.

HLR4.1 - Pedestrian warning during EVP
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/22/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR4.1 - Pedestrian warning during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP)
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination
From: LLR6 - Display lights : Requirement, Public
To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public

↗ Refine «refine» Source -> Destination
From: LLR2 - Set carriageway state : Requirement, Public
To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public

↗ Abstraction «Refine» Source -> Destination
From: ControllerStateMachine : StateMachine, Public
To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public

↗ Abstraction «Refine» Source -> Destination
From: LLR2 - Set carriageway state : Requirement, Public
To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public

HLR4.3 - After emergency vehicle pass through

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

After the emergency vehicle has passed through, the traffic light state shall transition from the emergency vehicle approaching state to the vehicle slow down state.

HLR4.3 - After emergency vehicle pass through
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/22/2022

EXTERNAL REQUIREMENTS

Requirement. HLR4.4 - Button requests during EVP

When the system receives an approach signal from an emergency vehicle on one carriageway, all button presses on that carriageway are ignored until after the emergency vehicle has passed through.
[Stereotype is «Functional».]

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Realization from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4.4 - Button requests during EVP	[Direction is 'Source -> Destination'.]
↳ Aggregation from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]
CONNECTORS	
↗ Abstraction «Refine» Source -> Destination	
From: LLR2 - Set carriageway state : Requirement, Public	
To: HLR4.3 - After emergency vehicle pass through : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination	
From: ControllerStateMachine : StateMachine, Public	
To: HLR4.3 - After emergency vehicle pass through : Requirement, Public	
↗ Refine «refine» Source -> Destination	
From: LLR2 - Set carriageway state : Requirement, Public	
To: HLR4.3 - After emergency vehicle pass through : Requirement, Public	

HLR4.4 - Button requests during EVP

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, all button presses on that carriageway are ignored until after the emergency vehicle has passed through.

HLR4.4 - Button requests during EVP
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/22/2022

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from «Functional» HLR4.4 - Button requests during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]
INCOMING STRUCTURAL RELATIONSHIPS	
➡ Realization from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4.4 - Button requests during EVP	[Direction is 'Source -> Destination'.]

CONNECTORS
<p>↗ Refine «refine» Source -> Destination From: LLR5 - Handle emergency vehicle signal received : Requirement, Public To: HLR4.4 - Button requests during EVP : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.4 - Button requests during EVP : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.4 - Button requests during EVP : Requirement, Public</p>

HLR5.1 - Green signal constraint

Requirement «Functional» owned by 'HLR5 - Traffic lights regime', in package 'Functional Requirement'

The vehicle and pedestrian light set on one carriageway must never be green at the same time.

HLR5.1 - Green signal constraint
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS
<p>↳ Aggregation from «Functional» HLR5.1 - Green signal constraint to «Functional» HLR5 - Traffic lights regime [Direction is 'Source -> Destination'.]</p>

CONNECTORS
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR5.1 - Green signal constraint : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR5.1 - Green signal constraint : Requirement, Public</p>

HLR5.2 - Single light display on light set at a given time

Requirement «Functional» owned by 'HLR5 - Traffic lights regime', in package 'Functional Requirement'

When the light switching occurs, each signal head must display no more than one light signal at a time.

HLR5.2 - Single light display on light set at a given time
Version 1.0 Phase 1.0 Proposed

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR5.2 - Single light display on light set at a given time to «Functional» HLR5 - Traffic lights regime

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination

From: LLR6 - Display lights : Requirement, Public
To: HLR5.2 - Single light display on light set at a given time : Requirement, Public

↗ Abstraction «Refine» Source -> Destination

From: ControllerStateMachine : StateMachine, Public
To: HLR5.2 - Single light display on light set at a given time : Requirement, Public

HLR5.3 - Vehicle passing efficiency

Requirement «Functional» owned by 'HLR5 - Traffic lights regime', in package 'Functional Requirement'

x seconds after the pedestrian light set has been switched to red, vehicle light set shall be switched to green.

HLR5.3 - Vehicle passing efficiency

Version 1.0 Phase 1.0 Proposed

chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR5.3 - Vehicle passing efficiency to «Functional» HLR5 - Traffic lights regime

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Abstraction «Refine» Source -> Destination

From: ControllerStateMachine : StateMachine, Public
To: HLR5.3 - Vehicle passing efficiency : Requirement, Public

↗ Refine «refine» Source -> Destination

From: LLR2 - Set carriageway state : Requirement, Public
To: HLR5.3 - Vehicle passing efficiency : Requirement, Public

HLR6 - System scope

ArchitecturalRequirement «ArchitecturalRequirement» in package 'Non-Functional Requirement'

The system's scope shall consist of a single controller that controls a pair of signal heads for motorists and a pair of signal heads for pedestrians on both carriageways

HLR6 - System scope
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

CONNECTORS

↗ **Abstraction** «Refine» Source -> Destination
From: TrafficLightController : Class, Public
To: HLR6 - System scope : ArchitecturalRequirement, Public

HLR7 - IR sensor range

ArchitecturalRequirement «*ArchitecturalRequirement*» in package 'Non-Functional Requirement'

The system shall detect infrared signals through passive infrared sensors within a minimum range of X meters.

HLR7 - IR sensor range
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

CONNECTORS

↗ **Abstraction** «Refine» Source -> Destination
From: Carriageway : Class, Public
To: HLR7 - IR sensor range : ArchitecturalRequirement, Public

ControllerStateMachine

StateMachine in package 'System Architecture'

ControllerStateMachine
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/21/2022

ELEMENTS OWNED BY ControllerStateMachine

- █ Emergency Vehicle Approaching : State
- █ Emergency Vehicle Approaching : State
- █ Pedestrians Crossing : State

ELEMENTS OWNED BY ControllerStateMachine	
■ Pedestrians Crossing Almost Done : State	
■ Pedestrians Crossing Done : State	
■ State1 : State	
■ Vehicles Passing : State	
■ Vehicles Passing Almost Done : State	
■ Vehicles Passing Done : State	
■ Vehicles Passing Ready : State	
■ Vehicles Passing Slow Down : State	
■ Final : Final State	
■ Initial : Initial State	
■ Call Button Pressed : Trigger	
■ System Initialization : Trigger	
■ Time Elapsed : Trigger	
■ Time Elapsed : Trigger	
■ Time Elapsed : Trigger	
■ Time Elapsed : Trigger	
■ Time Elapsed : Trigger	
■ Time Elapsed : Trigger	
■ Time Elapsed : Trigger	
■ Time Elapsed : Trigger	
■ Time Elapsed : Trigger	
■ Time Elapsed : Trigger	

CONNECTORS	

CONNECTORS
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR5.1 - Green signal constraint : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR5.3 - Vehicle passing efficiency : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR4.3 - After emergency vehicle pass through : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR2 - Default light signals : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR5 - Traffic lights regime : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR5.2 - Single light display on light set at a given time : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public</p>

Emergency Vehicle Approaching

State owned by 'ControllerStateMachine', in package 'System Architecture'

Emergency Vehicle Approaching
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

Emergency Vehicle Approaching

State owned by 'ControllerStateMachine', in package 'System Architecture'

Emergency Vehicle Approaching
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

Pedestrians Crossing

State owned by 'ControllerStateMachine', in package 'System Architecture'

Pedestrians Crossing
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

- ↳ Transition from Pedestrians Crossing to Pedestrians Crossing Almost Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 10000 seconds

INCOMING BEHAVIORAL RELATIONSHIPS

- ↖ Transition from Vehicles Passing Done to Pedestrians Crossing
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

OPERATIONS

- ♀ entry () : entry Public

Changes VLS and PLS state for the specified carriageway.
Vehicle Light Set State = {color: 'red', flash: false}
Pedestrian Light Set State = {color: 'green', flash: false}

Duration before Trigger = 10 seconds

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Pedestrians Crossing Almost Done

State owned by 'ControllerStateMachine', in package 'System Architecture'

Pedestrians Crossing Almost Done
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

- ↳ Transition from Pedestrians Crossing Almost Done to Pedestrians Crossing Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 10000 seconds

OUTGOING BEHAVIORAL RELATIONSHIPS
INCOMING BEHAVIORAL RELATIONSHIPS
<p>↳ Transition from Pedestrians Crossing to Pedestrians Crossing Almost Done Effect: setCarriagewayState(Carriageway, String)</p> <p>Triggers: Time Elapsed Time > 10000 seconds</p>
OPERATIONS
<p>♀ entry () : entry Public</p> <p>Changes VLS and PLS state for the specified carriageway. Vehicle Light Set State = {color: 'red', flash: false} Pedestrian Light Set State = {color: 'green', flash: true}</p> <p>Duration before Trigger = 5 seconds [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<h2>Pedestrians Crossing Done</h2> <p>State owned by 'ControllerStateMachine', in package 'System Architecture'</p> <p style="text-align: right;">Pedestrians Crossing Done Version 1.0 Phase 1.0 Proposed chioma created on 5/4/2022. Last modified 5/5/2022</p>
OUTGOING BEHAVIORAL RELATIONSHIPS
<p>↳ Transition from Pedestrians Crossing Done to Vehicles Passing Ready Effect: setCarriagewayState(Carriageway, String)</p> <p>Triggers: Time Elapsed Time > 3000 seconds</p>
INCOMING BEHAVIORAL RELATIONSHIPS
<p>↳ Transition from Pedestrians Crossing Almost Done to Pedestrians Crossing Done Effect: setCarriagewayState(Carriageway, String)</p> <p>Triggers: Time Elapsed Time > 10000 seconds</p>
OPERATIONS
<p>♀ entry () : entry Public</p> <p>Changes VLS and PLS state for the specified carriageway.</p>

OPERATIONS

Vehicle Light Set State = {color: 'red', flash: false}
Pedestrian Light Set State = {color: 'red', flash: false}

Duration before Trigger = 3 seconds
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

State1

State owned by 'ControllerStateMachine', in package 'System Architecture'

State1
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

INCOMING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing Slow Down to State1

Vehicles Passing

State owned by 'ControllerStateMachine', in package 'System Architecture'

Vehicles Passing
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing to Vehicles Passing Slow Down
Guard: callButtonReady == true
Effect: processCallButtonPress(Carriageway)

Triggers:
Call Button Pressed Call Carriageway.onCallButtonPress

↳ Transition from Vehicles Passing to Final

INCOMING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing Ready to Vehicles Passing
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

OPERATIONS

entry () : entry Public

Changes VLS and PLS state for specified carriageway. When duration is complete, changes "callButtonReady" to true

Vehicle Light Set State = {color: 'green', flash: false}
Pedestrian Light Set State = {color: 'red', flash: false}
Duration before Trigger = 10 seconds

Default system state. Triggered by a call button press. However, state transition only happens after an amount of time has passed.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Vehicles Passing Almost Done

State owned by 'ControllerStateMachine', in package 'System Architecture'

Vehicles Passing Almost Done
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

Transition from Vehicles Passing Almost Done to Vehicles Passing Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

INCOMING BEHAVIORAL RELATIONSHIPS

Transition from Vehicles Passing Slow Down to Vehicles Passing Almost Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 5000 seconds

OPERATIONS

entry () : entry Public

Changes VLS and PLS state for the specified carriageway.

Vehicle Light Set State = {color: 'yellow', flash: true}
Pedestrian Light Set State = {color: 'red', flash: false}

Duration before Trigger = 3 seconds

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Vehicles Passing Done

State owned by 'ControllerStateMachine', in package 'System Architecture'

Vehicles Passing Done
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing Done to Pedestrians Crossing
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

INCOMING BEHAVIORAL RELATIONSHIPS

↗ Transition from Vehicles Passing Almost Done to Vehicles Passing Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

OPERATIONS

EntryPoint : entry Public

Changes VLS and PLS state for the specified carriageway.
Vehicle Light Set State = {color: 'red', flash: true}
Pedestrian Light Set State = {color: 'red', flash: false}

Duration before Trigger = 3 seconds
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Vehicles Passing Ready

State owned by 'ControllerStateMachine', in package 'System Architecture'

Vehicles Passing Ready
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing Ready to Vehicles Passing
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

OUTGOING BEHAVIORAL RELATIONSHIPS
INCOMING BEHAVIORAL RELATIONSHIPS
<p>↳ Transition from Initial to Vehicles Passing Ready Effect: startUpSequence()</p> <p>↳ Transition from Pedestrians Crossing Done to Vehicles Passing Ready Effect: setCarriagewayState(Carriageway, String)</p> <p>Triggers: Time Elapsed Time > 3000 seconds</p>
OPERATIONS
<p>◊ entry () : entry Public</p> <p>Changes VLS and PLS state for the specified carriageway. Vehicle Light Set State = {color: 'yellow', flash: false} Pedestrian Light Set State = {color: 'red', flash: false}</p> <p>Duration before Trigger = 3 seconds First state after downtime system initialization. Signals that system is about to move into its default state [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]</p>
<h2>Vehicles Passing Slow Down</h2> <p><i>State owned by 'ControllerStateMachine', in package 'System Architecture'</i></p>
Vehicles Passing Slow Down Version 1.0 Phase 1.0 Proposed chioma created on 5/4/2022. Last modified 5/5/2022
OUTGOING BEHAVIORAL RELATIONSHIPS
<p>↳ Transition from Vehicles Passing Slow Down to Vehicles Passing Almost Done Effect: setCarriagewayState(Carriageway, String)</p> <p>Triggers: Time Elapsed Time > 5000 seconds</p>
<p>↳ Transition from Vehicles Passing Slow Down to State1</p>
INCOMING BEHAVIORAL RELATIONSHIPS
<p>↳ Transition from Vehicles Passing to Vehicles Passing Slow Down Guard: callButtonReady == true</p>

INCOMING BEHAVIORAL RELATIONSHIPS

Effect: processCallButtonPress(Carriageway)

Triggers:
Call Button Pressed Call Carriageway.onCallButtonPress**OPERATIONS**

entry () : entry Public

Changes VLS and PLS state for the specified carriageway.

Vehicle Light Set State = {color: 'green', flash: true}

Pedestrian Light Set State = {color: 'red', flash: false}

Duration before Trigger = 5 seconds

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Final*Final State owned by 'ControllerStateMachine', in package 'System Architecture'***INCOMING BEHAVIORAL RELATIONSHIPS**

Transition from Vehicles Passing to Final

Initial*Initial State owned by 'ControllerStateMachine', in package 'System Architecture'***OUTGOING BEHAVIORAL RELATIONSHIPS**

Transition from Initial to Vehicles Passing Ready

Effect: startUpSequence()

Call Button Pressed*Trigger owned by 'ControllerStateMachine', in package 'System Architecture'*Call Button Pressed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022**System Initialization***Trigger owned by 'ControllerStateMachine', in package 'System Architecture'*

System Initialization
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Carriageway

Class in package 'Controller'

Carriageway
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/22/2022

STRUCTURAL PART OF Carriageway

cbEvent : Port

evpEvent : Port

EXTERNAL REQUIREMENTS

Requirement. LLR5 - Handle emergency vehicle signal received

When invoked, this operation sets the emergencySignalReceived flag to true. It returns no value. It is important to note that when the controller is changing the carriageway state, it checks to see if the emergencySignalReceived flag is true. If it is, it changes the state to the 'emergencyVehicleApproachingState'.

[Stereotype is «Functional».]

Requirement. LLR6 - Display lights

This operation accepts the colour and flashing state configuration for the vehicle and pedestrian lights on a specific carriageway. It then changes the current colour and flashing effect state to the new one. It returns no value.

[Stereotype is «Functional».]

EXTERNAL REQUIREMENTS

- Requirement. LLR7 - Handle call button press

When invoked, this operation sets the callButtonPressed flag to true. If the callButtonReady flag is true, it then proceeds to call the processCallButtonPress function on the controller. It passes a reference to itself as an argument. If the callButtonReady flag is false, it does nothing. This operation returns no value.

[Stereotype is «Functional».]

OUTGOING STRUCTURAL RELATIONSHIPS

- ↳ Realization from Carriageway to «Functional» LLR6 - Display lights
[Direction is 'Source -> Destination'.]
- ↳ Realization from Carriageway to «Functional» LLR7 - Handle call button press
[Direction is 'Source -> Destination'.]
- ↳ Realization from Carriageway to «Functional» LLR5 - Handle emergency vehicle signal received
[Direction is 'Source -> Destination'.]

CONNECTORS

- ↗ Abstraction «Refine» Source -> Destination
From: Carriageway : Class, Public
To: HLR7 - IR sensor range : ArchitecturalRequirement, Public

ATTRIBUTES

- ◆ callButtonPressed : Boolean Private = false
[Is static True. Containment is Not Specified.]
- ◆ callButtonReady : Boolean Private = false
Properties:
ea_guid = {7CAA160E-0E80-4e84-B042-BB7CD692CBC0}
[Is static True. Containment is Not Specified.]
- ◆ controller : TrafficLightController Private
[Is static True. Containment is Not Specified.]
- ◆ emergencySignalReceived : Boolean Private = false
[Is static True. Containment is Not Specified.]
- ◆ id : Number Private
[Is static True. Containment is Not Specified.]
- ◆ pls : Object Private = null
[Is static True. Containment is Not Specified.]

ATTRIBUTES
❖ vls : Object Private = null [Is static True. Containment is Not Specified.]
OPERATIONS
❖ display (vls : Object , pls : Object) : void Public Properties: ea_guid = {09765566-B1F5-4449-B163-C26BC52FED4A} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
❖ getCallButtonPressed () : Boolean Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
❖ getCallButtonReady () : Boolean Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
❖ getEmergencySignalReceived () : Boolean Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
❖ onCallButtonPress () : void Public Properties: ea_guid = {2D10D394-CDF9-4ea-A6D3-6ED057886444} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
❖ onEmergencyVehicleSignalReceived () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
❖ setCallButtonPressed (callButtonPressed : Boolean) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
❖ setCallButtonReady (callButtonReady : Boolean) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
❖ setEmergencySignalReceived (emergencySignalReceived : Boolean) : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

cbEvent

Port owned by 'Carriageway', in package 'Controller'

cbEvent
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

evpEvent

Port owned by 'Carriageway', in package 'Controller'

evpEvent
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

TrafficLightController

Class in package 'Controller'

TrafficLightController
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/22/2022

STRUCTURAL PART OF TrafficLightController

 timerEvent : Port

EXTERNAL REQUIREMENTS

Requirement. LLR1 - Startup sequence

This operation gets the downtime (in milliseconds) from the external system, DowntimeTracker. If the downtime is less than the configured startup delay threshold, it waits for the configured startup delay before changing the state to 'vehiclesPassingReady'. Otherwise, if the downtime is greater than the configured startup delay threshold, it changes the state immediately to 'vehiclesPassingReady'. This operation returns no value.

[Stereotype is «Functional».]

Requirement. LLR2 - Set carriageway state

The operation accepts as parameters the carriageway on which the state is to be changed and a string indicating the state. It sets the callButtonReady flag to false, indicating that any button presses should be ignored for the time being. It also retrieves the state's properties (i.e. colour and flashing states of the vehicle and pedestrian lights, duration of the state, next state, next action) from the state config object and then invokes the carriageway display function to display the appropriate lights. After this, it delays for the time specified in the state's duration attribute and then gets the emergencySignalReceived flag from the carriageway class. If the flag is true, and the state is not 'pedestriansCrossing' and 'pedestriansCrossingAlmostDone', then it performs the following operations:

- Sets the emergencySignalReceived flag to false
- Sets the nextState property to 'emergencyVehicleApproaching'
- Sets the nextAction property to false

If the nextState property exists, it calls itself with the carriageway argument and the nextState. However, if the nextAction property exists, indicating that the "vehiclesPassing" state duration is complete, the operation vehiclePassingDurationComplete is called. The operation returns no value.

[Stereotype is «Functional».]

EXTERNAL REQUIREMENTS

Requirement. LLR3 - Process call button press

This operation accepts the carriageway on which the call button press is to be processed as a parameter. It gets the callButtonPressed flag from the carriageway class. If the callButtonPressed flag is true, it performs the following operations:

- It changes the callButtonPressed flag to false
- Because the carriageway whose call button was pressed is passed as an argument, it gets a reference to the 'other' carriageway.
- It invokes the setCarriagewayState function with the state 'vehiclesPassingSlowDown' for the carriageway passed in.
- It calls the setCarriagewayState function for the 'other' carriageway after a defined carriagewayWaitPeriod.

If the callButtonPressed flag is false, indicating that there are no button presses to process, no action is taken. This operation is only performed when the state, vehiclePassing's wait duration has expired, or after the system has received a call button press and the callButtonReady flag is set to true. This operation returns no value.

[Stereotype is «Functional».]

Requirement. LLR4 - Vehicle passing duration complete

This operation accepts the carriageway on which the call button press is to be processed as a parameter. It sets the callButtonReady flag to true, which indicates that the minimum wait period for vehicle passing has elapsed. It also invokes the processCallButtonPress function passing in a reference to the carriageway on which the call button was pressed. This operation returns no value.

[Stereotype is «Functional».]

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Realization from TrafficLightController to «Functional» LLR1 - Startup sequence
[Direction is 'Source -> Destination'.]

↳ Realization from TrafficLightController to «Functional» LLR3 - Process call button press
[Direction is 'Source -> Destination'.]

↳ Realization from TrafficLightController to «Functional» LLR2 - Set carriageway state
[Direction is 'Source -> Destination'.]

↳ Realization from TrafficLightController to «Functional» LLR4 - Vehicle passing duration complete
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Abstraction «Refine» Source -> Destination
From: TrafficLightController : Class, Public
To: HLR6 - System scope : ArchitecturalRequirement, Public

ATTRIBUTES

❖ c1 : Carriageway Private
[Is static True. Containment is Not Specified.]

ATTRIBUTES	
◆ c2 : Carriageway Private	[Is static True. Containment is Not Specified.]
◆ carriagewayWaitPeriod : Number Private Const = 3000	[Is static True. Containment is Not Specified.]
◆ emergencyStateDuration : Number Private Const = 20000	[Is static True. Containment is Not Specified.]
◆ maxStateDuration : Number Private Const = 10000	[Is static True. Containment is Not Specified.]
◆ midStateDuration : Number Private Const = 5000	[Is static True. Containment is Not Specified.]
◆ minStateDuration : Number Private Const = 3000	[Is static True. Containment is Not Specified.]
◆ startUpDelay : Number Private Const = 10000	[Is static True. Containment is Not Specified.]
◆ startUpDelayThreshold : Number Private Const = 3000	[Is static True. Containment is Not Specified.]
◆ stateConfig : Object Private Const	[Is static True. Containment is Not Specified.]

OPERATIONS	
◆ processCallButtonPress (carriageway : Carriageway) : void Public	
Properties:	
ea_guid = {52705713-51E6-4862-9026-879D15BA0FDB}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ pulseReceived () : void Public	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setCarriagewayState (carriageway : Carriageway , state : String) : void Private	
Properties:	
ea_guid = {AE926857-86A9-4018-97E5-2AA4B9D14D10}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ startUpSequence () : void Private	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

❖ vehiclePassingDurationComplete (carriageway : Carriageway) : void Private
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

timerEvent

Port owned by 'TrafficLightController', in package 'Controller'

timerEvent
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

timerEvent

Port owned by 'TrafficLightController', in package 'Controller'

timerEvent
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

cbEvent

Port owned by 'Carriageway', in package 'Controller'

cbEvent
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

evpEvent

Port owned by 'Carriageway', in package 'Controller'

evpEvent
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

LLR1 - Startup sequence

Requirement «Functional» in package 'Low Level Requirements'

This operation gets the downtime (in milliseconds) from the external system, DowntimeTracker. If the downtime is less than the configured startup delay threshold, it waits for the configured startup delay before changing the state to 'vehiclesPassingReady'. Otherwise, if the downtime is greater than the configured startup delay threshold, it changes the state immediately to 'vehiclesPassingReady'. This operation returns no value.

LLR1 - Startup sequence
Version 1.0 Phase 1.0 Proposed
chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from TrafficLightController to «Functional» LLR1 - Startup sequence
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ **Abstraction** «Refine» Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: HLR1.1 - Startup delay condition : Requirement, Public

↗ **Abstraction** «Refine» Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: HLR1 - System initialization : Requirement, Public

↗ **Refine** «refine» Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: HLR1.2 - Resumption of normal operations : Requirement, Public

↗ **Refine** «refine» Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: HLR1.1 - Startup delay condition : Requirement, Public

↗ **Refine** «refine» Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: HLR1 - System initialization : Requirement, Public

↗ **Abstraction** «Refine» Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: HLR1.2 - Resumption of normal operations : Requirement, Public

↗ **Dependency** Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: DowntimeTracker : Class, Public

LLR2 - Set carriageway state

Requirement «Functional» in package 'Low Level Requirements'

The operation accepts as parameters the carriageway on which the state is to be changed and a string indicating the state. It sets the callButtonReady flag to false, indicating that any button presses should be ignored for the time being. It also retrieves the state's properties (i.e. colour and flashing states of the vehicle and pedestrian lights, duration of the state, next state, next action) from the state config object and then invokes the carriageway display function to display the appropriate lights. After this, it delays for the time

specified in the state's duration attribute and then gets the emergencySignalReceived flag from the carriageway class. If the flag is true, and the state is not 'pedestriansCrossing' and 'pedestriansCrossingAlmostDone', then it performs the following operations:

- Sets the emergencySignalReceived flag to false
- Sets the nextState property to 'emergencyVehicleApproaching'
- Sets the nextAction property to false

If the nextState property exists, it calls itself with the carriageway argument and the nextState. However, if the nextAction property exists, indicating that the "vehiclesPassing" state duration is complete, the operation vehiclePassingDurationComplete is called. The operation returns no value.

LLR2 - Set carriageway state
Version 1.0 Phase 1.0 Proposed
chioma created on 5/5/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS
<p>→ Realization from TrafficLightController to «Functional» LLR2 - Set carriageway state [Direction is 'Source -> Destination'.]</p>
CONNECTORS
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.3 - After emergency vehicle pass through : Requirement, Public</p> <p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR5 - Traffic lights regime : Requirement, Public</p> <p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public</p> <p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.3 - After emergency vehicle pass through : Requirement, Public</p> <p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public</p> <p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.4 - Button requests during EVP : Requirement, Public</p> <p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR2 - Default light signals : Requirement, Public</p>

CONNECTORS
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR5.3 - Vehicle passing efficiency : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.2 - EVP state transition : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR5.1 - Green signal constraint : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.4 - Button requests during EVP : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR2 - Default light signals : Requirement, Public</p>

LLR3 - Process call button press

Requirement «Functional» in package 'Low Level Requirements'

This operation accepts the carriageway on which the call button press is to be processed as a parameter. It gets the callButtonPressed flag from the carriageway class. If the callButtonPressed flag is true, it performs the following operations:

- It changes the callButtonPressed flag to false
- Because the carriageway whose call button was pressed is passed as an argument, it gets a reference to the 'other' carriageway.
- It invokes the setCarriagewayState function with the state 'vehiclesPassingSlowDown' for the carriageway passed in.
- It calls the setCarriagewayState function for the 'other' carriageway after a defined carriagewayWaitPeriod.

If the callButtonPressed flag is false, indicating that there are no button presses to process, no action is taken. This operation is only performed when the state, vehiclePassing's wait duration has expired, or after the system has received a call button press and the callButtonReady flag is set to true. This operation returns no value.

ELEMENTS OWNED BY LLR3 - Process call button press	
█	User : Actor «Functional»
█	InteractionFragment : InteractionFragment «Functional»
█	c1 : Sequence «Functional»
█	c2 : Sequence «Functional»
█	controller : Sequence «Functional»
█	Object1 : Sequence «control»

INCOMING STRUCTURAL RELATIONSHIPS	
➡	Realization from TrafficLightController to «Functional» LLR3 - Process call button press [Direction is 'Source -> Destination'.]

CONNECTORS	
↗	Refine «refine» Source -> Destination From: LLR3 - Process call button press : Requirement, Public To: HLR2 - Default light signals : Requirement, Public
↗	Refine «refine» Source -> Destination From: LLR3 - Process call button press : Requirement, Public To: HLR3.2 - Dual carriageway response : Requirement, Public
↗	Abstraction «Refine» Source -> Destination From: LLR3 - Process call button press : Requirement, Public To: HLR3.2 - Dual carriageway response : Requirement, Public
↗	Abstraction «Refine» Source -> Destination From: LLR3 - Process call button press : Requirement, Public To: HLR2 - Default light signals : Requirement, Public

User

Actor owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

User

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: onCallButtonPress()
↳ Sequence from User to c2

Name: onCallButtonPress()
↳ Sequence from User to c1

InteractionFragment

InteractionFragment owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

c1

Sequence owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

STRUCTURAL PART OF c1: Carriageway

❶ Port : Port

❶ Port : Port

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: processCallButtonPress(Carriageway)
↳ Sequence from c1 to controller

INCOMING BEHAVIORAL RELATIONSHIPS

Name: display(Object, Object)
↗ Sequence from controller to c1

Name: onCallButtonPress()
↗ Sequence from User to c1

Name: display(Object, Object)

INCOMING BEHAVIORAL RELATIONSHIPS Sequence from controller to c1**Port***Port owned by 'c1: Carriageway', in package 'Low Level Requirements'*

Port

Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022**Port***Port owned by 'c1: Carriageway', in package 'Low Level Requirements'*

Port

Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022**c2***Sequence owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'*

c2

Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022**STRUCTURAL PART OF c2: Carriageway** Port : Port Port : Port**OUTGOING BEHAVIORAL RELATIONSHIPS**

Name: processCallButtonPress(Carriageway)

 Sequence from c2 to controller**INCOMING BEHAVIORAL RELATIONSHIPS**

Name: onCallButtonPress()

 Sequence from User to c2

Name: display(Object, Object)

 Sequence from controller to c2

INCOMING BEHAVIORAL RELATIONSHIPS

Name: display(Object, Object)
↳ Sequence from controller to c2

Port

Port owned by 'c2: Carriageway', in package 'Low Level Requirements'

Port
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

Port

Port owned by 'c2: Carriageway', in package 'Low Level Requirements'

Port
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

controller

Sequence owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

controller
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

STRUCTURAL PART OF controller: TrafficLightController

↳ Port : Port

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: display(Object, Object)
↳ Sequence from controller to c1

Name: display(Object, Object)
↳ Sequence from controller to c2

Name: display(Object, Object)
↳ Sequence from controller to c2

Name: setCarriagewayState(Carriageway, String)
↳ Sequence from controller to controller

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: display(Object, Object)	Sequence from controller to c1
Name: setCarriagewayState(Carriageway, String)	Sequence from controller to controller
INCOMING BEHAVIORAL RELATIONSHIPS	
Name: processCallButtonPress(Carriageway)	Sequence from c1 to controller
Name: setCarriagewayState(Carriageway, String)	Sequence from controller to controller
Name: setCarriagewayState(Carriageway, String)	Sequence from controller to controller
Name: processCallButtonPress(Carriageway)	Sequence from c2 to controller

Port

Port owned by 'controller: TrafficLightController', in package 'Low Level Requirements'

Port
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

Object1

Sequence «control» owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

Object1
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

LLR4 - Vehicle passing duration complete

Requirement «Functional» in package 'Low Level Requirements'

This operation accepts the carriageway on which the call button press is to be processed as a parameter. It sets the callButtonReady flag to true, which indicates that the minimum wait period for vehicle passing has elapsed. It also invokes the processCallButtonPress function passing in a reference to the carriageway on which the call button was pressed. This operation returns no value.

LLR4 - Vehicle passing duration complete
Version 1.0 Phase 1.0 Proposed
chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS

- Realization from TrafficLightController to «Functional» LLR4 - Vehicle passing duration complete
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination
From: LLR4 - Vehicle passing duration complete : Requirement, Public
To: HLR3 - Call button response : Requirement, Public

↗ Abstraction «Refine» Source -> Destination
From: LLR4 - Vehicle passing duration complete : Requirement, Public
To: HLR3 - Call button response : Requirement, Public

↗ Refine «refine» Source -> Destination
From: LLR4 - Vehicle passing duration complete : Requirement, Public
To: HLR2 - Default light signals : Requirement, Public

↗ Abstraction «Refine» Source -> Destination
From: LLR4 - Vehicle passing duration complete : Requirement, Public
To: HLR2 - Default light signals : Requirement, Public

LLR5 - Handle emergency vehicle signal received

Requirement «Functional» in package 'Low Level Requirements'

When invoked, this operation sets the emergencySignalReceived flag to true. It returns no value. It is important to note that when the controller is changing the carriageway state, it checks to see if the emergencySignalReceived flag is true. If it is, it changes the state to the 'emergencyVehicleApproachingState'.

LLR5 - Handle emergency vehicle signal received
Version 1.0 Phase 1.0 Proposed
chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS

- Realization from Carriageway to «Functional» LLR5 - Handle emergency vehicle signal received
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination
From: LLR5 - Handle emergency vehicle signal received : Requirement, Public
To: HLR4.4 - Button requests during EVP : Requirement, Public

CONNECTORS
<p>↗ Refine «refine» Source -> Destination From: LLR5 - Handle emergency vehicle signal received : Requirement, Public To: HLR4.2 - EVP state transition : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR5 - Handle emergency vehicle signal received : Requirement, Public To: HLR4 - Emergency vehicle preemption (EVP) : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR5 - Handle emergency vehicle signal received : Requirement, Public To: HLR4 - Emergency vehicle preemption (EVP) : Requirement, Public</p>

LLR6 - Display lights

Requirement «Functional» in package 'Low Level Requirements'

This operation accepts the colour and flashing state configuration for the vehicle and pedestrian lights on a specific carriageway. It then changes the current colour and flashing effect state to the new one. It returns no value.

LLR6 - Display lights
 Version 1.0 Phase 1.0 Proposed
 chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Realization from Carriageway to «Functional» LLR6 - Display lights [Direction is 'Source -> Destination'.]</p>
CONNECTORS
<p>↗ Refine «refine» Source -> Destination From: LLR6 - Display lights : Requirement, Public To: HLR5.2 - Single light display on light set at a given time : Requirement, Public</p> <p>↗ Refine «refine» Source -> Destination From: LLR6 - Display lights : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public</p>

LLR7 - Handle call button press

Requirement «Functional» in package 'Low Level Requirements'

When invoked, this operation sets the callButtonPressed flag to true. If the callButtonReady flag is true, it then proceeds to call the processCallButtonPress function on the controller. It passes a reference to itself as an argument. If the callButtonReady flag is false, it does nothing. This operation returns no value.

LLR7 - Handle call button press
Version 1.0 Phase 1.0 Proposed
chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS

- ➡ Realization from Carriageway to «Functional» LLR7 - Handle call button press
[Direction is 'Source -> Destination'.]

CONNECTORS

- ↗ Refine «refine» Source -> Destination
From: LLR7 - Handle call button press : Requirement, Public
To: HLR3 - Call button response : Requirement, Public
- ↗ Refine «refine» Source -> Destination
From: LLR7 - Handle call button press : Requirement, Public
To: HLR3.1 - Pedestrian crossing wait time : Requirement, Public
- ↗ Abstraction «Refine» Source -> Destination
From: LLR7 - Handle call button press : Requirement, Public
To: HLR3 - Call button response : Requirement, Public

DowntimeTracker

Class in package 'External'

DowntimeTracker
Version 1.0 Phase 1.0 Proposed
chioma created on 5/21/2022. Last modified 5/22/2022

CONNECTORS

- ↗ Dependency Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: DowntimeTracker : Class, Public

ATTRIBUTES

- ❖ downtime : Number Private
[Is static True. Containment is Not Specified.]

OPERATIONS

- ♀ getDownTime () : Number Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

PulseGenerator

Class in package 'External'

PulseGenerator
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/22/2022

External

Boundary in package 'Design'

External
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022
Extends

System

Boundary in package 'Design'

System
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022
Extends

Low Level Requirements

Package in package 'Design'

Low Level Requirements
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

Low Level Requirements diagram

Requirements diagram in package 'Low Level Requirements'

Low Level Requirements
Version 1.0
chioma created on 5/4/2022. Last modified 5/21/2022



Figure 3: Low Level Requirements

LLR1 - Startup sequence

Requirement «Functional» in package 'Low Level Requirements'

This operation gets the downtime (in milliseconds) from the external system, DowntimeTracker. If the downtime is less than the configured startup delay threshold, it waits for the configured startup delay before changing the state to 'vehiclesPassingReady'. Otherwise, if the downtime is greater than the configured startup delay threshold, it changes the state immediately to 'vehiclesPassingReady'. This operation returns no value.

LLR1 - Startup sequence
Version 1.0 Phase 1.0 Proposed
chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS	
<p>→ Realization from TrafficLightController to «Functional» LLR1 - Startup sequence [Direction is 'Source -> Destination'.]</p>	

CONNECTORS	
<p>↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.1 - Startup delay condition : Requirement, Public</p>	
<p>↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1 - System initialization : Requirement, Public</p>	
<p>↗ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.2 - Resumption of normal operations : Requirement, Public</p>	
<p>↗ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.1 - Startup delay condition : Requirement, Public</p>	

CONNECTORS
<p>↗ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1 - System initialization : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.2 - Resumption of normal operations : Requirement, Public</p>
<p>↗ Dependency Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: DowntimeTracker : Class, Public</p>

LLR2 - Set carriageway state

Requirement «Functional» in package 'Low Level Requirements'

The operation accepts as parameters the carriageway on which the state is to be changed and a string indicating the state. It sets the callButtonReady flag to false, indicating that any button presses should be ignored for the time being. It also retrieves the state's properties (i.e. colour and flashing states of the vehicle and pedestrian lights, duration of the state, next state, next action) from the state config object and then invokes the carriageway display function to display the appropriate lights. After this, it delays for the time specified in the state's duration attribute and then gets the emergencySignalReceived flag from the carriageway class. If the flag is true, and the state is not 'pedestriansCrossing' and 'pedestriansCrossingAlmostDone', then it performs the following operations:

- Sets the emergencySignalReceived flag to false
- Sets the nextState property to 'emergencyVehicleApproaching'
- Sets the nextAction property to false

If the nextState property exists, it calls itself with the carriageway argument and the nextState. However, if the nextAction property exists, indicating that the "vehiclesPassing" state duration is complete, the operation vehiclePassingDurationComplete is called. The operation returns no value.

LLR2 - Set carriageway state
Version 1.0 Phase 1.0 Proposed
chioma created on 5/5/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS
<p>⇒ Realization from TrafficLightController to «Functional» LLR2 - Set carriageway state [Direction is 'Source -> Destination'.]</p>
CONNECTORS
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.3 - After emergency vehicle pass through : Requirement, Public</p>

CONNECTORS
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR5 - Traffic lights regime : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.3 - After emergency vehicle pass through : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.4 - Button requests during EVP : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR2 - Default light signals : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR5.3 - Vehicle passing efficiency : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.2 - EVP state transition : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR5.1 - Green signal constraint : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.4 - Button requests during EVP : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public</p>

CONNECTORS

↗ Refine «refine» Source -> Destination
 From: LLR2 - Set carriageway state : Requirement, Public
 To: HLR2 - Default light signals : Requirement, Public

LLR3 - Process call button press

Requirement «Functional» in package 'Low Level Requirements'

This operation accepts the carriageway on which the call button press is to be processed as a parameter. It gets the callButtonPressed flag from the carriageway class. If the callButtonPressed flag is true, it performs the following operations:

- It changes the callButtonPressed flag to false
- Because the carriageway whose call button was pressed is passed as an argument, it gets a reference to the 'other' carriageway.
- It invokes the setCarriagewayState function with the state 'vehiclesPassingSlowDown' for the carriageway passed in.
- It calls the setCarriagewayState function for the 'other' carriageway after a defined carriagewayWaitPeriod.

If the callButtonPressed flag is false, indicating that there are no button presses to process, no action is taken. This operation is only performed when the state, vehiclePassing's wait duration has expired, or after the system has received a call button press and the callButtonReady flag is set to true. This operation returns no value.

LLR3 - Process call button press
 Version 1.0 Phase 1.0 Proposed
 chioma created on 5/21/2022. Last modified 5/22/2022

ELEMENTS OWNED BY LLR3 - Process call button press

- █ User : Actor «Functional»
- █ InteractionFragment : InteractionFragment «Functional»
- █ c1 : Sequence «Functional»
- █ c2 : Sequence «Functional»
- █ controller : Sequence «Functional»
- █ Object1 : Sequence «control»

INCOMING STRUCTURAL RELATIONSHIPS

INCOMING STRUCTURAL RELATIONSHIPS

- Realization from TrafficLightController to «Functional» LLR3 - Process call button press
 [Direction is 'Source -> Destination'.]

CONNECTORS

- ↗ Refine «refine» Source -> Destination

From: LLR3 - Process call button press : Requirement, Public
 To: HLR2 - Default light signals : Requirement, Public
- ↗ Refine «refine» Source -> Destination

From: LLR3 - Process call button press : Requirement, Public
 To: HLR3.2 - Dual carriageway response : Requirement, Public
- ↗ Abstraction «Refine» Source -> Destination

From: LLR3 - Process call button press : Requirement, Public
 To: HLR3.2 - Dual carriageway response : Requirement, Public
- ↗ Abstraction «Refine» Source -> Destination

From: LLR3 - Process call button press : Requirement, Public
 To: HLR2 - Default light signals : Requirement, Public

Call Button Event Sequence diagram

Interaction diagram in package 'Low Level Requirements'

Call Button Event Sequence
 Version 1.0
 chioma created on 5/5/2022. Last modified 5/22/2022

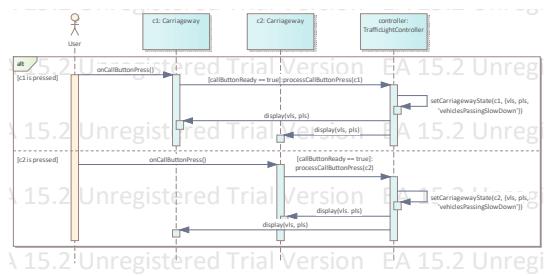


Figure 4: Call Button Event Sequence

INTERACTION MESSAGES

<p>✉ 1.0 'onCallButtonPress()' from 'User' sent to 'c1: Carriageway'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.1 'processCallButtonPress(Carriageway)' from 'c1: Carriageway' sent to 'controller: TrafficLightController'.</p> <p>When "callButtonReady == true". Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.2 'setCarriagewayState(Carriageway, String)' from 'controller: TrafficLightController' sent to 'controller: TrafficLightController'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.3 'display(Object, Object)' from 'controller: TrafficLightController' sent to 'c1: Carriageway'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.4 'display(Object, Object)' from 'controller: TrafficLightController' sent to 'c2: Carriageway'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.5 'onCallButtonPress()' from 'User' sent to 'c2: Carriageway'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.6 'processCallButtonPress(Carriageway)' from 'c2: Carriageway' sent to 'controller: TrafficLightController'.</p> <p>When "callButtonReady == true". Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.7 'setCarriagewayState(Carriageway, String)' from 'controller: TrafficLightController' sent to 'controller: TrafficLightController'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]
<p>✉ 1.8 'display(Object, Object)' from 'controller: TrafficLightController' sent to 'c2: Carriageway'.</p> <p>Synchronous Call. Returns void.</p>	[Return is False. Iteration is False. New group is False.]

✉ 1.9 'display(Object, Object)' from 'controller: TrafficLightController' sent to 'c1: Carriageway'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

User

Actor owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

User
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: onCallButtonPress()
↳ Sequence from User to c2

Name: onCallButtonPress()
↳ Sequence from User to c1

InteractionFragment

InteractionFragment owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

InteractionFragment
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

c1

Sequence owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

c1
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

STRUCTURAL PART OF c1: Carriageway

❶ Port : Port

❷ Port : Port

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: processCallButtonPress(Carriageway)

OUTGOING BEHAVIORAL RELATIONSHIPS
↳ Sequence from c1 to controller
INCOMING BEHAVIORAL RELATIONSHIPS
Name: display(Object, Object) ↗ Sequence from controller to c1
Name: onCallButtonPress() ↗ Sequence from User to c1
Name: display(Object, Object) ↗ Sequence from controller to c1

Port

Port owned by 'c1: Carriageway', in package 'Low Level Requirements'

Port
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

Port

Port owned by 'c1: Carriageway', in package 'Low Level Requirements'

Port
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

c2

Sequence owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

c2
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

STRUCTURAL PART OF c2: Carriageway
✖ Port : Port
✖ Port : Port

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: processCallButtonPress(Carriageway) ↳ Sequence from c2 to controller
INCOMING BEHAVIORAL RELATIONSHIPS
Name: onCallButtonPress() ↗ Sequence from User to c2
Name: display(Object, Object) ↗ Sequence from controller to c2
Name: display(Object, Object) ↗ Sequence from controller to c2

Port

Port owned by 'c2: Carriageway', in package 'Low Level Requirements'

Port
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

Port

Port owned by 'c2: Carriageway', in package 'Low Level Requirements'

Port
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

controller

Sequence owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

controller
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

STRUCTURAL PART OF controller: TrafficLightController
✖ Port : Port

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: display(Object, Object) ↳ Sequence from controller to c1

OUTGOING BEHAVIORAL RELATIONSHIPS
Name: display(Object, Object) ↳ Sequence from controller to c2
Name: display(Object, Object) ↳ Sequence from controller to c2
Name: setCarriagewayState(Carriageway, String) ↳ Sequence from controller to controller
Name: display(Object, Object) ↳ Sequence from controller to c1
Name: setCarriagewayState(Carriageway, String) ↳ Sequence from controller to controller

INCOMING BEHAVIORAL RELATIONSHIPS
Name: processCallButtonPress(Carriageway) ↗ Sequence from c1 to controller
Name: setCarriagewayState(Carriageway, String) ↗ Sequence from controller to controller
Name: setCarriagewayState(Carriageway, String) ↗ Sequence from controller to controller
Name: processCallButtonPress(Carriageway) ↗ Sequence from c2 to controller

Port

Port owned by 'controller: TrafficLightController', in package 'Low Level Requirements'

Port
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

Object1

Sequence «control» owned by 'LLR3 - Process call button press', in package 'Low Level Requirements'

Object1
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

LLR4 - Vehicle passing duration complete

Requirement «Functional» in package 'Low Level Requirements'

This operation accepts the carriageway on which the call button press is to be processed as a parameter. It sets the callButtonReady flag to true, which indicates that the minimum wait period for vehicle passing has elapsed. It also invokes the processCallButtonPress function passing in a reference to the carriageway on which the call button was pressed. This operation returns no value.

LLR4 - Vehicle passing duration complete

Version 1.0 Phase 1.0 Proposed

chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS

- ⇒ Realization from TrafficLightController to «Functional» LLR4 - Vehicle passing duration complete
 - [Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination

From: LLR4 - Vehicle passing duration complete : Requirement, Public
To: HLR3 - Call button response : Requirement, Public

↗ Abstraction «Refine» Source -> Destination

From: LLR4 - Vehicle passing duration complete : Requirement, Public
To: HLR3 - Call button response : Requirement, Public

↗ Refine «refine» Source -> Destination

From: LLR4 - Vehicle passing duration complete : Requirement, Public
To: HLR2 - Default light signals : Requirement, Public

↗ Abstraction «Refine» Source -> Destination

From: LLR4 - Vehicle passing duration complete : Requirement, Public
To: HLR2 - Default light signals : Requirement, Public

LLR5 - Handle emergency vehicle signal received

Requirement «Functional» in package 'Low Level Requirements'

When invoked, this operation sets the emergencySignalReceived flag to true. It returns no value. It is important to note that when the controller is changing the carriageway state, it checks to see if the emergencySignalReceived flag is true. If it is, it changes the state to the 'emergencyVehicleApproachingState'.

LLR5 - Handle emergency vehicle signal received

Version 1.0 Phase 1.0 Proposed

chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS	
→ Realization from Carriageway to «Functional» LLR5 - Handle emergency vehicle signal received [Direction is 'Source -> Destination'.]	
CONNECTORS	
<p>↗ Refine «refine» Source -> Destination From: LLR5 - Handle emergency vehicle signal received : Requirement, Public To: HLR4.4 - Button requests during EVP : Requirement, Public</p> <p>↗ Refine «refine» Source -> Destination From: LLR5 - Handle emergency vehicle signal received : Requirement, Public To: HLR4.2 - EVP state transition : Requirement, Public</p> <p>↗ Refine «refine» Source -> Destination From: LLR5 - Handle emergency vehicle signal received : Requirement, Public To: HLR4 - Emergency vehicle preemption (EVP) : Requirement, Public</p> <p>↗ Abstraction «Refine» Source -> Destination From: LLR5 - Handle emergency vehicle signal received : Requirement, Public To: HLR4 - Emergency vehicle preemption (EVP) : Requirement, Public</p>	

LLR6 - Display lights

Requirement «Functional» in package 'Low Level Requirements'

This operation accepts the colour and flashing state configuration for the vehicle and pedestrian lights on a specific carriageway. It then changes the current colour and flashing effect state to the new one. It returns no value.

LLR6 - Display lights
Version 1.0 Phase 1.0 Proposed
chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS	
→ Realization from Carriageway to «Functional» LLR6 - Display lights [Direction is 'Source -> Destination'.]	
CONNECTORS	
<p>↗ Refine «refine» Source -> Destination From: LLR6 - Display lights : Requirement, Public To: HLR5.2 - Single light display on light set at a given time : Requirement, Public</p>	

CONNECTORS

↗ **Refine** «refine» Source -> Destination
From: LLR6 - Display lights : Requirement, Public
To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public

LLR7 - Handle call button press

Requirement «Functional» in package 'Low Level Requirements'

When invoked, this operation sets the callButtonPressed flag to true. If the callButtonReady flag is true, it then proceeds to call the processCallButtonPress function on the controller. It passes a reference to itself as an argument. If the callButtonReady flag is false, it does nothing. This operation returns no value.

LLR7 - Handle call button press
Version 1.0 Phase 1.0 Proposed
chioma created on 5/21/2022. Last modified 5/22/2022

INCOMING STRUCTURAL RELATIONSHIPS

→ Realization from Carriageway to «Functional» LLR7 - Handle call button press
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ **Refine** «refine» Source -> Destination
From: LLR7 - Handle call button press : Requirement, Public
To: HLR3 - Call button response : Requirement, Public

↗ **Refine** «refine» Source -> Destination
From: LLR7 - Handle call button press : Requirement, Public
To: HLR3.1 - Pedestrian crossing wait time : Requirement, Public

↗ **Abstraction** «Refine» Source -> Destination
From: LLR7 - Handle call button press : Requirement, Public
To: HLR3 - Call button response : Requirement, Public

System Architecture

Package in package 'Design'

System Architecture
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

Controller

Package in package 'System Architecture'

Controller
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Controller diagram

Class diagram in package 'Controller'

Controller
Version 1.0

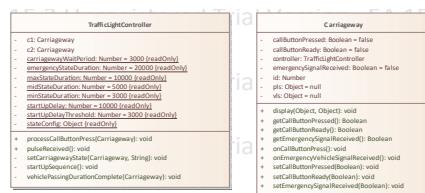


Figure 5: Controller

Carriageway

Class in package 'Controller'

Carriageway
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/22/2022

STRUCTURAL PART OF Carriageway

❖ cbEvent : Port

❖ evpEvent : Port

EXTERNAL REQUIREMENTS	
<input checked="" type="checkbox"/> Requirement. LLR5 - Handle emergency vehicle signal received	When invoked, this operation sets the emergencySignalReceived flag to true. It returns no value. It is important to note that when the controller is changing the carriageway state, it checks to see if the emergencySignalReceived flag is true. If it is, it changes the state to the 'emergencyVehicleApproachingState'. [Stereotype is «Functional».]
<input checked="" type="checkbox"/> Requirement. LLR6 - Display lights	This operation accepts the colour and flashing state configuration for the vehicle and pedestrian lights on a specific carriageway. It then changes the current colour and flashing effect state to the new one. It returns no value. [Stereotype is «Functional».]
<input checked="" type="checkbox"/> Requirement. LLR7 - Handle call button press	When invoked, this operation sets the callButtonPressed flag to true. If the callButtonReady flag is true, it then proceeds to call the processCallButtonPress function on the controller. It passes a reference to itself as an argument. If the callButtonReady flag is false, it does nothing. This operation returns no value. [Stereotype is «Functional».]

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Realization from Carriageway to «Functional» LLR6 - Display lights	[Direction is 'Source -> Destination'.]
↳ Realization from Carriageway to «Functional» LLR7 - Handle call button press	[Direction is 'Source -> Destination'.]
↳ Realization from Carriageway to «Functional» LLR5 - Handle emergency vehicle signal received	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Abstraction «Refine» Source -> Destination From: Carriageway : Class, Public To: HLR7 - IR sensor range : ArchitecturalRequirement, Public	

ATTRIBUTES	
◆ callButtonPressed : Boolean Private = false	[Is static True. Containment is Not Specified.]
◆ callButtonReady : Boolean Private = false Properties: ea_guid = {7CAA160E-0E80-4e84-B042-BB7CD692CBC0}	[Is static True. Containment is Not Specified.]

ATTRIBUTES	
◆ controller : TrafficLightController	Private [Is static True. Containment is Not Specified.]
◆ emergencySignalReceived : Boolean	Private = false [Is static True. Containment is Not Specified.]
◆ id : Number	Private [Is static True. Containment is Not Specified.]
◆ pls : Object	Private = null [Is static True. Containment is Not Specified.]
◆ vls : Object	Private = null [Is static True. Containment is Not Specified.]
OPERATIONS	
◆ display (vls : Object , pls : Object) : void	Public
Properties:	
ea_guid = {09765566-B1F5-4449-B163-C26BC52FED4A}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getCallButtonPressed () : Boolean	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getCallButtonReady () : Boolean	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ getEmergencySignalReceived () : Boolean	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ onCallButtonPress () : void	Public
Properties:	
ea_guid = {2D10D394-CDF9-4caa-A6D3-6ED057886444}	[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ onEmergencyVehicleSignalReceived () : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setCallButtonPressed (callButtonPressed : Boolean) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◆ setCallButtonReady (callButtonReady : Boolean) : void	Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

OPERATIONS

◆ setEmergencySignalReceived (emergencySignalReceived : Boolean) : void Public
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

cbEvent

Port owned by 'Carriageway', in package 'Controller'

cbEvent
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

evpEvent

Port owned by 'Carriageway', in package 'Controller'

evpEvent
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

TrafficLightController

Class in package 'Controller'

TrafficLightController
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/22/2022

STRUCTURAL PART OF TrafficLightController

⌚ timerEvent : Port

EXTERNAL REQUIREMENTS

Requirement. LLR1 - Startup sequence

This operation gets the downtime (in milliseconds) from the external system, DowntimeTracker. If the downtime is less than the configured startup delay threshold, it waits for the configured startup delay before changing the state to 'vehiclesPassingReady'. Otherwise, if the downtime is greater than the configured startup delay threshold, it changes the state immediately to 'vehiclesPassingReady'. This operation returns no value.

[Stereotype is «Functional».]

EXTERNAL REQUIREMENTS

Requirement. LLR2 - Set carriageway state

The operation accepts as parameters the carriageway on which the state is to be changed and a string indicating the state. It sets the callButtonReady flag to false, indicating that any button presses should be ignored for the time being. It also retrieves the state's properties (i.e. colour and flashing states of the vehicle and pedestrian lights, duration of the state, next state, next action) from the state config object and then invokes the carriageway display function to display the appropriate lights. After this, it delays for the time specified in the state's duration attribute and then gets the emergencySignalReceived flag from the carriageway class. If the flag is true, and the state is not 'pedestriansCrossing' and 'pedestriansCrossingAlmostDone', then it performs the following operations:

- Sets the emergencySignalReceived flag to false
- Sets the nextState property to 'emergencyVehicleApproaching'
- Sets the nextAction property to false

If the nextState property exists, it calls itself with the carriageway argument and the nextState. However, if the nextAction property exists, indicating that the "vehiclesPassing" state duration is complete, the operation vehiclePassingDurationComplete is called. The operation returns no value.

[Stereotype is «Functional».]

Requirement. LLR3 - Process call button press

This operation accepts the carriageway on which the call button press is to be processed as a parameter. It gets the callButtonPressed flag from the carriageway class. If the callButtonPressed flag is true, it performs the following operations:

- It changes the callButtonPressed flag to false
- Because the carriageway whose call button was pressed is passed as an argument, it gets a reference to the 'other' carriageway.
- It invokes the setCarriagewayState function with the state 'vehiclesPassingSlowDown' for the carriageway passed in.
- It calls the setCarriagewayState function for the 'other' carriageway after a defined carriagewayWaitPeriod.

If the callButtonPressed flag is false, indicating that there are no button presses to process, no action is taken. This operation is only performed when the state, vehiclePassing's wait duration has expired, or after the system has received a call button press and the callButtonReady flag is set to true. This operation returns no value.

[Stereotype is «Functional».]

Requirement. LLR4 - Vehicle passing duration complete

This operation accepts the carriageway on which the call button press is to be processed as a parameter. It sets the callButtonReady flag to true, which indicates that the minimum wait period for vehicle passing has elapsed. It also invokes the processCallButtonPress function passing in a reference to the carriageway on which the call button was pressed. This operation returns no value.

[Stereotype is «Functional».]

OUTGOING STRUCTURAL RELATIONSHIPS

- ↳ Realization from TrafficLightController to «Functional» LLR1 - Startup sequence
 [Direction is 'Source -> Destination'.]
- ↳ Realization from TrafficLightController to «Functional» LLR3 - Process call button press
 [Direction is 'Source -> Destination'.]

OUTGOING STRUCTURAL RELATIONSHIPS
↳ Realization from TrafficLightController to «Functional» LLR2 - Set carriageway state [Direction is 'Source -> Destination'.]
↳ Realization from TrafficLightController to «Functional» LLR4 - Vehicle passing duration complete [Direction is 'Source -> Destination'.]

CONNECTORS
↗ Abstraction «Refine» Source -> Destination From: TrafficLightController : Class, Public To: HLR6 - System scope : ArchitecturalRequirement, Public

ATTRIBUTES
◆ c1 : Carriageway Private [Is static True. Containment is Not Specified.]
◆ c2 : Carriageway Private [Is static True. Containment is Not Specified.]
◆ carriagewayWaitPeriod : Number Private Const = 3000 [Is static True. Containment is Not Specified.]
◆ emergencyStateDuration : Number Private Const = 20000 [Is static True. Containment is Not Specified.]
◆ maxStateDuration : Number Private Const = 10000 [Is static True. Containment is Not Specified.]
◆ midStateDuration : Number Private Const = 5000 [Is static True. Containment is Not Specified.]
◆ minStateDuration : Number Private Const = 3000 [Is static True. Containment is Not Specified.]
◆ startUpDelay : Number Private Const = 10000 [Is static True. Containment is Not Specified.]
◆ startUpDelayThreshold : Number Private Const = 3000 [Is static True. Containment is Not Specified.]
◆ stateConfig : Object Private Const [Is static True. Containment is Not Specified.]

OPERATIONS
◊ processCallButtonPress (carriageway : Carriageway) : void Public Properties: ea_guid = {52705713-51E6-4862-9026-879D15BA0FDB} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ pulseReceived () : void Public [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ setCarriagewayState (carriageway : Carriageway , state : String) : void Private Properties: ea_guid = {AE926857-86A9-4018-97E5-2AA4B9D14D10} [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ startUpSequence () : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]
◊ vehiclePassingDurationComplete (carriageway : Carriageway) : void Private [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

timerEvent

Port owned by 'TrafficLightController', in package 'Controller'

timerEvent
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

External

Package in package 'System Architecture'

External
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

External diagram

Class diagram in package 'External'

External
Version 1.0
chioma created on 5/4/2022. Last modified 5/21/2022



Figure 6: External

DowntimeTracker

Class in package 'External'

DowntimeTracker
Version 1.0 Phase 1.0 Proposed
chioma created on 5/21/2022. Last modified 5/22/2022

CONNECTORS

- ↗ **Dependency** Source -> Destination
 - From: LLR1 - Startup sequence : Requirement, Public
 - To: DowntimeTracker : Class, Public

ATTRIBUTES

- ❖ downtime : Number Private
 - [Is static True. Containment is Not Specified.]

OPERATIONS

- ❖ getDownTime() : Number Public
 - [Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

PulseGenerator

Class in package 'External'

PulseGenerator
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/22/2022

ControllerStateMachine

StateMachine in package 'System Architecture'

ControllerStateMachine
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/21/2022

ELEMENTS OWNED BY ControllerStateMachine	
■	Emergency Vehicle Approaching : State
■	Emergency Vehicle Approaching : State
■	Pedestrians Crossing : State
■	Pedestrians Crossing Almost Done : State
■	Pedestrians Crossing Done : State
■	State1 : State
■	Vehicles Passing : State
■	Vehicles Passing Almost Done : State
■	Vehicles Passing Done : State
■	Vehicles Passing Ready : State
■	Vehicles Passing Slow Down : State
■	Final : Final State
■	Initial : Initial State
■	Call Button Pressed : Trigger

ELEMENTS OWNED BY ControllerStateMachine	
■	System Initialization : Trigger
■	Time Elapsed : Trigger
■	Time Elapsed : Trigger
■	Time Elapsed : Trigger
■	Time Elapsed : Trigger
■	Time Elapsed : Trigger
■	Time Elapsed : Trigger
■	Time Elapsed : Trigger
■	Time Elapsed : Trigger
■	Time Elapsed : Trigger
CONNECTORS	
↗	Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR5.1 - Green signal constraint : Requirement, Public
↗	Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR5.3 - Vehicle passing efficiency : Requirement, Public
↗	Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR4.3 - After emergency vehicle pass through : Requirement, Public
↗	Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR2 - Default light signals : Requirement, Public
↗	Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR5 - Traffic lights regime : Requirement, Public
↗	Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR5.2 - Single light display on light set at a given time : Requirement, Public

CONNECTORS

 **Abstraction** «Refine» Source -> Destination
 From: ControllerStateMachine : StateMachine, Public
 To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public

LightSequence diagram

StateMachine diagram in package 'System Architecture'

LightSequence

Version 1.0

chioma created on 5/4/2022. Last modified 5/22/2022

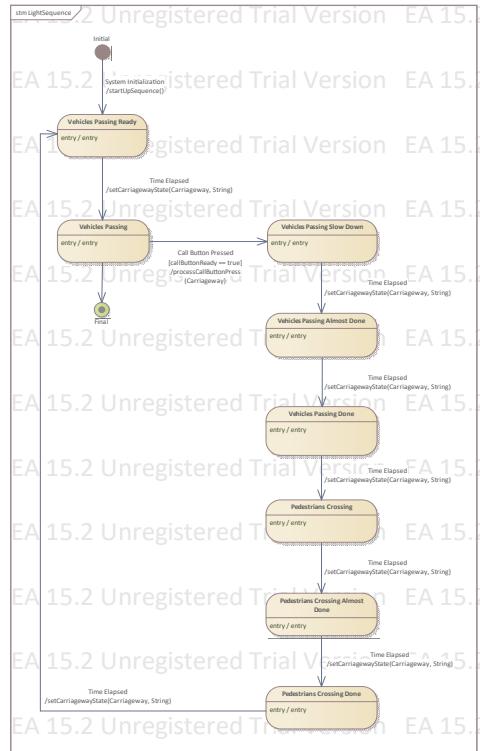


Figure 7: LightSequence

Emergency Vehicle Approaching

State owned by 'ControllerStateMachine', in package 'System Architecture'

Emergency Vehicle Approaching

State owned by 'ControllerStateMachine', in package 'System Architecture'

Emergency Vehicle Approaching
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

Pedestrians Crossing

State owned by 'ControllerStateMachine', in package 'System Architecture'

Pedestrians Crossing
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Transition from Pedestrians Crossing to Pedestrians Crossing Almost Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 10000 seconds

INCOMING BEHAVIORAL RELATIONSHIPS

↗ Transition from Vehicles Passing Done to Pedestrians Crossing
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

OPERATIONS

◊ entry () : entry Public

Changes VLS and PLS state for the specified carriageway.
Vehicle Light Set State = {color: 'red', flash: false}
Pedestrian Light Set State = {color: 'green', flash: false}

Duration before Trigger = 10 seconds

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Pedestrians Crossing Almost Done

State owned by 'ControllerStateMachine', in package 'System Architecture'

Pedestrians Crossing Almost Done
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

- ↳ Transition from Pedestrians Crossing Almost Done to Pedestrians Crossing Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 10000 seconds

INCOMING BEHAVIORAL RELATIONSHIPS

- ↗ Transition from Pedestrians Crossing to Pedestrians Crossing Almost Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 10000 seconds

OPERATIONS

- ♀ entry () : entry Public

Changes VLS and PLS state for the specified carriageway.
Vehicle Light Set State = {color: 'red', flash: false}
Pedestrian Light Set State = {color: 'green', flash: true}

Duration before Trigger = 5 seconds
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Pedestrians Crossing Done

State owned by 'ControllerStateMachine', in package 'System Architecture'

Pedestrians Crossing Done
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

- ↳ Transition from Pedestrians Crossing Done to Vehicles Passing Ready
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

↳ Transition from Pedestrians Crossing Almost Done to Pedestrians Crossing Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 10000 seconds

OPERATIONS

entry () : entry Public

Changes VLS and PLS state for the specified carriageway.
Vehicle Light Set State = {color: 'red', flash: false}
Pedestrian Light Set State = {color: 'red', flash: false}

Duration before Trigger = 3 seconds

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

State1

State owned by 'ControllerStateMachine', in package 'System Architecture'

State1
Version 1.0 Phase 1.0 Proposed
chioma created on 5/22/2022. Last modified 5/22/2022

INCOMING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing Slow Down to State1

Vehicles Passing

State owned by 'ControllerStateMachine', in package 'System Architecture'

Vehicles Passing
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing to Vehicles Passing Slow Down
Guard: callButtonReady == true
Effect: processCallButtonPress(Carriageway)

Triggers:
Call Button Pressed Call Carriageway.onCallButtonPress

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing to Final

INCOMING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing Ready to Vehicles Passing
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

OPERATIONS

♀ entry () : entry Public

Changes VLS and PLS state for specified carriageway. When duration is complete, changes "callButtonReady" to true

Vehicle Light Set State = {color: 'green', flash: false}
Pedestrian Light Set State = {color: 'red', flash: false}
Duration before Trigger = 10 seconds

Default system state. Triggered by a call button press. However, state transition only happens after an amount of time has passed.

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Vehicles Passing Almost Done

State owned by 'ControllerStateMachine', in package 'System Architecture'

Vehicles Passing Almost Done
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing Almost Done to Vehicles Passing Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

INCOMING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing Slow Down to Vehicles Passing Almost Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 5000 seconds

OPERATIONS

◊ entry () : entry Public

Changes VLS and PLS state for the specified carriageway.
Vehicle Light Set State = {color: 'yellow', flash: true}
Pedestrian Light Set State = {color: 'red', flash: false}

Duration before Trigger = 3 seconds

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Vehicles Passing Done

State owned by 'ControllerStateMachine', in package 'System Architecture'

Vehicles Passing Done
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Transition from Vehicles Passing Done to Pedestrians Crossing
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

INCOMING BEHAVIORAL RELATIONSHIPS

↗ Transition from Vehicles Passing Almost Done to Vehicles Passing Done
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

OPERATIONS

◊ entry () : entry Public

Changes VLS and PLS state for the specified carriageway.
Vehicle Light Set State = {color: 'red', flash: true}
Pedestrian Light Set State = {color: 'red', flash: false}

Duration before Trigger = 3 seconds

[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

Vehicles Passing Ready

State owned by 'ControllerStateMachine', in package 'System Architecture'

Vehicles Passing Ready
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

- ◆ Transition from Vehicles Passing Ready to Vehicles Passing
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

INCOMING BEHAVIORAL RELATIONSHIPS

- ◆ Transition from Initial to Vehicles Passing Ready
Effect: startUpSequence()
- ◆ Transition from Pedestrians Crossing Done to Vehicles Passing Ready
Effect: setCarriagewayState(Carriageway, String)

Triggers:
Time Elapsed Time > 3000 seconds

OPERATIONS

- entry () : entry Public
Changes VLS and PLS state for the specified carriageway.
Vehicle Light Set State = {color: 'yellow', flash: false}
Pedestrian Light Set State = {color: 'red', flash: false}
- Duration before Trigger = 3 seconds
First state after downtime system initialization.
Signals that system is about to move into its default state
[Is static False. Is abstract False. Is return array False. Is query False. Is synchronized False.]

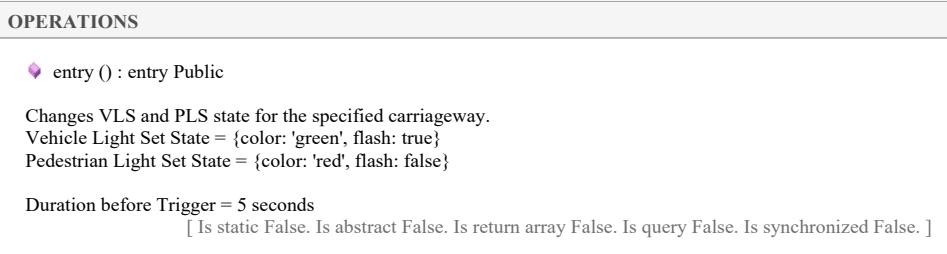
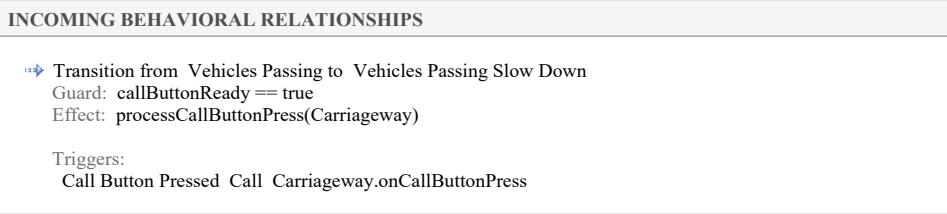
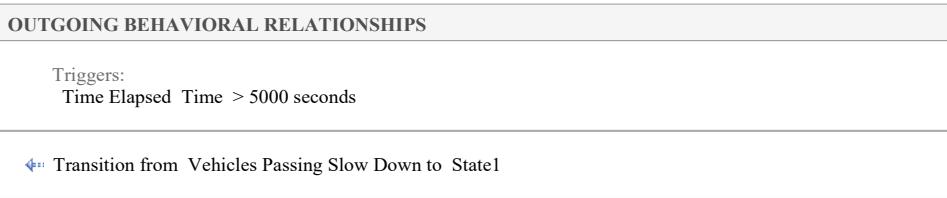
Vehicles Passing Slow Down

State owned by 'ControllerStateMachine', in package 'System Architecture'

Vehicles Passing Slow Down
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/5/2022

OUTGOING BEHAVIORAL RELATIONSHIPS

- ◆ Transition from Vehicles Passing Slow Down to Vehicles Passing Almost Done
Effect: setCarriagewayState(Carriageway, String)



Final

Final State owned by 'ControllerStateMachine', in package 'System Architecture'



Initial

Initial State owned by 'ControllerStateMachine', in package 'System Architecture'



Call Button Pressed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Call Button Pressed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

System Initialization

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

System Initialization
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Time Elapsed

Trigger owned by 'ControllerStateMachine', in package 'System Architecture'

Time Elapsed
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

High Level Requirements

Package in package 'Traffic Light System Model'

High Level Requirements
Version 1.0 Phase 1.0 Proposed
chioma created on 2/19/2022. Last modified 5/4/2022

Functional Requirement

Package in package 'High Level Requirements'

Functional Requirement
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Functional Requirement Hierarchy diagram

Requirements diagram in package 'Functional Requirement'

Functional Requirement Hierarchy
Version 1.0
chioma created on 2/19/2022. Last modified 5/4/2022

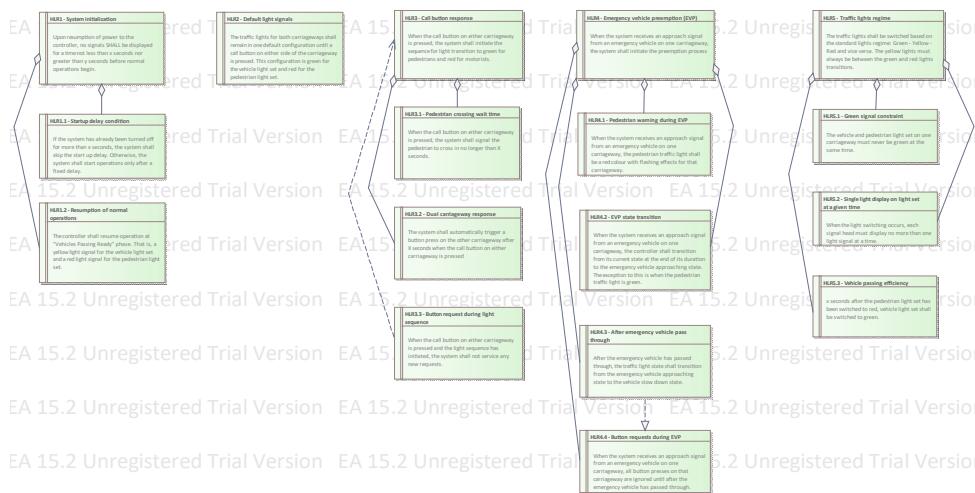


Figure 8: Functional Requirement Hierarchy

HLR1 - System initialization

Requirement «Functional» in package 'Functional Requirement'

Upon resumption of power to the controller, no signals SHALL be displayed for a time not less than x seconds nor greater than y seconds before normal operations begin.

HLR1 - System initialization
Version 1.0 Phase 1.0 Proposed
chioma created on 2/19/2022. Last modified 5/21/2022

ELEMENTS OWNED BY HLR1 - System initialization

■ HLR1.1 - Startup delay condition : Requirement «Functional»

If the system has already been turned off for more than x seconds, the system shall skip the start up delay. Otherwise, the system shall start operations only after a fixed delay.

■ HLR1.2 - Resumption of normal operations : Requirement «Functional»

The controller shall resume operation at "Vehicles Passing Ready" phase. That is, a yellow light signal for the vehicle light set and a red light signal for the pedestrian light set.

INCOMING STRUCTURAL RELATIONSHIPS

→ Aggregation from «Functional» HLR1.1 - Startup delay condition to «Functional» HLR1 - System initialization
[Direction is 'Source -> Destination'.]

→ Aggregation from «Functional» HLR1.2 - Resumption of normal operations to «Functional» HLR1 - System initialization
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Abstraction «Refine» Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: HLR1 - System initialization : Requirement, Public

↗ Refine «refine» Source -> Destination
From: LLR1 - Startup sequence : Requirement, Public
To: HLR1 - System initialization : Requirement, Public

HLR1.1 - Startup delay condition

Requirement «Functional» owned by 'HLR1 - System initialization', in package 'Functional Requirement'

If the system has already been turned off for more than x seconds, the system shall skip the start up delay. Otherwise, the system shall start operations only after a fixed delay.

HLR1.1 - Startup delay condition
Version 1.0 Phase 1.0 Proposed
chioma created on 2/19/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from «Functional» HLR1.1 - Startup delay condition to «Functional» HLR1 - System initialization	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.1 - Startup delay condition : Requirement, Public	
↗ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.1 - Startup delay condition : Requirement, Public	

HLR1.2 - Resumption of normal operations

Requirement «Functional» owned by 'HLR1 - System initialization', in package 'Functional Requirement'

The controller shall resume operation at "Vehicles Passing Ready" phase. That is, a yellow light signal for the vehicle light set and a red light signal for the pedestrian light set.

HLR1.2 - Resumption of normal operations
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS	
↳ Aggregation from «Functional» HLR1.2 - Resumption of normal operations to «Functional» HLR1 - System initialization	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Refine «refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.2 - Resumption of normal operations : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination From: LLR1 - Startup sequence : Requirement, Public To: HLR1.2 - Resumption of normal operations : Requirement, Public	

HLR2 - Default light signals

Requirement «Functional» in package 'Functional Requirement'

The traffic lights for both carriageways shall remain in one default configuration until a call button on either

side of the carriageway is pressed. This configuration is green for the vehicle light set and red for the pedestrian light set.

HLR2 - Default light signals
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

CONNECTORS
<p>↗ Refine «refine» Source -> Destination From: LLR3 - Process call button press : Requirement, Public To: HLR2 - Default light signals : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR4 - Vehicle passing duration complete : Requirement, Public To: HLR2 - Default light signals : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR4 - Vehicle passing duration complete : Requirement, Public To: HLR2 - Default light signals : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR3 - Process call button press : Requirement, Public To: HLR2 - Default light signals : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR2 - Default light signals : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR2 - Default light signals : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR2 - Default light signals : Requirement, Public</p>

HLR3 - Call button response

Requirement «Functional» in package 'Functional Requirement'

When the call button on either carriageway is pressed, the system shall initiate the sequence for light transition to green for pedestrians and red for motorists.

HLR3 - Call button response
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

ELEMENTS OWNED BY HLR3 - Call button response

ELEMENTS OWNED BY HLR3 - Call button response	
HLR3.1 - Pedestrian crossing wait time : Requirement «Functional»	When the call button on either carriageway is pressed, the system shall signal the pedestrian to cross in no longer than X seconds.
HLR3.2 - Dual carriageway response : Requirement «Functional»	The system shall automatically trigger a button press on the other carriageway after X seconds when the call button on either carriageway is pressed
HLR3.3 - Button request during light sequence : Requirement «Functional»	When the call button on either carriageway is pressed and the light sequence has initiated, the system shall not service any new requests.

INCOMING STRUCTURAL RELATIONSHIPS	
→ Aggregation from «Functional» HLR3.1 - Pedestrian crossing wait time to «Functional» HLR3 - Call button response	[Direction is 'Source -> Destination'.]
→ Aggregation from «Functional» HLR3.2 - Dual carriageway response to «Functional» HLR3 - Call button response	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Refine «refine» Source -> Destination	
From: LLR4 - Vehicle passing duration complete : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	
↗ Refine «refine» Source -> Destination	
From: LLR7 - Handle call button press : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	
↗ Dependency Source -> Destination	
From: HLR3.3 - Button request during light sequence : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination	
From: LLR4 - Vehicle passing duration complete : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination	
From: LLR7 - Handle call button press : Requirement, Public	
To: HLR3 - Call button response : Requirement, Public	

HLR3.1 - Pedestrian crossing wait time

Requirement «Functional» owned by 'HLR3 - Call button response', in package 'Functional Requirement'

When the call button on either carriageway is pressed, the system shall signal the pedestrian to cross in no longer than X seconds.

HLR3.1 - Pedestrian crossing wait time
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR3.1 - Pedestrian crossing wait time to «Functional» HLR3 - Call button response

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination

From: LLR7 - Handle call button press : Requirement, Public
To: HLR3.1 - Pedestrian crossing wait time : Requirement, Public

HLR3.2 - Dual carriageway response

Requirement «Functional» owned by 'HLR3 - Call button response', in package 'Functional Requirement'

The system shall automatically trigger a button press on the other carriageway after X seconds when the call button on either carriageway is pressed

HLR3.2 - Dual carriageway response
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR3.2 - Dual carriageway response to «Functional» HLR3 - Call button response

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination

From: LLR3 - Process call button press : Requirement, Public
To: HLR3.2 - Dual carriageway response : Requirement, Public

↗ Abstraction «Refine» Source -> Destination

From: LLR3 - Process call button press : Requirement, Public
To: HLR3.2 - Dual carriageway response : Requirement, Public

HLR3.3 - Button request during light sequence

Requirement «Functional» owned by 'HLR3 - Call button response', in package 'Functional Requirement'

When the call button on either carriageway is pressed and the light sequence has initiated, the system shall not service any new requests.

HLR3.3 - Button request during light sequence
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

CONNECTORS	
 Dependency	Source -> Destination From: HLR3.3 - Button request during light sequence : Requirement, Public To: HLR3 - Call button response : Requirement, Public
 Refine «refine»	Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public
 Abstraction «Refine»	Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR3.3 - Button request during light sequence : Requirement, Public

HLR4 - Emergency vehicle preemption (EVP)

Requirement «Functional» in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, the system shall initiate the preemption process

HLR4 - Emergency vehicle preemption (EVP)
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

ELEMENTS OWNED BY HLR4 - Emergency vehicle preemption (EVP)	
 HLR4.1 - Pedestrian warning during EVP : Requirement «Functional»	When the system receives an approach signal from an emergency vehicle on one carriageway, the pedestrian traffic light shall be a red colour with flashing effects for that carriageway.
 HLR4.2 - EVP state transition : Requirement «Functional»	When the system receives an approach signal from an emergency vehicle on one carriageway, the controller shall transition from its current state at the end of its duration to the emergency vehicle approaching state. The exception to this is when the pedestrian traffic light is green.

ELEMENTS OWNED BY HLR4 - Emergency vehicle preemption (EVP)	
■ HLR4.3 - After emergency vehicle pass through : Requirement «Functional»	After the emergency vehicle has passed through, the traffic light state shall transition from the emergency vehicle approaching state to the vehicle slow down state.
■ HLR4.4 - Button requests during EVP : Requirement «Functional»	When the system receives an approach signal from an emergency vehicle on one carriageway, all button presses on that carriageway are ignored until after the emergency vehicle has passed through.
INCOMING STRUCTURAL RELATIONSHIPS	
→ Aggregation from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]
→ Aggregation from «Functional» HLR4.4 - Button requests during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]
→ Aggregation from «Functional» HLR4.2 - EVP state transition to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]
→ Aggregation from «Functional» HLR4.1 - Pedestrian warning during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP)	[Direction is 'Source -> Destination'.]

CONNECTORS	
↗ Refine «refine» Source -> Destination	
From: LLR5 - Handle emergency vehicle signal received : Requirement, Public	
To: HLR4 - Emergency vehicle preemption (EVP) : Requirement, Public	
↗ Abstraction «Refine» Source -> Destination	
From: LLR5 - Handle emergency vehicle signal received : Requirement, Public	
To: HLR4 - Emergency vehicle preemption (EVP) : Requirement, Public	

HLR4.1 - Pedestrian warning during EVP

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, the pedestrian traffic light shall be a red colour with flashing effects for that carriageway.

HLR4.1 - Pedestrian warning during EVP

OUTGOING STRUCTURAL RELATIONSHIPS
↳ Aggregation from «Functional» HLR4.1 - Pedestrian warning during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP) [Direction is 'Source -> Destination'.]

CONNECTORS
↗ Refine «refine» Source -> Destination From: LLR6 - Display lights : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public
↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public
↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public
↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.1 - Pedestrian warning during EVP : Requirement, Public

HLR4.2 - EVP state transition

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, the controller shall transition from its current state at the end of its duration to the emergency vehicle approaching state. The exception to this is when the pedestrian traffic light is green.

OUTGOING STRUCTURAL RELATIONSHIPS
↳ Aggregation from «Functional» HLR4.2 - EVP state transition to «Functional» HLR4 - Emergency vehicle preemption (EVP) [Direction is 'Source -> Destination'.]

CONNECTORS

CONNECTORS
<p>↗ Refine «refine» Source -> Destination From: LLR5 - Handle emergency vehicle signal received : Requirement, Public To: HLR4.2 - EVP state transition : Requirement, Public</p>
<p>↗ Refine «refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.2 - EVP state transition : Requirement, Public</p>

HLR4.3 - After emergency vehicle pass through

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

After the emergency vehicle has passed through, the traffic light state shall transition from the emergency vehicle approaching state to the vehicle slow down state.

HLR4.3 - After emergency vehicle pass through
 Version 1.0 Phase 1.0 Proposed
 chioma created on 2/20/2022. Last modified 5/22/2022

EXTERNAL REQUIREMENTS
<p><input checked="" type="checkbox"/> Requirement. HLR4.4 - Button requests during EVP</p> <p>When the system receives an approach signal from an emergency vehicle on one carriageway, all button presses on that carriageway are ignored until after the emergency vehicle has passed through. [Stereotype is «Functional».]</p>

OUTGOING STRUCTURAL RELATIONSHIPS
<p>↳ Realization from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4.4 - Button requests during EVP [Direction is 'Source -> Destination'.]</p>
<p>↳ Aggregation from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4 - Emergency vehicle preemption (EVP) [Direction is 'Source -> Destination'.]</p>

CONNECTORS
<p>↗ Abstraction «Refine» Source -> Destination From: LLR2 - Set carriageway state : Requirement, Public To: HLR4.3 - After emergency vehicle pass through : Requirement, Public</p>
<p>↗ Abstraction «Refine» Source -> Destination From: ControllerStateMachine : StateMachine, Public To: HLR4.3 - After emergency vehicle pass through : Requirement, Public</p>

CONNECTORS**↗ Refine «refine» Source -> Destination**

From: LLR2 - Set carriageway state : Requirement, Public
 To: HLR4.3 - After emergency vehicle pass through : Requirement, Public

HLR4.4 - Button requests during EVP

Requirement «Functional» owned by 'HLR4 - Emergency vehicle preemption (EVP)', in package 'Functional Requirement'

When the system receives an approach signal from an emergency vehicle on one carriageway, all button presses on that carriageway are ignored until after the emergency vehicle has passed through.

HLR4.4 - Button requests during EVP

Version 1.0 Phase 1.0 Proposed

chioma created on 2/20/2022. Last modified 5/22/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↖ Aggregation from «Functional» HLR4.4 - Button requests during EVP to «Functional» HLR4 - Emergency vehicle preemption (EVP)

[Direction is 'Source -> Destination'.]

INCOMING STRUCTURAL RELATIONSHIPS

↗ Realization from «Functional» HLR4.3 - After emergency vehicle pass through to «Functional» HLR4.4 - Button requests during EVP

[Direction is 'Source -> Destination'.]

CONNECTORS**↗ Refine «refine» Source -> Destination**

From: LLR5 - Handle emergency vehicle signal received : Requirement, Public
 To: HLR4.4 - Button requests during EVP : Requirement, Public

↗ Refine «refine» Source -> Destination

From: LLR2 - Set carriageway state : Requirement, Public
 To: HLR4.4 - Button requests during EVP : Requirement, Public

↗ Abstraction «Refine» Source -> Destination

From: LLR2 - Set carriageway state : Requirement, Public
 To: HLR4.4 - Button requests during EVP : Requirement, Public

HLR5 - Traffic lights regime

Requirement «Functional» in package 'Functional Requirement'

The traffic lights shall be switched based on the standard lights regime: Green - Yellow - Red and vice versa.
The yellow lights must always be between the green and red lights transitions.

HLR5 - Traffic lights regime
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

ELEMENTS OWNED BY HLR5 - Traffic lights regime	
 HLR5.1 - Green signal constraint : Requirement «Functional»	The vehicle and pedestrian light set on one carriageway must never be green at the same time.
 HLR5.2 - Single light display on light set at a given time : Requirement «Functional»	When the light switching occurs, each signal head must display no more than one light signal at a time.
 HLR5.3 - Vehicle passing efficiency : Requirement «Functional»	x seconds after the pedestrian light set has been switched to red, vehicle light set shall be switched to green.

INCOMING STRUCTURAL RELATIONSHIPS	
 Aggregation from «Functional» HLR5.2 - Single light display on light set at a given time to «Functional» HLR5 - Traffic lights regime	[Direction is 'Source > Destination'.]
 Aggregation from «Functional» HLR5.3 - Vehicle passing efficiency to «Functional» HLR5 - Traffic lights regime	[Direction is 'Source > Destination'.]
 Aggregation from «Functional» HLR5.1 - Green signal constraint to «Functional» HLR5 - Traffic lights regime	[Direction is 'Source > Destination'.]

CONNECTORS	
 Refine «refine» Source -> Destination	
From: LLR2 - Set carriageway state : Requirement, Public	
To: HLR5 - Traffic lights regime : Requirement, Public	
 Abstraction «Refine» Source -> Destination	
From: ControllerStateMachine : StateMachine, Public	
To: HLR5 - Traffic lights regime : Requirement, Public	

HLR5.1 - Green signal constraint

Requirement «Functional» owned by 'HLR5 - Traffic lights regime', in package 'Functional Requirement'

The vehicle and pedestrian light set on one carriageway must never be green at the same time.

HLR5.1 - Green signal constraint
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR5.1 - Green signal constraint to «Functional» HLR5 - Traffic lights regime
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Abstraction «Refine» Source -> Destination
From: ControllerStateMachine : StateMachine, Public
To: HLR5.1 - Green signal constraint : Requirement, Public

↗ Refine «refine» Source -> Destination
From: LLR2 - Set carriageway state : Requirement, Public
To: HLR5.1 - Green signal constraint : Requirement, Public

HLR5.2 - Single light display on light set at a given time

Requirement «Functional» owned by 'HLR5 - Traffic lights regime', in package 'Functional Requirement'

When the light switching occurs, each signal head must display no more than one light signal at a time.

HLR5.2 - Single light display on light set at a given time
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR5.2 - Single light display on light set at a given time to «Functional» HLR5 - Traffic lights regime
[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Refine «refine» Source -> Destination
From: LLR6 - Display lights : Requirement, Public
To: HLR5.2 - Single light display on light set at a given time : Requirement, Public

↗ Abstraction «Refine» Source -> Destination
From: ControllerStateMachine : StateMachine, Public
To: HLR5.2 - Single light display on light set at a given time : Requirement, Public

HLR5.3 - Vehicle passing efficiency

Requirement «Functional» owned by 'HLR5 - Traffic lights regime', in package 'Functional Requirement'

x seconds after the pedestrian light set has been switched to red, vehicle light set shall be switched to green.

HLR5.3 - Vehicle passing efficiency
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

OUTGOING STRUCTURAL RELATIONSHIPS

↳ Aggregation from «Functional» HLR5.3 - Vehicle passing efficiency to «Functional» HLR5 - Traffic lights regime

[Direction is 'Source -> Destination'.]

CONNECTORS

↗ Abstraction «Refine» Source -> Destination

From: ControllerStateMachine : StateMachine, Public

To: HLR5.3 - Vehicle passing efficiency : Requirement, Public

↗ Refine «refine» Source -> Destination

From: LLR2 - Set carriageway state : Requirement, Public

To: HLR5.3 - Vehicle passing efficiency : Requirement, Public

Non-Functional Requirement

Package in package 'High Level Requirements'

Non-Functional Requirement
Version 1.0 Phase 1.0 Proposed
chioma created on 5/4/2022. Last modified 5/4/2022

Non-Functional Requirement Hierarchy diagram

Requirements diagram in package 'Non-Functional Requirement'

Non-Functional Requirement Hierarchy
Version 1.0
chioma created on 2/20/2022. Last modified 5/23/2022

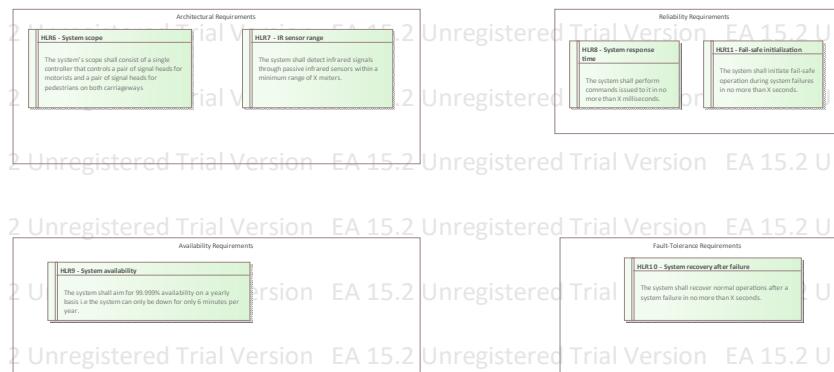


Figure 9: Non-Functional Requirement Hierarchy

Architectural Requirements

Boundary in package 'Non-Functional Requirement'

Architectural Requirements
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 2/20/2022
Extends

Availability Requirements

Boundary in package 'Non-Functional Requirement'

Availability Requirements
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 2/20/2022
Extends

Fault-Tolerance Requirements

Boundary in package 'Non-Functional Requirement'

Fault-Tolerance Requirements
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 2/20/2022
Extends

Reliability Requirements

Boundary in package 'Non-Functional Requirement'

Reliability Requirements
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 2/20/2022
Extends

HLR10 - System recovery after failure

NonfunctionalRequirement «NonfunctionalRequirement» in package 'Non-Functional Requirement'

The system shall recover normal operations after a system failure in no more than X seconds.

HLR10 - System recovery after failure
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

HLR11 - Fail-safe initialization

NonfunctionalRequirement «NonfunctionalRequirement» in package 'Non-Functional Requirement'

The system shall initiate fail-safe operation during system failures in no more than X seconds.

HLR11 - Fail-safe initialization
Version 1.0 Phase 1.0 Proposed
chioma created on 2/20/2022. Last modified 5/13/2022

HLR6 - System scope

ArchitecturalRequirement «ArchitecturalRequirement» in package 'Non-Functional Requirement'

The system's scope shall consist of a single controller that controls a pair of signal heads for motorists and a pair of signal heads for pedestrians on both carriageways

HLR6 - System scope
Version 1.0 Phase 1.0 Proposed

CONNECTORS **Abstraction** «Refine» Source -> DestinationFrom: TrafficLightController : Class, Public
To: HLR6 - System scope : ArchitecturalRequirement, Public

HLR7 - IR sensor range

ArchitecturalRequirement «ArchitecturalRequirement» in package 'Non-Functional Requirement'

The system shall detect infrared signals through passive infrared sensors within a minimum range of X meters.

HLR7 - IR sensor range

Version 1.0 Phase 1.0 Proposed

chioma created on 2/20/2022. Last modified 5/13/2022

CONNECTORS **Abstraction** «Refine» Source -> DestinationFrom: Carriageway : Class, Public
To: HLR7 - IR sensor range : ArchitecturalRequirement, Public

HLR8 - System response time

NonfunctionalRequirement «NonfunctionalRequirement» in package 'Non-Functional Requirement'

The system shall perform commands issued to it in no more than X milliseconds.

HLR8 - System response time

Version 1.0 Phase 1.0 Proposed

chioma created on 2/20/2022. Last modified 5/13/2022

HLR9 - System availability

NonfunctionalRequirement «NonfunctionalRequirement» in package 'Non-Functional Requirement'

The system shall aim for 99.999% availability on a yearly basis i.e the system can only be down for only 6 minutes per year.

HLR9 - System availability

Version 1.0 Phase 1.0 Proposed

chioma created on 2/20/2022. Last modified 5/13/2022

