GROUP 09
MOK Zhi Kin

MyProjectName : HighwayToSafety
Messir Analysis Document
- v 1.0 -

(*Report type: Default*)

MOK Zhi Kin

Friday 16th December, 2016 - 22:45

Contents

List of Figures

Listings

Chapter 1

Introduction

1.1 Overview

HighwayToSafety is a simple system intended to help users organising and managing the aftermath of an highway car crash as effectively as possible. It assures a quick communication between the workforce, who are on the site of the accident and their colleagues who work at the emergency centrals.

1.2 Purpose and recipients of the document

This document is an analysis document complying with the MESSIR methodology. Its intent is to provide a first simple example of a precise specification of the functional properties of the HTS system.

The recipients of this document are:

- the HTS system's buyer company (ABC): this document is used as a contractual document jointly with any other document considered as useful (as requirement elicitation document, ...) in order to have a higher degree of precision in requirement description. It is also used as a basis document for the HTS system validation using specification based testing.

- the HTS system development company (ADC) will use this document as the basis for development (mainly design, implementation, maintenance). It is also used for verification and validation using test plans defined using the analysis models described in this document and according to the HTS methodology.

1.3 Application Domain

The HTS system belongs to the Crisis Management Systems Domain. It is a system dedicated to only crisis professional end users.

It is not an institutional system certified and guaranteed by any governmental entity and thus must be used with caution.

1.4 Definitions, acronyms and abbreviations

HTS - HighwayToSafety

1.5 Document structure

The document structure is designed to be coherent with the Messir methodology [1]. Section 2 provides a general description of the system purpose, its users, its environment and some general non functional requirements. A more detailed description of the non functional requirements related to dependability, if any, are provided in section 7. The system operations triggered by input events coming from the external actors belonging to the environment are described in Section 3. The HTS concepts used to represent the any persistent or transient information is given in Section 4. The precise specification of the system operations in term of system's state changes, events sent together with the constraints on the allowed sequences of system operations are described in Section 5.

Chapter 2

General Description

2.1 Domain Stakeholders

All stakeholders of the system are detailed in this section. After a brief description of a stakeholder, its objectives are first stated. Thereafter, the responsibilities of the stakeholder are detailed which help to achieve the stakeholder objectives to a certain degree. While the objectives characterize the general problems addressed by the *iCrashMini* system, the responsibilities describe concrete actions that are expected from a stakeholder. Some of these responsibilities can be traced looking at the use case described in Section 2.3, and hence must be supported by the *iCrashMini* system. All stakeholders listed in this section have an interest in the system or are affected by the system in some way, but only a subset of the stakeholders are directly involved in the use cases described. Let us remind that use case diagrams or descriptions are not Messir analysis phase mandatory outputs. They are proposed as informal means to help understanding the semantics of the system specification made of the mandatory analysis models, which provide a complete executable specification.

2.2 System's Actors

2.2.1 Emergency Central

Emergency Central Administrator Most likely a chief or someone with high autorities in the Emergency Central. His job is to assign fellow colleagues as coordinators.

Emergency Central Coordinator An Emergency Central Coordinator is an employee who has been given special rights by the administrator. His job is:

1. To check the incident's location with the notifier.
2. To create the new event in the system.
3. To send a dispatch message to the corresponding coordinators.

2.2.2 Firemen

General idea : The most important dispatched coordinators, because they are the ones sitting in the ambulances and providing first aids to the injured individuals.

Firemen Administrator Most likely a chief or someone with high autorities in the Firemen Department. His job is to assign fellow colleagues as coordinators.

Firemen Coordinator A Firemen Coordinator is a Team Leader who has been given special rights by the administrator. His job is:

1. To assemble his team as soon as possible when a dispatch request is received.
2. To provide information on his Team's whereabouts.
3. To call for backup if needed when on-site.
4. To send messages if needed.

2.2.3 Tow Service

General idea : Dispatched coordinators responsible of clearing the highway of the destroyed cars.

Tow Service Administrator Most likely a chief or someone with high autorities in the Tow Service. His job is to assign fellow colleagues as coordinators.

Tow Service Coordinator A Tow Service Coordinator is a Team Leader who has been given special rights by the administrator. His job is:

1. To assemble his team as soon as possible when a dispatch request is received.
2. To provide information on his Team's whereabouts.
3. To send messages if needed.

2.2.4 Police

General idea : General idea : Dispatched coordinators responsible of solving eventual conflicts between the victims.

Police Administrator Most likely a chief or someone with high autorities in the Police Department. His job is to assign fellow colleagues as coordinators.

Police Coordinator A Police Coordinator is a Team Leader who has been given special rights by the administrator. His job is:

1. To assemble his team as soon as possible when a dispatch request is received.
2. To provide information on his Team's whereabouts.
3. To send messages if needed.

2.2.5 Communication Company

A Communication Company is a company who provides fixed and mobile voice services like POST, TANGO, ORANGE, etc. Its job is to transfer any emergency calls to the Emergency Central and to provide the notifier's geographical postion via its network to the system.

2.3 Use Cases Model

This section contains the use cases elicited during the requirements elicitation phase. The use cases are textually described as suggested by the Messir method and inspired by the standard Cokburn template [2].

2.3.1 Use Cases

2.3.1.1 summary-suGlobalManagementOfEvent

The goal is to manage the creation of a new crisis event including all the necessary information and to have the requested coordinators arrive on the crisis event's location.

Use-Case Description	
Name	suGlobalManagementOfEvent
Scope	system
Level	summary
Primary actor(s)	
1	`actCentralCoordinator[active]`
Secondary actor(s)	
1	`actCommunicationCompany[active]`
2	`actFiremenCoordinator[active]`
3	`actTowServiceCoordinator[active]`
Goal(s) description	
The goal is to manage the creation of a new crisis event including all the necessary information and to have the requested coordinators arrive on the crisis event's location.	
Reuse	

continues in next page . . .

... Use-Case Description table continuation

1	ugCreateNewCrisisEvent [1..*]
2	ugGlobalDispatchManagement [1..*]
Protocol condition(s)	
1	none.
Pre-condition(s)	
1	none.
Main post-condition(s)	
1	a new crisis event has been created and modifications have been made by the coordinators to the system and its environment concerning a crisis event.
Main Steps	
a	the actor actCentralCoordinator executes the ugCreateNewCrisisEvent use case
b	the actor actFiremenCoordinator executes the ugGlobalDispatchManagement use case
c	the actor actTowServiceCoordinator executes the ugGlobalDispatchManagement use case
Steps Ordering Constraints	
1	step (a) must be executed before step (b) or step (c)
2	step (b) XOR step (c)
Additional Information	
none	

Figure 2.1 Shows the suGlobaManagementOfEvent use-case and its actors.

2.3.1.2 usergoal-ugCreateNewCrisisEvent

The goal is to manage the creation of a new crisis event including all the necessary information.

Use-Case Description	
Name	ugCreateNewCrisisEvent
Scope	system
Level	usergoal
Primary actor(s)	
1	actCentralCoordinator[active]
Secondary actor(s)	
1	actCommunicationCompany[active]
2	actFiremenCoordinator[passive]
3	actTowServiceCoordinator[passive]
Goal(s) description	
The goal is to manage the creation of a new crisis event including all the necessary information.	
Reuse	
1	oeAddNewCrisisEvent [1..*]
2	oeRequestCrisisEventLocation [1..*]
3	oeReceiveCrisisEventLocation [1..*]
4	oeCreateNewCrisisEvent [1..*]
5	oeMovePinOnMap [0..*]
Protocol condition(s)	

continues in next page ...

... Use-Case Description table continuation

1	none.
Pre-condition(s)	
1	none.
Main post-condition(s)	
1	a dispatch order including the crisis event's information such as the id, map with pin, witness's phone number, etc. is sent to nearest, free Firemen Team and Tow Service Team.
Main Steps	
a	the actor `actCentralCoordinator` executes the `oeAddNewCrisisEvent` use case
b	the actor `actCentralCoordinator` executes the `oeRequestCrisisEventLocation` use case
c	the actor `actCommunicationCompany` executes the `oeReceiveCrisisEventLocation` use case
d	the actor `actCentralCoordinator` executes the `oeMovePinOnMap` use case
e	the actor `actCentralCoordinator` executes the `oeCreateNewCrisisEvent` use case
Steps Ordering Constraints	
1	step (a) must be executed first
2	if step (c) then previously step (b)
3	if step (d) then previously step (c)
4	step (e) executed as last
5	step (a), (b), (c), (e) must be executed at least once
Additional Information	
none	

Figure 2.2 Shows the ugCreateNewCrisisEvent use-case and its actors.

2.3.1.3 usergoal-ugGlobalDispatchManagement

The goal is to have the requested coordinators arrive on the crisis event's location.

USE-CASE DESCRIPTION	
Name	ugGlobalDispatchManagement
Scope	system
Level	usergoal
Primary actor(s)	
1	`actFiremenCoordinator[active]`
2	`actTowServiceCoordinator[active]`
Secondary actor(s)	
1	`actCentralCoordinator[passive]`
2	`actPoliceCoordinator[active]`
Goal(s) description	
The goal is to have the requested coordinators arrive on the crisis event's location.	
Reuse	
1	`oeGetCrisisEventInformation [2..*]`
2	`oeUpdateDispatchStatus [4..*]`
3	`oeRefreshMap [0..*]`
4	`oeMessage [0..*]`

continues in next page ...

... Use-Case Description table continuation

5	oeRequestHelp [0..*]
6	oeCloseCrisisEvent [2..*]
7	oeAddRequestHelp [0..*]
Protocol condition(s)	
1	none.
Pre-condition(s)	
1	the sender is associated to a crisis event.
Main post-condition(s)	
1	modifications have been made to the system and its environment concerning a crisis event.
Main Steps	
a	the actor actFiremenCoordinator executes the oeGetCrisisEventInformation use case
b	the actor actFiremenCoordinator executes the oeUpdateDispatchStatus use case
c	the actor actTowServiceCoordinator executes the oeGetCrisisEventInformation use case
d	the actor actTowServiceCoordinator executes the oeUpdateDispatchStatus use case
e	the actor actTowServiceCoordinator executes the oeRefreshMap use case
f	the actor actTowServiceCoordinator executes the oeMessage use case
g	the actor actFiremenCoordinator executes the oeAddRequestHelp use case
h	the actor actFiremenCoordinator executes the oeRequestHelp use case
i	the actor actPoliceCoordinator executes the oeGetCrisisEventInformation use case
j	the actor actPoliceCoordinator executes the oeUpdateDispatchStatus use case
k	the actor actFiremenCoordinator executes the oeCloseCrisisEvent use case
l	the actor actTowServiceCoordinator executes the oeCloseCrisisEvent use case
m	the actor actPoliceCoordinator executes the oeCloseCrisisEvent use case
Steps Ordering Constraints	
1	if step (b),(d),(j) then previously step (a),(c),(i) respectively
2	if step (k),(l),(m) then previously step (b),(d),(j) at least two times respectively
3	step (h) can only be executed if step (g) has at least been executed once previously
4	if step (i) then previously step (h)
Additional Information	
none	

Figure 2.3 Shows the ugGlobalDispatchManagement use-case and its actors.

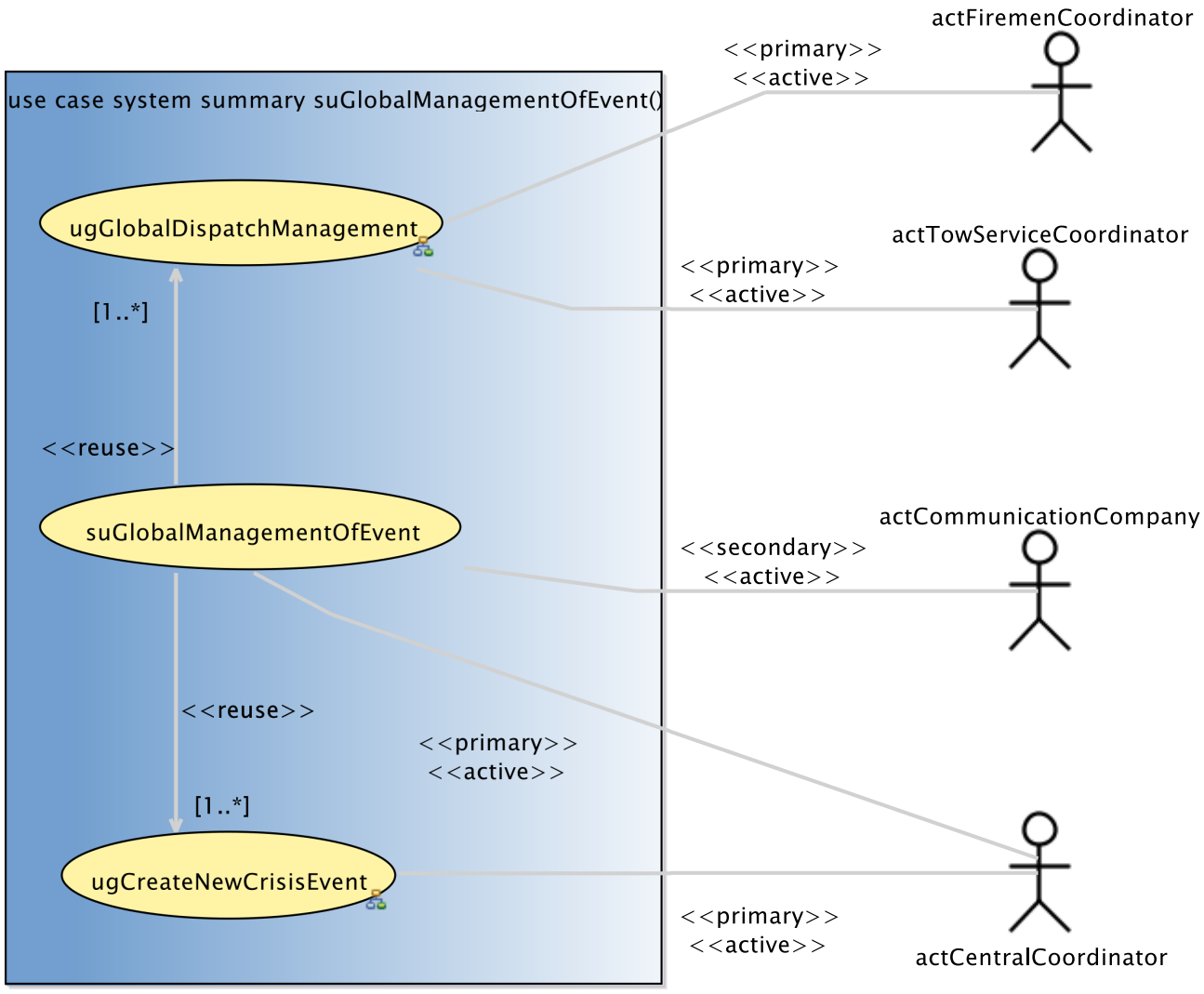

use case system summary suGlobalManagementOfEvent()
ugGlobalDispatchManagement
suGlobalManagementOfEvent
ugCreateNewCrisisEvent
[1..*]
<<reuse>>
[1..*]
<<reuse>>
<<primary>>
<<active>>
actFiremenCoordinator
<<primary>>
<<active>>
actTowServiceCoordinator
<<primary>>
<<active>>
actCommunicationCompany
<<secondary>>
<<active>>
actCentralCoordinator
<<primary>>
<<active>>

Figure 2.1:

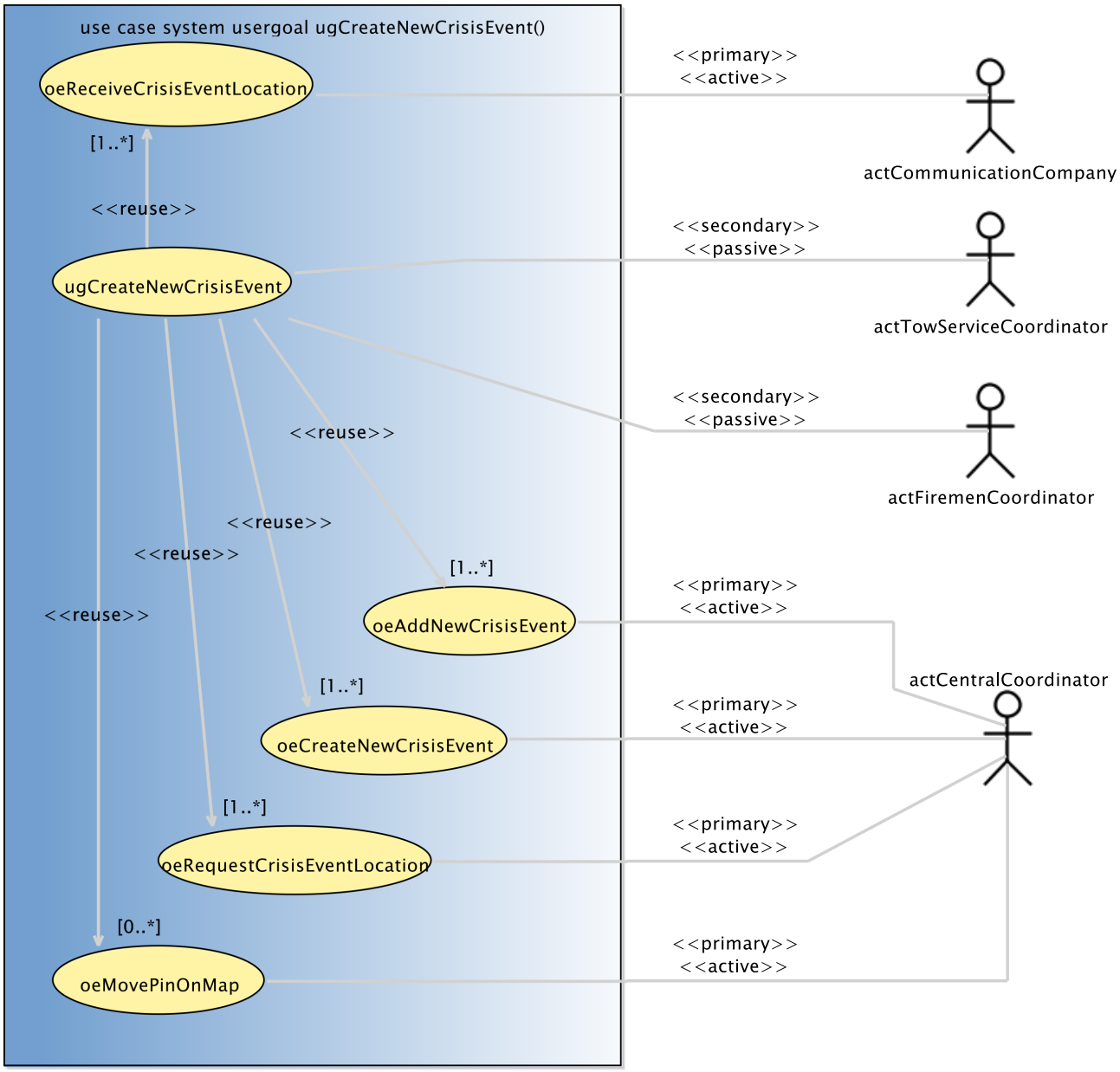


Figure 2.2: ugCreateNewCrisisEvent

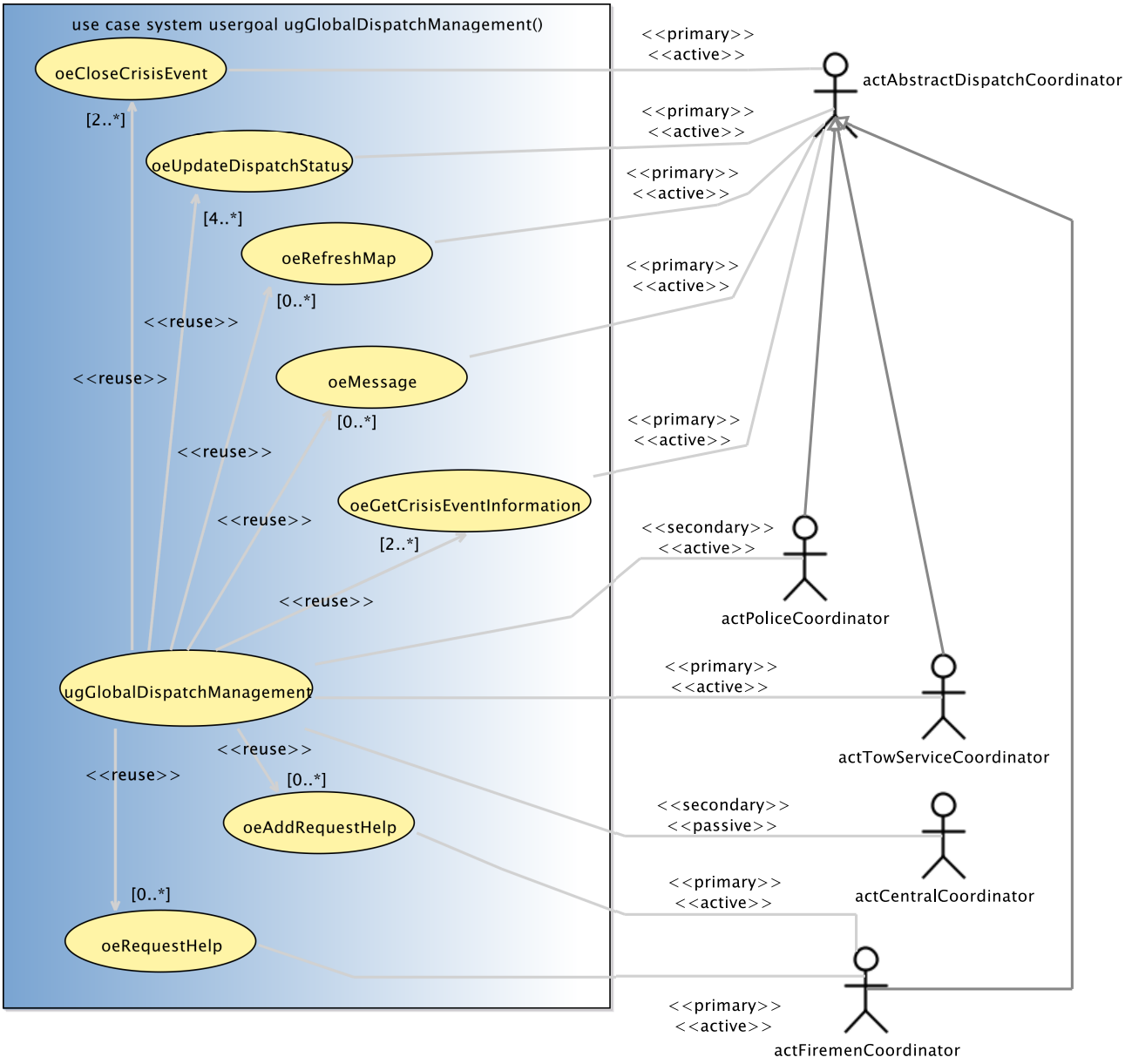


Figure 2.3: ugGlobalDispatchManagement

2.3.2 Use Case Instance(s)

2.3.2.1 Use-Case Instance - uciugCreateNewCrisiEvent:ugCreateNewCrisisEvent

Shows the ugCreateNewCrisisEvent instance.

USERGOAL USE-CASE INSTANCE
Instantiated Use Case ugCreateNewCrisisEvent
Instance ID uciugCreateNewCrisiEvent

Figure 2.4 Shows the ugCreateNewCrisisEvent instance.

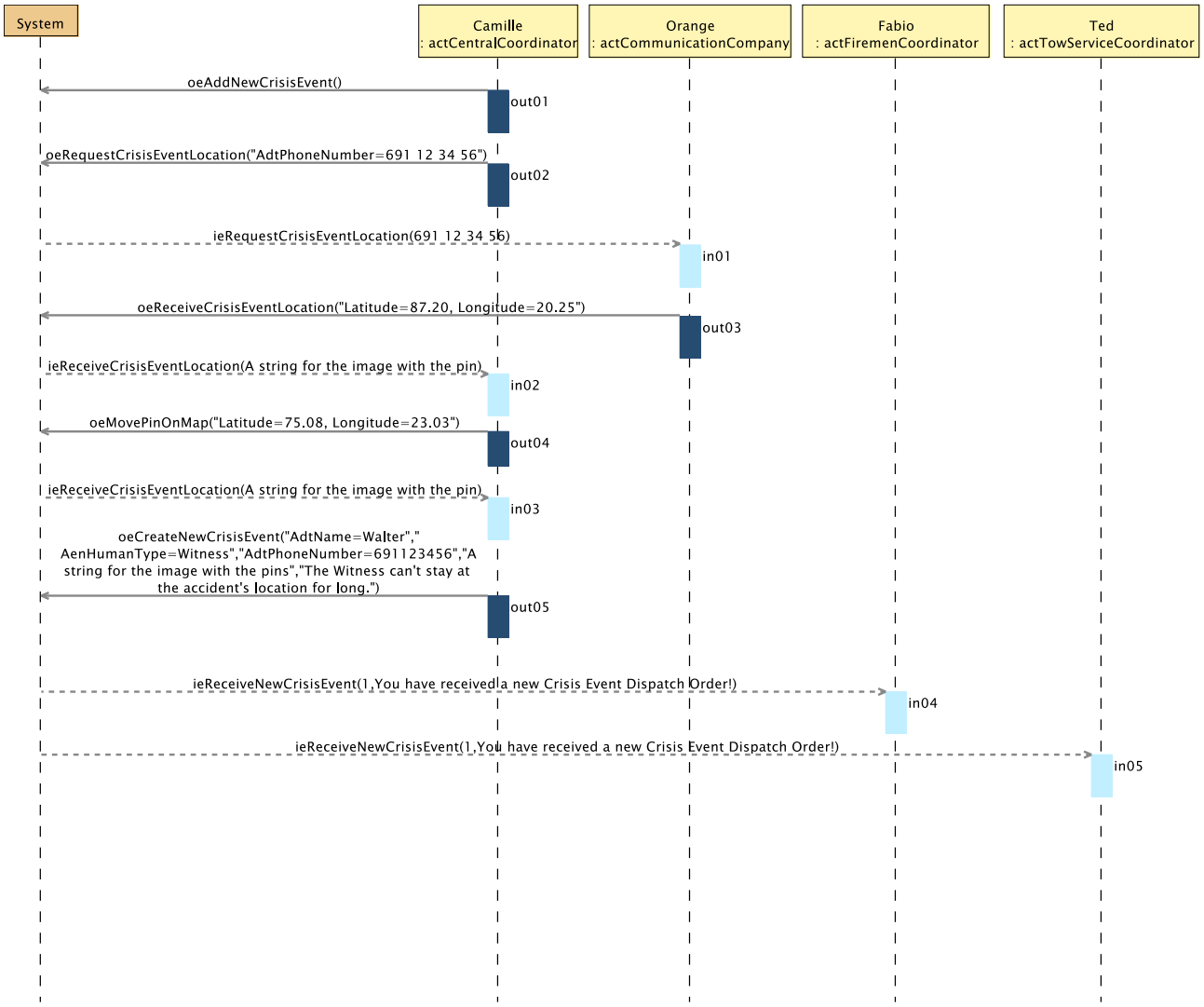


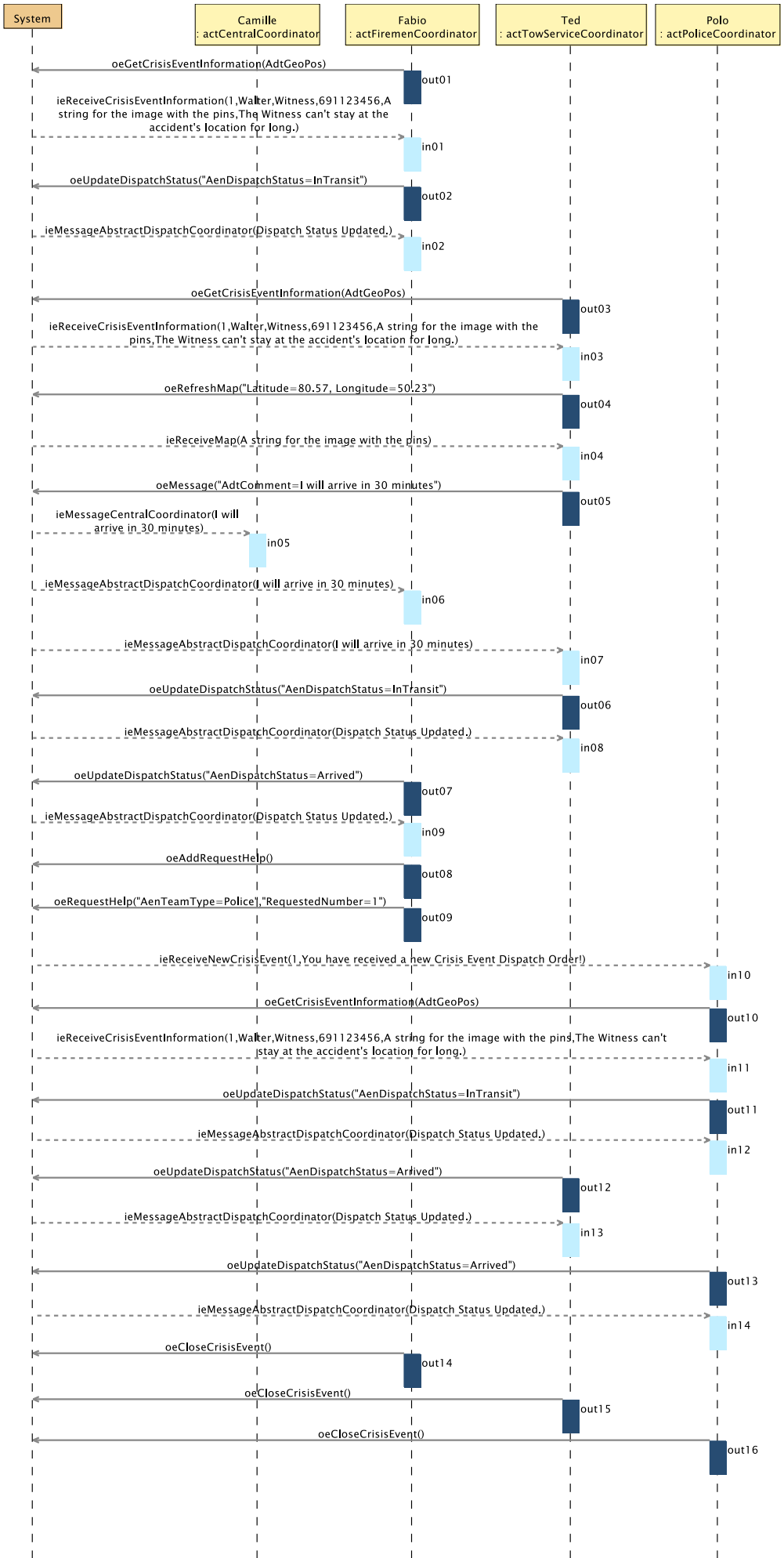
Figure 2.4: ugCreateNewCrisisEvent

2.3.2.2 Use-Case Instance - uciugGlobalDispatchManagement:ugGlobalDispatchManagement

Shows the ugGlobalDispatchManagement instance.

USERGOAL USE-CASE INSTANCE
Instantiated Use Case ugGlobalDispatchManagement
Instance ID uciugGlobalDispatchManagement

Figure 2.5 Shows the ugGlobalDispatchManagement instance.


System
Camille
: actCentralCoordinator
Fabio
: actFiremenCoordinator
Ted
: actTowServiceCoordinator
Polo
: actPoliceCoordinator
oeGetCrisisEventInformation(AdtGeoPos)
out01
ieReceiveCrisisEventInformation(1,Walter,Witness,691123456,A string for the image with the pins,The Witness can't stay at the accident's location for long.)
in01
oeUpdateDispatchStatus("AenDispatchStatus=InTransit")
out02
ieMessageAbstractDispatchCoordinator(Dispatch Status Updated.)
in02
oeGetCrisisEventInformation(AdtGeoPos)
out03
ieReceiveCrisisEventInformation(1,Walter,Witness,691123456,A string for the image with the pins,The Witness can't stay at the accident's location for long.)
in03
oeRefreshMap("Latitude=80.57, Longitude=50.23")
out04
ieReceiveMap(A string for the image with the pins)
in04
oeMessage("AdtComment=I will arrive in 30 minutes")
out05
ieMessageCentralCoordinator(I will arrive in 30 minutes)
in05
ieMessageAbstractDispatchCoordinator(I will arrive in 30 minutes)
in06
ieMessageAbstractDispatchCoordinator(I will arrive in 30 minutes)
in07
oeUpdateDispatchStatus("AenDispatchStatus=InTransit")
out06
ieMessageAbstractDispatchCoordinator(Dispatch Status Updated.)
in08
oeUpdateDispatchStatus("AenDispatchStatus=Arrived")
out07
ieMessageAbstractDispatchCoordinator(Dispatch Status Updated.)
in09
oeAddRequestHelp()
out08
oeRequestHelp("AenTeamType=Police","RequestedNumber=1")
out09
ieReceiveNewCrisisEvent(1,You have received a new Crisis Event Dispatch Order!)
in10
oeGetCrisisEventInformation(AdtGeoPos)
out10
ieReceiveCrisisEventInformation(1,Walter,Witness,691123456,A string for the image with the pins,The Witness can't stay at the accident's location for long.)
in11
oeUpdateDispatchStatus("AenDispatchStatus=InTransit")
out11
ieMessageAbstractDispatchCoordinator(Dispatch Status Updated.)
in12
oeUpdateDispatchStatus("AenDispatchStatus=Arrived")
out12
ieMessageAbstractDispatchCoordinator(Dispatch Status Updated.)
in13
oeUpdateDispatchStatus("AenDispatchStatus=Arrived")
out13
ieMessageAbstractDispatchCoordinator(Dispatch Status Updated.)
in14
oeCloseCrisisEvent()
out14
oeCloseCrisisEvent()
out15
oeCloseCrisisEvent()
out16

Chapter 3

Environment Model

3.1 Environment model view(s)

There are no view(s) for the Messir environment model.

3.2 Actors and Interfaces Descriptions

We provide for the given views the description of the actors together with their associated input and output interface descriptions.

3.2.1 `actAbstractDispatchCoordinator` Actor

Actor	
actAbstractDispatchCoordinator An abstract Actor which brings together the common operations of the FiremenCoordinator, the PoliceCoordinator and the TowServiceCoordinator.	
OutputInterfaces	
OUT 1	**oeMessage(AdtComment:dtComment):ptBoolean**
OUT 2	**oeUpdateDispatchStatus(AetDispatchStatus:etDispatchStatus):ptBoolean**
InputInterfaces	
IN 1	**ieReceiveNewCrisisEvent(AdtCrisisID:dtCrisisID, AMessage:dtString):ptBoolean**
IN 2	**ieMessageAbstractDispatchCoordinator(AdtComment:dtComment):ptBoolean**

3.2.2 `actCentralCoordinator` Actor

Actor	
actCentralCoordinator Is representing the person that receives the victim's or witness' call in the emergency central.	
OutputInterfaces	
OUT 1	**oeRequestCrisisEventLocation(AdtPhoneNumber:dtPhoneNumber):ptBoolean**

continues in next page . . .

... Actor table continuation

OUT 2	**oeCreateNewCrisisEvent(AName:dtString, AetHumanType:etHumanType, AdtPhoneNumber:dtPhoneNumber, AdtMapWithPin:dtMapWithPin, AdtComment:dtComment):ptBoolean**
InputInterfaces	
IN 1	**ieReceiveCrisisEventLocation(AdtMapWithPin:dtMapWithPin):ptBoolean**
IN 2	**ieMessageCentralCoordinator(AdtComment:dtComment):ptBoolean**

3.2.3 `actCommunicationCompany` Actor

ACTOR	
actCommunicationCompany	
Is representing any communication company in Luxembourg.	
OutputInterfaces	
OUT 1	**oeReceiveCrisisEventLocation(AdtGeoPos:dtGeoPos):ptBoolean**
InputInterfaces	
IN 1	**ieRequestCrisisEventLocation(AdtPhoneNumber:dtPhoneNumber):ptBoolean**

3.2.4 `actFiremenCoordinator` Actor

ACTOR	
actFiremenCoordinator	
Is representing any firemen team leader able to manage a two Ambulances.	
Extends	
lu.uni.lassy.excalibur.group09.spec.environment.actAbstractDispatchCoordinator	
OutputInterfaces	
OUT 1	**oeRequestHelp(AetTeamType:etTeamType, ARequestedNumber:ptInteger):ptBoo**

3.2.5 `actPoliceCoordinator` Actor

ACTOR	
actPoliceCoordinator	
Is representing a police team leader.	
Extends	
lu.uni.lassy.excalibur.group09.spec.environment.actAbstractDispatchCoordinator	

3.2.6 `actTowServiceCoordinator` Actor

Actor
actTowServiceCoordinator
Is representing a tow service driver.
Extends
lu.uni.lassy.excalibur.group09.spec.environment.actAbstractDispatchCoordinator

Chapter 4

Concept Model

4.1 PrimaryTypes-Classes

4.1.1 Local view 12

Figure 4.1 Main view of the concept model

4.1.2 Local view 18

Figure 4.2 View of the actors and operations regarding a dispatched coordinator

4.2 PrimaryTypes-Datatypes

4.2.1 Local view 15

Figure 4.3 View of all the datatypes

4.2.2 Local view 16

Figure 4.4 View of all the different modes for the coordinators/actors

4.2.3 Local view 17

Figure 4.5 View of 'mapWithPin' and its relationships

4.3 Concept Model Types Descriptions

This section provides the textual descriptions of all the types defined in the concept model and that can be part of the graphical views provided.

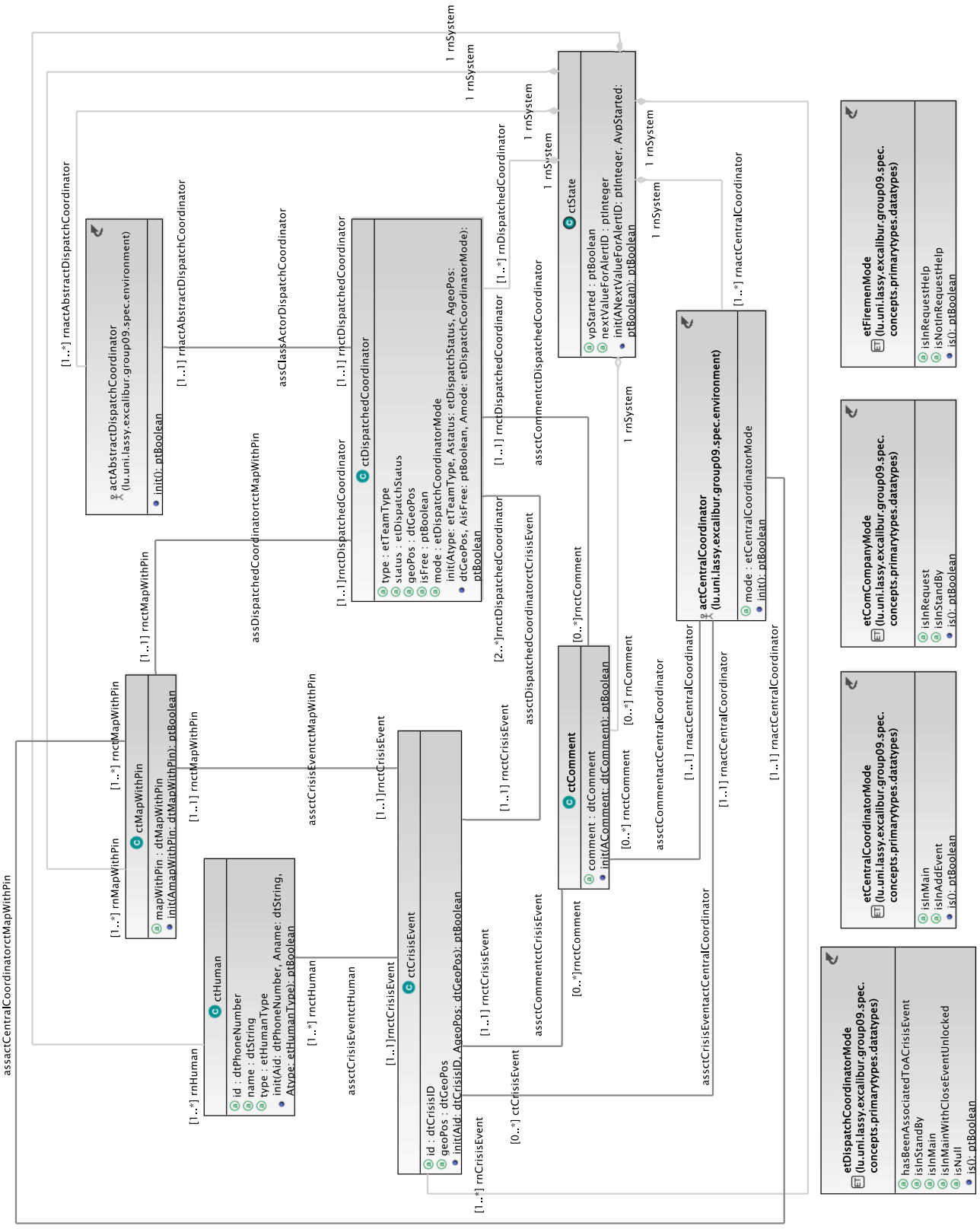


Figure 4.1: Concept Model - PrimaryTypes-Classes local view 12. Main view of the concept model.

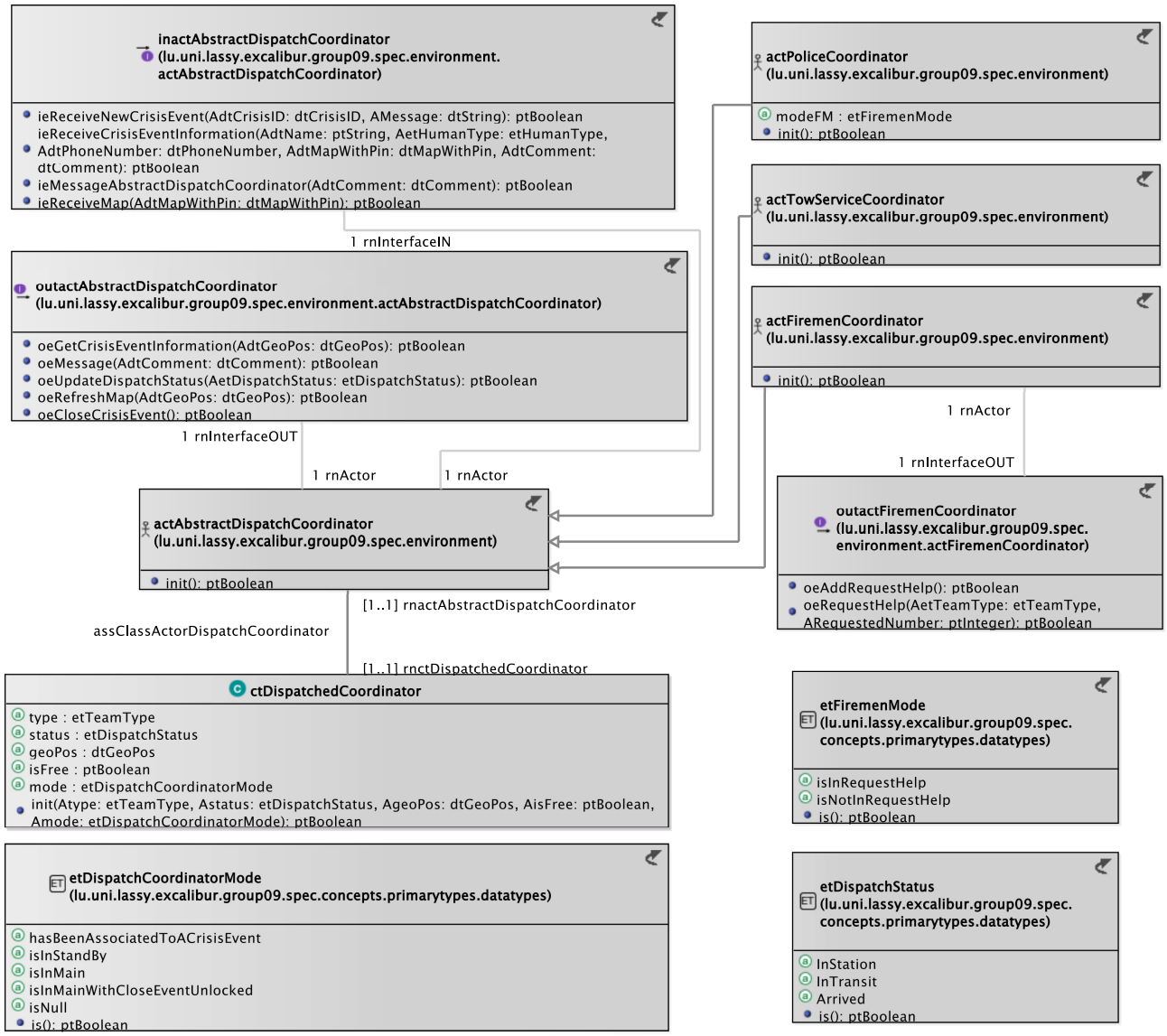


Figure 4.2: Concept Model - PrimaryTypes-Classes local view 18. View of the actors and operations regarding a dispatched coordinator.

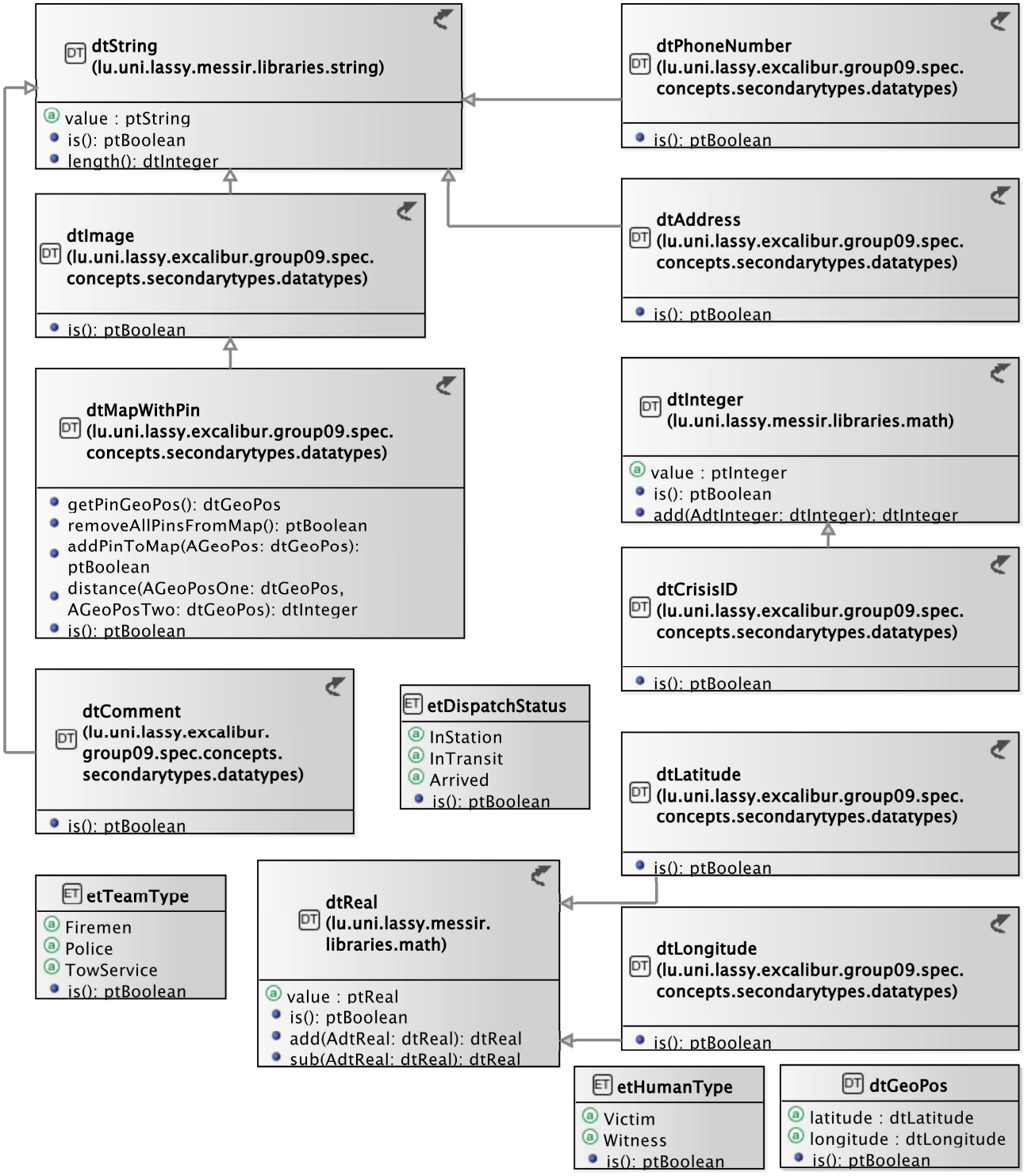


Figure 4.3: Concept Model - PrimaryTypes-Datatypes local view 15. View of all the datatypes.

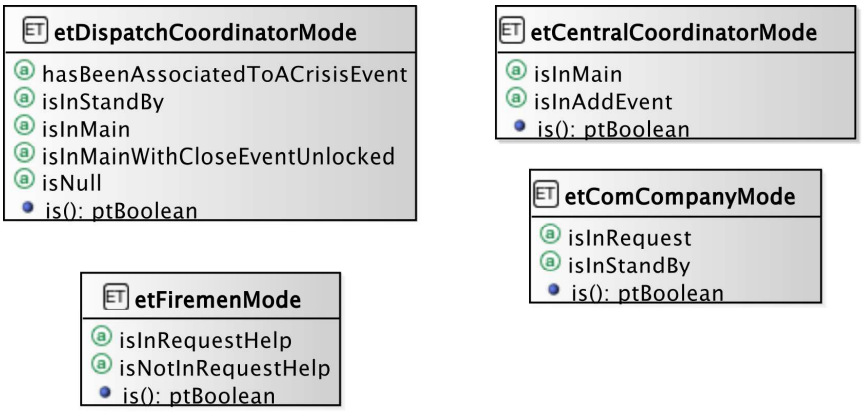


Figure 4.4: Concept Model - PrimaryTypes-Datatypes local view 16. View of all the different modes for the coordinators/actors.

4.3.1 Primary types - Class types descriptions

The table below is providing comments on the graphical views given for the class types of the primary types. Type logical operations are precisely specified in the operation model.

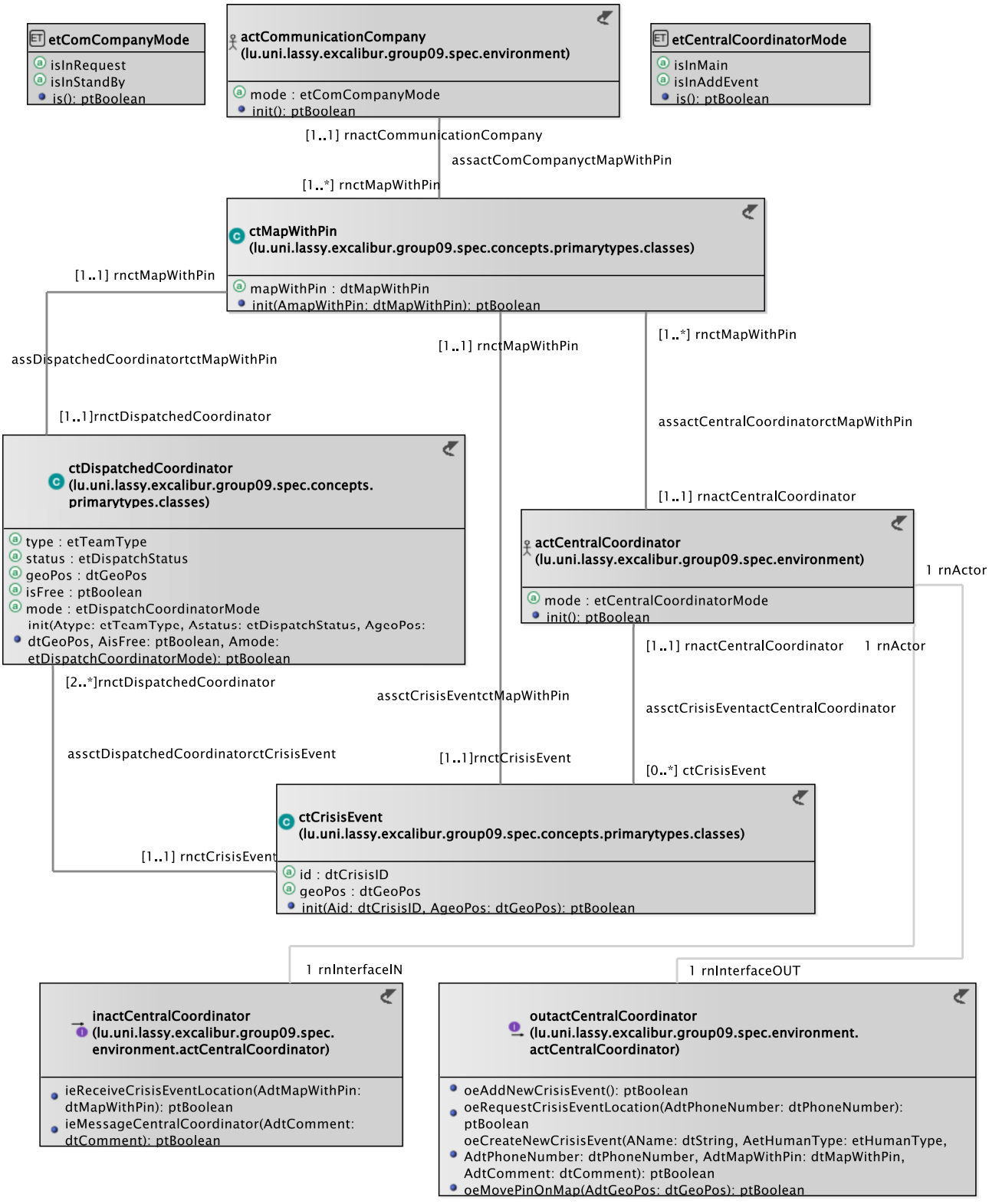


Figure 4.5: Concept Model - PrimaryTypes-Datatypes local view 17. View of 'mapWithPin' and its relationships.

Classes	
ctComment	
A class containing a comment.	
attribute	**comment: dtComment**
operation	**init(AComment:dtComment):ptBoolean**
ctCrisisEvent	
A class containing the attributes identifying a crisis event.	
attribute	**id: dtCrisisID**
operation	**init(Aid:dtCrisisID, AgeoPos:dtGeoPos):ptBoolean**
ctDispatchedCoordinator	
A class containing the attributes identifying a dispatched team.	
attribute	**status: etDispatchStatus**
attribute	**type: etTeamType**
operation	**init(Atype:etTeamType, Astatus:etDispatchStatus, AgeoPos:dtGeoPos, AisFree:ptBoolean, Amode:etDispatchCoordinatorMode):pt**
ctHuman	
A class containing the attributes identifying an human.	
attribute	**id: dtPhoneNumber**
attribute	**name: dtString**
attribute	**type: etHumanType**
operation	**init(Aid:dtPhoneNumber, Aname:dtString, Atype:etHumanType):ptBoolean**
ctMapWithPin	
A class containing an image which is the map including the pins.	
attribute	**mapWithPin: dtMapWithPin**
operation	**init(AmapWithPin:dtMapWithPin):ptBoolean**
ctState	
used to model the system.	
attribute	**vpStarted: ptBoolean**
operation	**init(ANextValueForAlertID:ptInteger, AvpStarted:ptBoolean):ptBoolean**

4.3.2 Primary types - Datatypes types descriptions

The table below is providing comments on the graphical views given for the datatype types of the primary types.

DATATYPES	
dtGeoPos	
Two Real numbers used to identify a geographical position on earth.	
attribute	**latitude: dtLatitude**
attribute	**longitude: dtLongitude**
operation	**is():ptBoolean**

ENUMERATIONS	
etCentralCoordinatorMode	
Modes of the Central Coordinator to identify what operations he/she can do at what moment.	
operation	**is():ptBoolean**
etComCompanyMode	
Modes of the Communication Company to identify what operations it can do at what moment.	
operation	**is():ptBoolean**
etDispatchCoordinatorMode	
Modes of a dispatched Coordinator to identify what operations he/she can do at what moment.	
operation	**is():ptBoolean**
etDispatchStatus	
A String used to identify a dispatch status.	
etFiremenMode	
Additional mode for the Firemen Coordinator to identify what operations he/she can do at what moment.	
operation	**is():ptBoolean**
etHumanType	
A String used to identify an Human type.	
etTeamType	
A String used to identify a team type.	

4.3.3 Primary types - Association types descriptions

The table below is providing comments on the association types of the primary types.

UNDIRECTED ASSOCIATIONS
assactCentralCoordinatorctMapWithPin
association between a Central Coordinator and a map with pin.
assactComCompanyctMapWithPin
Association of a communication company to a map with pin.
assClassActorDispatchCoordinator
Association of a dispatched coordinator to an actor of the same type.
assctCommentactCentralCoordinator

continues in next page . . .

... Undirected associations table continuation

Association of a comment to a central coordinator actor.
assctCommentctCrisisEvent Association of a comment to a crisis event.
assctCommentctDispatchedCoordinator Association of a comment to a dispatched coordinator.
assctCrisisEventactCentralCoordinator Association of a crisis event to the central coordinator actor (who created it).
assctCrisisEventctHuman Association of a crisis event to an human.
assctCrisisEventctMapWithPin Association of a crisis event with a MapWithPin image.
assctDispatchedCoordinatorctCrisisEvent Association of a dispatched coordinator to a crisis event.
assDispatchedCoordinatortctMapWithPin Association of a dispatched coordinator with a MapWithPin image.

4.3.4 Primary types - Aggregation types descriptions

There are no aggregation types for the primary types.

4.3.4.1 Primary types - Composition types descriptions

There are no composition types for the primary types.

4.3.5 Secondary types - Class types descriptions

There are no elements in this category in the system analysed.

4.3.6 Secondary types - Datatypes types descriptions

The table below is providing comments on the graphical views given for the datatype types of the secondary types.

DATATYPES	
dtAddress A String used to identify an address.	
extends operation	dtString **`is():ptBoolean`**
dtComment A String used to identify a comment.	
extends operation	dtString **`is():ptBoolean`**
dtCrisisID An Integer used to identify a crisis id.	

continues in next page . . .

... Datatypes table continuation

extends	dtInteger
operation	**is():ptBoolean**
dtImage	
A String used to identify an image.	
extends	dtString
operation	**is():ptBoolean**
dtLatitude	
used to define a latitude value of a geograpical positions on earth.	
extends	dtReal
operation	**is():ptBoolean**
dtLongitude	
used to define a longitude value of a geograpical positions on earth.	
extends	dtReal
operation	**is():ptBoolean**
dtMapWithPin	
An image which is a map including pins.	
extends	dtImage
operation	**is():ptBoolean**
dtPhoneNumber	
A String used to store a phone number.	
extends	dtString
operation	**is():ptBoolean**

4.3.7 Secondary types - Association types descriptions

There are no association types for the secondary types.

4.3.8 Secondary types - Aggregation types descriptions

There are no aggregation types for the secondary types.

4.3.9 Secondary types - Composition types descriptions

There are no composition types for the secondary types.

Chapter 5

Operation Model

This section contains the operation schemes of each operation defined in either an actor, its output interface, in a primary or secondary type (class, datatype or enumeration types). The Messir OCL code listing is joined to the comment table.

5.1 Environment - Out Interface Operation Scheme for actAbstractDispatchCoordinator

5.1.1 Operation Model for oeCloseCrisisEvent

The `oeCloseCrisisEvent` operation has the following properties:

Operation	
oeCloseCrisisEvent sent to close up the associated crisis event for the current coordinator.	
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The dispatch coordinator's mode is `isInMainWithCloseEventUnlocked`.
Pre-Condition (functional)	
PreF 1	it is supposed that a dispatched coordinator can only be associated to a single crisis event at the same time.
Post-Condition (functional)	
PostF 1	The coordinator's attribute `isFree` is set back to true and can thus be associated to another crisis event.
Post-Condition (protocol)	
PostP 1	The dispatch coordinator is no longer associated to the current crisis event.
PostP 2	The dispatch coordinator's mode has is to `isInStandBy`.

5.1.2 Operation Model for oeGetCrisisEventInformation

The `oeGetCrisisEventInformation` operation has the following properties:

Operation
oeGetCrisisEventInformation sent to get the stored information of the crisis event to which the dispatch coordinator is associated.

continues in next page . . .

. . . Operation table continuation

Parameters	
1	**AdtGeoPos: dtGeoPos** a geographical position that identifies the actor's current position.
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The dispatch coordinator's mode is `hasBeenAssociatedToACrisisEvent`.
Pre-Condition (functional)	
PreF 1	it is supposed that a dispatched coordinator can only be associated to a single crisis event at the same time.
PreF 2	the `GeoPos` given by the coordinator is a valid one.
Post-Condition (functional)	
PostF 1	the map with pins returned to the coordinator includes a pin of the actor's current position and another one of the crisis event's location.
Post-Condition (protocol)	
PostP 1	The dispatch coordinator's mode has been set to `isInMain`.

5.1.3 Operation Model for oeMessage

The `oeMessage` operation has the following properties:

Operation	
oeMessage sent to transmit a message.	
Parameters	
1	**AdtComment: dtComment** A comment or message that the coordinator wants to transmit.
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The dispatch coordinator's mode is `isInMain` or `isInMainWithCloseEventUnlocked`.
Pre-Condition (functional)	
PreF 1	it is supposed that a dispatched coordinator can only be associated to a single crisis event at the same time.

5.1.4 Operation Model for oeRefreshMap

The `oeRefreshMap` operation has the following properties:

Operation	
oeRefreshMap sent to refresh the map.	
Parameters	
1	**AdtGeoPos: dtGeoPos** the coordinator's current geographical position.

continues in next page . . .

. . . Operation table continuation

Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The dispatch coordinator's is `isInMain` or `isInMainWithCloseEventUnlocked`.
Pre-Condition (functional)	
PreF 1	it is supposed that a dispatched coordinator can only be associated to a single crisis event at the same time.
PreF 2	the `GeoPos` given by the coordinator is a valid one.
Post-Condition (functional)	
PostF 1	the map with pins returned to the coordinator includes a pin of the actor's current position and another one of the crisis event's location.

5.1.5 Operation Model for oeUpdateDispatchStatus

The `oeUpdateDispatchStatus` operation has the following properties:

OPERATION	
oeUpdateDispatchStatus sent to update the dispatch status.	
Parameters	
1	**AetDispatchStatus: etDispatchStatus**
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The dispatch coordinator's mode is `isInMain`.
Pre-Condition (functional)	
PreF 1	it is supposed that a dispatched coordinator can only be associated to a single crisis event at the same time.
Post-Condition (functional)	
PostF 1	the attribute `status` of the coordinator is modified either from 'InStation' to 'InTransit' or from 'InTransit' to 'Arrived'
Post-Condition (protocol)	
PostP 1	when the attribute `status` of the coordinator is set to 'Arrived', the actor's mode is set to `isInMainWithCloseEventUnlocked`.

5.2 Environment - Out Interface Operation Scheme for actCentralCoordinator

5.2.1 Operation Model for oeAddNewCrisisEvent

The `oeAddNewCrisisEvent` operation has the following properties:

OPERATION
oeAddNewCrisisEvent sent with the intention to add a new crisis event.

continues in next page . . .

. . . Operation table continuation

Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The actor's mode is `isInMain`.
Post-Condition (protocol)	
PostP 1	The actor's mode has been set to `isInAddEvent`.

5.2.2 Operation Model for oeCreateNewCrisisEvent

The `oeCreateNewCrisisEvent` operation has the following properties:

OPERATION	
oeCreateNewCrisisEvent sent to create a new crisis event and to alert the corresponding coordinators.	
Parameters	
1	**AName: dtString** the name of the notifier that informed the Central Coordinator of the crisis event.
2	**AetHumanType: etHumanType** the notifier can be either a victim or a witness.
3	**AdtPhoneNumber: dtPhoneNumber** the phone number of the notifier.
4	**AdtMapWithPin: dtMapWithPin** a map with pin showing the crisis event's location.
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The actor's mode is `isInAddEvent`.
Pre-Condition (functional)	
PreF 1	The map with pin provided by the central coordinator only has a single pin.
Post-Condition (functional)	
PostF 1	A new crisis event is created and initialised with a new crisis event id attribute which is the `nextValueForAlertID`@pre in `ctState` and the geographical position attribute of the crisis event which is extracted from the map with pin provided by the central coordinator.
PostF 2	An alert message 'You have received a new dispatch order!' together with new crisis event id is sent to a free FiremenCoordinator and a free TowServiceCoordinator that are geographically the nearest of the crisis event's location and their dispatch status attributes are set to 'InStation'.
PostF 3	The new crisis event is then associated to : the central coordinator who created this instance, the two selected dispatch coordinators, the map with pin instance that has the map with pin attriubute communicated by the central central coordinator in the parameters, the human instance, who may be initialised, if he/she is not yet in the system state, with his/her phone number as the unique id, a name and a type (witness/victim) which are given as parameters by the central coordinator, comment instance(s) which is/are initalised if the central coordinator has given some additional comments to the crisis event.

continues in next page . . .

. . . Operation table continuation

PostF 4	The two selected dispatch coordinators' attribute `isFree` is set to false and can thus no more be associated to another crisis event.
PostF 5	the attribute `nextValueForAltertID` in ctState instance should be equal to the one @pre incremented by one.
Post-Condition (protocol)	
PostP 1	The actor's mode is set to `isInMain`.
PostP 2	the two selected dispatch coordinators' modes are set to `hasBeenAssociatedToACrisisEvent`.
PostP 3	the selected firemen coordinator's modeFM is set to `isNotInRequestHelp`.

5.2.3 Operation Model for oeMovePinOnMap

The `oeMovePinOnMap` operation has the following properties:

OPERATION	
oeMovePinOnMap	
sent to move the pin on the map (adjustments).	
Parameters	
1	**AdtGeoPos: dtGeoPos** the new geographical position of where the pin should be moved to.
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The actor's mode is `isInAddEvent`.
Pre-Condition (functional)	
PreF 1	It is supposed that the actor's is already associated with a newly created map with pin.
Post-Condition (functional)	
PostF 1	The returned map has the previous pin replaced by a new one using the geographical position given by the actor.
PostF 2	The returned map only has a single pin on it.

5.2.4 Operation Model for oeRequestCrisisEventLocation

The `oeRequestCrisisEventLocation` operation has the following properties:

OPERATION	
oeRequestCrisisEventLocation	
sent to request a crisis event's location.	
Parameters	
1	**AdtPhoneNumber: dtPhoneNumber** The phone number of the witness/victim used to get the geographical position.
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The actor's mode is `isInAddEvent`.
Pre-Condition (functional)	

continues in next page . . .

. . . Operation table continuation

PreF 1	it is supposed that the phone nuumber given by the `CentralCoordinator` is always sent to the correct communication company.
Post-Condition (functional)	
PostF 1	the phone number can be identified by the communication company.
Post-Condition (protocol)	
PostP 1	The communication company's mode is set to `isInRequest`.

5.3 Environment - Out Interface Operation Scheme for actCommunicationCompany

5.3.1 Operation Model for oeReceiveCrisisEventLocation

The `oeReceiveCrisisEventLocation` operation has the following properties:

OPERATION	
oeReceiveCrisisEventLocation sent to get a map with pin returned to the central coordinator.	
Parameters	
1	**AdtGeoPos: dtGeoPos** the geographical position used to initialise the map with pin.
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The actor's mode is `isInRequest`.
Pre-Condition (functional)	
PreF 1	the `GeoPos` given by the communication company is a valid one.
Post-Condition (functional)	
PostF 1	A new map with pin instance is created and initialised using the given geographical position as base.
PostF 2	The new map with pin returned to the `CentralCoordinator` only has a single pin, which is the one corresponding to the geographical position given in the parameters.
PostF 3	The new map with pin is then associated to the central coordinator who received it and the communication company who initialised it,
Post-Condition (protocol)	
PostP 1	The actor's mode has been set to `isInStandBy`.

5.4 Environment - Out Interface Operation Scheme for actFiremenCoordinator

5.4.1 Operation Model for oeAddRequestHelp

The `oeAddRequestHelp` operation has the following properties:

OPERATION	
oeAddRequestHelp	
sent with the intention to request help from additional dispatch coordinators.	
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The firemen coordinator's mode is `isInMain` or `isInMainWithCloseEventUnlocked`.
PreP 2	The firemen coordinator's modeFM is `isNotInRequestHelp`.
Post-Condition (protocol)	
PostP 1	The firemen coordinator's mode is set to `isNull`.
PostP 2	The firemen coordinator's modeFM is set to `isInRequestHelp`.

5.4.2 Operation Model for oeRequestHelp

The `oeRequestHelp` operation has the following properties:

OPERATION	
oeRequestHelp	
sent to assign additional dispatch coordinators to the associated crisis event.	
Parameters	
1	**AetTeamType: etTeamType**
2	**ARequestedNumber: ptInteger**
Return type	
ptBoolean	
Pre-Condition (protocol)	
PreP 1	The firemen coordinator's mode has is `isNull`.
PreP 2	The firemen coordinator's modeFM is `isInRequestHelp`.
Pre-Condition (functional)	
PreF 1	it is supposed that a dispatched (or firemen) coordinator can only be associated to a single crisis event at the same time.
Post-Condition (functional)	
PostF 1	An alert message 'You have received a new dispatch order!' is sent to one or more free FiremenCoordinator(s) and/or one or more free TowServiceCoordinator(s) and/or one or more free PoliceCoordinator(s) that are geographically the nearest of the crisis event's location and their dispatch status attributes are set to 'InStation'.
PostF 2	The crisis event (self.rnActor.rnctCrisisEvent) is then associated to the selected dispatch coordinator(s).
PostF 3	The selected dispatch coordinator(s)' attribute `isFree` is set to false and can thus no more be associated to another crisis event.
Post-Condition (protocol)	
PostP 1	The actor's mode is set to `isInMain` or `isInMainWithCloseEventUnlocked`.
PostP 2	The selected dispatch coordinator(s)' modes are set to `hasBeenAssociatedToACrisisEvent`.

continues in next page . . .

... Operation table continuation

PostP 3	If a firemen coordinator has been selected, his modeFM is set to `isNotInRequestHelp`.

5.5 Environment - Actor Operation Schemes

There are no elements in this category in the system analysed.

5.6 Primary Types - Operation Schemes for Classes

There are no elements in this category in the system analysed.

5.7 Primary Types - Operation Schemes for Datatypes

There are no elements in this category in the system analysed.

5.8 Primary Types - Operation Schemes for Enumerations

There are no elements in this category in the system analysed.

5.9 Secondary Types - Operation Schemes for Classes

There are no elements in this category in the system analysed.

5.10 Secondary Types - Operation Schemes for Datatype dtMapWithPin

5.10.1 Operation Model for addPinToMap

The `addPinToMap` operation has the following properties:

OPERATION	
addPinToMap	
used to add a pin to the map.	
Parameters	
1	**AGeoPos: dtGeoPos** the geographical position of the pin we want to add.
Return type	
ptBoolean	
Post-Condition (functional)	
PostF 1	The given geographical position is converted into a pin on the map.

5.10.2 Operation Model for getPinGeoPos

The `getPinGeoPos` operation has the following properties:

OPERATION

continues in next page ...

. . . Operation table continuation

getPinGeoPos
used to get the geographical position of the pin that is currently shown on the map (used only when there's a single pin on the map).
Return type
dtGeoPos
Post-Condition (functional)
PostF 1 Returns a geographical position of the pin that was on the map.

5.10.3 Operation Model for removeAllPinsFromMap

The `removeAllPinsFromMap` operation has the following properties:

Operation
removeAllPinsFromMap
used to remove all pins on the map (usually used to afterwards regenerate new pins, so that the map stays clean).
Return type
ptBoolean
Post-Condition (functional)
PostF 1 All pins are removed from the map.

5.11 Secondary Types - Operation Schemes for Enumerations

There are no elements in this category in the system analysed.

Chapter 6

Test Model(s)

There are no elements in this category in the system analysed.

Chapter 7

Additional Constraints

Appendix A

Undocumented Messir Specification Elements

A.1 Undocumented Use Cases

A.1.1 Undocumented Subfunction Level Use Cases

- lu.uni.lassy.excalibur.group09.spec.usecases.oeAddRequestHelp
- lu.uni.lassy.excalibur.group09.spec.usecases.oeCloseCrisisEvent
- lu.uni.lassy.excalibur.group09.spec.usecases.oeCreateNewCrisisEvent
- lu.uni.lassy.excalibur.group09.spec.usecases.oeGetCrisisEventInformation
- lu.uni.lassy.excalibur.group09.spec.usecases.oeAddNewCrisisEvent
- lu.uni.lassy.excalibur.group09.spec.usecases.oeMessage
- lu.uni.lassy.excalibur.group09.spec.usecases.oeMovePinOnMap
- lu.uni.lassy.excalibur.group09.spec.usecases.oeRefreshMap
- lu.uni.lassy.excalibur.group09.spec.usecases.oeRequestCrisisEventLocation
- lu.uni.lassy.excalibur.group09.spec.usecases.oeRequestHelp
- lu.uni.lassy.excalibur.group09.spec.usecases.oeReceiveCrisisEventLocation
- lu.uni.lassy.excalibur.group09.spec.usecases.oeUpdateDispatchStatus

Appendix B

Messir Specification Files Listing

B.1 File ./src-gen/messir-spec/.views.msr

```
//
//DON'T TOUCH THIS FILE !!!
//
package uuidff8a216549a64951bf055c8b5a9dde2a {
 Concept Model {}
}
```

Listing B.1: Messir Spec. file .views.msr.

B.2 File ./src-gen/messir-spec/operations/environment/environment-actAbstractDispatchCoordinator-oeCloseCrisisEvent.msr

```
package lu.uni.lassy.excalibur.group09.spec.environment.operations.actAbstractDispatchCoordinator.
    outactAbstractDispatchCoordinator.oeCloseCrisisEvent {

import lu.uni.lassy.messir.libraries.primitives
import lu.uni.lassy.messir.libraries.math
import lu.uni.lassy.messir.libraries.string
import lu.uni.lassy.messir.libraries.calendar
import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes

 Operation Model {

  operation: lu.uni.lassy.excalibur.group09.spec.environment.actAbstractDispatchCoordinator.
      outactAbstractDispatchCoordinator.oeCloseCrisisEvent():ptBoolean{
  // include below the specification information (pre,post or ocl or prolog)

  }
 }
}
```

Listing B.2: Messir Spec. file environment-actAbstractDispatchCoordinator-oeCloseCrisisEvent.msr.

B.3 File ./src-gen/messir-spec/operations/environment/environment-actAbstractDispatchCoordinator-oeGetCrisisEventInformation.msr

```
package lu.uni.lassy.excalibur.group09.spec.environment.operations.actAbstractDispatchCoordinator.
    outactAbstractDispatchCoordinator.oeGetCrisisEventInformation {

import lu.uni.lassy.messir.libraries.primitives
import lu.uni.lassy.messir.libraries.math
import lu.uni.lassy.messir.libraries.string
import lu.uni.lassy.messir.libraries.calendar
```

```
7 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
9
10 Operation Model {
11
12  operation: lu.uni.lassy.excalibur.group09.spec.environment.actAbstractDispatchCoordinator.
        outactAbstractDispatchCoordinator.oeGetCrisisEventInformation(AdtGeoPos:dtGeoPos):ptBoolean{
13  // include below the specification information (pre,post or ocl or prolog)
14
15  }
16 }
17 }
```

Listing B.3: Messir Spec. file environment-actAbstractDispatchCoordinator-oeGetCrisisEventInformation.msr.

B.4 File ./src-gen/messir-spec/operations/environment/environment-actAbstractDispatchCoordinator-oeMessage.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.environment.operations.actAbstractDispatchCoordinator.
      outactAbstractDispatchCoordinator.oeMessage {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
9
10 Operation Model {
11
12  operation: lu.uni.lassy.excalibur.group09.spec.environment.actAbstractDispatchCoordinator.
        outactAbstractDispatchCoordinator.oeMessage(AdtComment:dtComment):ptBoolean{
13  // include below the specification information (pre,post or ocl or prolog)
14
15  }
16 }
17 }
```

Listing B.4: Messir Spec. file environment-actAbstractDispatchCoordinator-oeMessage.msr.

B.5 File ./src-gen/messir-spec/operations/environment/environment-actAbstractDispatchCoordinator-oeRefreshMap.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.environment.operations.actAbstractDispatchCoordinator.
      outactAbstractDispatchCoordinator.oeRefreshMap {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
9
10 Operation Model {
11
12  operation: lu.uni.lassy.excalibur.group09.spec.environment.actAbstractDispatchCoordinator.
        outactAbstractDispatchCoordinator.oeRefreshMap(AdtGeoPos:dtGeoPos):ptBoolean{
13  // include below the specification information (pre,post or ocl or prolog)
14
15  }
16 }
17 }
```

Listing B.5: Messir Spec. file environment-actAbstractDispatchCoordinator-oeRefreshMap.msr.

B.6 File ./src-gen/messir-spec/operations/environment/environment-actAbstractDispatchCoordinator-oeUpdateDispatchStatus.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.environment.operations.actAbstractDispatchCoordinator.
      outactAbstractDispatchCoordinator.oeUpdateDispatchStatus {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
9
10  Operation Model {
11
12   operation: lu.uni.lassy.excalibur.group09.spec.environment.actAbstractDispatchCoordinator.
        outactAbstractDispatchCoordinator.oeUpdateDispatchStatus(AetDispatchStatus:etDispatchStatus):
        ptBoolean{
13   // include below the specification information (pre,post or ocl or prolog)
14
15   }
16  }
17 }
```

Listing B.6: Messir Spec. file environment-actAbstractDispatchCoordinator-oeUpdateDispatchStatus.msr.

B.7 File ./src-gen/messir-spec/operations/environment/environment-actCentralCoordinator-oeAddNewCrisisEvent.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.environment.operations.actCentralCoordinator.
      outactCentralCoordinator.oeAddNewCrisisEvent {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7
8  Operation Model {
9
10   operation: lu.uni.lassy.excalibur.group09.spec.environment.actCentralCoordinator.
        outactCentralCoordinator.oeAddNewCrisisEvent():ptBoolean{
11   // include below the specification information (pre,post or ocl or prolog)
12
13   }
14  }
15 }
```

Listing B.7: Messir Spec. file environment-actCentralCoordinator-oeAddNewCrisisEvent.msr.

B.8 File ./src-gen/messir-spec/operations/environment/environment-actCentralCoordinator-oeConfirmCrisisEventLocation.msr

```
1 //package lu.uni.lassy.excalibur.group09.spec.environment.operations.actCentralCoordinator.
      outactCentralCoordinator.oeConfirmCrisisEventLocation {
2 //
3 //import lu.uni.lassy.messir.libraries.primitives
4 //import lu.uni.lassy.messir.libraries.math
5 //import lu.uni.lassy.messir.libraries.string
6 //import lu.uni.lassy.messir.libraries.calendar
7 //
8 // Operation Model {
9 //
10 //  operation: lu.uni.lassy.excalibur.group09.spec.environment.actCentralCoordinator.
      outactCentralCoordinator.oeConfirmCrisisEventLocation():ptBoolean{
```

```
//  // include below the specification information (pre,post or ocl or prolog)
//
//  }
// }
//}
```

Listing B.8: Messir Spec. file environment-actCentralCoordinator-oeConfirmCrisisEventLocation.msr.

B.9 File ./src-gen/messir-spec/operations/environment/environment-actCentralCoordinator-oeCreateNewCrisisEvent.msr

```
package lu.uni.lassy.excalibur.group09.spec.environment.operations.actCentralCoordinator.
    outactCentralCoordinator.oeCreateNewCrisisEvent {

import lu.uni.lassy.messir.libraries.primitives
import lu.uni.lassy.messir.libraries.math
import lu.uni.lassy.messir.libraries.string
import lu.uni.lassy.messir.libraries.calendar
import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.classes
import lu.uni.lassy.excalibur.group09.spec.environment

 Operation Model {

  operation: lu.uni.lassy.excalibur.group09.spec.environment.actCentralCoordinator.
      outactCentralCoordinator.oeCreateNewCrisisEvent(
   AName:dtString,
   AetHumanType:etHumanType,
   AdtPhoneNumber:dtPhoneNumber,
   AdtMapWithPin:dtMapWithPin,
   AdtComment:dtComment
  ):ptBoolean{
  // include below the specification information (pre,post or ocl or prolog)
  preP{
   let AetCentralCoordinatorMode:etCentralCoordinatorMode in

   //PreP 01
   self.rnActor.mode@pre = AetCentralCoordinatorMode
   and 'isInAddEvent' = AetCentralCoordinatorMode

  }

  postF{
   let TheCrisisEvent:ctCrisisEvent in
   let TheMessage:dtString in

   let AdtGeoPosOfCrisisEvent:dtGeoPos in
   let AdtCrisisID:dtCrisisID in

   let TheFiremenCoordinator:ctDispatchedCoordinator in
   let TheTowServiceCoordinator:ctDispatchedCoordinator in
   let TheFiremenCoordinatorIN:actAbstractDispatchCoordinator in
   let TheTowServiceCoordinatorIN:actAbstractDispatchCoordinator in

   let AetFiremenMode:etFiremenMode in
   let AFirstTeamType:etTeamType in
   let ASecondTeamType:etTeamType in

   let TheHuman:ctHuman in

   //PostF 01
   self.rnActor.rnSystem.nextValueForAlertID@pre = AdtCrisisID
   and self.rnActor.rnctMapWithPin.mapWithPin.getPinGeoPos() = AdtGeoPosOfCrisisEvent
   and TheCrisisEvent.init(
    AdtCrisisID,
    AdtGeoPosOfCrisisEvent
   )
```

```
56
57    //PostF 02
58    and TheFiremenCoordinator.type@pre = AFirstTeamType
59    and 'Firemen' = AFirstTeamType
60    and TheTowServiceCoordinator.type@pre = ASecondTeamType
61    and 'TowService' = ASecondTeamType
62    and 'TRUE' = TheFiremenCoordinator.isFree@pre
63    and 'TRUE' = TheTowService.isFree@pre
64
65    /*
66    ctDispatchCoordinator such that there doesn't exist another ctDispatchCoordinator that has a
          shorter distance
67    between its geographical position and the geographical position of the newly instanciated crisis
          event.
68    */
69    and TheFiremenCoordinator = self.rnActor.rnSystem.rnctDispatchedCoordinator->select(
70       dc1 / dtMapWithPin.distance(dc1.geoPos,AdtGeoPosOfCrisisEvent) > distance(self.rnActor.
              rnSystem.rnctDispatchedCoordinator.geoPos,AdtGeoPosOfCrisisEvent)->isEmpty()
71    )
72
73    and TheTowServiceCoordinator = self.rnActor.rnSystem.rnctDispatchedCoordinator->select(
74       dc1 / dtMapWithPin.distance(dc1.geoPos,AdtGeoPosOfCrisisEvent) > distance(self.rnActor.
              rnSystem.rnctDispatchedCoordinator.geoPos,AdtGeoPosOfCrisisEvent)->isEmpty()
75    )
76
77    and 'InStation' = TheFiremen.status@post
78    and 'InStation' = TheFiremen.status@post
79
80    and 'You have received a new dispatch order!' = TheMessage.value
81    and TheFiremenIN^ieReceiveNewCrisisEvent(AdtCrisisID, 'You have received a new dispatch order!')
82    and TheTowServiceIN^ieReceiveNewCrisisEvent(AdtCrisisID, 'You have received a new dispatch order
          !')
83
84    //PostF 03
85    and TheCrisisEvent.rnactCentralCoordinator = self.rnActor@pre
86
87    and TheCrisisEvent.rnctDispatchedCoordinator = TheFiremenCoordinator
88    and TheCrisisEvent.rnctDispatchedCoordinator = TheTowServiceCoordinator
89
90    and TheCrisisEvent.rnctMapWithPin = self.rnActor.rnctMapWithPin@post
91
92    //IF an ctHuman instance with the same id (PhoneNumber) is already in the system state, associate
          it to that instance
93    //ELSE initialise a new ctHuman instance and associate it
94    and if(
95      self.rnActor.rnSystem.rnHuman -> exists(h / h.id = AdtPhoneNumber)
96     )
97     then(
98      TheHuman = self.rnActor.rnSystem.rnHuman -> select(h / h.id = AdtPhoneNumber)
99      TheCrisisEvent.rnHuman = TheHuman
100    )
101    else(
102     TheHuman.init(
103      AdtPhoneNumber,
104      AName,
105      AetHumanType
106     )
107     TheCrisisEvent.rnHuman = TheHuman
108    )
109    endif
110
111   and TheCrisisEvent.rnctComment = self.rnActor.rnctComment@post
112
113   //PostF 04
114   and 'FALSE' = TheFiremenCoordinator.isFree@post
115   and 'FALSE' = TheTowServiceCoordinator.isFree@post
116
117   //PostF 05
118   and self.rnActor.rnSystem.nextValueForAlertID@post = self.rnActor.rnSystem.
          nextValueForAlertID@pre + 1
```

```
119
120   }
121
122   postP{
123    let TheCrisisEvent:ctCrisisEvent in
124
125    let AetCentralCoordinatorMode:etCentralCoordinatorMode in
126    let AetDispatchCoordinatorMode:etDispatchCoordinatorMode in
127
128    let AFirstTeamType:etTeamType in
129    let ASecondTeamType:etTeamType in
130
131    let TheFiremenCoordinator:ctDispatchedCoordinator in
132    let TheTowServiceCoordinator:ctDispatchedCoordinator in
133
134    let AetFiremenMode:etFiremenMode in
135
136    //PostP 01
137    and self.rnActor.mode@post = AetCentralCoordinatorMode
138    and 'isInMain' = AetCentralCoordinatorMode
139
140    /*PostP 02 & 03*/
141    and self.rnActor.rnctCrisisEvent = TheCrisisEvent
142
143    //To make sure who is who
144    and TheFiremenCoordinator.type@pre = AFirstTeamType
145    and 'Firemen' = AFirstTeamType
146    and TheTowServiceCoordinator.type@pre = ASecondTeamType
147    and 'TowService' = ASecondTeamType
148
149    and TheCrisisEvent.rnctDispatchedCoordinator = TheFiremenCoordinator
150    and TheCrisisEvent.rnctDispatchedCoordinator = TheTowServiceCoordinator
151
152    //PostP 02
153    and TheFiremenCoordinator.mode@post = AetDispatchCoordinatorMode
154    and TheTowServiceCoordinator.mode@post = AetDispatchCoordinatorMode
155    and 'hasBeenAssociatedToACrisisEvent' = AetDispatchCoordinatorMode
156
157    //PostP 03
158    and TheFirementCoordinator.modeFM@post = AetFiremenMode
159    and 'IsNotInRequestHelp' = AetFiremenMode
160   }
161
162   }
163  }
164 }
```

Listing B.9: Messir Spec. file environment-actCentralCoordinator-oeCreateNewCrisisEvent.msr.

B.10 File ./src-gen/messir-spec/operations/environment/environment-actCentralCoordinator-oeInitialiseNewCrisisEvent.msr

```
1 //package lu.uni.lassy.excalibur.group09.spec.environment.operations.actCentralCoordinator.
      outactCentralCoordinator.oeInitialiseNewCrisisEvent {
2 //
3 //import lu.uni.lassy.messir.libraries.primitives
4 //import lu.uni.lassy.messir.libraries.math
5 //import lu.uni.lassy.messir.libraries.string
6 //import lu.uni.lassy.messir.libraries.calendar
7 //
8 // Operation Model {
9 //
10 //  operation: lu.uni.lassy.excalibur.group09.spec.environment.actCentralCoordinator.
      outactCentralCoordinator.oeInitialiseNewCrisisEvent():ptBoolean{
11 //  // include below the specification information (pre,post or ocl or prolog)
12 //
13 //  }
14 // }
```

```
15 //}
```

Listing B.10: Messir Spec. file environment-actCentralCoordinator-oeInitialiseNewCrisisEvent.msr.

B.11 File ./src-gen/messir-spec/operations/environment/environment-actCentralCoordinator-oeMovePinOnMap.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.environment.operations.actCentralCoordinator.
      outactCentralCoordinator.oeMovePinOnMap {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
8
9  Operation Model {
10
11   operation: lu.uni.lassy.excalibur.group09.spec.environment.actCentralCoordinator.
         outactCentralCoordinator.oeMovePinOnMap(AdtGeoPos:dtGeoPos):ptBoolean{
12   // include below the specification information (pre,post or ocl or prolog)
13
14   }
15  }
16 }
```

Listing B.11: Messir Spec. file environment-actCentralCoordinator-oeMovePinOnMap.msr.

B.12 File ./src-gen/messir-spec/operations/environment/environment-actCentralCoordinator-oeRequestCrisisEventLocation.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.environment.operations.actCentralCoordinator.
      outactCentralCoordinator.oeRequestCrisisEventLocation {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
9
10  Operation Model {
11
12   operation: lu.uni.lassy.excalibur.group09.spec.environment.actCentralCoordinator.
         outactCentralCoordinator.oeRequestCrisisEventLocation(AdtPhoneNumber:dtPhoneNumber):ptBoolean{
13   // include below the specification information (pre,post or ocl or prolog)
14  preP {
15    let AvpStarted: ptBoolean in
16    self.rnActor.rnSystem.vpStarted = AvpStarted
17    and AvpStarted = true
18  }
19
20  preF { true }
21
22  postF {
23   let TheactYou:lu.uni.lassy.excalibur.group09.spec.environment.actCentralCoordinator in
24   let AptString:ptString in
25   /* Post Functional:*/
26   /* PostF01 */
27   AptString = 'Hello World !'
28   and TheactYou.InterfaceIN = self.rnActor.InterfaceIN
29   and TheactYou.InterfaceIN^ieHelloWorld(AptString)
30  }
31
32  postP { true }
33   }
```

```
34  }
35 }
```

Listing B.12: Messir Spec. file environment-actCentralCoordinator-oeRequestCrisisEventLocation.msr.

B.13 File ./src-gen/messir-spec/operations/environment/environment-actCommunicationCompany-oeReceiveCrisisEventLocation.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.environment.operations.actCommunicationCompany.
      outactCommunicationCompany.oeReceiveCrisisEventLocation {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
9
10  Operation Model {
11
12   operation: lu.uni.lassy.excalibur.group09.spec.environment.actCommunicationCompany.
        outactCommunicationCompany.oeReceiveCrisisEventLocation(AdtGeoPos:dtGeoPos):ptBoolean{
13   // include below the specification information (pre,post or ocl or prolog)
14
15   }
16  }
17 }
```

Listing B.13: Messir Spec. file environment-actCommunicationCompany-oeReceiveCrisisEventLocation.msr.

B.14 File ./src-gen/messir-spec/operations/environment/environment-actFiremenCoordinator-oeAddRequestHelp.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.environment.operations.actFiremenCoordinator.
      outactFiremenCoordinator.oeAddRequestHelp {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7
8  Operation Model {
9
10   operation: lu.uni.lassy.excalibur.group09.spec.environment.actFiremenCoordinator.
        outactFiremenCoordinator.oeAddRequestHelp():ptBoolean{
11   // include below the specification information (pre,post or ocl or prolog)
12
13   }
14  }
15 }
```

Listing B.14: Messir Spec. file environment-actFiremenCoordinator-oeAddRequestHelp.msr.

B.15 File ./src-gen/messir-spec/operations/environment/environment-actFiremenCoordinator-oeRequestHelp.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.environment.operations.actFiremenCoordinator.
      outactFiremenCoordinator.oeRequestHelp {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
```

```
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
9
10  Operation Model {
11
12   operation: lu.uni.lassy.excalibur.group09.spec.environment.actFiremenCoordinator.
        outactFiremenCoordinator.oeRequestHelp(AetTeamType:etTeamType, ARequestedNumber:ptInteger):
        ptBoolean{
13   // include below the specification information (pre,post or ocl or prolog)
14
15   }
16  }
17 }
```

Listing B.15: Messir Spec. file environment-actFiremenCoordinator-oeRequestHelp.msr.

B.16 File ./src-gen/messir-spec/environment/environment.msr

```
1 /*
2 * @author Kira
3 * @date Tue Oct 25 23:54:03 CEST 2016
4 */
5
6 package lu.uni.lassy.excalibur.group09.spec.environment {
7
8 import lu.uni.lassy.messir.libraries.calendar
9 import lu.uni.lassy.messir.libraries.math
10 import lu.uni.lassy.messir.libraries.primitives
11 import lu.uni.lassy.messir.libraries.string
12 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
13 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
14
15 Environment Model {
16
17   actor actCentralCoordinator role rnactCentralCoordinator cardinality [1..*] {
18
19    attribute mode: etCentralCoordinatorMode
20    operation init():ptBoolean
21
22    input interface inactCentralCoordinator {
23     operation ieReceiveCrisisEventLocation(AdtMapWithPin:dtMapWithPin) : ptBoolean
24     operation ieMessageCentralCoordinator(AdtComment:dtComment) : ptBoolean
25    }
26
27    output interface outactCentralCoordinator {
28     operation oeAddNewCrisisEvent() : ptBoolean
29     operation oeRequestCrisisEventLocation(AdtPhoneNumber:dtPhoneNumber) : ptBoolean
30     operation oeCreateNewCrisisEvent(AName:dtString, AetHumanType:etHumanType, AdtPhoneNumber:
           dtPhoneNumber, AdtMapWithPin:dtMapWithPin, AdtComment:dtComment) : ptBoolean
31     operation oeMovePinOnMap (AdtGeoPos:dtGeoPos) : ptBoolean
32    }
33   }
34
35   actor actCommunicationCompany role rnactCommunicationCompany cardinality [1..*] {
36
37    attribute mode: etComCompanyMode
38    operation init() : ptBoolean
39
40    input interface inactCommunicationCompany {
41     operation ieRequestCrisisEventLocation(AdtPhoneNumber:dtPhoneNumber) : ptBoolean
42    }
43
44    output interface outactCommunicationCompany {
45     operation oeReceiveCrisisEventLocation(AdtGeoPos:dtGeoPos) : ptBoolean
46    }
47   }
48
```

```
49   actor actAbstractDispatchCoordinator role rnactAbstractDispatchCoordinator cardinality [1..*] {
50
51    operation init() : ptBoolean
52
53    input interface inactAbstractDispatchCoordinator {
54     operation ieReceiveNewCrisisEvent(AdtCrisisID:dtCrisisID, AMessage:dtString) : ptBoolean
55     operation ieReceiveCrisisEventInformation(AdtName:ptString, AetHumanType:etHumanType,
           AdtPhoneNumber:dtPhoneNumber, AdtMapWithPin:dtMapWithPin, AdtComment:dtComment) : ptBoolean
56     operation ieMessageAbstractDispatchCoordinator(AdtComment: dtComment) : ptBoolean
57     operation ieReceiveMap(AdtMapWithPin: dtMapWithPin) : ptBoolean
58    }
59
60    output interface outactAbstractDispatchCoordinator {
61     operation oeGetCrisisEventInformation(AdtGeoPos:dtGeoPos) : ptBoolean
62     operation oeMessage(AdtComment:dtComment) : ptBoolean
63     operation oeUpdateDispatchStatus(AetDispatchStatus:etDispatchStatus): ptBoolean
64     operation oeRefreshMap(AdtGeoPos:dtGeoPos) : ptBoolean
65     operation oeCloseCrisisEvent(): ptBoolean
66    }
67   }
68
69   actor actFiremenCoordinator role rnactFiremenCoordinator cardinality [1..*] extends
         actAbstractDispatchCoordinator {
70
71    operation init() : ptBoolean
72
73    input interface inactFiremenCoordinator {
74    }
75
76    output interface outactFiremenCoordinator {
77     operation oeAddRequestHelp() : ptBoolean
78     operation oeRequestHelp(AetTeamType: etTeamType, ARequestedNumber:ptInteger) : ptBoolean
79    }
80   }
81
82   actor actPoliceCoordinator role rnPoliceCoordinator cardinality [1..*] extends
         actAbstractDispatchCoordinator {
83
84    attribute modeFM: etFiremenMode
85    operation init() : ptBoolean
86
87    input interface inactPoliceCoordinator {
88    }
89
90    output interface outactPoliceCoordinator {
91    }
92   }
93
94   actor actTowServiceCoordinator role rnTowServiceCoordinator cardinality [1..*] extends
         actAbstractDispatchCoordinator {
95
96    operation init() : ptBoolean
97
98    input interface inactTowServiceCoordinator {
99    }
100
101   output interface outactTowServiceCoordinator {
102   }
103  }
104
105 }
106 }
```

Listing B.16: Messir Spec. file environment.msr.

B.17 File ./src-gen/messir-spec/concepts/primarytypes-associations/primarytypes-associations.msr

```
1 /*
```

```
2 * @author Kira
3 * @date Tue Oct 25 23:54:03 CEST 2016
4 */
5
6 package lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.associations {
7
8 import lu.uni.lassy.messir.libraries.calendar
9 import lu.uni.lassy.messir.libraries.math
10 import lu.uni.lassy.messir.libraries.primitives
11 import lu.uni.lassy.messir.libraries.string
12 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.classes
13 import lu.uni.lassy.excalibur.group09.spec.environment
14
15 Concept Model {
16
17  Primary Types {
18
19   association assctCrisisEventctHuman
20    ctCrisisEvent(rnctCrisisEvent)[1..1]
21    ctHuman(rnctHuman)[1..*]
22
23   association assctCrisisEventctMapWithPin
24    ctCrisisEvent(rnctCrisisEvent)[1..1]
25    ctMapWithPin(rnctMapWithPin)[1..1]
26
27   association assDispatchedCoordinatortctMapWithPin
28    ctDispatchedCoordinator(rnctDispatchedCoordinator)[1..1]
29    ctMapWithPin(rnctMapWithPin)[1..1]
30
31    association assactComCompanyctMapWithPin
32   actCommunicationCompany(rnactCommunicationCompany)[1..1]
33   ctMapWithPin(rnctMapWithPin)[1..*]
34
35    association assactCentralCoordinatorctMapWithPin
36   actCentralCoordinator(rnactCentralCoordinator)[1..1]
37   ctMapWithPin(rnctMapWithPin)[1..*]
38
39   association assClassActorDispatchCoordinator
40    ctDispatchedCoordinator(rnctDispatchedCoordinator)[1..1]
41    actAbstractDispatchCoordinator(rnactAbstractDispatchCoordinator)[1..1]
42
43   association assctDispatchedCoordinatorctCrisisEvent
44    ctDispatchedCoordinator(rnctDispatchedCoordinator)[2..*]
45    ctCrisisEvent(rnctCrisisEvent)[1..1]
46
47   association assctCommentctCrisisEvent
48    ctComment(rnctComment)[0..*]
49    ctCrisisEvent(rnctCrisisEvent)[1..1]
50
51   association assctCommentctDispatchedCoordinator
52    ctComment(rnctComment)[0..*]
53    ctDispatchedCoordinator(rnctDispatchedCoordinator)[1..1]
54
55   association assctCommentactCentralCoordinator
56    ctComment(rnctComment)[0..*]
57    actCentralCoordinator(rnactCentralCoordinator)[1..1]
58
59   association assctCrisisEventactCentralCoordinator
60    ctCrisisEvent(ctCrisisEvent)[0..*]
61    actCentralCoordinator(rnactCentralCoordinator)[1..1]
62  }
63 }
64 }
```

Listing B.17: Messir Spec. file primarytypes-associations.msr.

B.18 File ./src-gen/messir-spec/concepts/primarytypes-classes/primarytypes-classes.msr

```
/*
* @author Kira
* @date Tue Oct 25 23:54:03 CEST 2016
*/

package lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.classes {

import lu.uni.lassy.messir.libraries.calendar
import lu.uni.lassy.messir.libraries.math
import lu.uni.lassy.messir.libraries.primitives
import lu.uni.lassy.messir.libraries.string
import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes

import lu.uni.lassy.messir.libraries.primitives

Concept Model {

 Primary Types {

  state class ctState {
   attribute vpStarted: ptBoolean
   attribute nextValueForAlertID:ptInteger
   operation init( ANextValueForAlertID:ptInteger,
       AvpStarted:ptBoolean
   ): ptBoolean
  }

  class ctHuman role rnHuman cardinality [1..*] {
   attribute id: dtPhoneNumber
   attribute name: dtString
   attribute type: etHumanType

   operation init( Aid:dtPhoneNumber,
       Aname:dtString,
       Atype:etHumanType
   ): ptBoolean

  }

  class ctCrisisEvent role rnCrisisEvent cardinality [1..*] {
   attribute id: dtCrisisID
   attribute geoPos: dtGeoPos

   operation init( Aid:dtCrisisID,
       AgeoPos:dtGeoPos
   ): ptBoolean
  }

  class ctComment role rnComment cardinality [0..*] {
   attribute comment: dtComment

   operation init( AComment: dtComment

   ) : ptBoolean
  }

  class ctDispatchedCoordinator role rnDispatchedCoordinator cardinality [1..*] {
   attribute type: etTeamType
   attribute status: etDispatchStatus
   attribute geoPos: dtGeoPos
   attribute isFree: ptBoolean
   attribute mode: etDispatchCoordinatorMode

   operation init( Atype:etTeamType,
       Astatus:etDispatchStatus,
       AgeoPos:dtGeoPos,
       AisFree:ptBoolean,
       Amode:etDispatchCoordinatorMode
   ): ptBoolean
```

```
71   }
72
73   class ctMapWithPin role rnMapWithPin cardinality [1..*] {
74    attribute mapWithPin: dtMapWithPin
75
76    operation init( AmapWithPin:dtMapWithPin
77    ) : ptBoolean
78
79   }
80
81  }
82 }
83 }
```

Listing B.18: Messir Spec. file primarytypes-classes.msr.

B.19 File ./src-gen/messir-spec/concepts/primarytypes-datatypes/primarytypes-datatypes.msr

```
1 /*
2 * @author Kira
3 * @date Tue Oct 25 23:54:03 CEST 2016
4 */
5
6 package lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes {
7
8 import lu.uni.lassy.messir.libraries.calendar
9 import lu.uni.lassy.messir.libraries.math
10 import lu.uni.lassy.messir.libraries.primitives
11 import lu.uni.lassy.messir.libraries.string
12 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
13
14 Concept Model {
15
16  Primary Types {
17
18     datatype dtGeoPos {
19      attribute latitude:dtLatitude
20      attribute longitude:dtLongitude
21      operation is():ptBoolean
22     }
23
24     enum etDispatchStatus {
25      constants["InStation", "InTransit", "Arrived"]
26      operation is():ptBoolean
27     }
28
29     enum etHumanType {
30      constants["Victim", "Witness"]
31      operation is():ptBoolean
32     }
33
34     enum etTeamType {
35      constants["Firemen", "Police", "TowService"]
36      operation is():ptBoolean
37     }
38
39     enum etCentralCoordinatorMode {
40      constants["isInMain", "isInAddEvent"]
41      operation is():ptBoolean
42     }
43
44     enum etDispatchCoordinatorMode {
45      constants["hasBeenAssociatedToACrisisEvent", "isInStandBy", "isInMain", "
         isInMainWithCloseEventUnlocked", "isNull"]
46      operation is():ptBoolean
47     }
48
```

```
49     enum etFiremenMode {
50
51      constants["isInRequestHelp", "isNotInRequestHelp"]
52      operation is():ptBoolean
53
54     }
55
56     enum etComCompanyMode {
57      constants["isInRequest", "isInStandBy"]
58      operation is():ptBoolean
59     }
60
61  }
62 }
63 }
```

Listing B.19: Messir Spec. file primarytypes-datatypes.msr.

B.20 File ./src-gen/messir-spec/concepts/secondarytypes-associations/secondarytypes-associations.msr

```
1 /*
2 * @author Kira
3 * @date Tue Oct 25 23:54:03 CEST 2016
4 */
5
6 package lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.associations {
7
8 import lu.uni.lassy.messir.libraries.calendar
9 import lu.uni.lassy.messir.libraries.math
10 import lu.uni.lassy.messir.libraries.primitives
11 import lu.uni.lassy.messir.libraries.string
12
13 Concept Model {
14
15  Secondary Types {
16
17  }
18 }
19 }
```

Listing B.20: Messir Spec. file secondarytypes-associations.msr.

B.21 File ./src-gen/messir-spec/concepts/secondarytypes-classes/secondarytypes-classes.msr

```
1 /*
2 * @author Kira
3 * @date Tue Oct 25 23:54:03 CEST 2016
4 */
5
6 package lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.classes {
7
8 import lu.uni.lassy.messir.libraries.calendar
9 import lu.uni.lassy.messir.libraries.math
10 import lu.uni.lassy.messir.libraries.primitives
11 import lu.uni.lassy.messir.libraries.string
12
13 Concept Model {
14
15  Secondary Types {
16
17  }
18 }
19 }
```

Listing B.21: Messir Spec. file secondarytypes-classes.msr.

B.22 File ./src-gen/messir-spec/operations/concepts/secondarytypes-datatypes/secondarytypes-datatypes-dtMapWithPin-addPinToMap.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes.operations.datatypes.
      dtMapWithPin.addPinToMap {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
9
10  Operation Model {
11
12   operation: lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes.dtMapWithPin.
        addPinToMap(AGeoPos:dtGeoPos):ptBoolean{
13   // include below the specification information (pre,post or ocl or prolog)
14
15   }
16  }
17 }
```

Listing B.22: Messir Spec. file secondarytypes-datatypes-dtMapWithPin-addPinToMap.msr.

B.23 File ./src-gen/messir-spec/operations/concepts/secondarytypes-datatypes/secondarytypes-datatypes-dtMapWithPin-getPinGeoPos.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes.operations.datatypes.
      dtMapWithPin.getPinGeoPos {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
9
10  Operation Model {
11
12   operation: lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes.dtMapWithPin.
        getPinGeoPos():dtGeoPos{
13   // include below the specification information (pre,post or ocl or prolog)
14
15   }
16  }
17 }
```

Listing B.23: Messir Spec. file secondarytypes-datatypes-dtMapWithPin-getPinGeoPos.msr.

B.24 File ./src-gen/messir-spec/operations/concepts/secondarytypes-datatypes/secondarytypes-datatypes-dtMapWithPin-removeAllPinsFromMap.msr

```
1 package lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes.operations.datatypes.
      dtMapWithPin.removeAllPinsFromMap {
2
3 import lu.uni.lassy.messir.libraries.primitives
4 import lu.uni.lassy.messir.libraries.math
5 import lu.uni.lassy.messir.libraries.string
```

```
6 import lu.uni.lassy.messir.libraries.calendar
7 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
8 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
9
10  Operation Model {
11
12   operation: lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes.dtMapWithPin.
         removeAllPinsFromMap():ptBoolean{
13   // include below the specification information (pre,post or ocl or prolog)
14
15   }
16  }
17 }
```

Listing B.24: Messir Spec. file secondarytypes-datatypes-dtMapWithPin-removeAllPinsFromMap.msr.

B.25 File ./src-gen/messir-spec/concepts/secondarytypes-datatypes/secondarytypes-datatypes.msr

```
1 /*
2 * @author Kira
3 * @date Tue Oct 25 23:54:03 CEST 2016
4 */
5
6 package lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes {
7
8 import lu.uni.lassy.messir.libraries.calendar
9 import lu.uni.lassy.messir.libraries.math
10 import lu.uni.lassy.messir.libraries.primitives
11 import lu.uni.lassy.messir.libraries.string
12 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
13
14 Concept Model {
15
16  Secondary Types {
17
18   datatype dtPhoneNumber extends dtString {
19     operation is() : ptBoolean
20    }
21
22   datatype dtAddress extends dtString {
23     operation is() : ptBoolean
24    }
25
26   datatype dtCrisisID extends dtInteger {
27     operation is() : ptBoolean
28    }
29
30   datatype dtLongitude extends dtReal {
31       operation is() : ptBoolean
32      }
33
34   datatype dtLatitude extends dtReal {
35       operation is() : ptBoolean
36    }
37
38   datatype dtImage extends dtString {
39       operation is() : ptBoolean
40    }
41
42   datatype dtMapWithPin extends dtImage {
43     operation getPinGeoPos() : dtGeoPos
44     operation removeAllPinsFromMap() : ptBoolean
45     operation addPinToMap(AGeoPos:dtGeoPos) : ptBoolean
46     operation distance(AGeoPosOne:dtGeoPos, AGeoPosTwo:dtGeoPos) : dtInteger
47     operation is() : ptBoolean
48    }
```

```
49
50   datatype dtComment extends dtString {
51     operation is() : ptBoolean
52    }
53  }
54
55 }
56 }
```

Listing B.25: Messir Spec. file secondarytypes-datatypes.msr.

B.26 File ./src-gen/messir-spec/tests/tests.msr

```
1 /*
2 * @author Kira
3 * @date Tue Oct 25 23:54:03 CEST 2016
4 */
5
6 package lu.uni.lassy.excalibur.group09.spec.tests {
7
8 import lu.uni.lassy.messir.libraries.calendar
9 import lu.uni.lassy.messir.libraries.math
10 import lu.uni.lassy.messir.libraries.primitives
11 import lu.uni.lassy.messir.libraries.string
12
13  Test Model {
14
15  }
16
17 }
```

Listing B.26: Messir Spec. file tests.msr.

B.27 File ./src-gen/messir-spec/usecases/usecaseinstance-ugCreateNewCrisisEvent-uciugCreateNewCrisisEvent.msr

```
1 package usecases.uciugCreateNewCrisisEvent {
2  import lu.uni.lassy.excalibur.group09.spec.usecases
3  import lu.uni.lassy.excalibur.group09.spec.usecases
4  import lu.uni.lassy.excalibur.group09.spec.environment
5  import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
6
7  Use Case Model {
8   use case instance uciugCreateNewCrisiEvent : ugCreateNewCrisisEvent {
9    actors {
10    Camille : actCentralCoordinator
11    Orange : actCommunicationCompany
12    Fabio : actFiremenCoordinator
13    Ted : actTowServiceCoordinator
14   }
15
16   use case steps {
17    Camille executed instanceof subfunction oeAddNewCrisisEvent() {
18     }
19
20    Camille executed instanceof subfunction oeRequestCrisisEventLocation("AdtPhoneNumber=691 12 34
          56") {
21     ieRequestCrisisEventLocation("691 12 34 56") returned to Orange
22    }
23
24    Orange executed instanceof subfunction oeReceiveCrisisEventLocation("Latitude=87.20, Longitude
          =20.25") {
25     ieReceiveCrisisEventLocation("A string for the image with the pin") returned to Camille
26    }
27
28    Camille executed instanceof subfunction oeMovePinOnMap("Latitude=75.08, Longitude=23.03") {
29     ieReceiveCrisisEventLocation("A string for the image with the pin") returned to Camille
```

```
30     }
31
32     Camille executed instanceof subfunction oeCreateNewCrisisEvent("AdtName=Walter", "AenHumanType=
           Witness", "AdtPhoneNumber=691 12 34 56", "A string for the image with the pins", "The
           Witness can't stay at the accident's location for long.") {
33      ieReceiveNewCrisisEvent("1","You have received a new Crisis Event Dispatch Order!") returned to
            Fabio
34      ieReceiveNewCrisisEvent("1","You have received a new Crisis Event Dispatch Order!") returned to
            Ted
35     }
36
37    }
38
39   }
40  }
41 }
```

Listing B.27: Messir Spec. file usecaseinstance-ugCreateNewCrisisEvent-uciugCreateNewCrisisEvent.msr.

B.28 File ./src-gen/messir-spec/usecases/usecaseinstance-ugGlobalDispatchManagement-uciugGlobalDispatchManagement.msr

```
1 package usecases.uciugGlobalDispatchManagement {
2  import lu.uni.lassy.excalibur.group09.spec.usecases
3  import lu.uni.lassy.excalibur.group09.spec.usecases
4  import lu.uni.lassy.excalibur.group09.spec.environment
5  import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
6
7  Use Case Model {
8   use case instance uciugGlobalDispatchManagement : ugGlobalDispatchManagement {
9    actors {
10    Camille :  actCentralCoordinator
11    Fabio  :  actFiremenCoordinator
12    Ted  :  actTowServiceCoordinator
13    Polo  :  actPoliceCoordinator
14   }
15   use case steps {
16     Fabio executed instanceof subfunction oeGetCrisisEventInformation("Latitude=60.21, Longitude
           =40.23") {
17      ieReceiveCrisisEventInformation("1","Walter","Witness","691 12 34 56","A string for the image
            with the pins","The Witness can't stay at the accident's location for long.") returned to
            Fabio
18     }
19
20     Fabio executed instanceof subfunction oeUpdateDispatchStatus("AenDispatchStatus=InTransit") {
21      ieMessageAbstractDispatchCoordinator("Dispatch Status Updated.") returned to Fabio
22     }
23
24     Ted executed instanceof subfunction oeGetCrisisEventInformation("Latitude=81.45, Longitude
           =52.87") {
25      ieReceiveCrisisEventInformation("1","Walter","Witness","691 12 34 56","A string for the image
            with the pins","The Witness can't stay at the accident's location for long.") returned to
            Ted
26     }
27
28     Ted executed instanceof subfunction oeRefreshMap("Latitude=80.57, Longitude=50.23") {
29      ieReceiveMap("A string for the image with the pins") returned to Ted
30     }
31
32     Ted executed instanceof subfunction oeMessage("AdtComment=I will arrive in 30 minutes") {
33      ieMessageCentralCoordinator("I will arrive in 30 minutes") returned to Camille
34      ieMessageAbstractDispatchCoordinator("I will arrive in 30 minutes") returned to Fabio
35      ieMessageAbstractDispatchCoordinator("I will arrive in 30 minutes") returned to Ted
36     }
37
38     Ted executed instanceof subfunction oeUpdateDispatchStatus("AenDispatchStatus=InTransit") {
39      ieMessageAbstractDispatchCoordinator("Dispatch Status Updated.") returned to Ted
```

```
40      }
41
42      Fabio executed instanceof subfunction oeUpdateDispatchStatus("AenDispatchStatus=Arrived") {
43       ieMessageAbstractDispatchCoordinator("Dispatch Status Updated.") returned to Fabio
44      }
45
46      Fabio executed instanceof subfunction oeAddRequestHelp() {
47
48      }
49
50      Fabio executed instanceof subfunction oeRequestHelp("AenTeamType=Police", "RequestedNumber=1")
            {
51       ieReceiveNewCrisisEvent("1","You have received a new Crisis Event Dispatch Order!") returned
             to Polo
52      }
53
54      Polo executed instanceof subfunction oeGetCrisisEventInformation("Latitude=44.57, Longitude
            =63.23") {
55       ieReceiveCrisisEventInformation("1","Walter","Witness","691123456","A string for the image
             with the pins","The Witness can't stay at the accident's location for long.") returned to
             Polo
56      }
57
58      Polo executed instanceof subfunction oeUpdateDispatchStatus("AenDispatchStatus=InTransit") {
59       ieMessageAbstractDispatchCoordinator("Dispatch Status Updated.") returned to Polo
60      }
61
62      Ted executed instanceof subfunction oeUpdateDispatchStatus("AenDispatchStatus=Arrived") {
63       ieMessageAbstractDispatchCoordinator("Dispatch Status Updated.") returned to Ted
64      }
65
66      Polo executed instanceof subfunction oeUpdateDispatchStatus("AenDispatchStatus=Arrived") {
67       ieMessageAbstractDispatchCoordinator("Dispatch Status Updated.") returned to Polo
68      }
69
70      Fabio executed instanceof subfunction oeCloseCrisisEvent() {
71      }
72
73      Ted executed instanceof subfunction oeCloseCrisisEvent() {
74      }
75
76      Polo executed instanceof subfunction oeCloseCrisisEvent() {
77      }
78
79    }
80   }
81  }
82 }
```

Listing B.28: Messir Spec. file usecaseinstance-ugGlobalDispatchManagement-uciugGlobalDispatchManagement.msr.

B.29 File ./src-gen/messir-spec/usecases/usecases.msr

```
1 /*
2 * @author Kira
3 * @date Tue Oct 25 23:54:03 CEST 2016
4 */
5
6 package lu.uni.lassy.excalibur.group09.spec.usecases {
7
8 import lu.uni.lassy.messir.libraries.calendar
9 import lu.uni.lassy.messir.libraries.math
10 import lu.uni.lassy.messir.libraries.primitives
11 import lu.uni.lassy.messir.libraries.string
12 import lu.uni.lassy.excalibur.group09.spec.environment
13 import lu.uni.lassy.excalibur.group09.spec.concepts.primarytypes.datatypes
14 import lu.uni.lassy.excalibur.group09.spec.concepts.secondarytypes.datatypes
15
```

```
16 Use Case Model {
17
18  use case system summary suGlobalManagementOfEvent() {
19    actor actCentralCoordinator[primary, active]
20    actor actCommunicationCompany[secondary, active]
21    actor actFiremenCoordinator[secondary,active]
22    actor actTowServiceCoordinator[secondary,active]
23
24    reuse ugCreateNewCrisisEvent[1..*]
25    reuse ugGlobalDispatchManagement[1..*]
26
27    step a: actCentralCoordinator executes ugCreateNewCrisisEvent
28    step b: actFiremenCoordinator executes ugGlobalDispatchManagement
29    step c: actTowServiceCoordinator executes ugGlobalDispatchManagement
30
31    ordering constraint "step (a) must be executed before step (b) or step (c)"
32    ordering constraint "step (b) XOR step (c)"
33
34  }
35
36   use case system usergoal ugCreateNewCrisisEvent() {
37    actor actCentralCoordinator[primary, active]
38    actor actCommunicationCompany[secondary, active]
39    actor actFiremenCoordinator[secondary, passive]
40    actor actTowServiceCoordinator[secondary, passive]
41
42    reuse oeAddNewCrisisEvent[1..*]
43    reuse oeRequestCrisisEventLocation[1..*]
44    reuse oeReceiveCrisisEventLocation[1..*]
45    reuse oeCreateNewCrisisEvent[1..*]
46    reuse oeMovePinOnMap[0..*]
47
48    step a: actCentralCoordinator executes oeAddNewCrisisEvent
49    step b: actCentralCoordinator executes oeRequestCrisisEventLocation
50    step c: actCommunicationCompany executes oeReceiveCrisisEventLocation
51    step d: actCentralCoordinator executes oeMovePinOnMap
52    step e: actCentralCoordinator executes oeCreateNewCrisisEvent
53
54    ordering constraint "step (a) must be executed first"
55    ordering constraint "if step (c) then previously step (b)"
56    ordering constraint "if step (d) then previously step (c)"
57    ordering constraint "step (e) executed as last"
58    ordering constraint "step (a), (b), (c), (e) must be executed at least once"
59  }
60
61  use case system usergoal ugGlobalDispatchManagement() {
62    actor actFiremenCoordinator[primary, active]
63    actor actTowServiceCoordinator[primary,active]
64    actor actCentralCoordinator[secondary,passive]
65    actor actPoliceCoordinator[secondary,active]
66
67    reuse oeGetCrisisEventInformation[2..*]
68    reuse oeUpdateDispatchStatus[4..*]
69    reuse oeRefreshMap[0..*]
70    reuse oeMessage[0..*]
71    reuse oeRequestHelp[0..*]
72    reuse oeCloseCrisisEvent[2..*]
73    reuse oeAddRequestHelp[0..*]
74
75    step a: actFiremenCoordinator executes oeGetCrisisEventInformation
76    step b: actFiremenCoordinator executes oeUpdateDispatchStatus
77    step c: actTowServiceCoordinator executes oeGetCrisisEventInformation
78    step d: actTowServiceCoordinator executes oeUpdateDispatchStatus
79    step e: actTowServiceCoordinator executes oeRefreshMap
80    step f: actTowServiceCoordinator executes oeMessage
81    step g: actFiremenCoordinator executes oeAddRequestHelp
82    step h: actFiremenCoordinator executes oeRequestHelp
83    step i: actPoliceCoordinator executes oeGetCrisisEventInformation
84    step j: actPoliceCoordinator executes oeUpdateDispatchStatus
85    step k: actFiremenCoordinator executes oeCloseCrisisEvent
```

```
86    step l: actTowServiceCoordinator executes oeCloseCrisisEvent
87    step m: actPoliceCoordinator executes oeCloseCrisisEvent
88
89    ordering constraint "if step (b),(d),(j) then previously step (a),(c),(i) respectively"
90    ordering constraint "if step (k),(l),(m) then previously step (b),(d),(j) at least two times
          respectively"
91    ordering constraint "step (h) can only be executed if step (g) has at least been executed once
          previously"
92    ordering constraint "if step (i) then previously step (h)"
93
94  }
95
96  use case system subfunction oeAddNewCrisisEvent() {
97   actor actCentralCoordinator[primary,active]
98  }
99
100 use case system subfunction oeRequestCrisisEventLocation(AdtPhoneNumber:dtPhoneNumber) {
101  actor actCentralCoordinator[primary,active]
102  actor actCommunicationCompany[secondary, passive]
103  returned messages{
104   ieRequestCrisisEventLocation(AdtPhoneNumber) returned to actCommunicationCompany //Slide 208..
105  }
106 }
107
108 use case system subfunction oeReceiveCrisisEventLocation(AdtGeoPos:dtGeoPos) {
109  actor actCommunicationCompany[primary, active]
110  actor actCentralCoordinator[secondary, passive]
111  returned messages{
112   ieReceiveCrisisEventLocation(AdtMapWithPin) returned to actCentralCoordinator
113  }
114 }
115
116 use case system subfunction oeMovePinOnMap() {
117  actor actCentralCoordinator[primary,active]
118  returned messages{
119   ieReceiveCrisisEventLocation(AdtMapWithPin) returned to actCentralCoordinator
120  }
121 }
122
123 use case system subfunction oeCreateNewCrisisEvent(AName:dtString, AetHumanType:etHumanType,
        AdtPhoneNumber:dtPhoneNumber, AdtMapWithPin:dtMapWithPin, AdtComment:dtComment) {
124  actor actCentralCoordinator[primary,active]
125  actor actAbstractDispatchCoordinator[secondary,passive]
126  returned messages{
127   ieReceiveNewCrisisEvent(AdtCrisisID, AdtMessage) returned to actAbstractDispatchCoordinator
128  }
129 }
130
131 use case system subfunction oeGetCrisisEventInformation(AdtGeoPos:dtGeoPos) {
132  actor actAbstractDispatchCoordinator[primary,active]
133  returned messages{
134   ieReceiveCrisisEventInformation (AdtCrisisID, AdtName, AetHumanType, AdtPhoneNumber,
          AdtMapWithPin, AdtComment) returned to actAbstractDispatchCoordinator
135  }
136 }
137
138 use case system subfunction oeMessage(AdtComment:dtComment) {
139  actor actAbstractDispatchCoordinator[primary,active]
140  actor actCentralCoordinator[secondary, passive]
141  actor actAbstractDispatchCoordinator[secondary, multiple]
142  returned messages{
143   ieMessageAbstractDispatchCoordinator(AdtComment) returned to actAbstractDispatchCoordinator
144   ieMessageCentralCoordinator(AdtComment) returned to actCentralCoordinator
145  }
146 }
147
148 use case system subfunction oeUpdateDispatchStatus(AetDispatchStatus:etDispatchStatus){
149  actor actAbstractDispatchCoordinator[primary,active]
150  returned messages{
151   ieMessageAbstractDispatchCoordinator(AdtComment) returned to actAbstractDispatchCoordinator
```

```
152    }
153   }
154
155   use case system subfunction oeRefreshMap(AdtGeoPos:dtGeoPos){
156    actor actAbstractDispatchCoordinator[primary,active]
157    returned messages{
158     ieReceiveMap(AdtMapWithPin) returned to actAbstractDispatchCoordinator
159    }
160   }
161
162   use case system subfunction oeAddRequestHelp() {
163    actor actFiremenCoordinator[primary,active]
164   }
165
166   use case system subfunction oeRequestHelp(AetTeamType: etTeamType, RequestedNumber:ptInteger) {
167    actor actFiremenCoordinator[primary,active]
168    actor actAbstractDispatchCoordinator[secondary,passive]
169    returned messages{
170     ieReceiveNewCrisisEvent(AdtCrisisID, AdtMessage) returned to actAbstractDispatchCoordinator
171    }
172   }
173
174   use case system subfunction oeCloseCrisisEvent() {
175    actor actAbstractDispatchCoordinator[primary,active]
176   }
177
178  }
179
180 }
```

Listing B.29: Messir Spec. file usecases.msr.

Bibliography

[1] Guelfi, N.: Messir: A Scientific Method for the Software Engineer. to be published (2017)

[2] Armour, F., Miller, G.: Advanced Use Case Modeling: Software Systems. Addison-Wesley (2001)