Manual

[ATM system implementation]

Marcos Salgueiro Balsa

marcos.salgueiro@gmail.com

Table of Contents

General View

This emulated operational banking system is composed of three independent modules. Each one of them covers the required functionality by each intervening agent. They are: cash machines, consortium and banks.

The consortium and the banks are fault tolerant. This is implemented using mysql as a means to save session info in case of unexpected failure.

Transactions are partially supported at application level and fully supported DB level as expected.

Additional features:

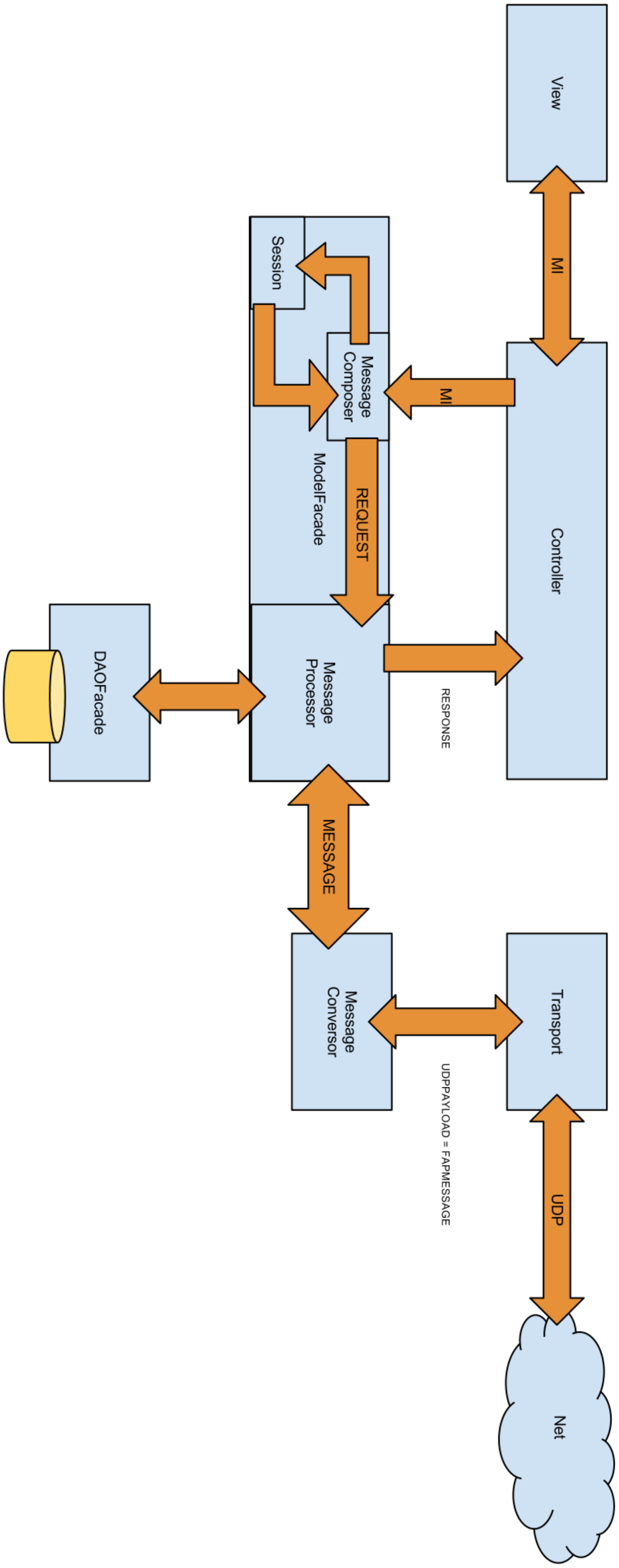
- 5 second timeout to support control message lose. The interface includes a way to emulate link failures.
- Message processing queue to both sockets in consortium and banks.
- Ability to inhibit message reception and emission in all three agents.
- The consortium is able to manage operation requests from multiple cash machines to multiple banks.

General diagram

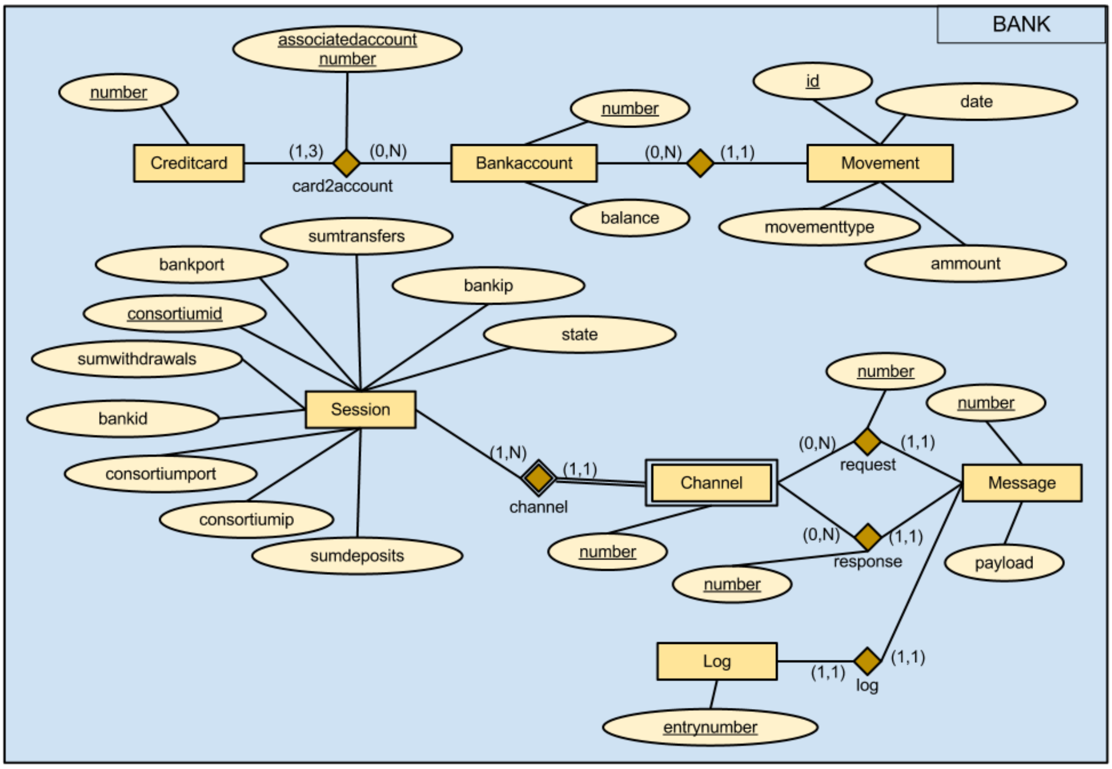
MVC pattern has been used to implement all behavior.

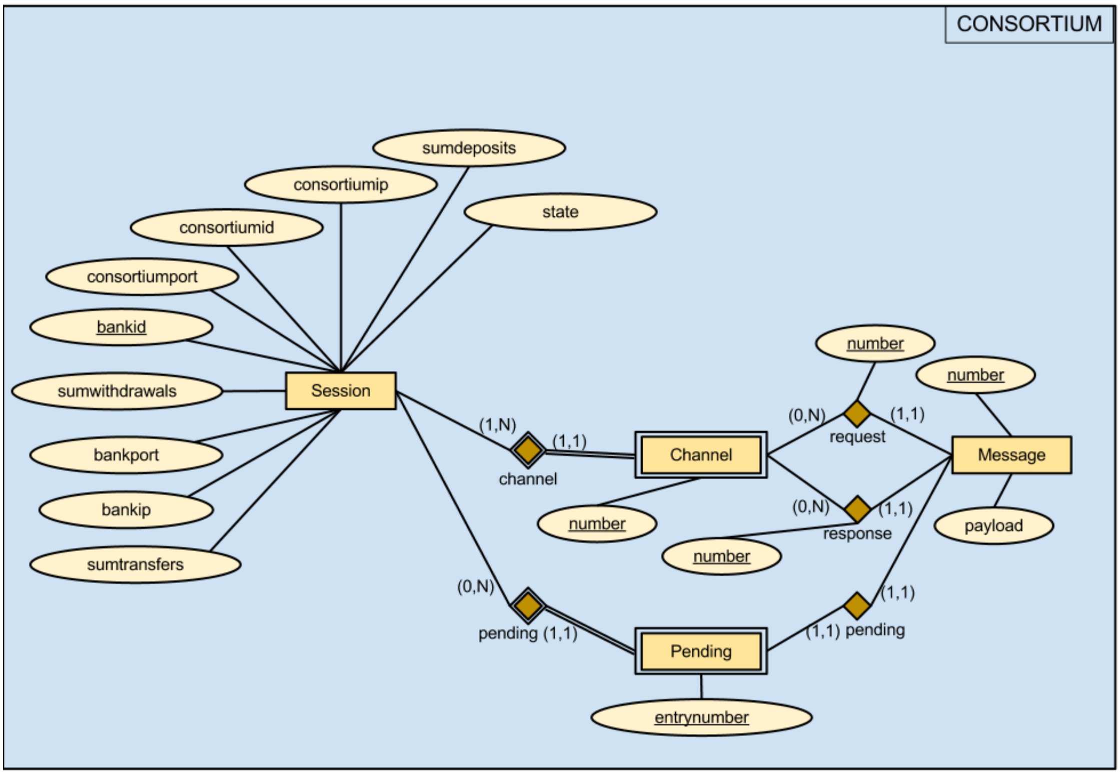
The Message Composer creates data or control request messages containing the required fields according to the kind of information interchange between agents. This newly created message is forwarded to the Message Processor that chooses the appropriate action: the one linked to the message.

An example recovery request message processing is show in the sequence diagram section, where the former interaction is made clearer.


View
MI
Controller
MI
Message Composer
Session
ModelFacade
REQUEST
Message Processor
RESPONSE
DAOFacade
MESSAGE
Message Conversor
Transport
UDPPAYLOAD = FAPMESSAGE
UDP
Net

Entity-Relationship Model





Relational Model

Primary keys appear underlined and foreign keys in italics.

Bank

Creditcard (number)

Bankaccount (number, balance)

Card2account (creditcardnumber, associatedaccountnumber, *realaccountnumber*)

Movement (id, movementtype, ammount, date, *accountnumber*)

Session (consortiumid, consortiumip, consortiumport, bankid, bankip, bankport, sumwithdrawals, sumdeposits, sumtransfers, numberofchannels, state)

ChannelMessageRequest (*consortiumid*, channelnumber, requestnumber, messageobjectpayload)

ChannelMessageResponse (*consortiumid*, channelnumber, responsenumber, messageobjectpayload)

LogMessage (entrynumber, messageobjectpayload)

Consortium

Session (bankid, bankip, bankport, consortiumid, consortiumip, consortiumport, sumwithdrawals, sumdeposits, sumtransfers, numberofchannels, state)

ChannelMessageRequest (*bankid*, channelnumber, requestnumber, messageobjectpayload)

ChannelMessageResponse (*bankid*, channelnumber, responsenumber, messageobjectpayload)

PendingMessage (*bankid*, pendingnumber, receivedtime, messageobjectpayload)

Table creation scripts

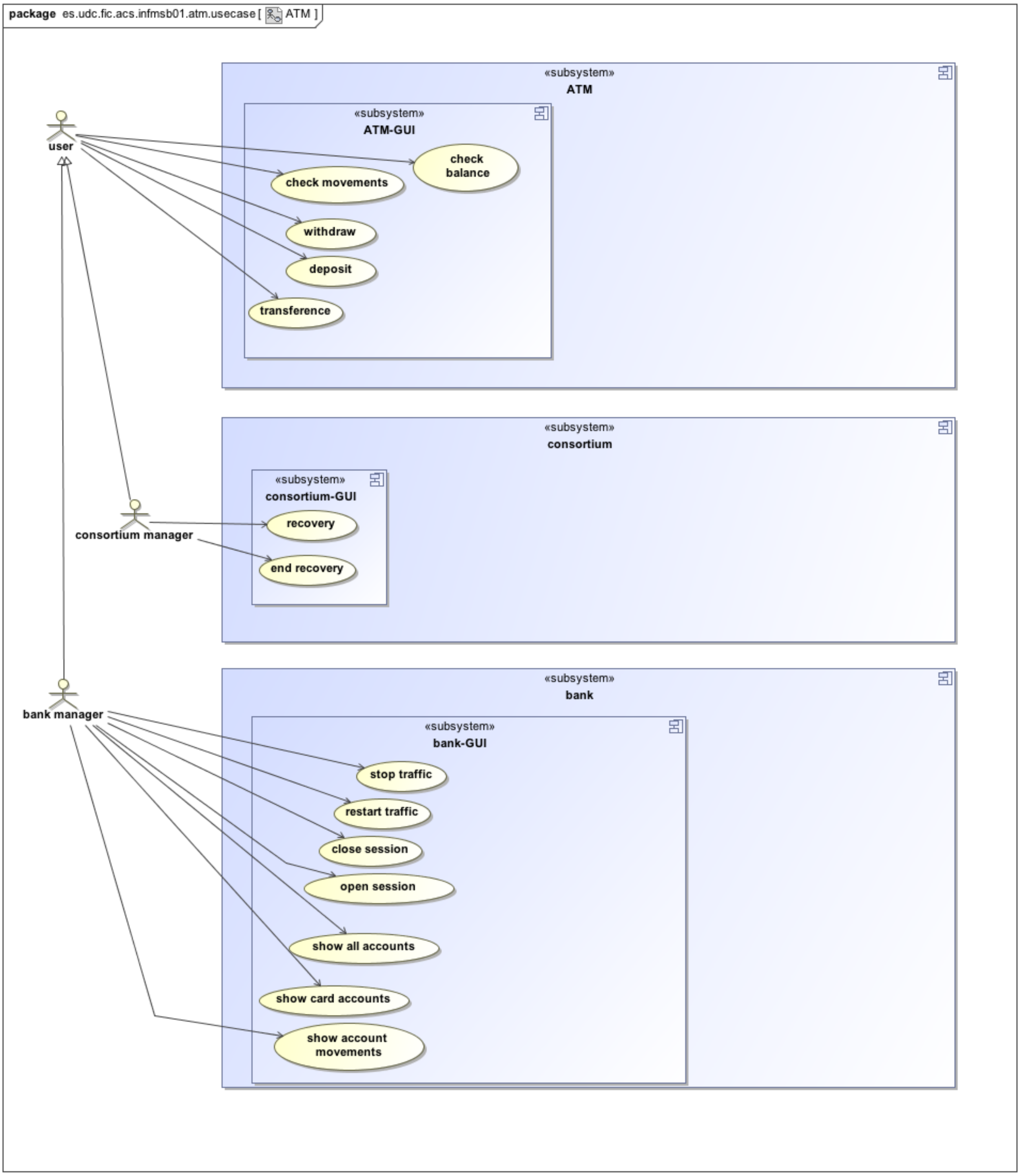
Consortium

```
-- CONSORTIUM DB
-- marcos.salgueiro@gmail.com
create database consortiumdb;
use consortiumdb;

create table session(
bankid varchar(8),
bankip varchar(15),
bankport numeric(5),
consortiumid varchar(8) not null,
consortiumip varchar(15), -- aaa.bbb.ccc.ddd
consortiumport numeric(5),
sumwithdrawals numeric(10),
sumdeposits numeric(10),
sumtransfers numeric(10),
numberofchannels numeric(2),
state numeric(2),
constraint pk_session primary key (bankid)
);

create table channelmessagerequest(
bankid varchar(8),
channelnumber numeric(2) not null,
requestnumber numeric(5) not null,
messageobjectpayload blob not null,
constraint pk_channelmessagerequest primary key (bankid, channelnumber, requestnumber),
constraint fk_chnnlmssgrqst_sessn foreign key (bankid) references session(bankid)
);

create table channelmessageresponse(
bankid varchar(8),
channelnumber numeric(2) not null,
responsenumber numeric(5) not null,
messageobjectpayload blob not null,
constraint pk_channelmessageresponse primary key (bankid, channelnumber,
responsenumber),
constraint fk_chnnlmssgrspns_sessn foreign key (bankid) references session(bankid)
);

create table pendingmessage(
bankid varchar(8),
pendingnumber numeric(4) not null,
receivedtime timestamp default now(),
messageobjectpayload blob not null,
constraint pk_pendingmessage primary key (bankid, pendingnumber),
constraint fk_pendingmssg_sessn foreign key (bankid) references session(bankid)
);
```

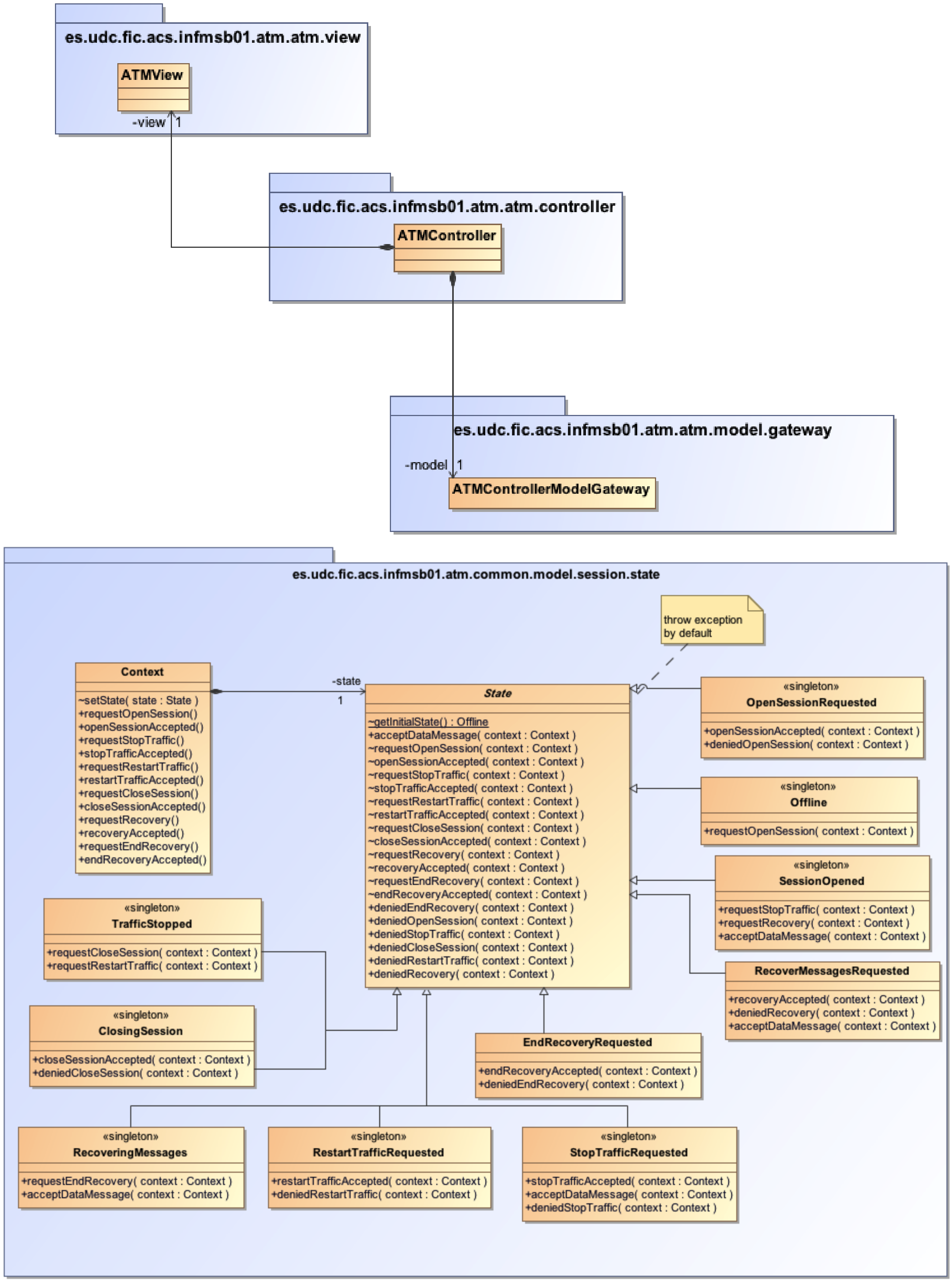
Bank

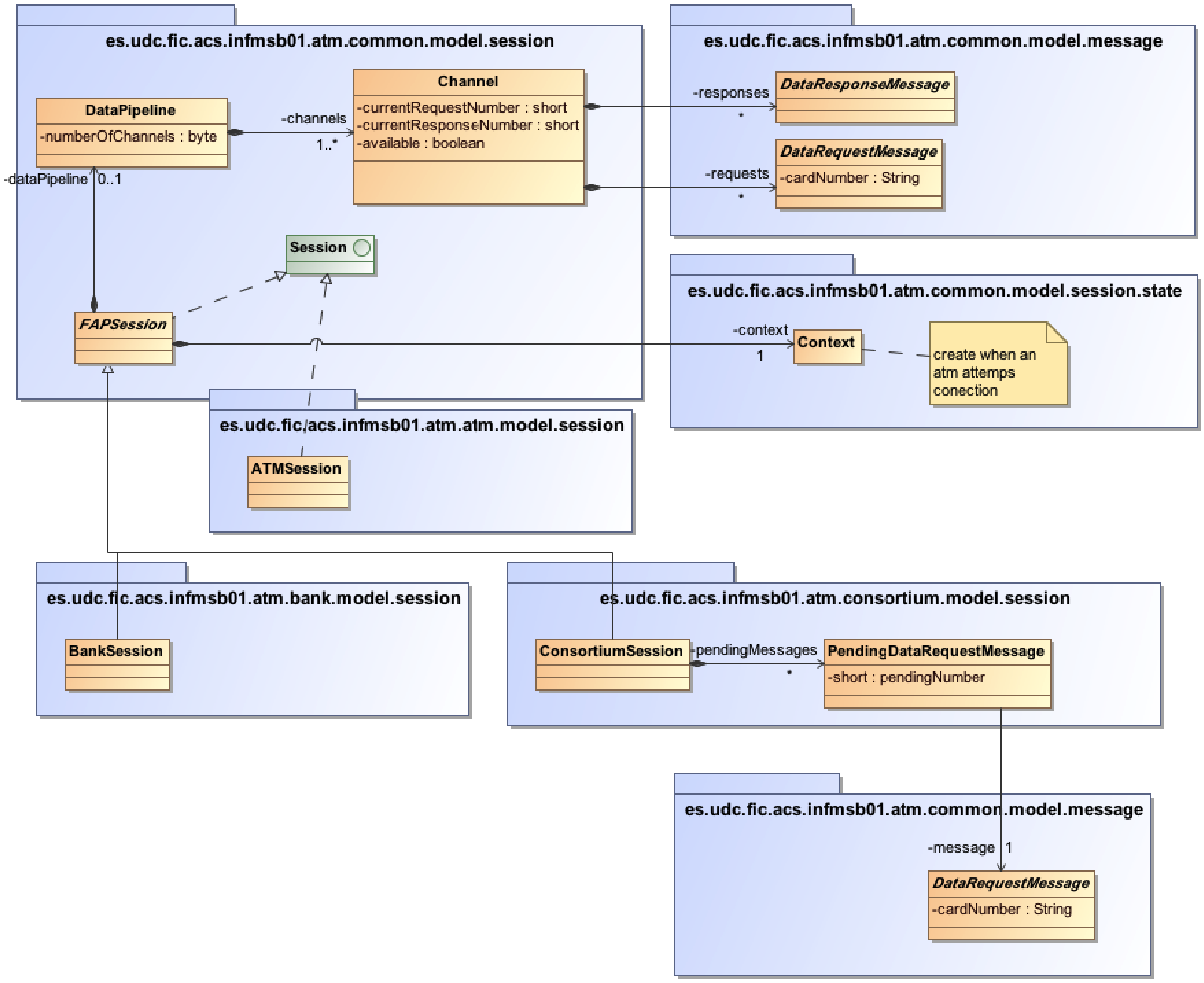
```
-- BANK DB --
-- marcos.salgueiro@gmail.com
create database bankdb;
use bankdb;

create table creditcard(
number varchar(20),
constraint pk_creditcard primary key (number)
);

create table bankaccount(
number varchar(20),
balance numeric(10,2) not null,
constraint pk_bankaccount primary key (number)
);

create table card2account(
creditcardnumber varchar(20),
associatedaccountnumber numeric(1),
realaccountnumber varchar(20),
constraint pk_card2account primary key (creditcardnumber, associatedaccountnumber),
constraint fk_card2account_card foreign key (creditcardnumber) references
creditcard(number),
constraint fk_card2account_bank foreign key (realaccountnumber) references
bankaccount(number)
);

create table movement(
id int auto_increment,
movementtype numeric(2) not null,
ammount numeric(8,2) not null,
date timestamp default now() not null,
accountnumber varchar(20),
constraint pk_movement primary key (id),
constraint fk_movement foreign key (accountnumber) references bankaccount(number)
);

create table session(
consortiumid varchar(8),
consortiumip varchar(15), -- aaa.bbb.ccc.ddd
consortiumport numeric(5),
bankid varchar(8) not null,
bankip varchar(15),
bankport numeric(5),
sumwithdrawals numeric(10),
sumdeposits numeric(10),
sumtransfers numeric(10),
numberofchannels numeric(2),
state numeric(2),
constraint pk_session primary key (consortiumid)
);

create table channelmessagerequest(
consortiumid varchar(8),
channelnumber numeric(2) not null,
requestnumber numeric(5) not null,
messageobjectpayload blob not null,
constraint pk_channelmessagerequest primary key (consortiumid, channelnumber,
requestnumber),
constraint fk_chnnlmssgrqst_sessn foreign key (consortiumid) references
session(consortiumid)
);

create table channelmessageresponse(
consortiumid varchar(8),
channelnumber numeric(2) not null,
responsenumber numeric(5) not null,
messageobjectpayload blob not null,
constraint pk_channelmessageresponse primary key (consortiumid, channelnumber,
responsenumber),
constraint fk_chnnlmssgrspns_sessn foreign key (consortiumid) references
session(consortiumid)
);

create table logmessage(
entrynumber int auto_increment,
messageobjectpayload blob,
constraint pk_logmessage primary key (entrynumber)
);
```

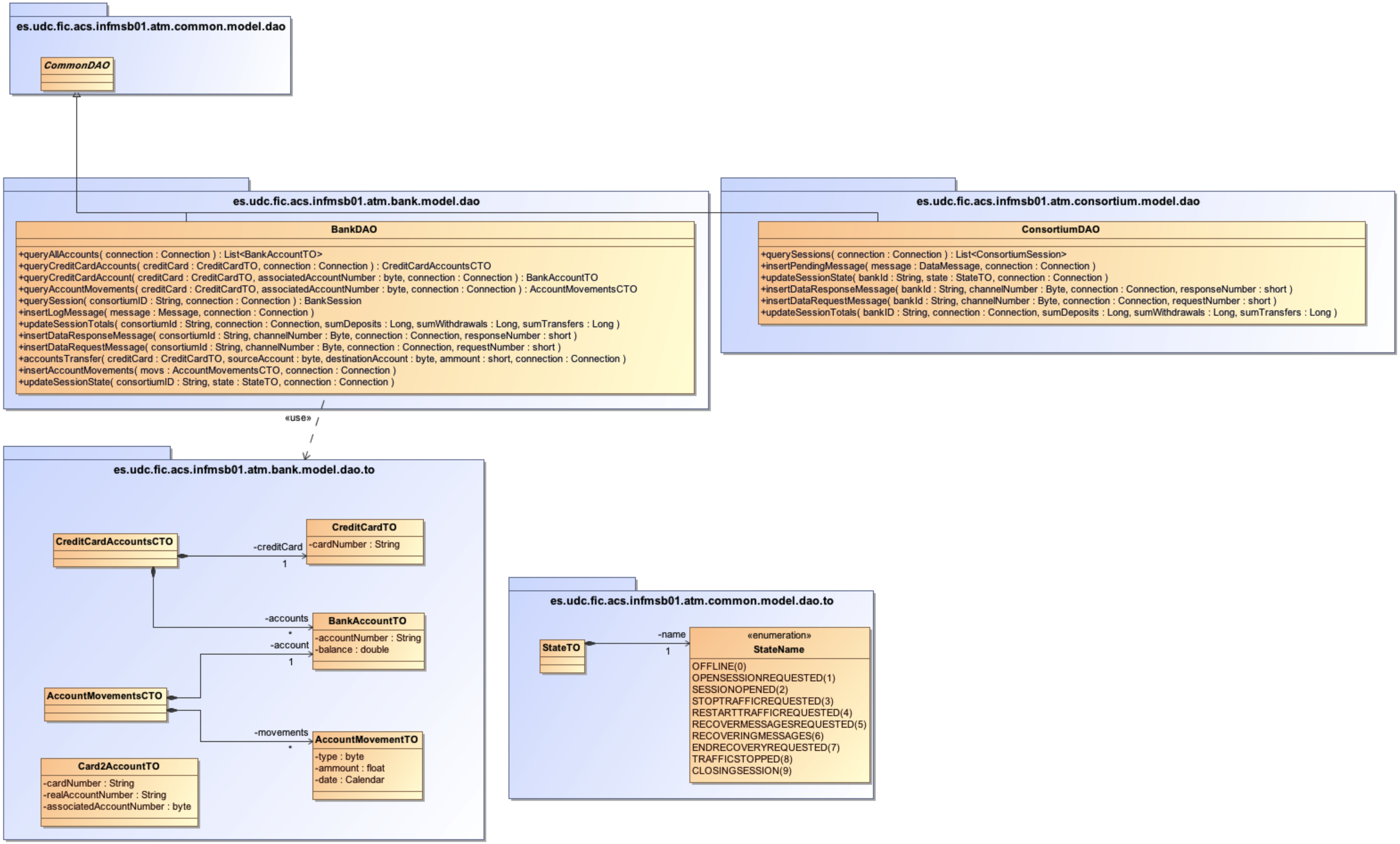
Use Cases Diagram

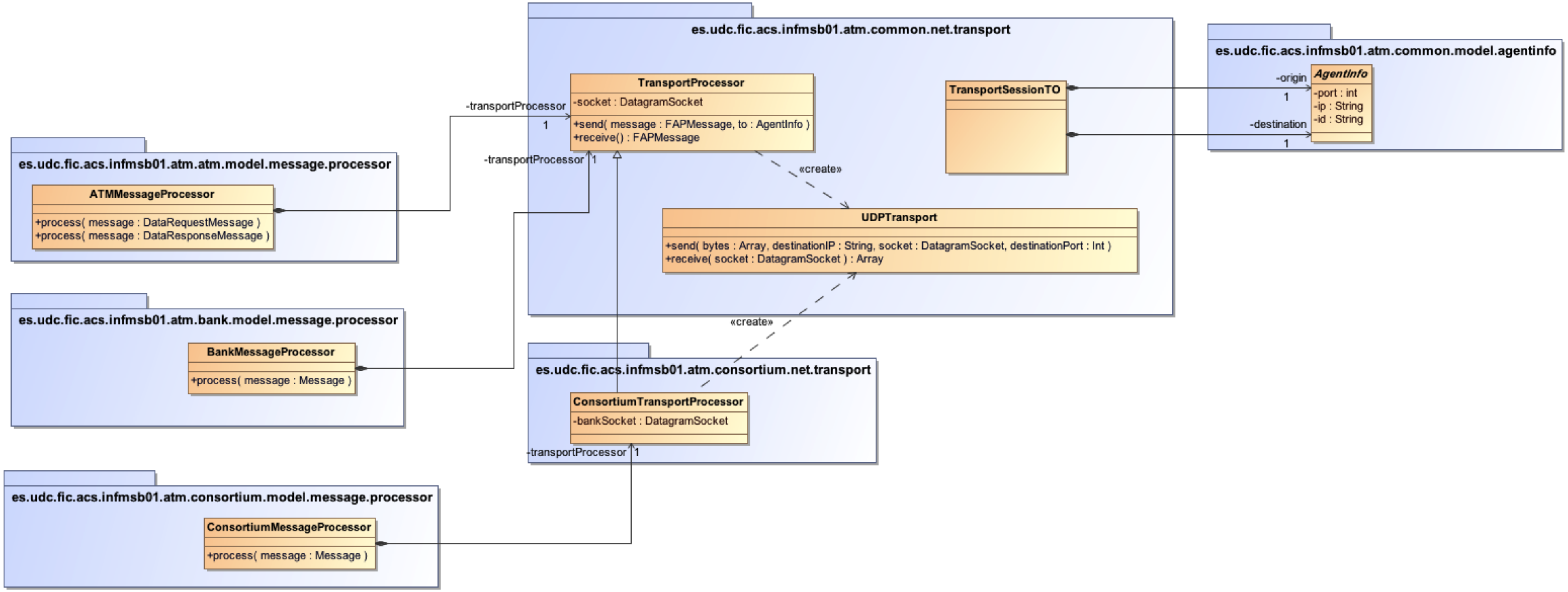


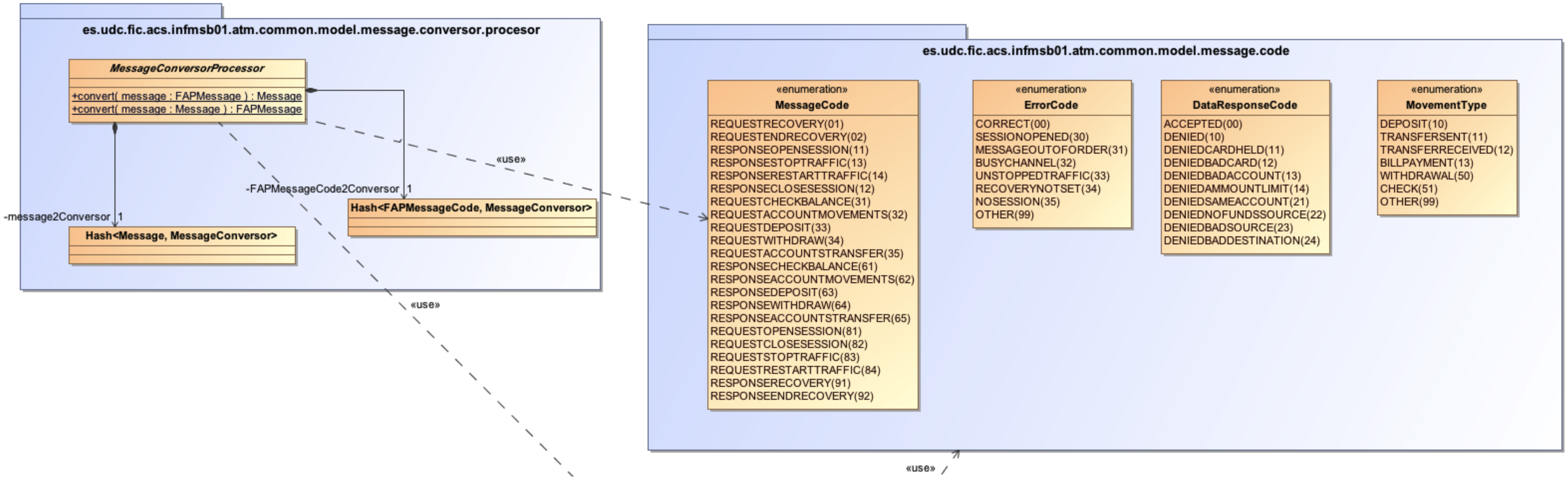
Class Diagrams

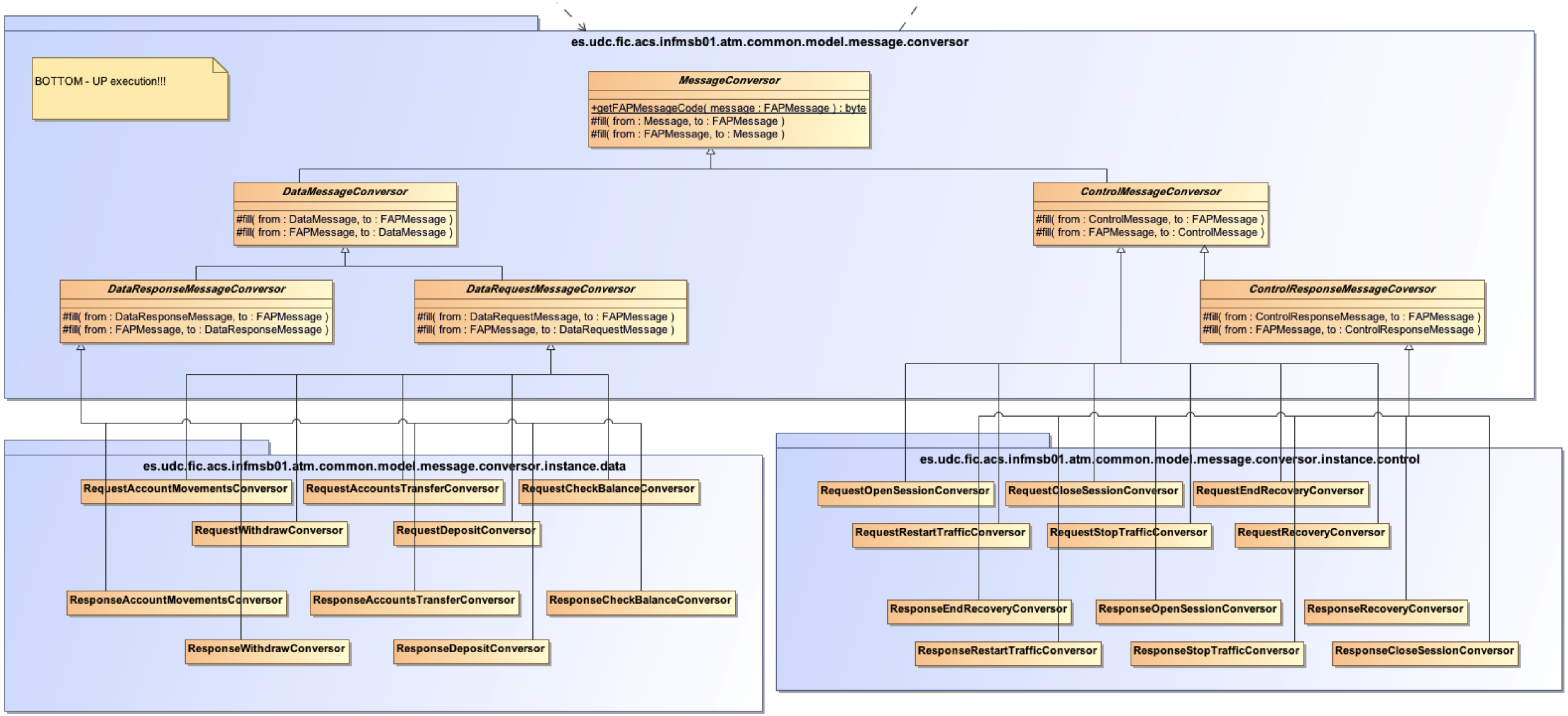

es.udc.fic.acs.infmsb01.atm.atm.view
ATMView
-view
1
es.udc.fic.acs.infmsb01.atm.atm.controller
ATMController
es.udc.fic.acs.infmsb01.atm.atm.model.gateway
-model
1
ATMControllerModelGateway
es.udc.fic.acs.infmsb01.atm.common.model.session.state
throw exception by default
Context
~setState(state : State)
+requestOpenSession()
+openSessionAccepted()
+requestStopTraffic()
+stopTrafficAccepted()
+requestRestartTraffic()
+restartTrafficAccepted()
+requestCloseSession()
+closeSessionAccepted()
+requestRecovery()
+recoveryAccepted()
+requestEndRecovery()
+endRecoveryAccepted()
-state
1
State
~getInitialState() : Offline
+acceptDataMessage(context : Context)
~requestOpenSession(context : Context)
~openSessionAccepted(context : Context)
~requestStopTraffic(context : Context)
~stopTrafficAccepted(context : Context)
~requestRestartTraffic(context : Context)
~restartTrafficAccepted(context : Context)
~requestCloseSession(context : Context)
~closeSessionAccepted(context : Context)
~requestRecovery(context : Context)
~recoveryAccepted(context : Context)
~requestEndRecovery(context : Context)
~endRecoveryAccepted(context : Context)
+deniedEndRecovery(context : Context)
+deniedOpenSession(context : Context)
+deniedStopTraffic(context : Context)
+deniedCloseSession(context : Context)
+deniedRestartTraffic(context : Context)
+deniedRecovery(context : Context)
«singleton»
OpenSessionRequested
+openSessionAccepted(context : Context)
+deniedOpenSession(context : Context)
«singleton»
Offline
+requestOpenSession(context : Context)
«singleton»
SessionOpened
+requestStopTraffic(context : Context)
+requestRecovery(context : Context)
+acceptDataMessage(context : Context)
RecoverMessagesRequested
+recoveryAccepted(context : Context)
+deniedRecovery(context : Context)
+acceptDataMessage(context : Context)
«singleton»
TrafficStopped
+requestCloseSession(context : Context)
+requestRestartTraffic(context : Context)
«singleton»
ClosingSession
+closeSessionAccepted(context : Context)
+deniedCloseSession(context : Context)
EndRecoveryRequested
+endRecoveryAccepted(context : Context)
+deniedEndRecovery(context : Context)
«singleton»
RecoveringMessages
+requestEndRecovery(context : Context)
+acceptDataMessage(context : Context)
«singleton»
RestartTrafficRequested
+restartTrafficAccepted(context : Context)
+deniedRestartTraffic(context : Context)
«singleton»
StopTrafficRequested
+stopTrafficAccepted(context : Context)
+acceptDataMessage(context : Context)
+deniedStopTraffic(context : Context)

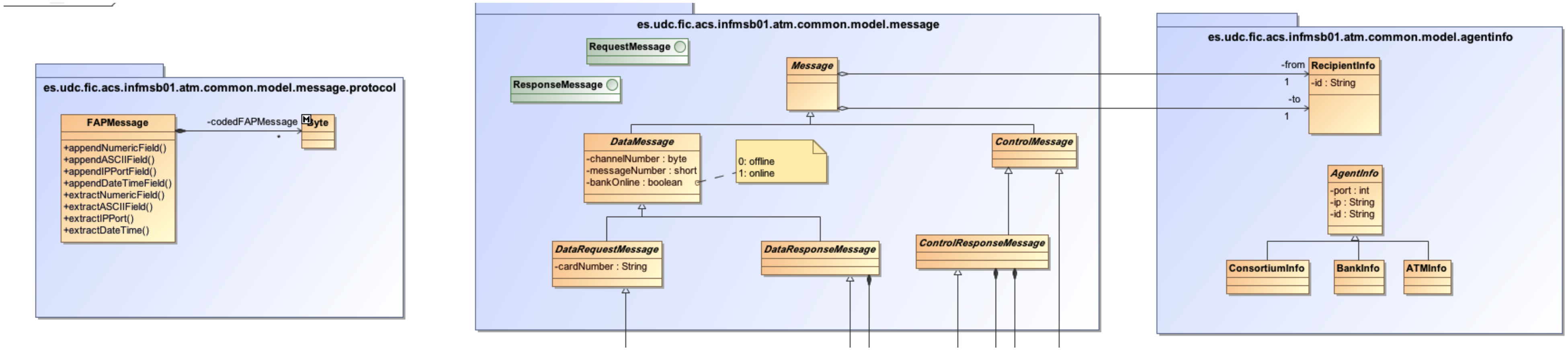

es.udc.fic.acs.infmsb01.atm.common.model.session
DataPipeline
-numberOfChannels : byte
-channels
1..*
Channel
-currentRequestNumber : short
-currentResponseNumber : short
-available : boolean
-dataPipeline
0..1
Session
FAPSession
es.udc.fic.acs.infmsb01.atm.common.model.message
-responses
*
DataResponseMessage
-requests
*
DataRequestMessage
-cardNumber : String
es.udc.fic.acs.infmsb01.atm.common.model.session.state
-context
1
Context
create when an
atm attemps
conection
es.udc.fic.acs.infmsb01.atm.atm.model.session
ATMSession
es.udc.fic.acs.infmsb01.atm.bank.model.session
BankSession
es.udc.fic.acs.infmsb01.atm.consortium.model.session
ConsortiumSession
-pendingMessages
*
PendingDataRequestMessage
-short : pendingNumber
es.udc.fic.acs.infmsb01.atm.common.model.message
-message
1
DataRequestMessage
-cardNumber : String

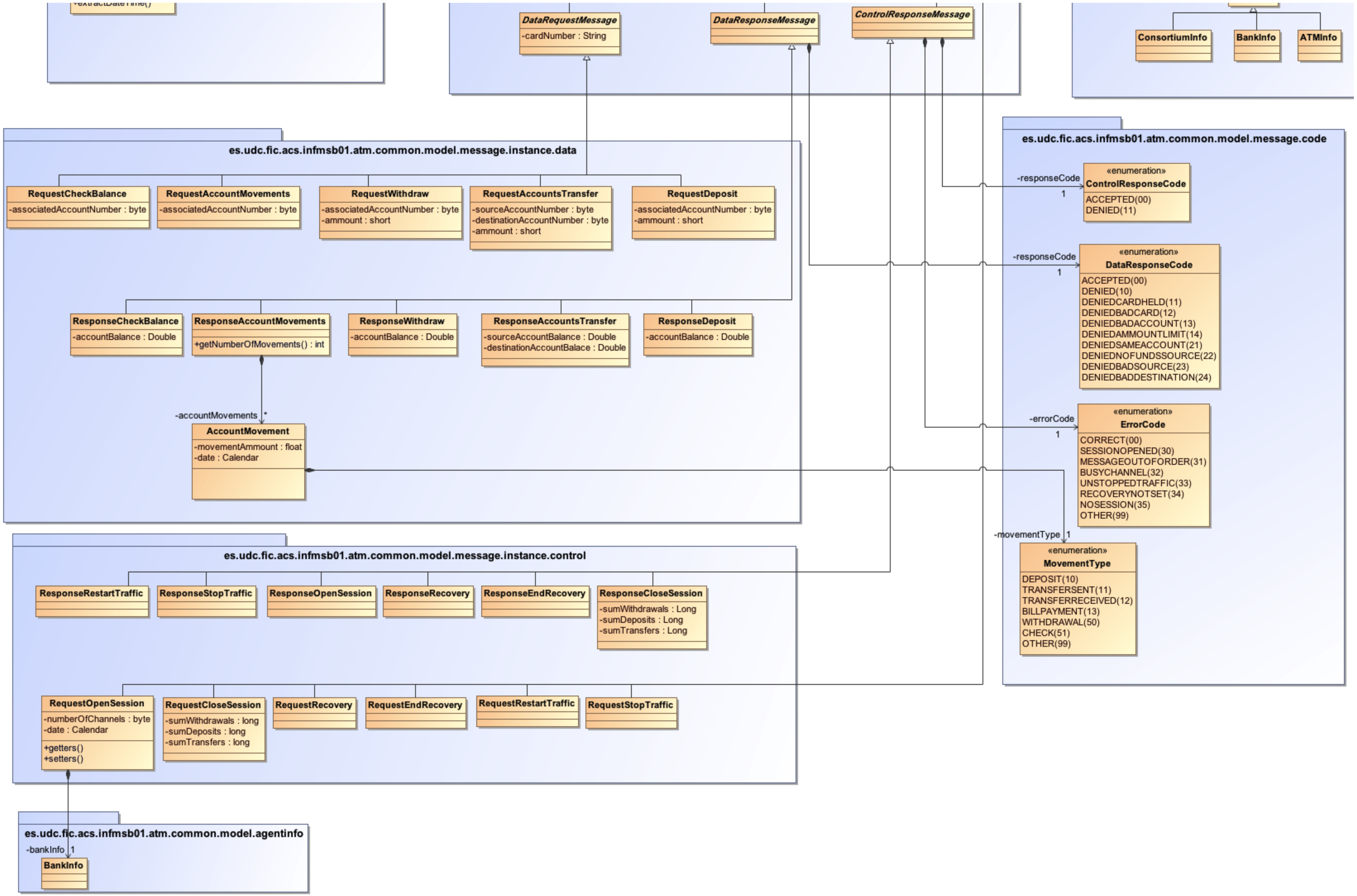

es.udc.fic.acs.infmsb01.atm.common.model.dao
CommonDAO
es.udc.fic.acs.infmsb01.atm.bank.model.dao
BankDAO
+queryAllAccounts(connection : Connection) : List<BankAccountTO>
+queryCreditCardAccounts(creditCard : CreditCardTO, connection : Connection) : CreditCardAccountsCTO
+queryCreditCardAccount(creditCard : CreditCardTO, associatedAccountNumber : byte, connection : Connection) : BankAccountTO
+queryAccountMovements(creditCard : CreditCardTO, associatedAccountNumber : byte, connection : Connection) : AccountMovementsCTO
+querySession(consortiumID : String, connection : Connection) : BankSession
+insertLogMessage(message : Message, connection : Connection)
+updateSessionTotals(consortiumId : String, connection : Connection, sumDeposits : Long, sumWithdrawals : Long, sumTransfers : Long)
+insertDataResponseMessage(consortiumId : String, channelNumber : Byte, connection : Connection, responseNumber : short)
+insertDataRequestMessage(consortiumId : String, channelNumber : Byte, connection : Connection, requestNumber : short)
+accountsTransfer(creditCard : CreditCardTO, sourceAccount : byte, destinationAccount : byte, ammount : short, connection : Connection)
+insertAccountMovements(movs : AccountMovementsCTO, connection : Connection)
+updateSessionState(consortiumID : String, state : StateTO, connection : Connection)
es.udc.fic.acs.infmsb01.atm.consortium.model.dao
ConsortiumDAO
+querySessions(connection : Connection) : List<ConsortiumSession>
+insertPendingMessage(message : DataMessage, connection : Connection)
+updateSessionState(bankId : String, state : StateTO, connection : Connection)
+insertDataResponseMessage(bankId : String, channelNumber : Byte, connection : Connection, responseNumber : short)
+insertDataRequestMessage(bankId : String, channelNumber : Byte, connection : Connection, requestNumber : short)
+updateSessionTotals(bankID : String, connection : Connection, sumDeposits : Long, sumWithdrawals : Long, sumTransfers : Long)
«use»
es.udc.fic.acs.infmsb01.atm.bank.model.dao.to
CreditCardAccountsCTO
-creditCard
1
CreditCardTO
-cardNumber : String
-accounts
*
BankAccountTO
-accountNumber : String
-balance : double
-account
1
AccountMovementsCTO
-movements
*
AccountMovementTO
-type : byte
-ammount : float
-date : Calendar
Card2AccountTO
-cardNumber : String
-realAccountNumber : String
-associatedAccountNumber : byte
es.udc.fic.acs.infmsb01.atm.common.model.dao.to
StateTO
-name
1
«enumeration»
StateName
OFFLINE(0)
OPENSESSIONREQUESTED(1)
SESSIONOPENED(2)
STOPTRAFFICREQUESTED(3)
RESTARTTRAFFICREQUESTED(4)
RECOVERMESSAGESREQUESTED(5)
RECOVERINGMESSAGES(6)
ENDRECOVERYREQUESTED(7)
TRAFFICSTOPPED(8)
CLOSINGSESSION(9)

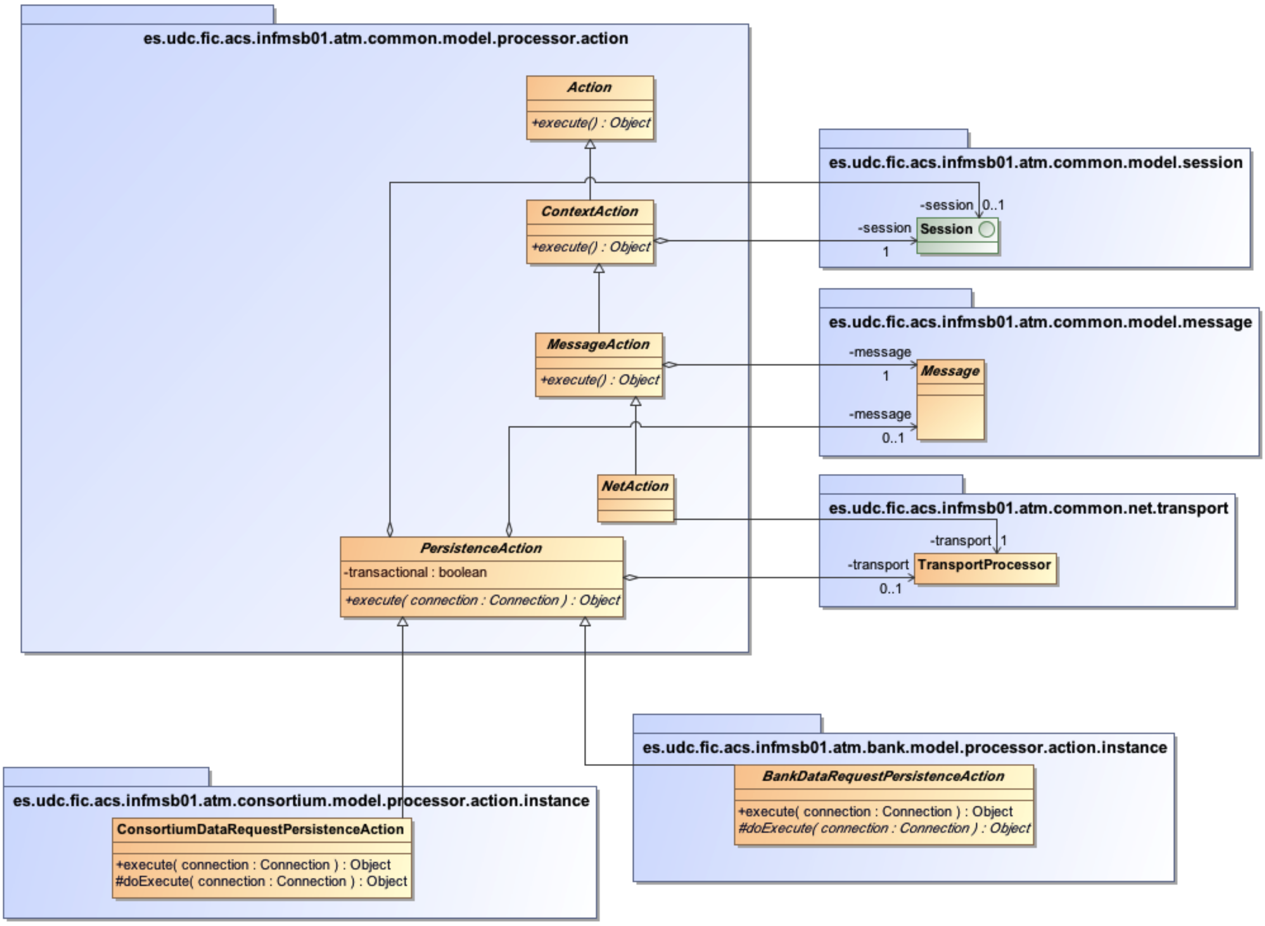

es.udc.fic.acs.infmsb01.atm.common.net.transport
TransportProcessor
-socket : DatagramSocket
+send(message : FAPMessage, to : AgentInfo)
+receive() : FAPMessage
TransportSessionTO
UDPTransport
+send(bytes : Array, destinationIP : String, socket : DatagramSocket, destinationPort : Int)
+receive(socket : DatagramSocket) : Array
«create»
«create»
es.udc.fic.acs.infmsb01.atm.common.model.agentinfo
AgentInfo
-port : int
-ip : String
-id : String
-origin
1
-destination
1
es.udc.fic.acs.infmsb01.atm.atm.model.message.processor
ATMMessageProcessor
+process(message : DataRequestMessage)
+process(message : DataResponseMessage)
-transportProcessor
1
es.udc.fic.acs.infmsb01.atm.bank.model.message.processor
BankMessageProcessor
+process(message : Message)
-transportProcessor
1
es.udc.fic.acs.infmsb01.atm.consortium.net.transport
ConsortiumTransportProcessor
-bankSocket : DatagramSocket
-transportProcessor
1
es.udc.fic.acs.infmsb01.atm.consortium.model.message.processor
ConsortiumMessageProcessor
+process(message : Message)

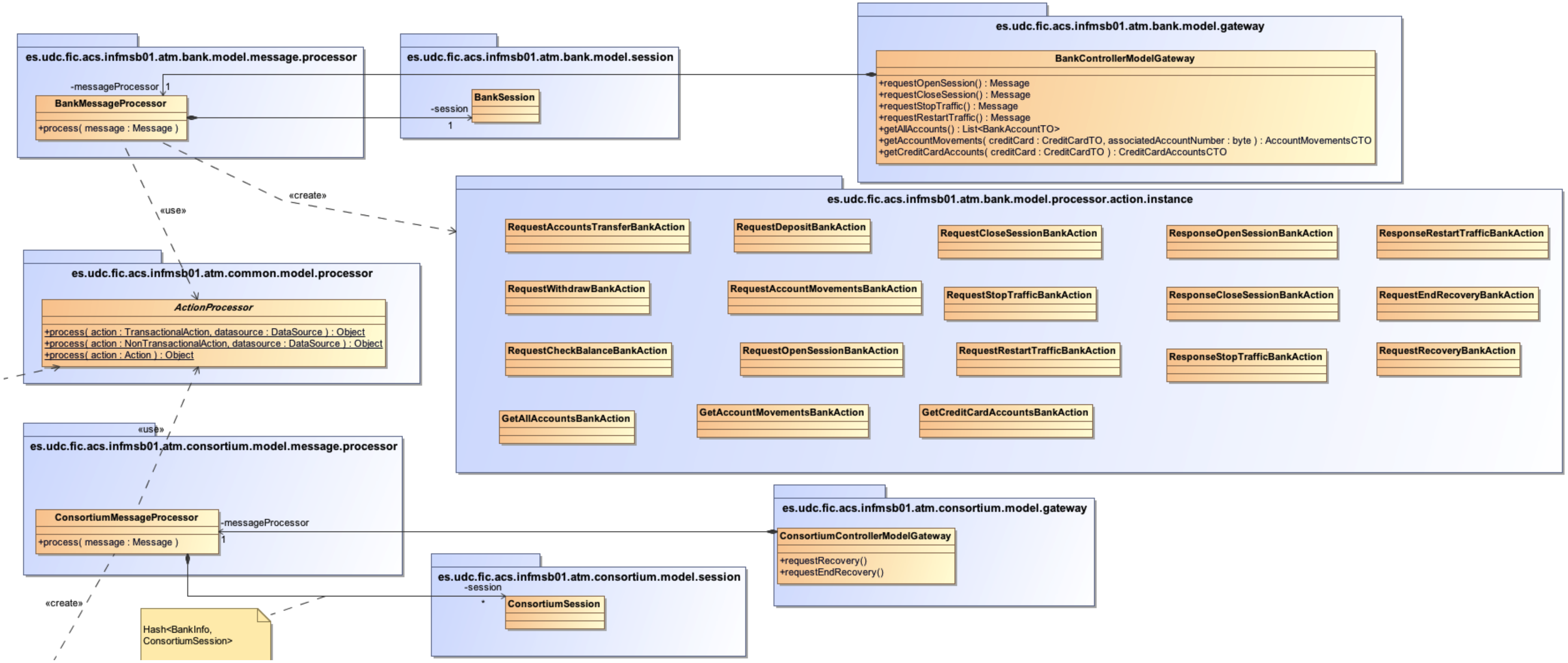

es.udc.fic.acs.infmsb01.atm.common.model.message.conversor.procesor
MessageConversorProcessor
+convert(message : FAPMessage) : Message
+convert(message : Message) : FAPMessage
-message2Conversor
1
Hash<Message, MessageConversor>
-FAPMessageCode2Conversor
1
Hash<FAPMessageCode, MessageConversor>
«use»
«use»
«use»
es.udc.fic.acs.infmsb01.atm.common.model.message.code
«enumeration»
MessageCode
REQUESTRECOVERY(01)
REQUESTENDRECOVERY(02)
RESPONSEOPENSESSION(11)
RESPONSESTOPTRAFFIC(13)
RESPONSERESTARTTRAFFIC(14)
RESPONSECLOSESESSION(12)
REQUESTCHECKBALANCE(31)
REQUESTACCOUNTMOVEMENTS(32)
REQUESTDEPOSIT(33)
REQUESTWITHDRAW(34)
REQUESTACCOUNTSTRANSFER(35)
RESPONSECHECKBALANCE(61)
RESPONSEACCOUNTMOVEMENTS(62)
RESPONSEDEPOSIT(63)
RESPONSEWITHDRAW(64)
RESPONSEACCOUNTSTRANSFER(65)
REQUESTOPENSESSION(81)
REQUESTCLOSESESSION(82)
REQUESTSTOPTRAFFIC(83)
REQUESTRESTARTTRAFFIC(84)
RESPONSERECOVERY(91)
RESPONSEENDRECOVERY(92)
«enumeration»
ErrorCode
CORRECT(00)
SESSIONOPENED(30)
MESSAGEOUTOFORDER(31)
BUSYCHANNEL(32)
UNSTOPPEDTRAFFIC(33)
RECOVERYNOTSET(34)
NOSESSION(35)
OTHER(99)
«enumeration»
DataResponseCode
ACCEPTED(00)
DENIED(10)
DENIEDCARDHELD(11)
DENIEDBADCARD(12)
DENIEDBADACCOUNT(13)
DENIEDAMMOUNTLIMIT(14)
DENIEDSAMEACCOUNT(21)
DENIEDNOFUNDSSOURCE(22)
DENIEDBADSOURCE(23)
DENIEDBADDESTINATION(24)
«enumeration»
MovementType
DEPOSIT(10)
TRANSFERSENT(11)
TRANSFERRECEIVED(12)
BILLPAYMENT(13)
WITHDRAWAL(50)
CHECK(51)
OTHER(99)

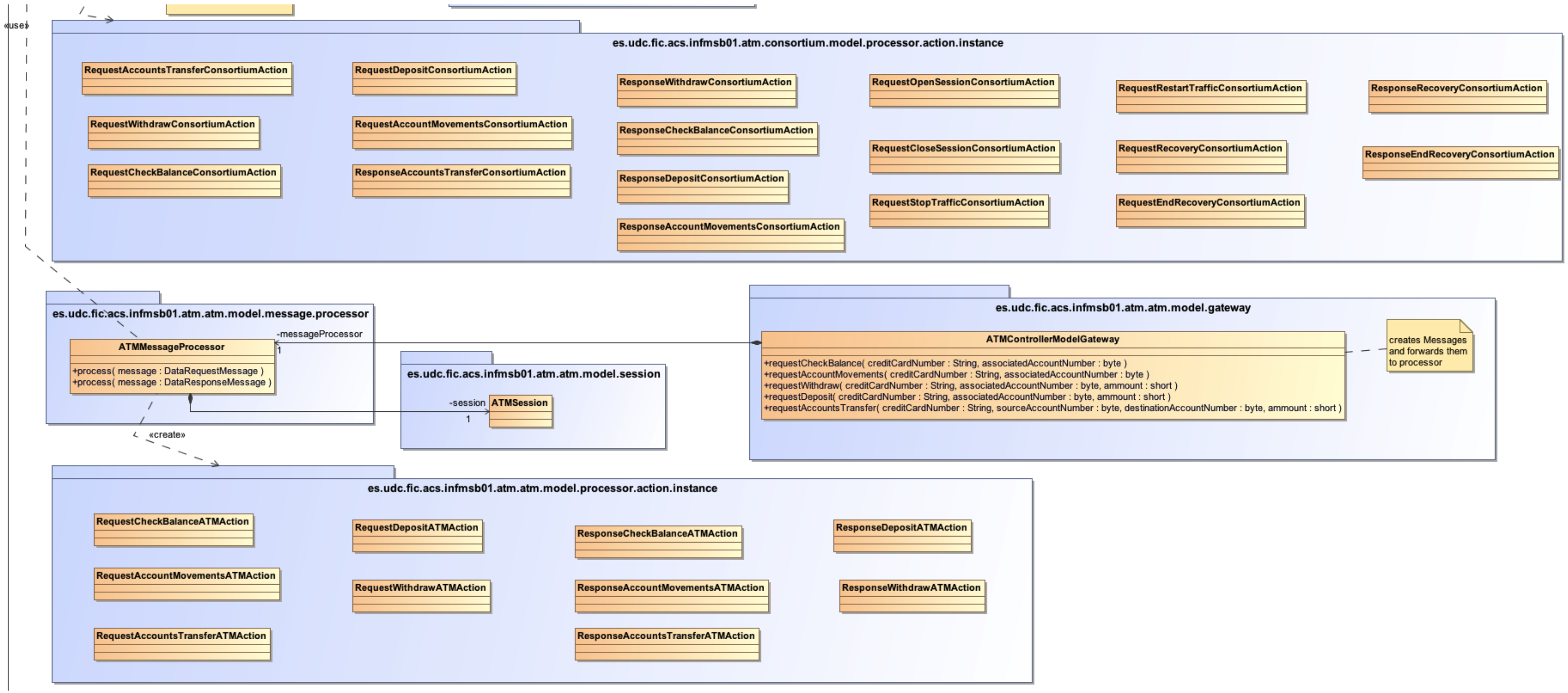

es.udc.fic.acs.infmsb01.atm.common.model.message.conversor
BOTTOM - UP execution!!!
MessageConversor
+getFAPMessageCode(message : FAPMessage) : byte
#fill(from : Message, to : FAPMessage)
#fill(from : FAPMessage, to : Message)
DataMessageConversor
#fill(from : DataMessage, to : FAPMessage)
#fill(from : FAPMessage, to : DataMessage)
ControlMessageConversor
#fill(from : ControlMessage, to : FAPMessage)
#fill(from : FAPMessage, to : ControlMessage)
DataResponseMessageConversor
#fill(from : DataResponseMessage, to : FAPMessage)
#fill(from : FAPMessage, to : DataResponseMessage)
DataRequestMessageConversor
#fill(from : DataRequestMessage, to : FAPMessage)
#fill(from : FAPMessage, to : DataRequestMessage)
ControlResponseMessageCoversor
#fill(from : ControlResponseMessage, to : FAPMessage)
#fill(from : FAPMessage, to : ControlResponseMessage)
es.udc.fic.acs.infmsb01.atm.common.model.message.conversor.instance.data
RequestAccountMovementsConversor
RequestAccountsTransferConversor
RequestCheckBalanceConversor
RequestWithdrawConversor
RequestDepositConversor
ResponseAccountMovementsConversor
ResponseAccountsTransferConversor
ResponseCheckBalanceConversor
ResponseWithdrawConversor
ResponseDepositConversor
es.udc.fic.acs.infmsb01.atm.common.model.message.conversor.instance.control
RequestOpenSessionConversor
RequestCloseSessionConversor
RequestEndRecoveryConversor
RequestRestartTrafficConversor
RequestStopTrafficConversor
RequestRecoveryConversor
ResponseEndRecoveryConversor
ResponseOpenSessionConversor
ResponseRecoveryConversor
ResponseRestartTrafficConversor
ResponseStopTrafficConversor
ResponseCloseSessionConversor


es.udc.fic.acs.infmsb01.atm.common.model.message.protocol
FAPMessage
+appendNumericField()
+appendASCIIField()
+appendIPPortField()
+appendDateTimeField()
+extractNumericField()
+extractASCIIField()
+extractIPPort()
+extractDateTime()
-codedFAPMessage
*
byte
es.udc.fic.acs.infmsb01.atm.common.model.message
RequestMessage
ResponseMessage
Message
DataMessage
-channelNumber : byte
-messageNumber : short
-bankOnline : boolean
0: offline
1: online
ControlMessage
DataRequestMessage
-cardNumber : String
DataResponseMessage
ControlResponseMessage
es.udc.fic.acs.infmsb01.atm.common.model.agentinfo
-from
1
-to
1
RecipientInfo
-id : String
AgentInfo
-port : int
-ip : String
-id : String
ConsortiumInfo
BankInfo
ATMInfo

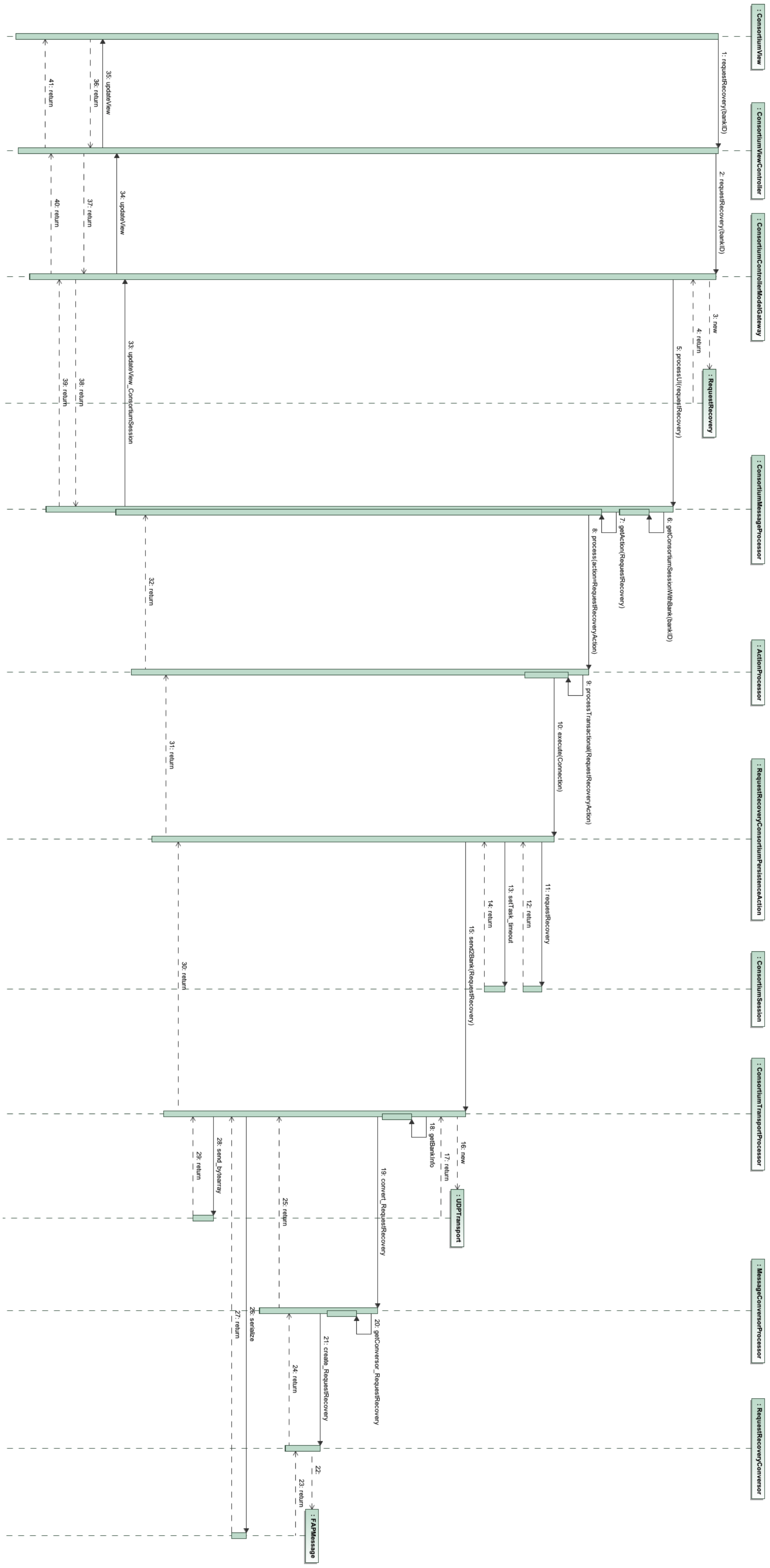

+extractDateTime()
DataRequestMessage
-cardNumber : String
DataResponseMessage
ControlResponseMessage
ConsortiumInfo
BankInfo
ATMInfo
es.udc.fic.acs.infmsb01.atm.common.model.message.instance.data
RequestCheckBalance
-associatedAccountNumber : byte
RequestAccountMovements
-associatedAccountNumber : byte
RequestWithdraw
-associatedAccountNumber : byte
-ammount : short
RequestAccountsTransfer
-sourceAccountNumber : byte
-destinationAccountNumber : byte
-ammount : short
RequestDeposit
-associatedAccountNumber : byte
-ammount : short
ResponseCheckBalance
-accountBalance : Double
ResponseAccountMovements
+getNumberOfMovements() : int
ResponseWithdraw
-accountBalance : Double
ResponseAccountsTransfer
-sourceAccountBalance : Double
-destinationAccountBalace : Double
ResponseDeposit
-accountBalance : Double
-accountMovements
*
AccountMovement
-movementAmmount : float
-date : Calendar
es.udc.fic.acs.infmsb01.atm.common.model.message.code
-responseCode
1
«enumeration»
ControlResponseCode
ACCEPTED(00)
DENIED(11)
-responseCode
1
«enumeration»
DataResponseCode
ACCEPTED(00)
DENIED(10)
DENIEDCARDHELD(11)
DENIEDBADCARD(12)
DENIEDBADACCOUNT(13)
DENIEDAMMOUNTLIMIT(14)
DENIEDSAMEACCOUNT(21)
DENIEDNOFUNDSSOURCE(22)
DENIEDBADSOURCE(23)
DENIEDBADDESTINATION(24)
-errorCode
1
«enumeration»
ErrorCode
CORRECT(00)
SESSIONOPENED(30)
MESSAGEOUTOFORDER(31)
BUSYCHANNEL(32)
UNSTOPPEDTRAFFIC(33)
RECOVERYNOTSET(34)
NOSESSION(35)
OTHER(99)
-movementType
1
«enumeration»
MovementType
DEPOSIT(10)
TRANSFERSENT(11)
TRANSFERRECEIVED(12)
BILLPAYMENT(13)
WITHDRAWAL(50)
CHECK(51)
OTHER(99)
es.udc.fic.acs.infmsb01.atm.common.model.message.instance.control
ResponseRestartTraffic
ResponseStopTraffic
ResponseOpenSession
ResponseRecovery
ResponseEndRecovery
ResponseCloseSession
-sumWithdrawals : Long
-sumDeposits : Long
-sumTransfers : Long
RequestOpenSession
-numberOfChannels : byte
-date : Calendar
+getters()
+setters()
RequestCloseSession
-sumWithdrawals : long
-sumDeposits : long
-sumTransfers : long
RequestRecovery
RequestEndRecovery
RequestRestartTraffic
RequestStopTraffic
es.udc.fic.acs.infmsb01.atm.common.model.agentinfo
-bankInfo
1
BankInfo

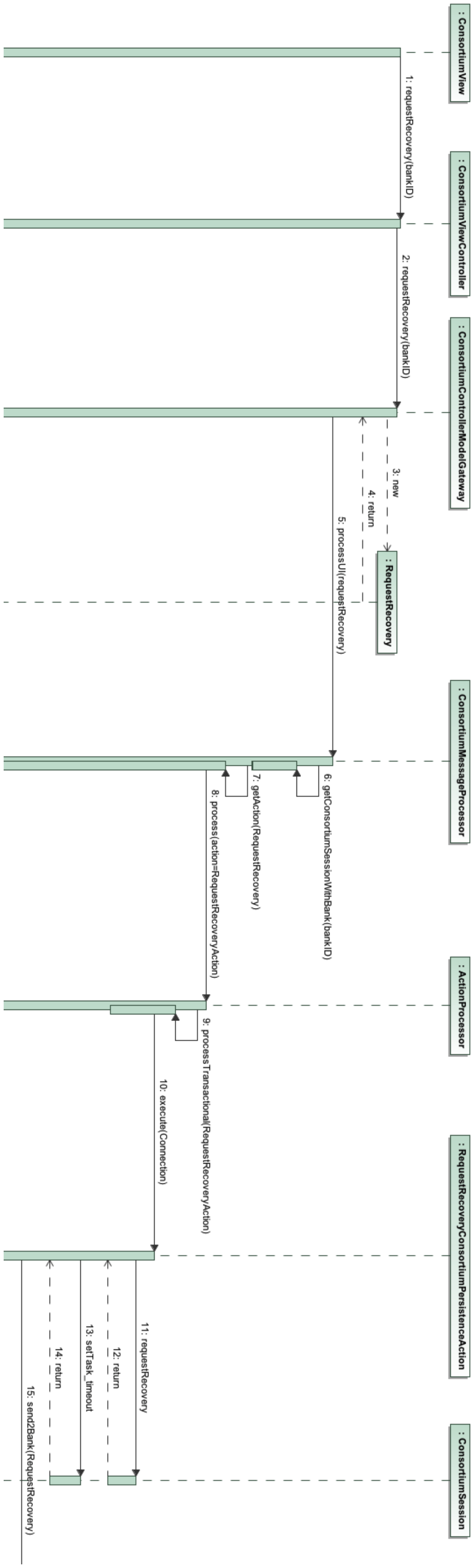

es.udc.fic.acs.infmsb01.atm.common.model.processor.action
Action
+execute() : Object
ContextAction
+execute() : Object
MessageAction
+execute() : Object
NetAction
PersistenceAction
-transactional : boolean
+execute(connection : Connection) : Object
es.udc.fic.acs.infmsb01.atm.common.model.session
-session
0..1
-session
1
Session
es.udc.fic.acs.infmsb01.atm.common.model.message
-message
1
Message
-message
0..1
es.udc.fic.acs.infmsb01.atm.common.net.transport
-transport
1
-transport
0..1
TransportProcessor
es.udc.fic.acs.infmsb01.atm.bank.model.processor.action.instance
BankDataRequestPersistenceAction
+execute(connection : Connection) : Object
#doExecute(connection : Connection) : Object
es.udc.fic.acs.infmsb01.atm.consortium.model.processor.action.instance
ConsortiumDataRequestPersistenceAction
+execute(connection : Connection) : Object
#doExecute(connection : Connection) : Object


es.udc.fic.acs.infmsb01.atm.bank.model.message.processor
-messageProcessor
1
BankMessageProcessor
+process(message : Message)
es.udc.fic.acs.infmsb01.atm.bank.model.session
-session
1
BankSession
es.udc.fic.acs.infmsb01.atm.bank.model.gateway
BankControllerModelGateway
+requestOpenSession() : Message
+requestCloseSession() : Message
+requestStopTraffic() : Message
+requestRestartTraffic() : Message
+getAllAccounts() : List<BankAccountTO>
+getAccountMovements(creditCard : CreditCardTO, associatedAccountNumber : byte) : AccountMovementsCTO
+getCreditCardAccounts(creditCard : CreditCardTO) : CreditCardAccountsCTO
«use»
«create»
es.udc.fic.acs.infmsb01.atm.common.model.processor
ActionProcessor
+process(action : TransactionalAction, datasource : DataSource) : Object
+process(action : NonTransactionalAction, datasource : DataSource) : Object
+process(action : Action) : Object
es.udc.fic.acs.infmsb01.atm.bank.model.processor.action.instance
RequestAccountsTransferBankAction
RequestDepositBankAction
RequestCloseSessionBankAction
ResponseOpenSessionBankAction
ResponseRestartTrafficBankAction
RequestWithdrawBankAction
RequestAccountMovementsBankAction
RequestStopTrafficBankAction
ResponseCloseSessionBankAction
RequestEndRecoveryBankAction
RequestCheckBalanceBankAction
RequestOpenSessionBankAction
RequestRestartTrafficBankAction
ResponseStopTrafficBankAction
RequestRecoveryBankAction
GetAllAccountsBankAction
GetAccountMovementsBankAction
GetCreditCardAccountsBankAction
«use»
es.udc.fic.acs.infmsb01.atm.consortium.model.message.processor
ConsortiumMessageProcessor
+process(message : Message)
-messageProcessor
1
es.udc.fic.acs.infmsb01.atm.consortium.model.gateway
ConsortiumControllerModelGateway
+requestRecovery()
+requestEndRecovery()
«create»
es.udc.fic.acs.infmsb01.atm.consortium.model.session
-session
*
ConsortiumSession
Hash<BankInfo,
ConsortiumSession>

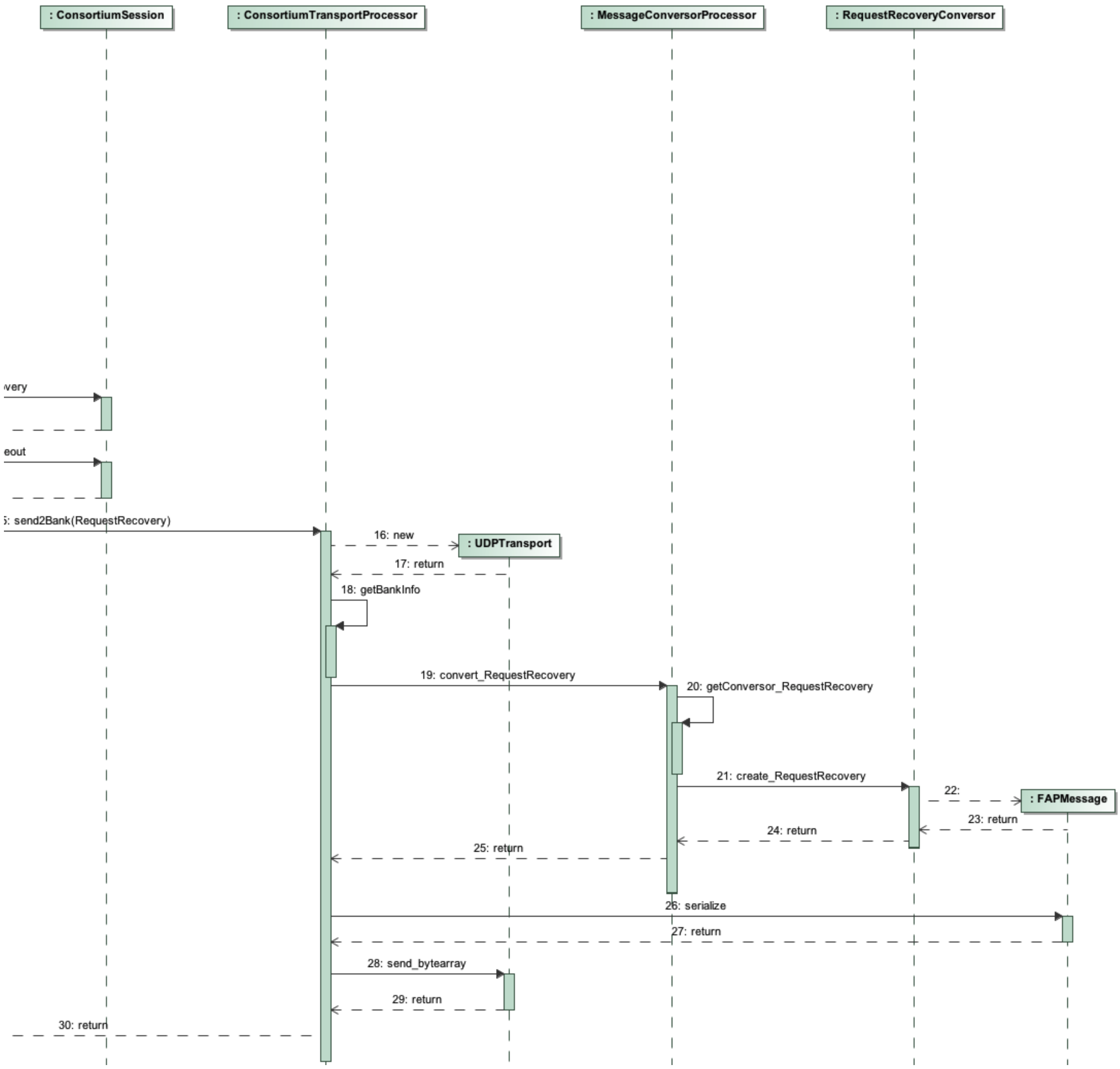

«use»
es.udc.fic.acs.infmsb01.atm.consortium.model.processor.action.instance
RequestAccountsTransferConsortiumAction
RequestWithdrawConsortiumAction
RequestCheckBalanceConsortiumAction
RequestDepositConsortiumAction
RequestAccountMovementsConsortiumAction
ResponseAccountsTransferConsortiumAction
ResponseWithdrawConsortiumAction
ResponseCheckBalanceConsortiumAction
ResponseDepositConsortiumAction
ResponseAccountMovementsConsortiumAction
RequestOpenSessionConsortiumAction
RequestCloseSessionConsortiumAction
RequestStopTrafficConsortiumAction
RequestRestartTrafficConsortiumAction
RequestRecoveryConsortiumAction
RequestEndRecoveryConsortiumAction
ResponseRecoveryConsortiumAction
ResponseEndRecoveryConsortiumAction
es.udc.fic.acs.infmsb01.atm.atm.model.message.processor
ATMMessageProcessor
+process(message : DataRequestMessage)
+process(message : DataResponseMessage)
-messageProcessor
1
es.udc.fic.acs.infmsb01.atm.atm.model.session
-session
1
ATMSession
es.udc.fic.acs.infmsb01.atm.atm.model.gateway
ATMControllerModelGateway
+requestCheckBalance(creditCardNumber : String, associatedAccountNumber : byte)
+requestAccountMovements(creditCardNumber : String, associatedAccountNumber : byte)
+requestWithdraw(creditCardNumber : String, associatedAccountNumber : byte, ammount : short)
+requestDeposit(creditCardNumber : String, associatedAccountNumber : byte, ammount : short)
+requestAccountsTransfer(creditCardNumber : String, sourceAccountNumber : byte, destinationAccountNumber : byte, ammount : short)
creates Messages
and forwards them
to processor
«create»
es.udc.fic.acs.infmsb01.atm.atm.model.processor.action.instance
RequestCheckBalanceATMAction
RequestAccountMovementsATMAction
RequestAccountsTransferATMAction
RequestDepositATMAction
RequestWithdrawATMAction
ResponseCheckBalanceATMAction
ResponseAccountMovementsATMAction
ResponseAccountsTransferATMAction
ResponseDepositATMAction
ResponseWithdrawATMAction

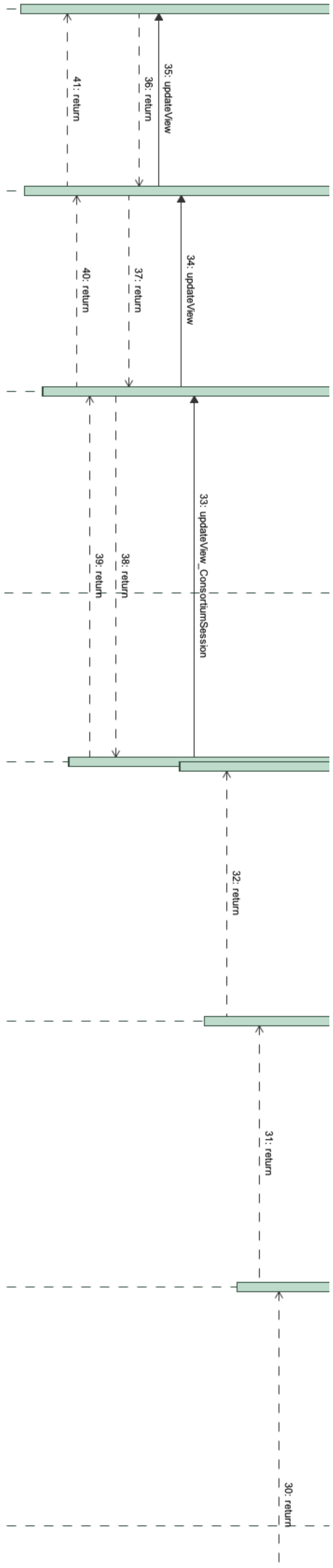
Sequence Diagrams

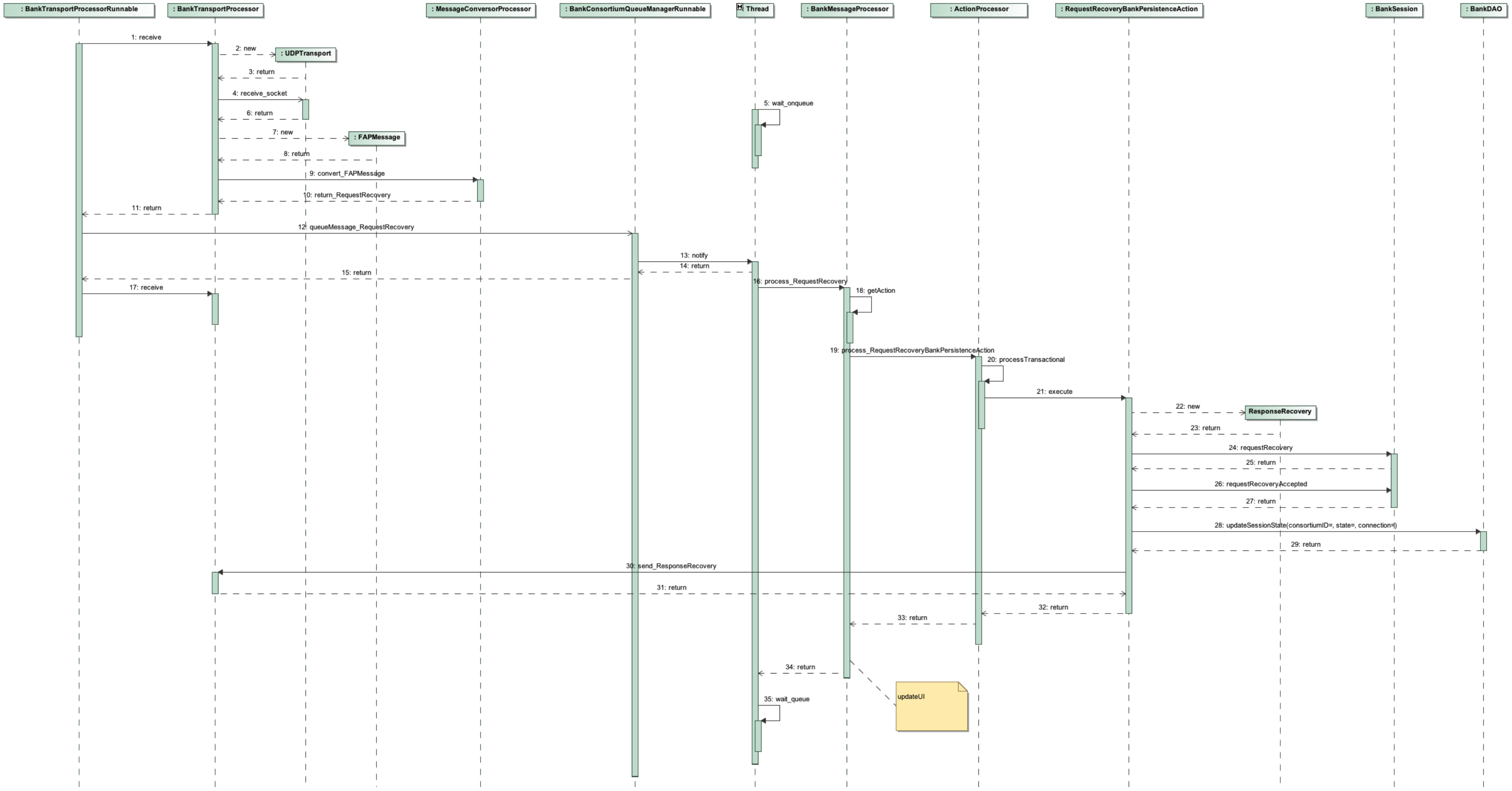

: ConsortiumView
: ConsortiumViewController
: ConsortiumControllerModelGateway
: RequestRecovery
: ConsortiumMessageProcessor
: ActionProcessor
: RequestRecoveryConsortiumPersistenceAction
: ConsortiumSession
: ConsortiumTransportProcessor
: UDPTransport
: MessageConversorProcessor
: RequestRecoveryConversor
: FAPMessage
1: requestRecovery(bankID)
2: requestRecovery(bankID)
3: new
4: return
5: processUI(requestRecovery)
6: getConsortiumSessionWithBank(bankID)
7: getAction(RequestRecovery)
8: process(actionRequestRecoveryAction)
9: processTransactional(RequestRecoveryAction)
10: execute(Connection)
11: requestRecovery
12: return
13: setTask_timeout
14: return
15: send2Bank(RequestRecovery)
16: new
17: return
18: getBankInfo
19: convert_RequestRecovery
20: getConversor_RequestRecovery
21: create_RequestRecovery
22:
23: return
24: return
25: return
26: serialize
27: return
28: send_bytearray
29: return
30: return
31: return
32: return
33: updateView_ConsortiumSession
34: updateView
35: updateView
36: return
37: return
38: return
39: return
40: return
41: return

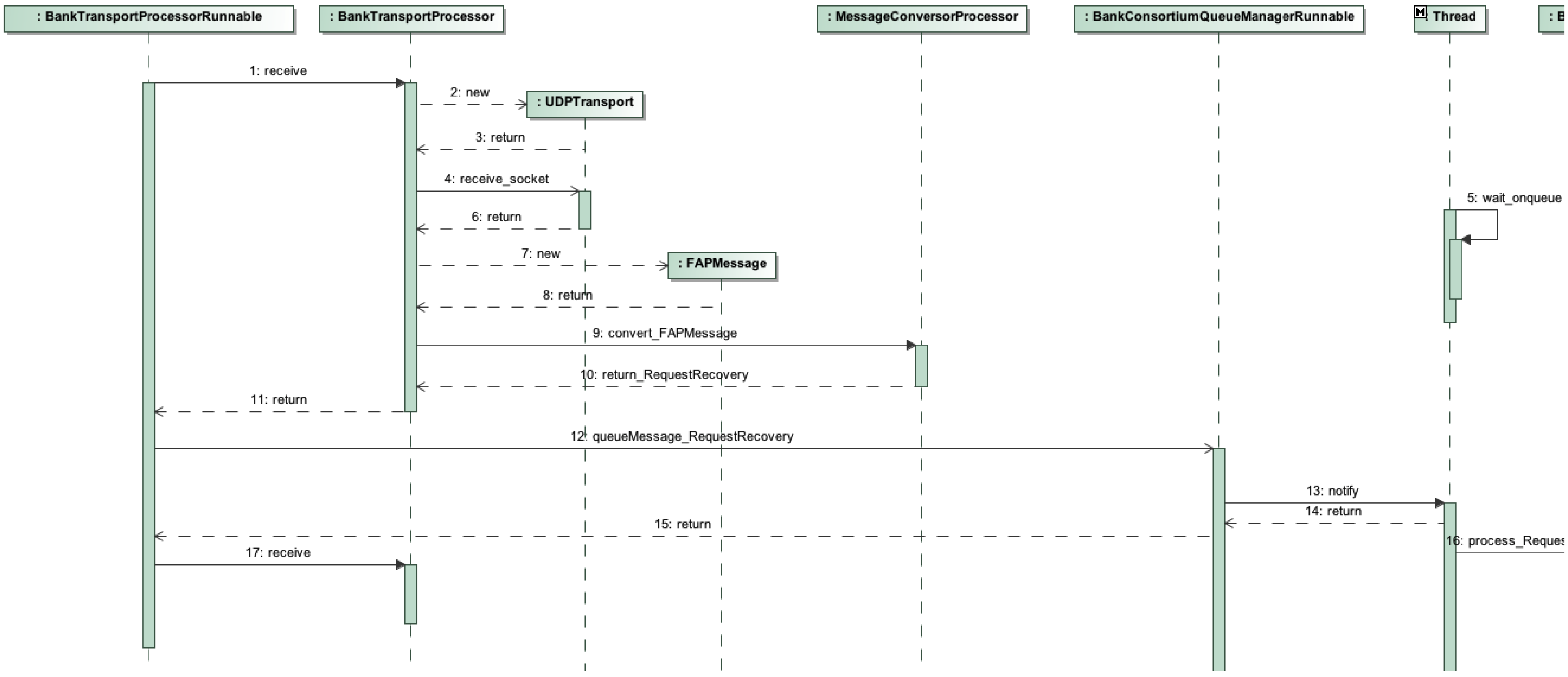


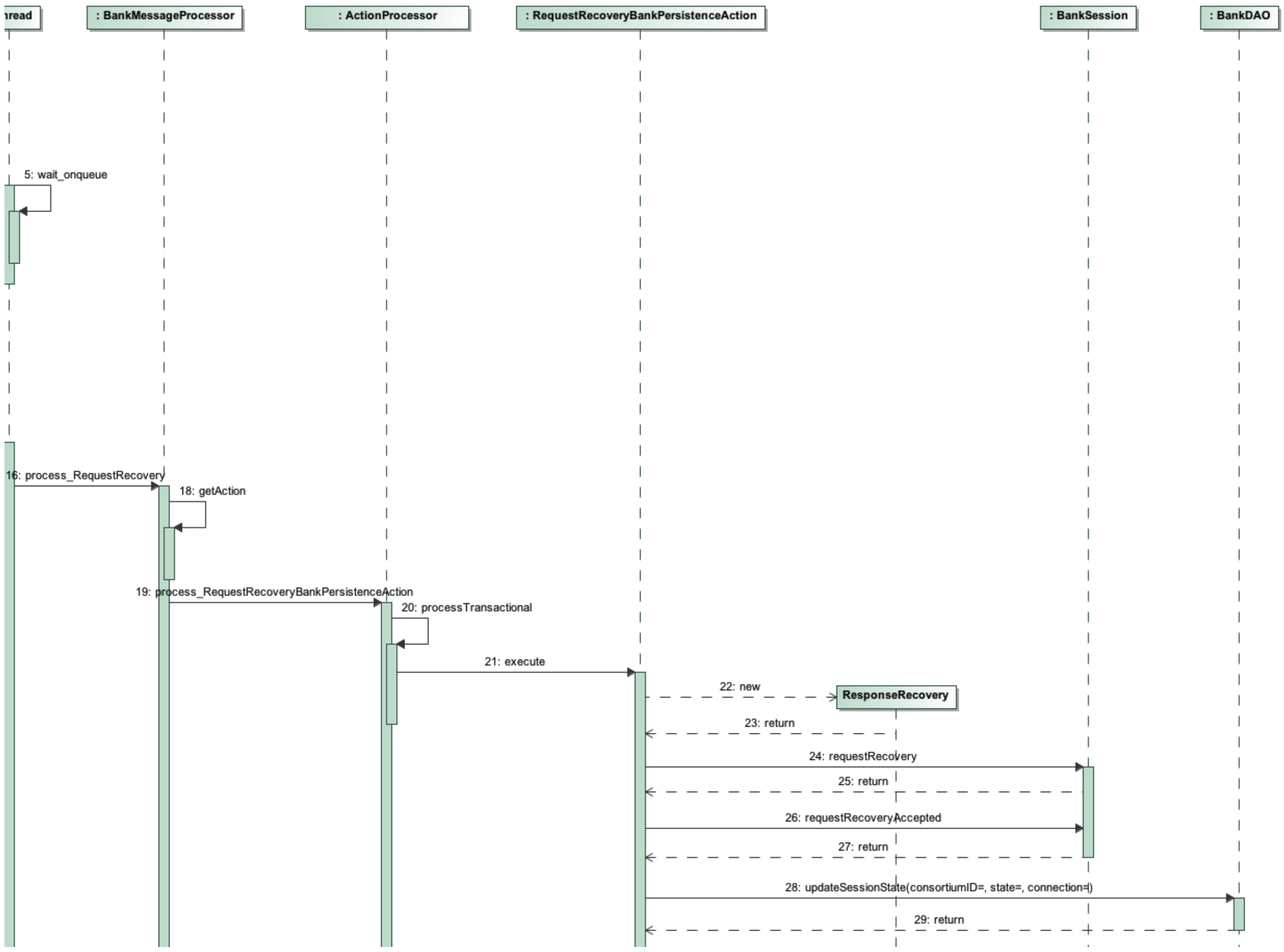
: ConsortiumView
: ConsortiumViewController
: ConsortiumControllerModelGateway
: RequestRecovery
: ConsortiumMessageProcessor
: ActionProcessor
: RequestRecoveryConsortiumPersistenceAction
: ConsortiumSession
1: requestRecovery(bankID)
2: requestRecovery(bankID)
3: new
4: return
5: processUI(requestRecovery)
6: getConsortiumSessionWithBank(bankID)
7: getAction(RequestRecovery)
8: process(action=RequestRecoveryAction)
9: processTransactional(RequestRecoveryAction)
10: execute(Connection)
11: requestRecovery
12: return
13: setTask_timeout
14: return
15: send2Bank(RequestRecovery)

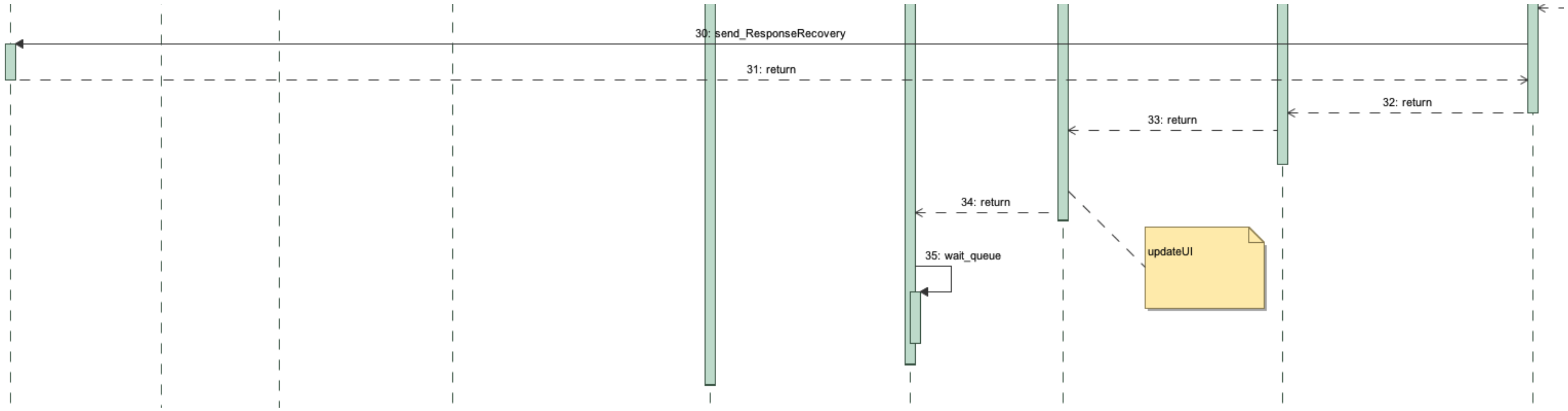

: ConsortiumSession
: ConsortiumTransportProcessor
: MessageConversorProcessor
: RequestRecoveryConversor
very
eout
5: send2Bank(RequestRecovery)
16: new
: UDPTransport
17: return
18: getBankInfo
19: convert_RequestRecovery
20: getConversor_RequestRecovery
21: create_RequestRecovery
22:
: FAPMessage
23: return
24: return
25: return
26: serialize
27: return
28: send_bytearray
29: return
30: return


30: return
31: return
32: return
33: updateView_ConsortiumSession
34: updateView
35: updateView
36: return
37: return
38: return
39: return
40: return
41: return


: BankTransportProcessorRunnable
: BankTransportProcessor
: MessageConversorProcessor
: BankConsortiumQueueManagerRunnable
Thread
: BankMessageProcessor
: ActionProcessor
: RequestRecoveryBankPersistenceAction
: BankSession
: BankDAO
1: receive
2: new
: UDPTransport
3: return
4: receive_socket
5: wait_onqueue
6: return
7: new
: FAPMessage
8: return
9: convert_FAPMessage
10: return_RequestRecovery
11: return
12: queueMessage_RequestRecovery
13: notify
14: return
15: return
16: process_RequestRecovery
17: receive
18: getAction
19: process_RequestRecoveryBankPersistenceAction
20: processTransactional
21: execute
22: new
ResponseRecovery
23: return
24: requestRecovery
25: return
26: requestRecoveryAccepted
27: return
28: updateSessionState(consortiumID=, state=, connection=)
29: return
30: send_ResponseRecovery
31: return
32: return
33: return
34: return
35: wait_queue
updateUI


: BankTransportProcessorRunnable
: BankTransportProcessor
: MessageConversorProcessor
: BankConsortiumQueueManagerRunnable
Thread
: B
1: receive
2: new
: UDPTransport
3: return
4: receive_socket
5: wait_onqueue
6: return
7: new
: FAPMessage
8: return
9: convert_FAPMessage
10: return_RequestRecovery
11: return
12: queueMessage_RequestRecovery
13: notify
14: return
15: return
16: process_Reques
17: receive


hread
: BankMessageProcessor
: ActionProcessor
: RequestRecoveryBankPersistenceAction
: BankSession
: BankDAO
5: wait_onqueue
16: process_RequestRecovery
18: getAction
19: process_RequestRecoveryBankPersistenceAction
20: processTransactional
21: execute
22: new
ResponseRecovery
23: return
24: requestRecovery
25: return
26: requestRecoveryAccepted
27: return
28: updateSessionState(consortiumID=, state=, connection=)
29: return


30: send_ResponseRecovery
31: return
32: return
33: return
34: return
35: wait_queue
updateUI