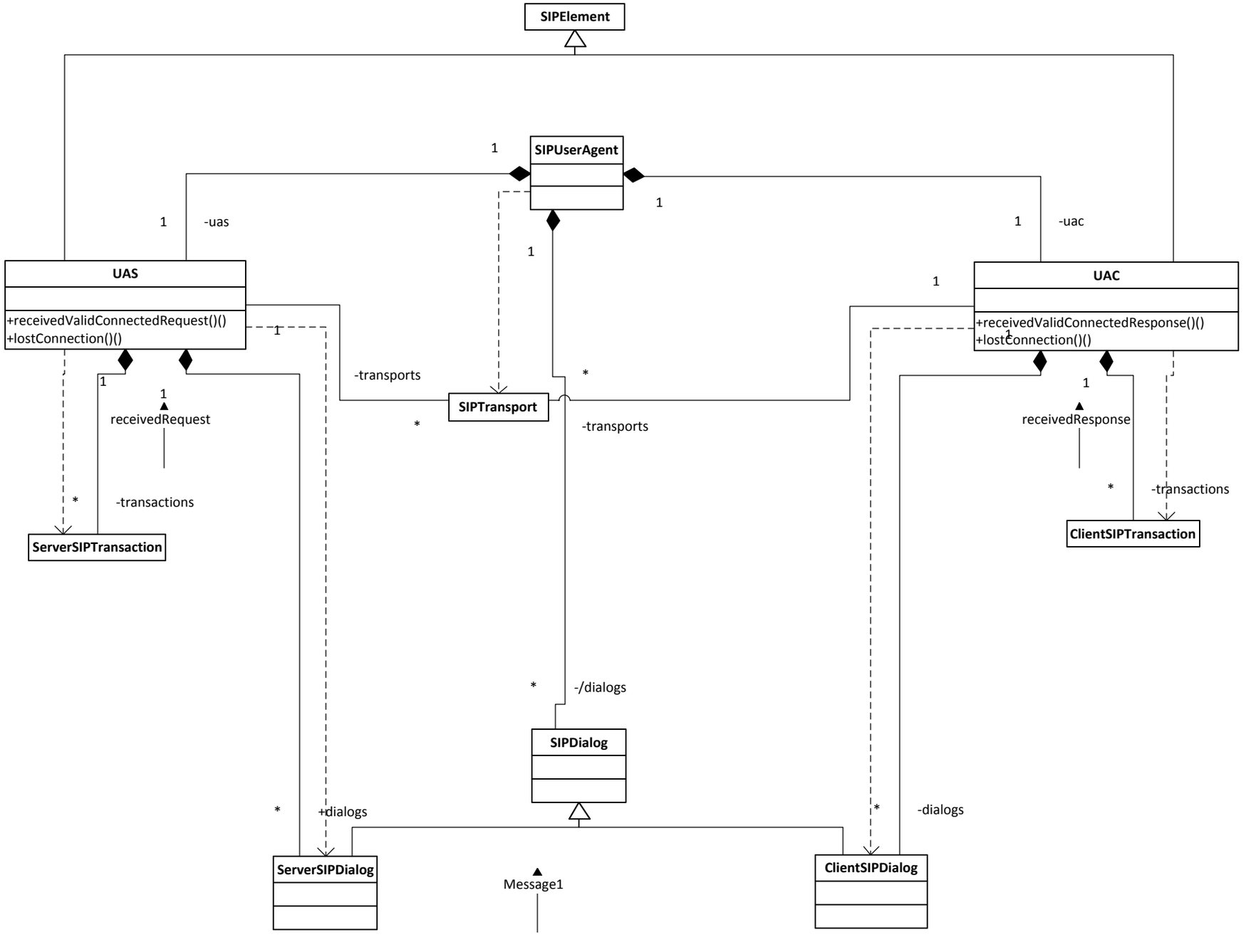
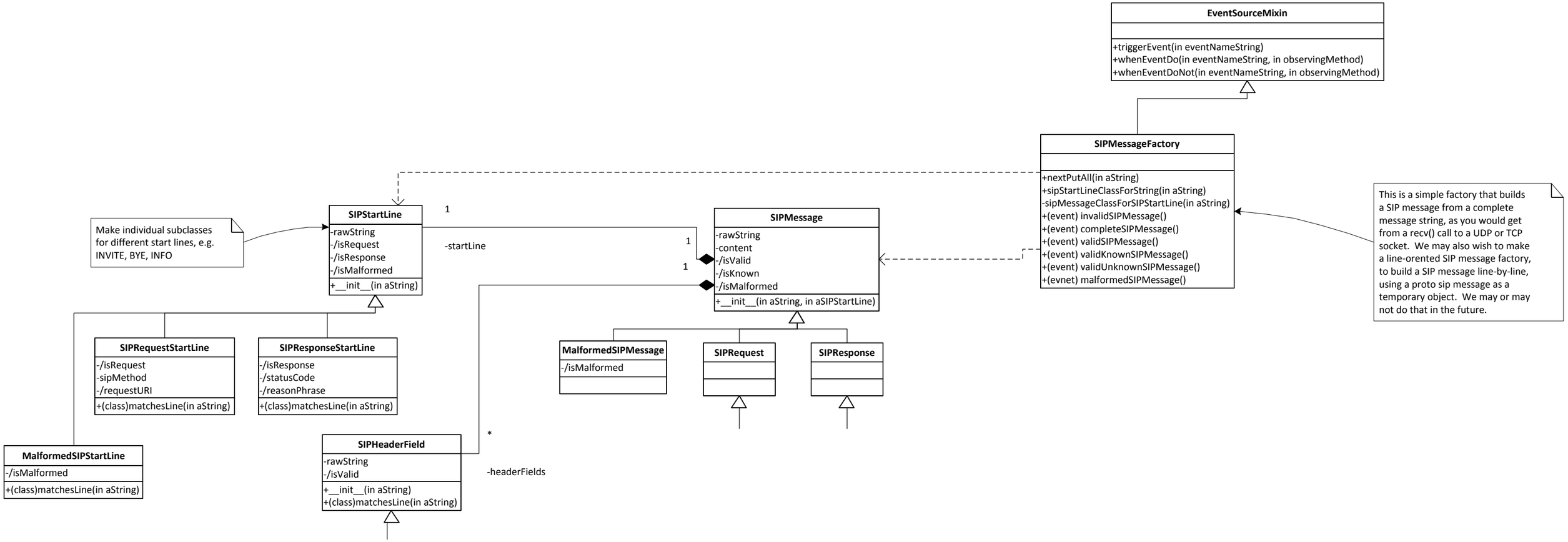
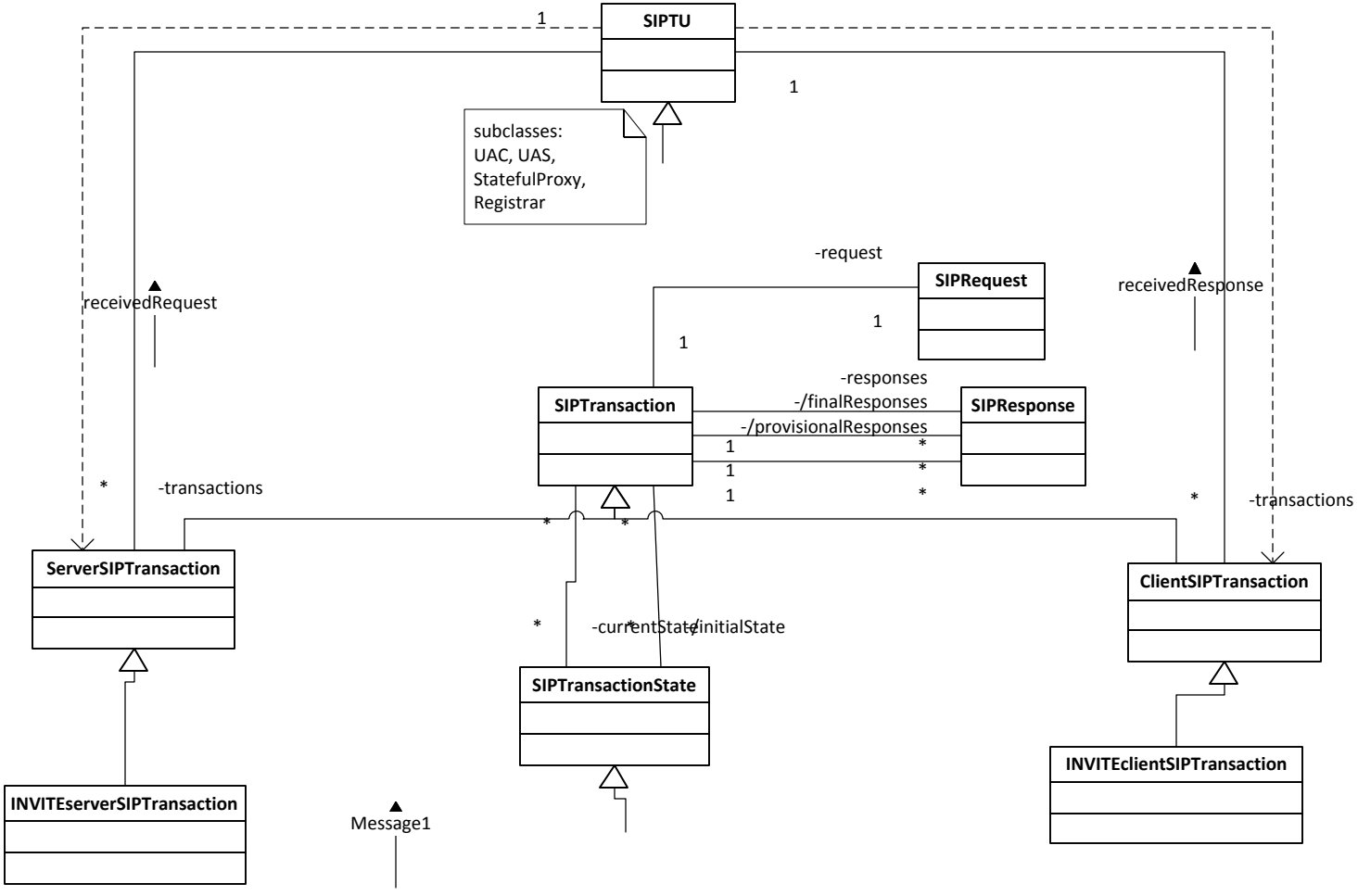
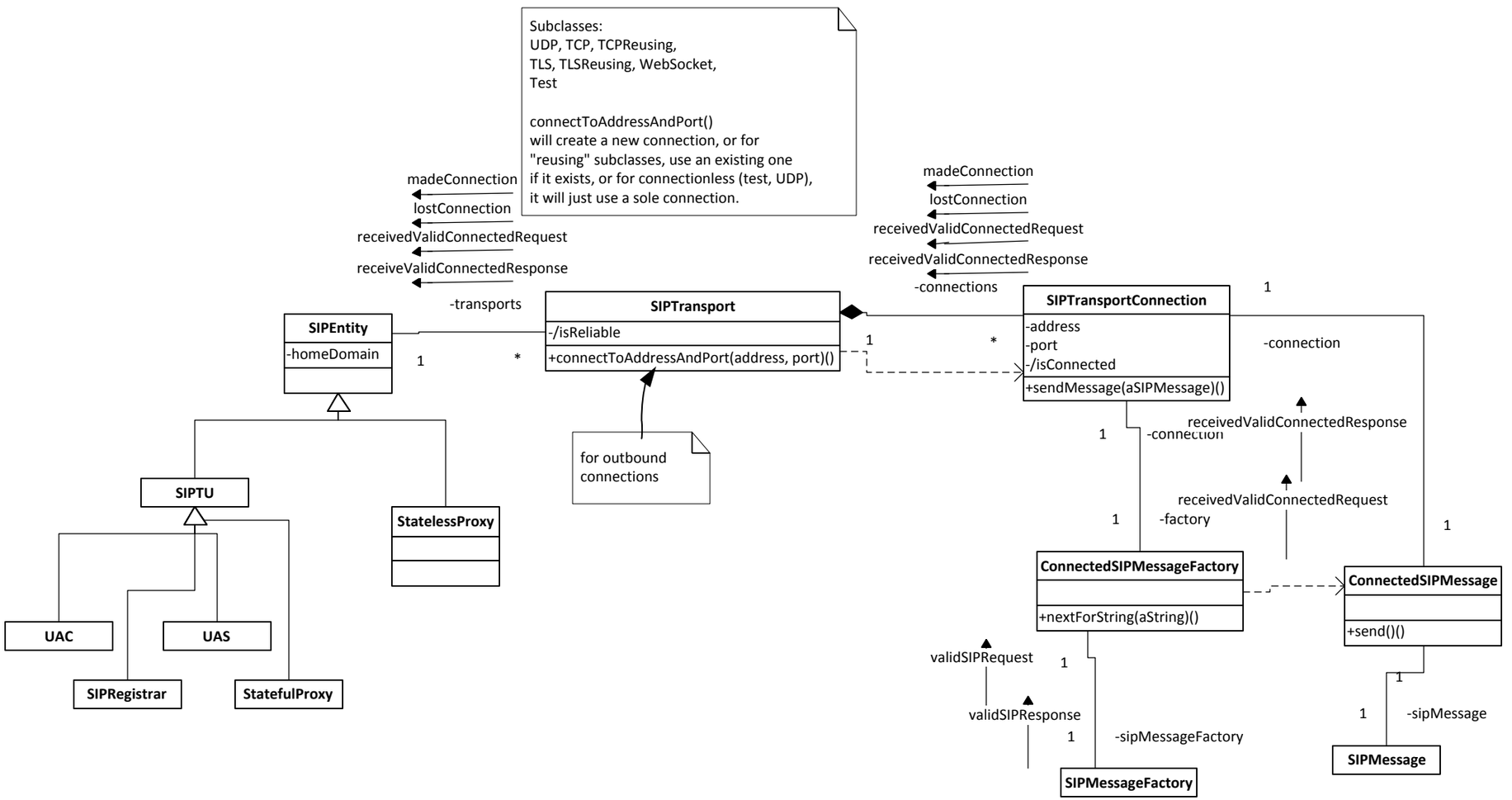

SIPElement
SIPUserAgent
1
1
1
-uas
1
-uac
UAS
+receivedValidConnectedRequest()()
+lostConnection()()
UAC
+receivedValidConnectedResponse()()
+lostConnection()()
1
1
1
1
-transports
*
SIPTransport
*
-transports
1
receivedRequest
1
receivedResponse
*
-transactions
*
-transactions
ServerSIPTransaction
ClientSIPTransaction
*
-/dialogs
SIPDialog
*
+dialogs
*
-dialogs
ServerSIPDialog
Message1
ClientSIPDialog


EventSourceMixin
+triggerEvent(in eventNameString)
+whenEventDo(in eventNameString, in observingMethod)
+whenEventDoNot(in eventNameString, in observingMethod)
SIPMessageFactory
+nextPutAll(in aString)
+sipStartLineClassForString(in aString)
-sipMessageClassForSIPStartLine(in aString)
+(event) invalidSIPMessage()
+(event) completeSIPMessage()
+(event) validSIPMessage()
+(event) validKnownSIPMessage()
+(event) validUnknownSIPMessage()
+(evnet) malformedSIPMessage()
This is a simple factory that builds a SIP message from a complete message string, as you would get from a recv() call to a UDP or TCP socket. We may also wish to make a line-orented SIP message factory, to build a SIP message line-by-line, using a proto sip message as a temporary object. We may or may not do that in the future.
Make individual subclasses for different start lines, e.g. INVITE, BYE, INFO
SIPStartLine
-rawString
-/isRequest
-/isResponse
-/isMalformed
+__init__(in aString)
1
-startLine
1
1
SIPMessage
-rawString
-content
-/isValid
-/isKnown
-/isMalformed
+__init__(in aString, in aSIPStartLine)
SIPRequestStartLine
-/isRequest
-sipMethod
-/requestURI
+(class)matchesLine(in aString)
SIPResponseStartLine
-/isResponse
-/statusCode
-/reasonPhrase
+(class)matchesLine(in aString)
MalformedSIPMessage
-/isMalformed
SIPRequest
SIPResponse
MalformedSIPStartLine
-/isMalformed
+(class)matchesLine(in aString)
SIPHeaderField
-rawString
-/isValid
+__init__(in aString)
+(class)matchesLine(in aString)
*
-headerFields


SIPTU
1
1
subclasses:
UAC, UAS,
StatefulProxy,
Registrar
-request
SIPRequest
1
1
receivedRequest
receivedResponse
SIPTransaction
-responses
-/finalResponses
-/provisionalResponses
SIPResponse
1
1
1
*
*
*
*
-transactions
*
*
*
-transactions
ServerSIPTransaction
ClientSIPTransaction
*
*
-currentState/initialState
SIPTransactionState
INVITEserverSIPTransaction
INVITEclientSIPTransaction
Message1


Subclasses:
UDP, TCP, TCPReusing,
TLS, TLSReusing, WebSocket,
Test
connectToAddressAndPort()
will create a new connection, or for
"reusing" subclasses, use an existing one
if it exists, or for connectionless (test, UDP),
it will just use a sole connection.
madeConnection
lostConnection
receivedValidConnectedRequest
receiveValidConnectedResponse
-transports
SIPEntity
-homeDomain
1
*
SIPTransport
-/isReliable
+connectToAddressAndPort(address, port)()
for outbound
connections
SIPTU
StatelessProxy
UAC
UAS
SIPRegistrar
StatefulProxy
madeConnection
lostConnection
receivedValidConnectedRequest
receivedValidConnectedResponse
-connections
SIPTransportConnection
-address
-port
-/isConnected
+sendMessage(aSIPMessage)()
1
*
1
-connection
receivedValidConnectedResponse
1
-connection
receivedValidConnectedRequest
1
-factory
1
ConnectedSIPMessageFactory
+nextForString(aString)()
ConnectedSIPMessage
+send()()
validSIPRequest
1
validSIPResponse
1
-sipMessageFactory
SIPMessageFactory
1
1
-sipMessage
SIPMessage