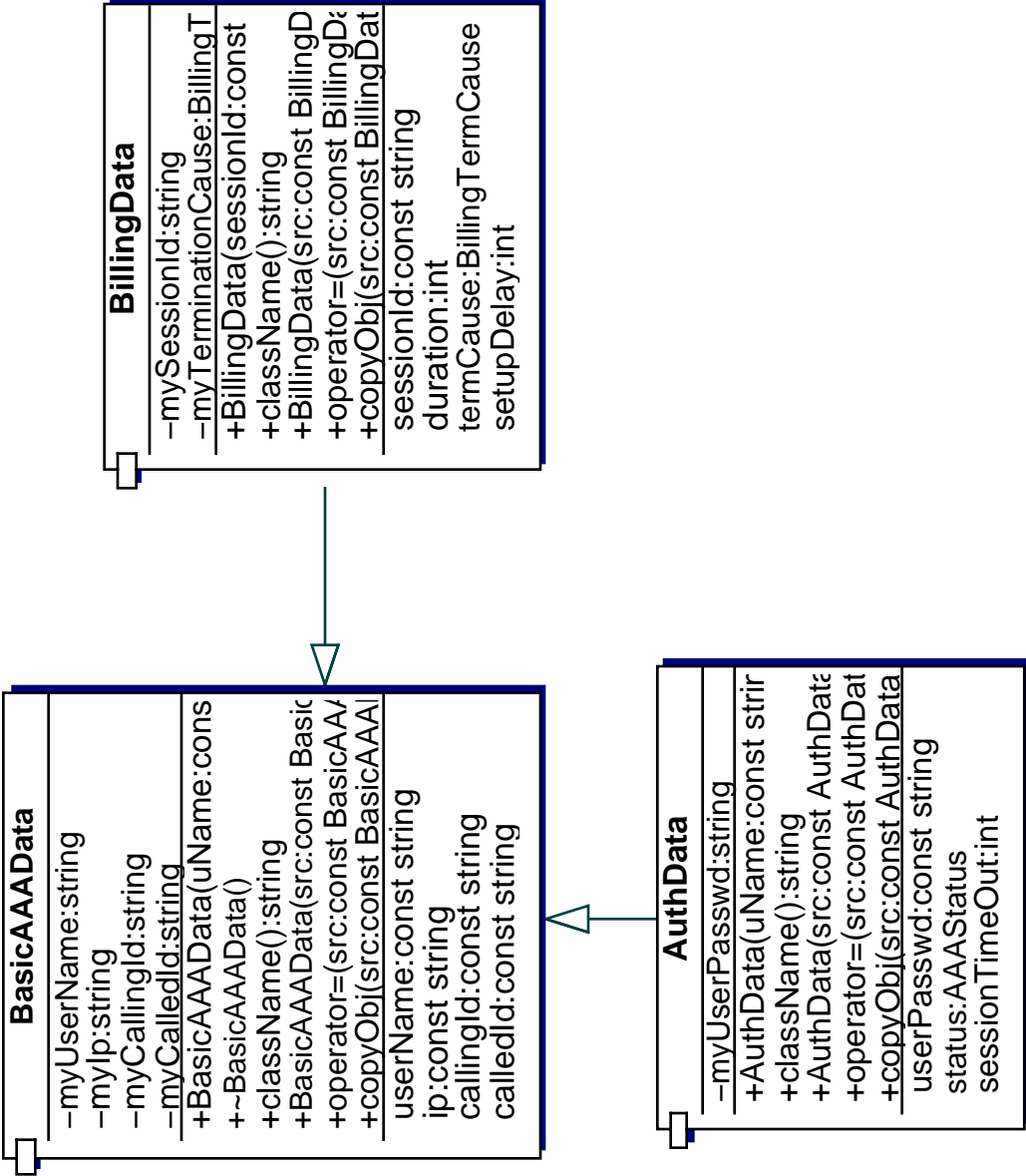


1.1, billingData



1.1, billingData

```

classDiagram
    class BasicAgent {
        +myAuthId:unsigned long
        +myInvokee:Sptr<UaBase>
        +myDelElgAid:bool
        +BasicAgent(authId:unsigned long)
        +BasicAgent(src:const BasicAgent)
        +operator=(src:const BasicAgent)
        +BasicAgent()
        +className():string
        +inCall():void
        +callFailed():void
        +doBye():void
        +doCancel():void
        +endCall():void
        +timerExpired():void
        +toBeDeleted():bool
        +setDeleted():void
        +copyObj(src:const BasicAgent):unsigned long
        +id:unsigned long
        +invokee:const Sptr<UaBase>
    }

    class SipClient {
        +UaClient(reqMsg:const Sptr)
        +className():string
        +sendBye():void
    }

    class SipServer {
        +UaServer(reqMsg:const Sptr)
        +className():string
        +sendBye():void
    }

    class SipCallLegData {
        +mySdpData:Sptr<SipSdp>
        +myRequestMsg:Sptr<SipMsg>
        +myResponseMsg:Sptr<SipMsg>
        +myAckMsg:Sptr<SipMsg>
        +mySipCallLeg:Sptr<SipCallLeg>
        +myContactVector:ContactVector
        +SipCallLegData(reqMsg:const Sptr)
        +SipCallLegData(src:const Sptr)
        +operator=(src:const Sptr)
        +copyObj(src:const Sptr)
        +pushContact(contact:const Sptr)
        +popContact():Sptr<Contact>
        +setRequest(sipMsg:const Sptr)
        +setAck(sipMsg:const Sptr)
        +getSdp():const Sptr<SipSdp>
        +getRequest():const Sptr<SipMsg>
        +getResponse():const Sptr<SipMsg>
        +getAck():const Sptr<SipMsg>
        +SipCallLegData()
        +setResponse(sipMsg:const Sptr)
        +numContacts:int
        +contactData:Sptr<ContactData>
        +callLegState:CallLegState
        +callLeg:const Sptr<SipCallLeg>
    }

    class AccountingData {
        +AccountingData(sessionId:const)
        +AccountingData(src:const AccountingData)
        +operator=(src:const AccountingData)
        +copyObj(src:const AccountingData):unsigned long
        +sessionId:unsigned long
        +unusedSeconds:int
        +startTime:time_t
        +endTime:time_t
    }

    class ContactData {
        +myCallPeerList:CallPeerList
        +ContactData()
        +ContactData(dLegData:const ContactData)
        +operator=(src:const ContactData)
        +copyObj(src:const ContactData):const Sptr<ContactData>
        +addPeer(callLegData:const Sptr):const Sptr<CallPeerList>
    }

    class MultiLegCallData {
        +myCallLegDataMap:CallLegDataMap
        +myTransactionPeerMap:TransactionPeerMap
        +myAccountingData:Sptr<AccountingData>
        +MultiLegCallData()
        +MultiLegCallData(src:const MultiLegCallData)
        +operator=(src:const MultiLegCallData)
        +copyObj(src:const MultiLegCallData):const Sptr<MultiLegCallData>
        +getCallLeg(callLeg:const Sptr):const Sptr<MultiLegCallData>
        +addCallLeg(callLeg:const Sptr)
        +removeCallLeg(callLeg:const Sptr)
        +findPeer(trid:const Sptr):const Sptr<TransactionPeer>
        +addTransactionPeer(peer:const Sptr)
        +removeTransactionPeer(peer:const Sptr)
        +getAccountingData():const Sptr<AccountingData>
        +setAccountingData(accData:const Sptr)
    }

    class CalIDB {
        +myAccountingDataMap:AccountingDataMap
        +myMultiLegCallDataMap:MultiLegCallDataMap
        +instance():CalIDB
        +className():string
        +CalIDB()
        +destroy():void
        +addCallLeg(userAgent:Sptr):const Sptr<MultiLegCallData>
        +addPeer(userAgent:Sptr):const Sptr<MultiLegCallData>
        +getMultiLegCallData(callLeg:const Sptr):const Sptr<MultiLegCallData>
        +setAccountingData(callLeg:const Sptr):const Sptr<AccountingData>
        +removeAccountingData(callLeg:const Sptr)
        +getAccountingData(sessionId:const):const Sptr<AccountingData>
        +findAllPeers(userAgent:Sptr):const Sptr<MultiLegCallData>
        +removePeer(srcAgent:const Sptr)
        +removePeer(agent:const Sptr)
        +removeCallData(agent:const Sptr)
        +CalIDB()
    }

    class UaState {
        +UaStateRinging()
        +UaState()
        +UaStateTrying()
        +UaStateIdle()
        +UaStateInCall()
        +UaStateEnd()
        +UaStateFailure()
        +UaStateTying()
    }

    class SipProxyEvent {
        +AAAEvent()
        +AAAEvent(agent:AuthAgent)
        +className():string
        +name():const char*
        +authAgent:AuthAgent
        +reqId:u_int16_t
    }

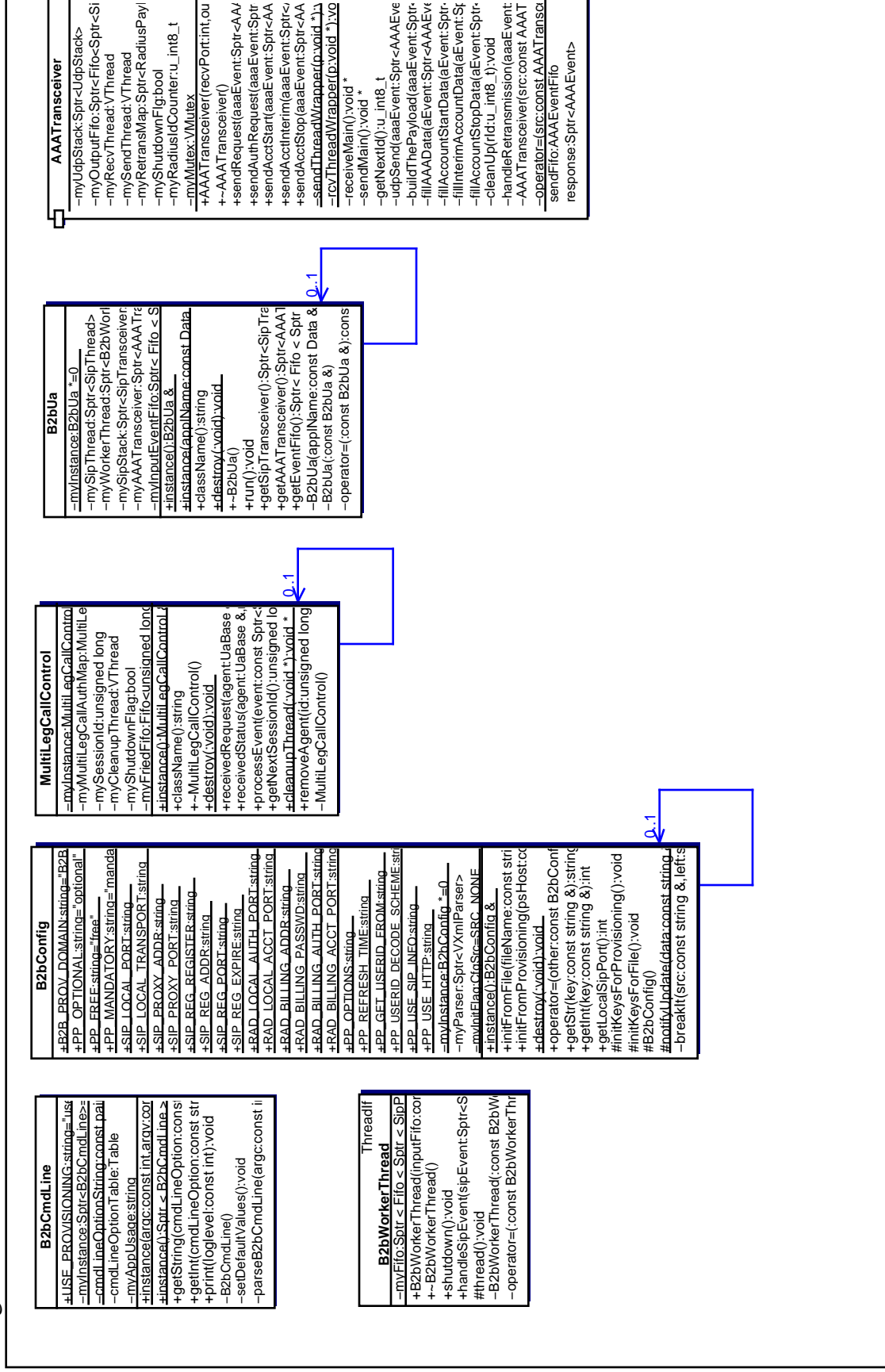
    class InvalidStateException {
        +InvalidStateException(msg:const)
        +getName():string
    }

    class RadiusPayload {
        +radiusCount:u_int16_t
        +myRadiusTransId:RadiusTransId
        +myMsg:Sptr<RadiusMessage>
        +authEvent:Sptr<AAAEvent>
        +radiusPayload()
    }

    BasicAgent --|> SipClient
    BasicAgent --|> SipServer
    BasicAgent --|> SipCallLegData
    BasicAgent --|> AccountingData
    BasicAgent --|> ContactData
    BasicAgent --|> MultiLegCallData
    BasicAgent --|> CalIDB
    BasicAgent --|> UaState
    SipClient --|> SipServer
    SipCallLegData --|> AccountingData
    SipCallLegData --|> ContactData
    SipCallLegData --|> MultiLegCallData
    SipCallLegData --|> CalIDB
    SipCallLegData --|> UaState
    SipCallLegData --|> SipProxyEvent
    SipCallLegData --|> InvalidStateException
    SipCallLegData --|> RadiusPayload
    SipCallLegData --|> AAAEvent
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
    SipCallLegData --|> InvalidStateException(msg:const)
    SipCallLegData --|> RadiusPayload(radiusCount:u_int16_t, myRadiusTransId:RadiusTransId, myMsg:Sptr<RadiusMessage>, authEvent:Sptr<AAAEvent>)
    SipCallLegData --|> AAAEvent()
    SipCallLegData --|> AAAEvent(agent:AuthAgent)
    SipCallLegData --|> AAAEvent(name:const char*, authAgent:AuthAgent, reqId:u_int16_t)
```

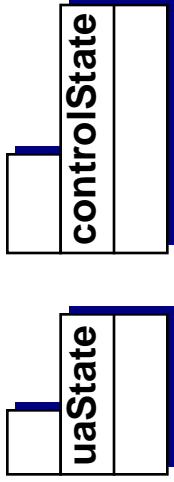
1.1, <default>

1.1, global



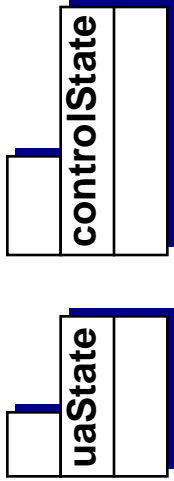
1.1, global

1.1, bin.debug.Linux.i686



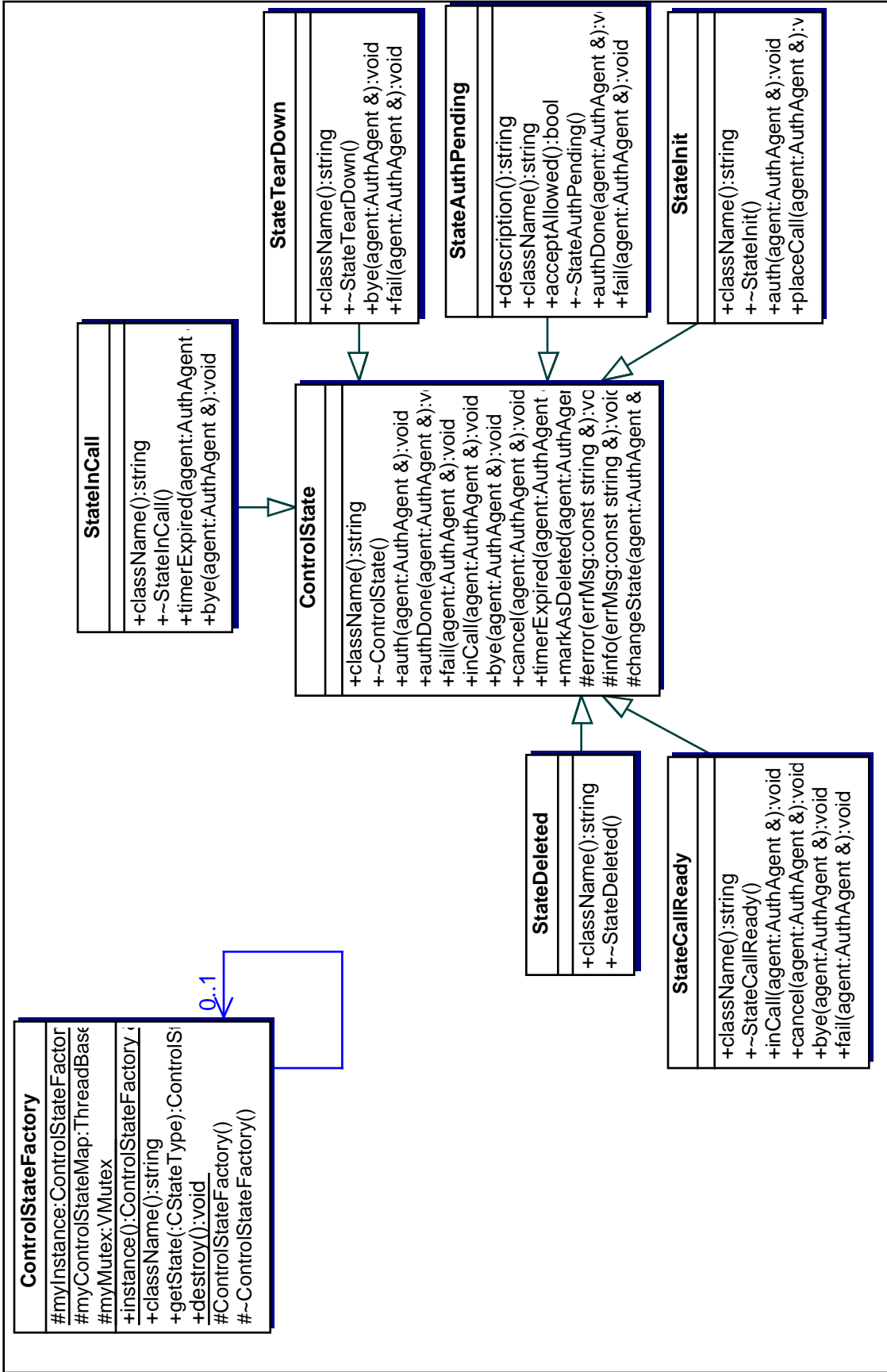
1.1, bin.debug.Linux.i686

1.1, bin.opt.Linux.i686



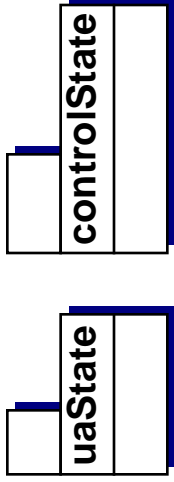
1.1, bin.opt.Linux.i686

1.1, controlState



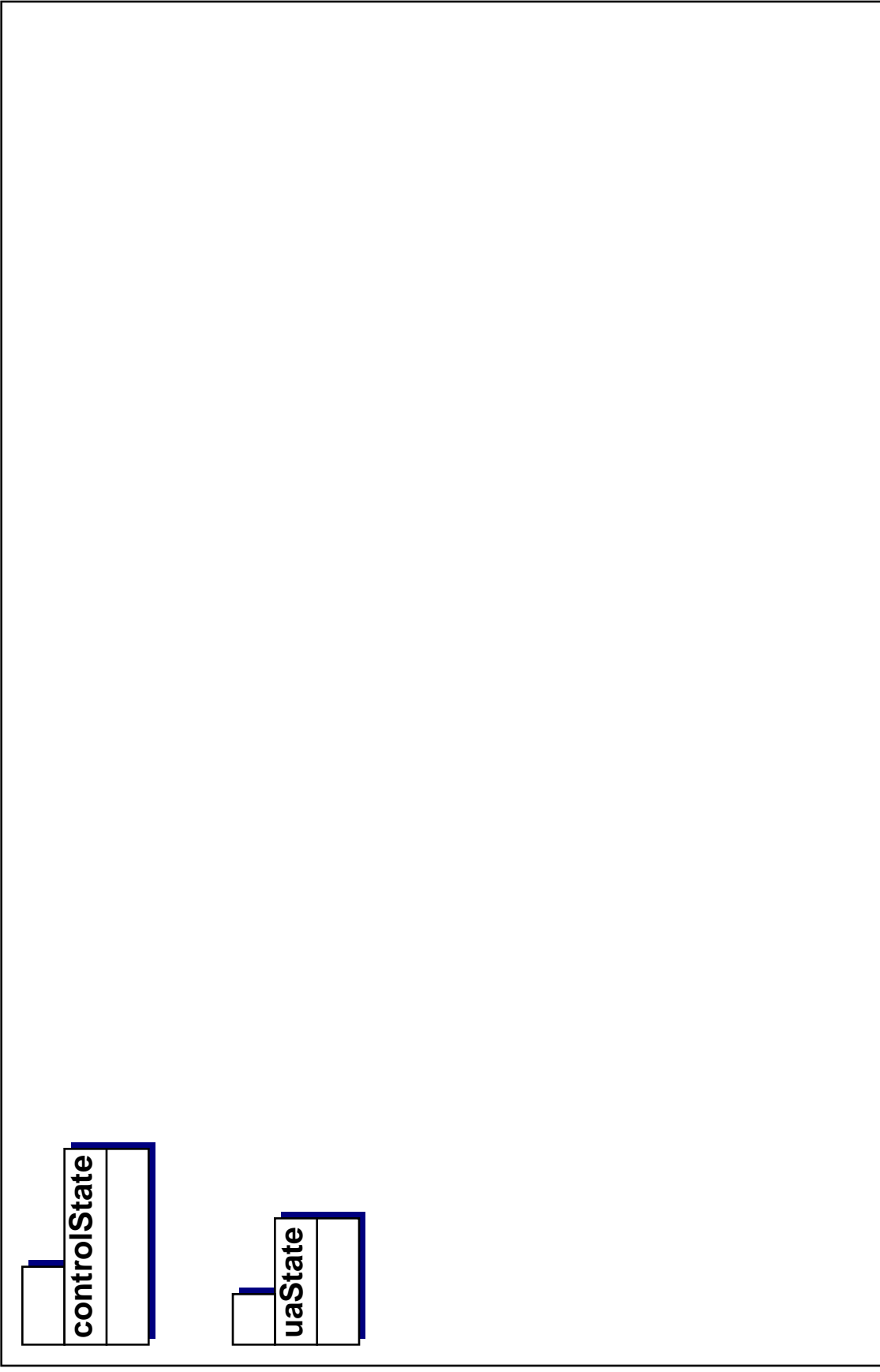
1.1, controlState

1.1, obj.debug.Linux.i686



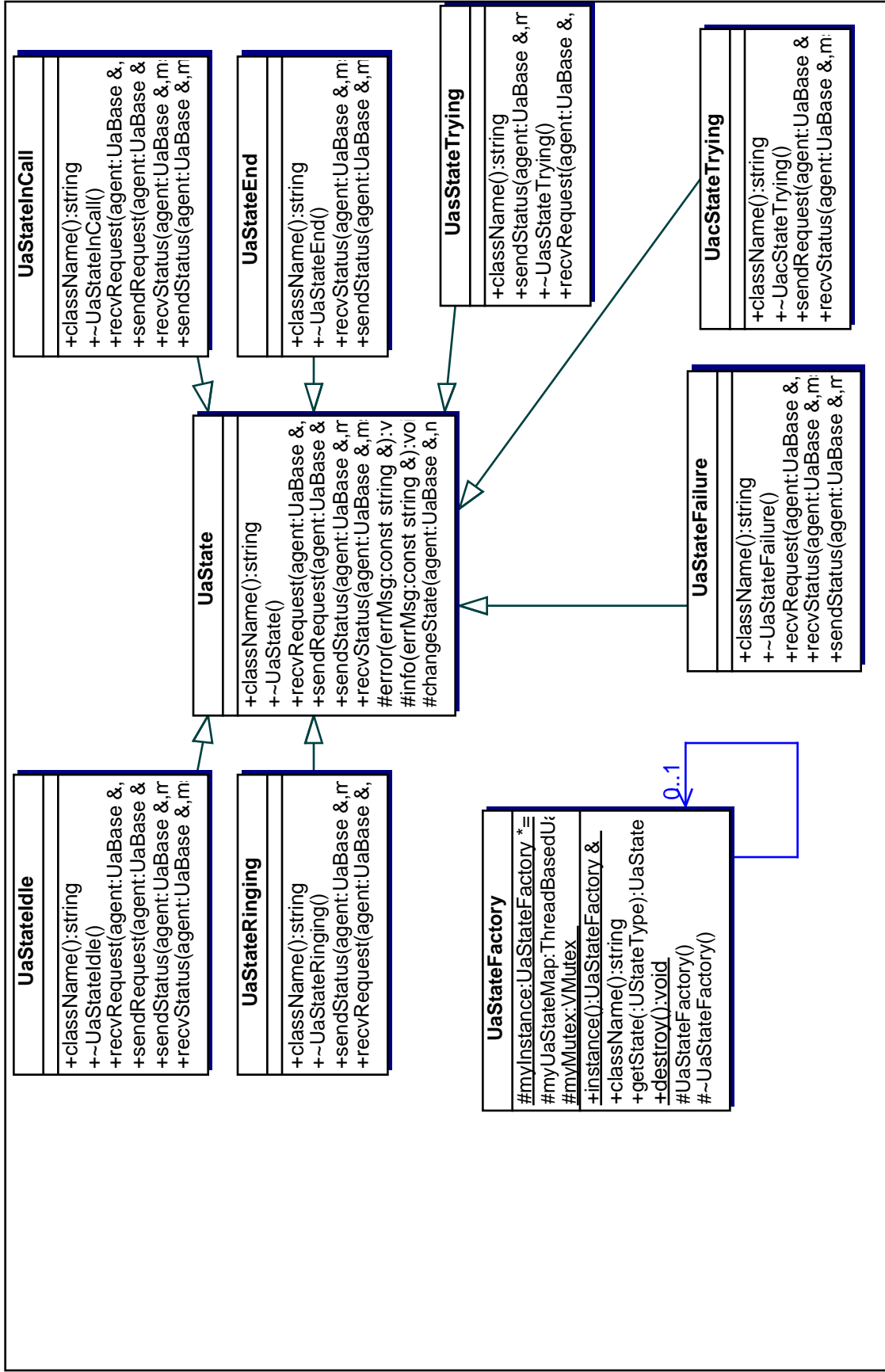
1.1, obj.debug.Linux.i686

1.1, obj.opt.Linux.i686



1.1, obj.opt.Linux.i686

1.1, uaState



1.1, uaState