Set of Models

SystemsEngineeringModel

subsystem models initially import by copy their portion of the System Engineering Model

«import»

«import»

«import»

«import»

FireControl_Subsystem_Model

HUD_Subsystem_Model

Targetting_Subsystem_Model

TrackManagement_Subsystem_Model

«import»

«import»

«usage»

«usage»

«usage»

«usage»

SharedModel

SystemRequirements

Domains

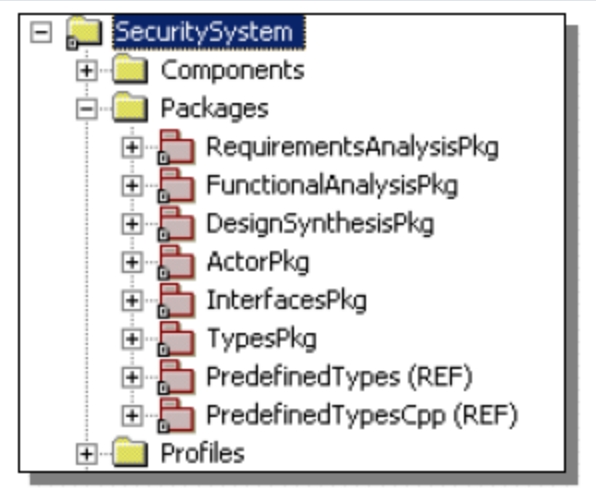
SystemsArchitecture

The subsystem models use by reference the packages in the domains

The Shared Model imports by copy and manages the subsystem interfaces and overall subsystem architecture.

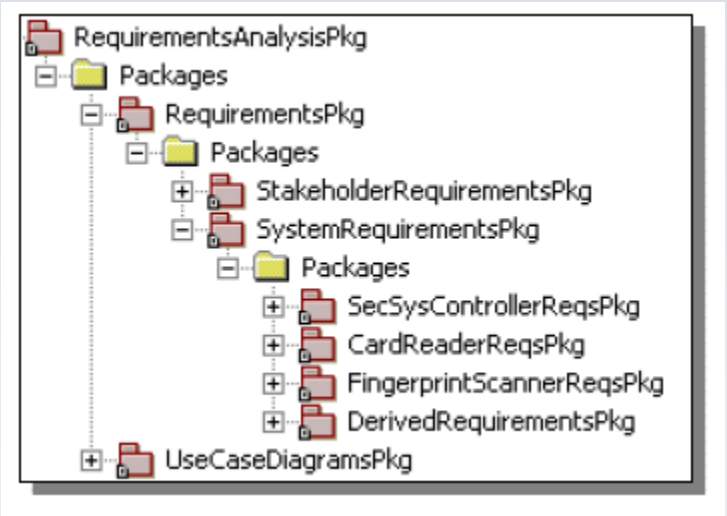
System Level Model Organization

- Recommended Model organization has packages for
  - Requirements
  - Functional (Black box) analysis)
    - Subpackage per use case
  - Architectural Design
    - Package per subsystem
      - This package is the primary hand off to downstream engineering



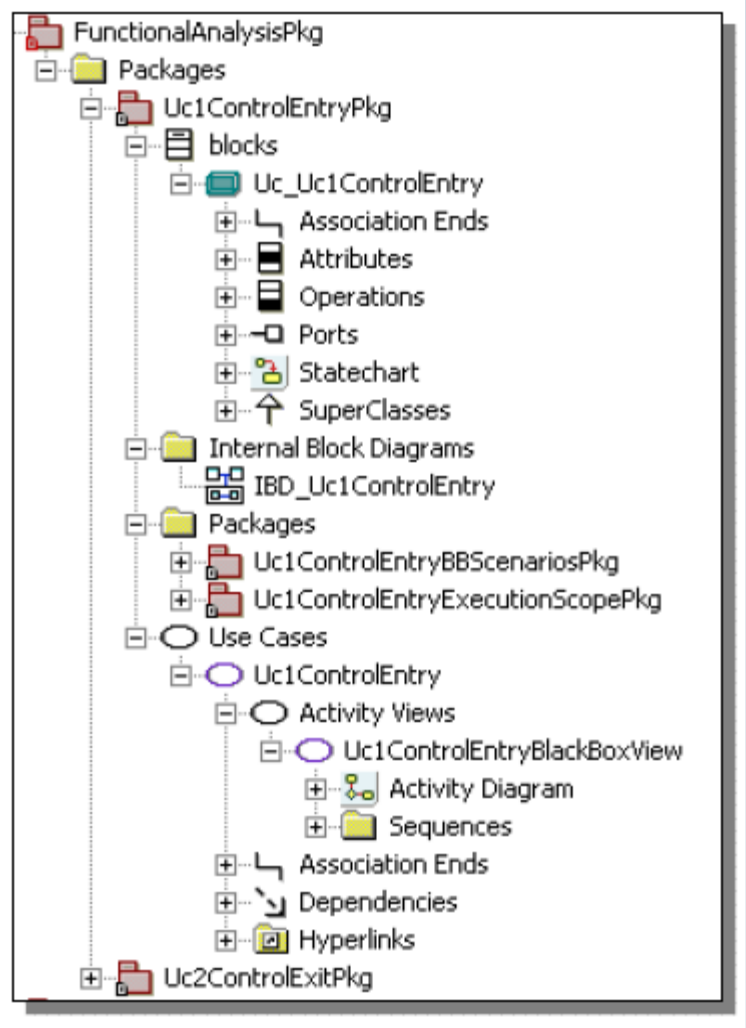
Requirements Analysis Package

- Generally organized around the use cases to which the requirements are allocated
- These requirements may be imported from DOORS or other requirements management tools
- This package has the use cases and use case diagrams but not the more detailed views (e.g. sequence, activity, and state diagrams)



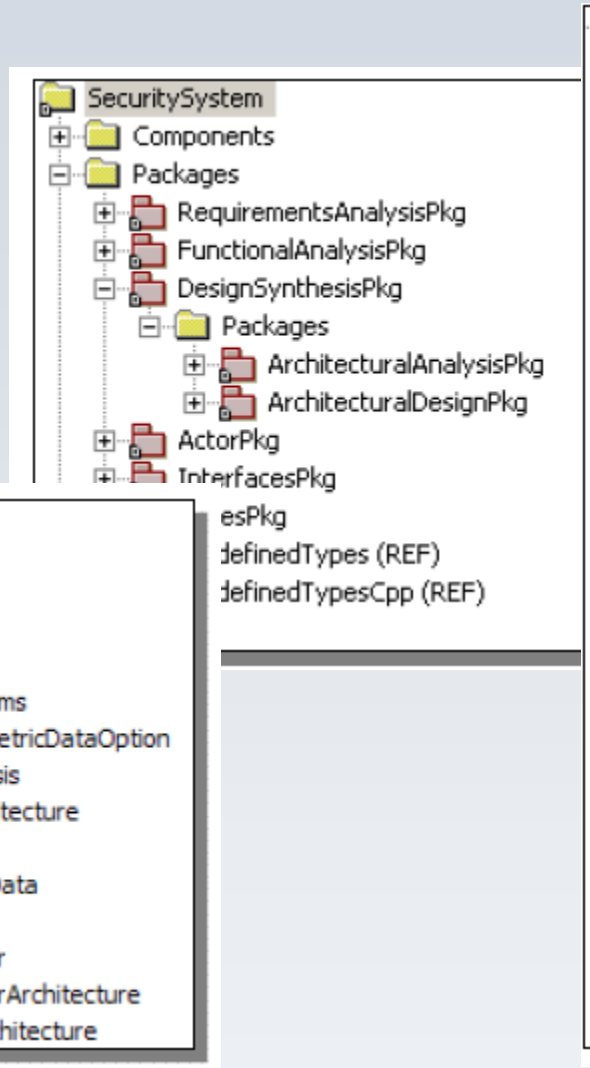
Functional Analysis Package

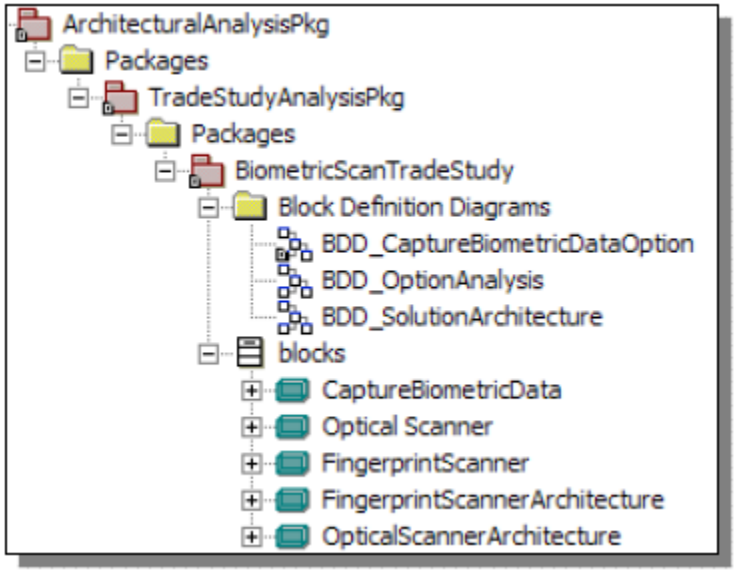
- This package holds the black box use case analysis.
- It is organized on a per use case basis

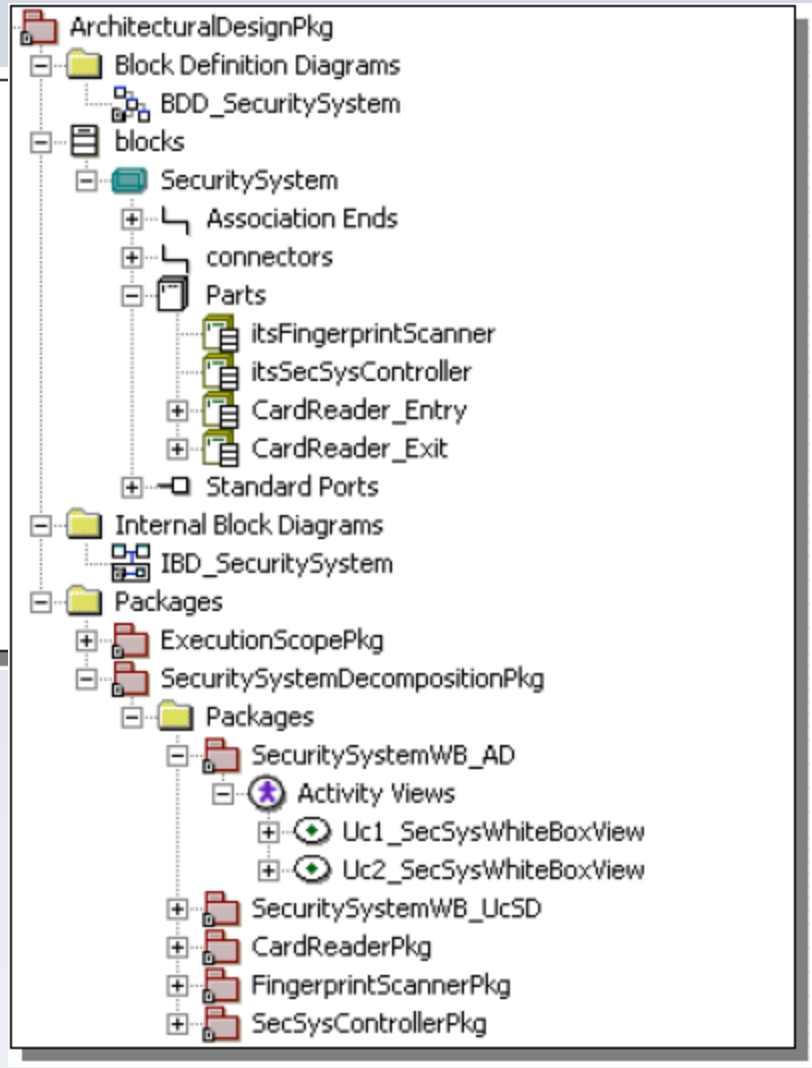


Design Synthesis Package

- Subdivided into
  - Architectural analysis package (for trade studies)
  - Architectural design package (for subsystem specification)

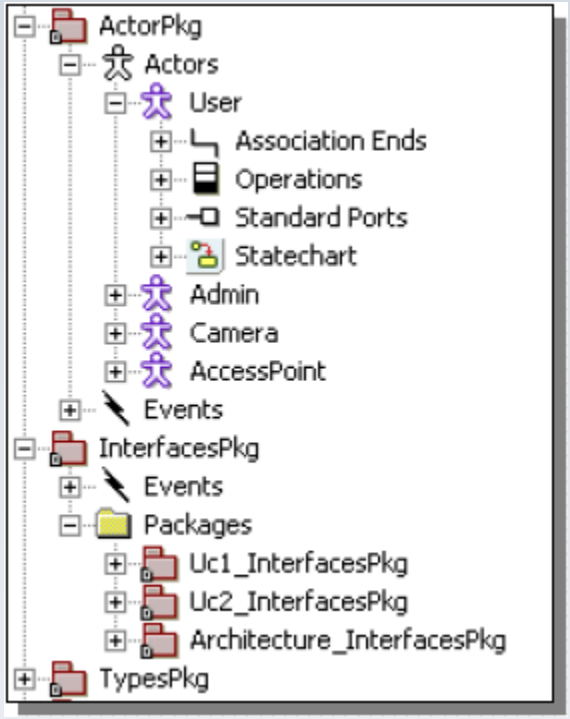




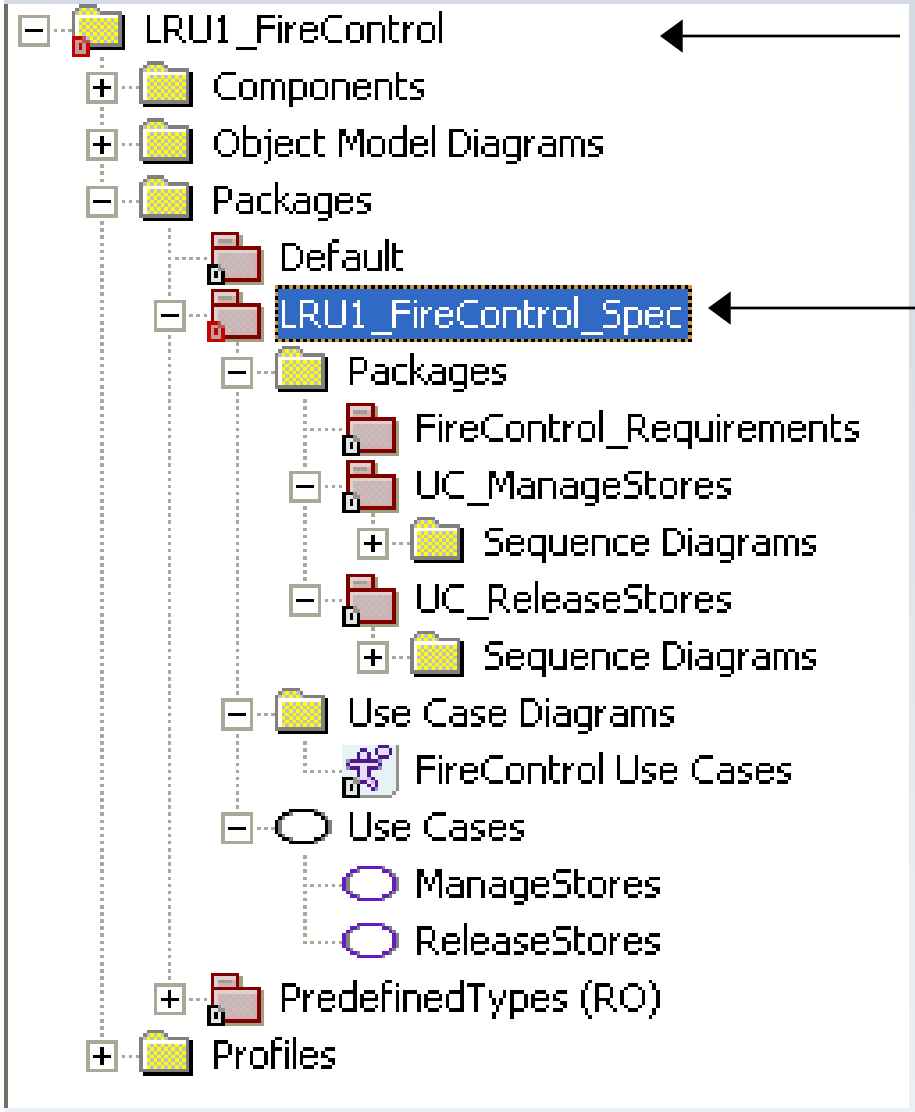


System Level Definitions

- Top level packages for
  - Actors
  - Interfaces
  - Data types



Subsystem Models

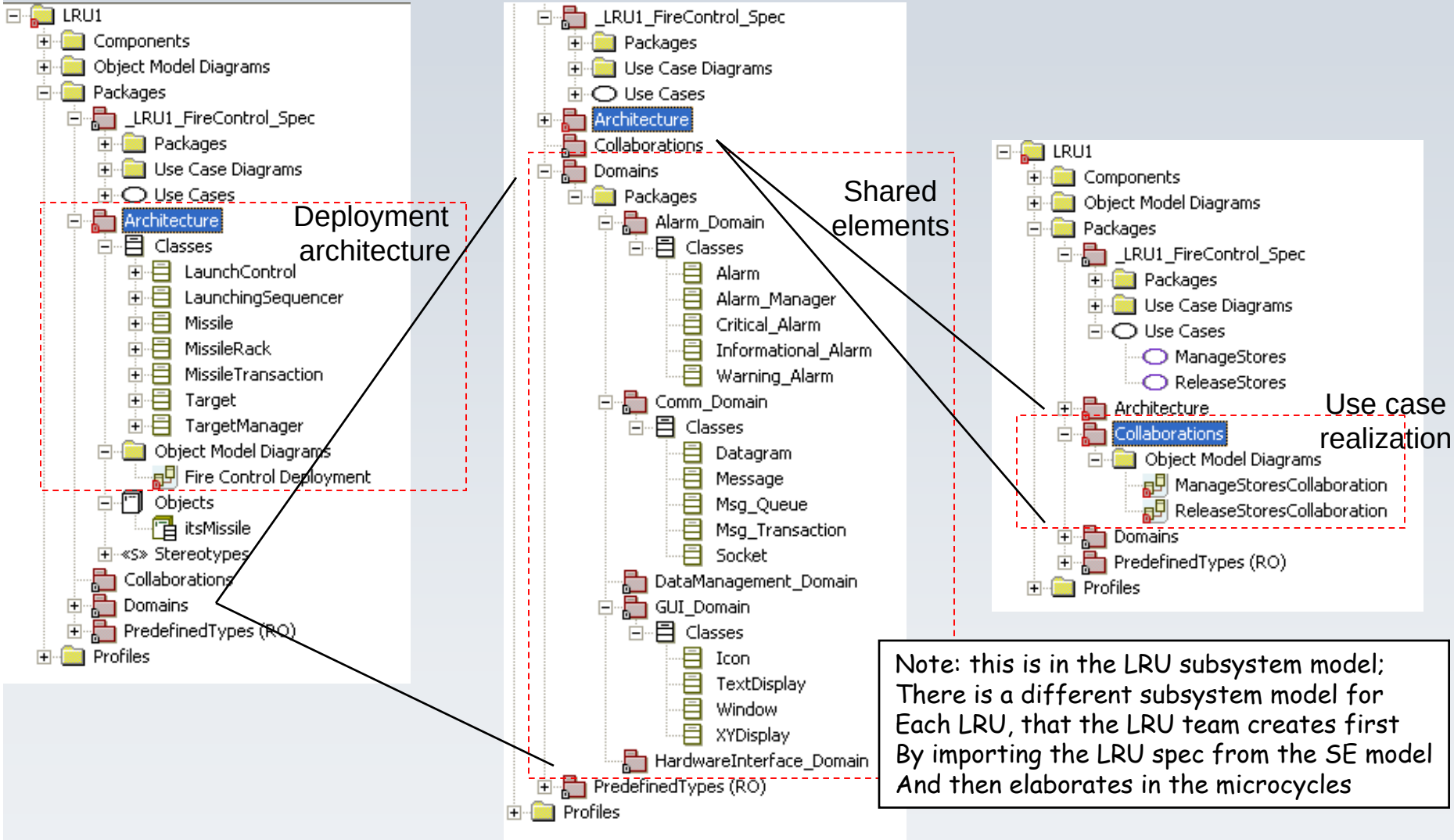


This is the model for this particular subsystem. This Starts with a new model being created with Add to Model To add just the specification for this subsystem

The specification imported from the SE model

The IPT then takes this model and identifies the hw/sw tradeoffs and formulates a deployment architecture for this LRU

Elaborated Subsystem Model Org



Shared Model

Shared Model

SubsystemInterfaces

Subsystem interfaces are used by the SE model to store the interaces used between subsystems. They are meant to be used by the SE model to store the information and used by the SE model to support execution. The subsystem models use these interfaces to specify what contracts they offer and require.

PhysicalArchitecture

Shows the overall subsystem architecture and contains the subsystems, their ports, and references to their interfaces.

Domains

Alarm_Domain

Comm_Domain

«Usage»

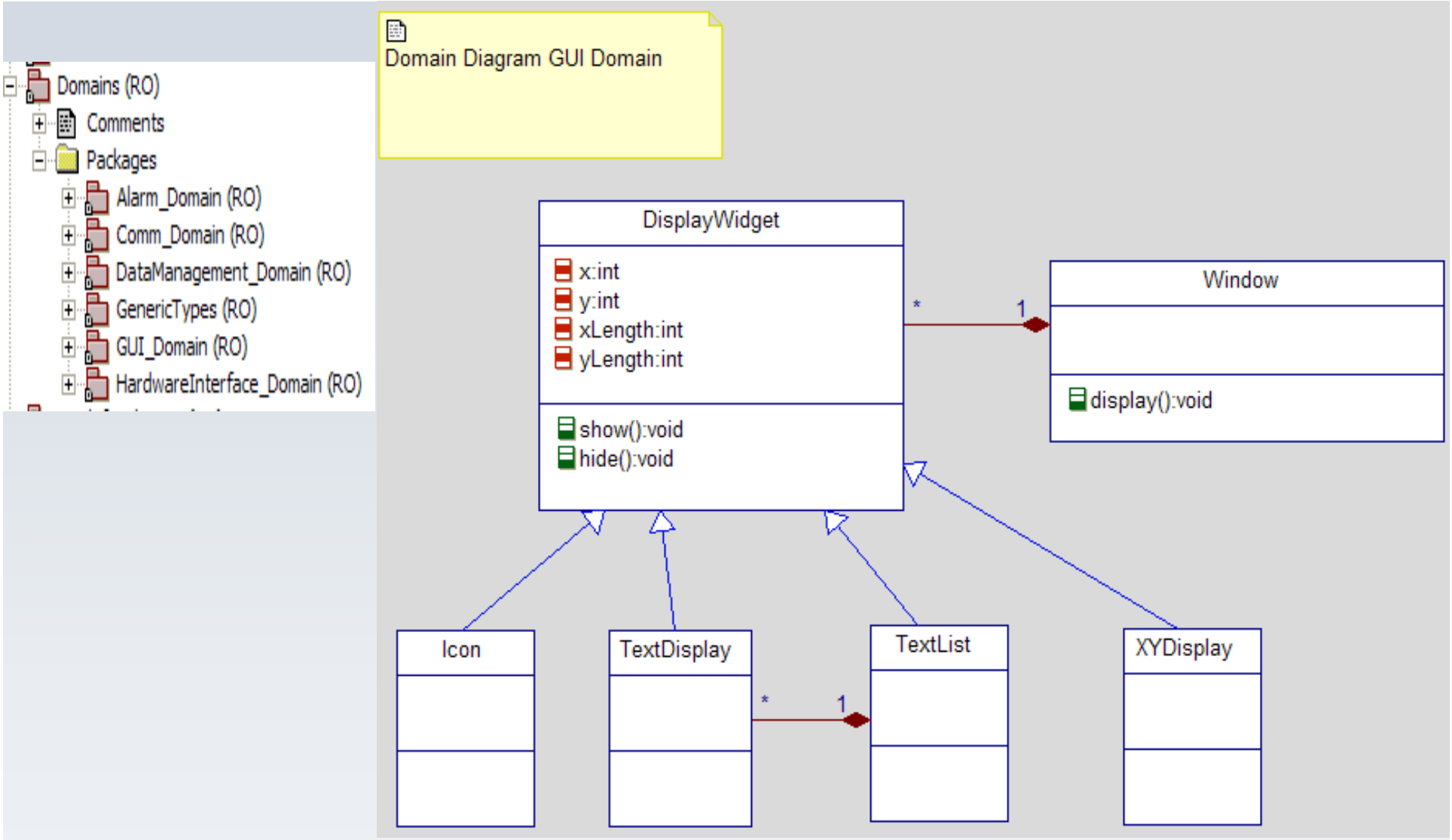
DataManagement_D

GUI_Domain

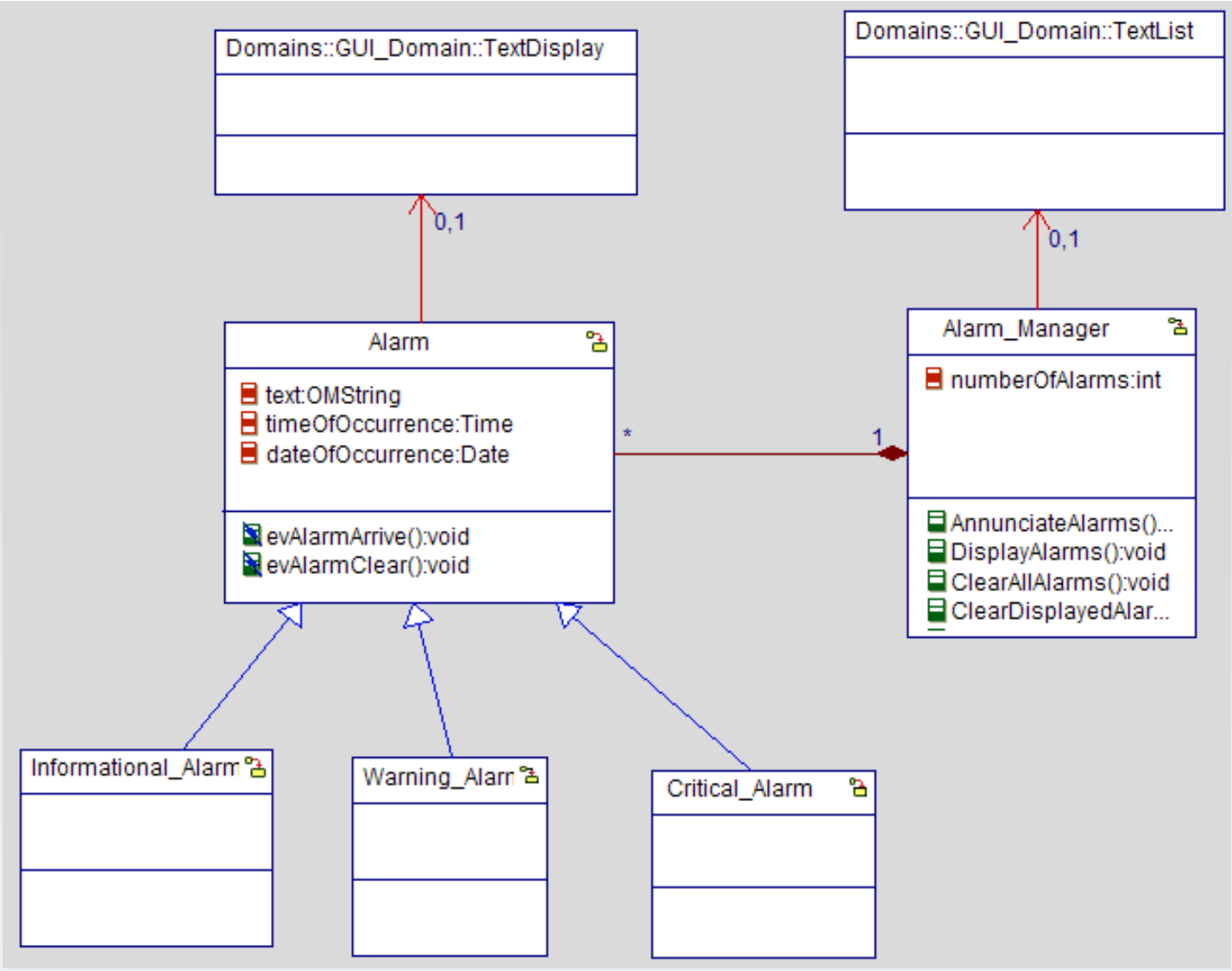
HardwareInterface_I

The Domains package stores classes shared, or potentially shared, by the subsystem teams in their internal collaborations. Primarily it is anticipated that they are "owned" (r/w rights) by a "domain owner" - a person in charge of maintaining the semantic integrity of the classes within the domain. They are used BY REFERENCE by the subsystem teams. Using them by reference ensures that as the domain is elaborated, the contents are published to the subsystem teams.

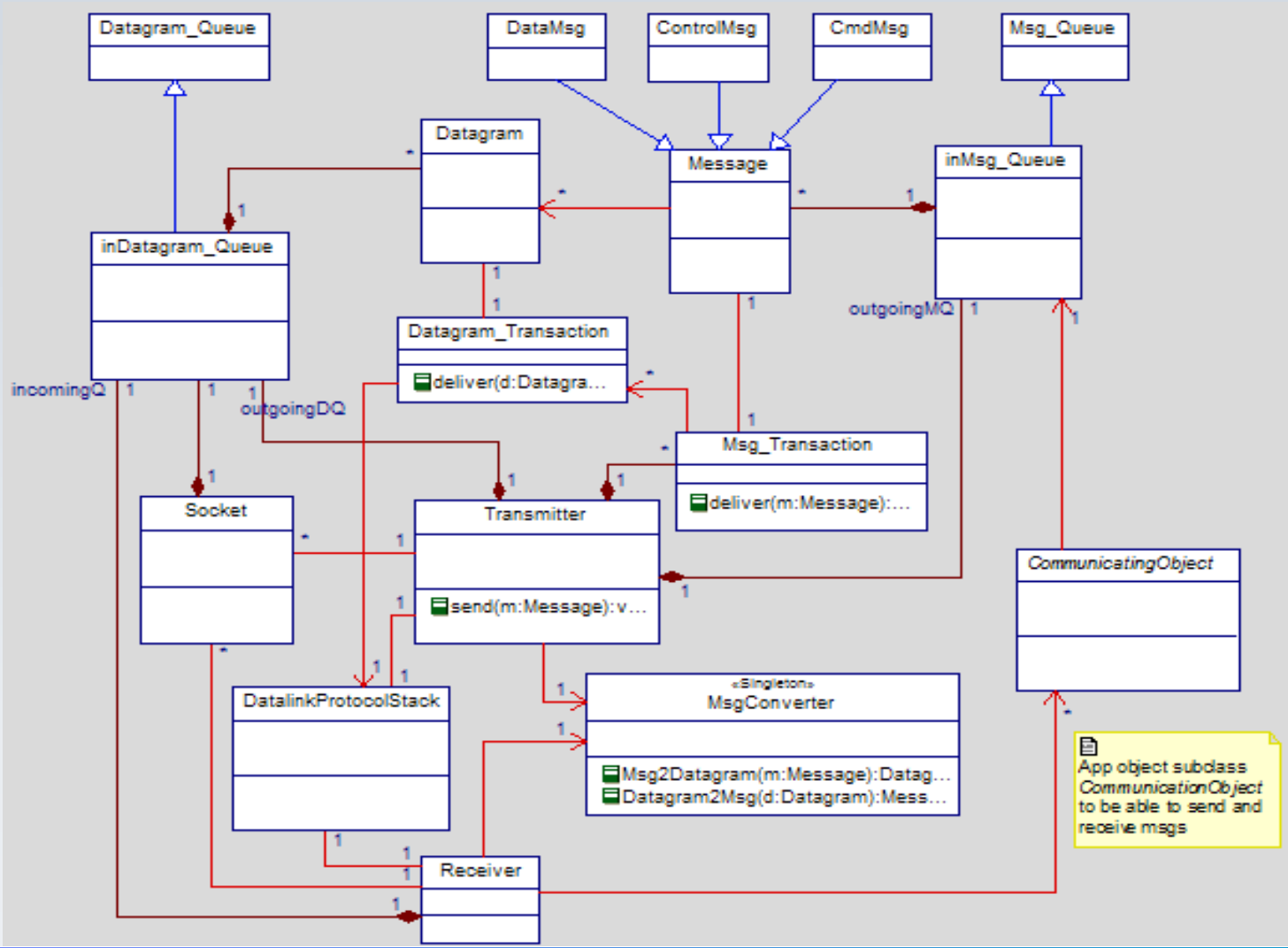
Domains


Domains (RO)
Comments
Packages
Alarm_Domain (RO)
Comm_Domain (RO)
DataManagement_Domain (RO)
GenericTypes (RO)
GUI_Domain (RO)
HardwareInterface_Domain (RO)
Domain Diagram GUI Domain
DisplayWidget
x:int
y:int
xLength:int
yLength:int
show():void
hide():void
Window
display():void
*
1
Icon
TextDisplay
TextList
XYDisplay
*
1

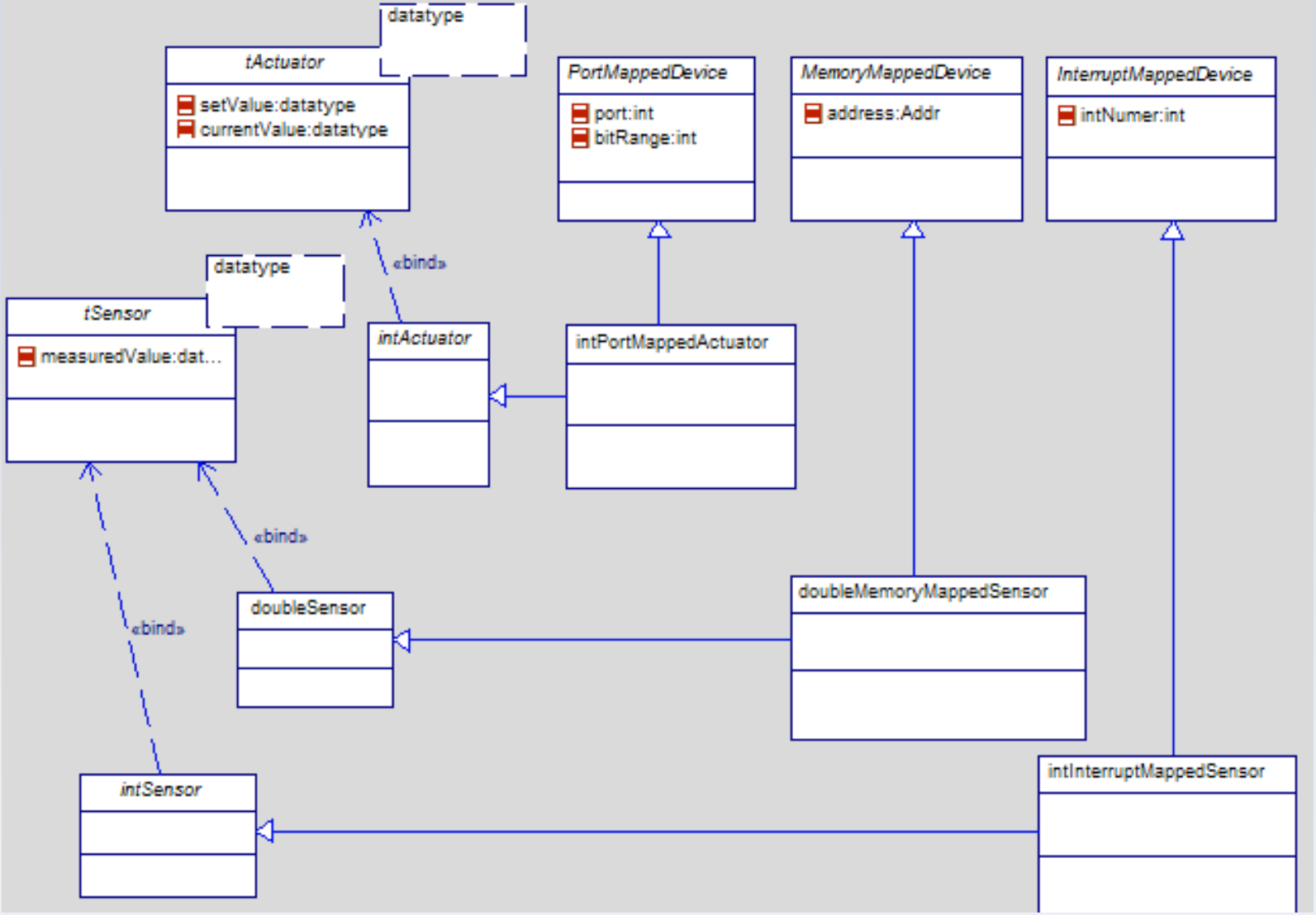
Domains


Domains::GUI_Domain::TextDisplay
Domains::GUI_Domain::TextList
0,1
0,1
Alarm
text:OMString
timeOfOccurrence:Time
dateOfOccurrence:Date
evAlarmArrive():void
evAlarmClear():void
*
1
Alarm_Manager
numberOfAlarms:int
AnnunciateAlarms()...
DisplayAlarms():void
ClearAllAlarms():void
ClearDisplayedAlar...
Informational_Alarm
Warning_Alarr
Critical_Alarm

Domains


Datagram_Queue
DataMsg
ControlMsg
CmdMsg
Msg_Queue
Datagram
Message
inMsg_Queue
inDatagram_Queue
outgoingMQ
Datagram_Transaction
deliver(d:Datagra...
incomingQ
outgoingDQ
Msg_Transaction
deliver(m:Message):...
Socket
Transmitter
send(m:Message):v...
CommunicatingObject
«Singleton»
MsgConverter
DatalinkProtocolStack
Msg2Datagram(m:Message):Datag...
Datagram2Msg(d:Datagram):Mess...
Receiver
App object subclass CommunicationObject to be able to send and receive msgs

Domains


datatype
tActuator
setValue:datatype
currentValue:datatype
PortMappedDevice
port:int
bitRange:int
MemoryMappedDevice
address:Addr
InterruptMappedDevice
intNumer:int
«bind»
datatype
tSensor
measuredValue:dat...
intActuator
intPortMappedActuator
«bind»
«bind»
doubleSensor
doubleMemoryMappedSensor
intSensor
intInterruptMappedSensor

Using Domains