Object-Oriented Systems Engineering Method (OOSEM)

Synthesize Candidate Physical Architectures

Sanford Friedenthal
safriedenthal@gmail.com

OOSEM Topics

- OOSEM Overview
- Method
  - Setup Model
  - Analyze Stakeholder Needs
  - Analyze System Requirements
  - Define Logical Architecture
  - Synthesize Candidate Physical Architectures
  - Optimize and Evaluate Alternatives
  - Manage Requirements Traceability
  - Integrate and Verify System
- Summary

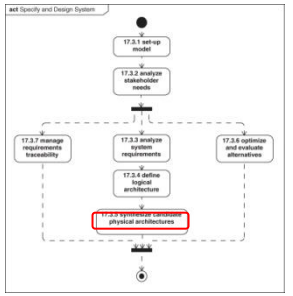
Module Objectives

- After completion of this module, student should understand
  - The primary modeling artifacts from Synthesize Candidate Physical Architectures
  - Considerations for partitioning system functionality, data, and control
  - How to distribute logical components across the system nodes
  - How to allocate logical components to physical components (i.e., hardware, software, data, procedures)
  - How to synthesize the system physical architecture
  - How to specify the physical components of the system

Motivation

- Synthesize alternative physical architectures that can satisfy the system requirements
- Specify the physical components of the system including the hardware, software, data, and procedures

Synthesize Candidate Physical Architectures



- Define how logical components are distributed across nodes
- Allocate the logical components to physical components that are implemented in hardware, software, data, and procedures
- Define hardware, software, data, and other cross-cutting views of the system architecture

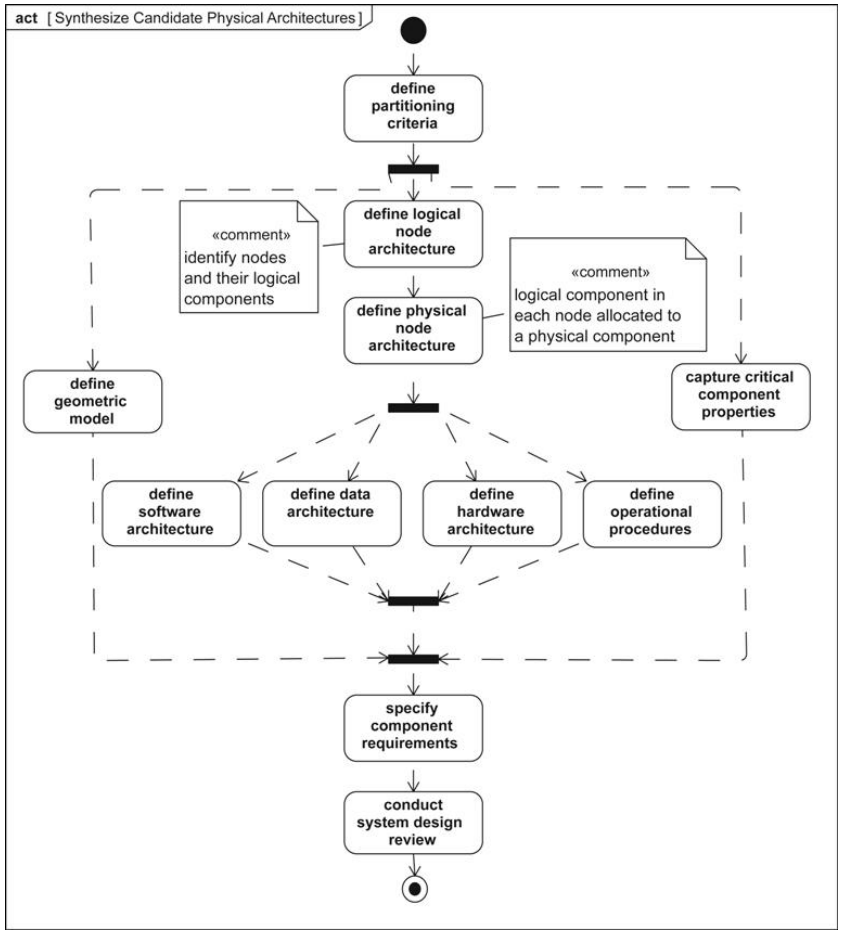


Figure 17.26

Define Partitioning Criteria

- Define criteria to:
  - Partition functionality, data, and control among the logical and physical components
  - Partition the components among subsystems, nodes, and layers of the architecture
- Apply partitioning criteria to:
  - Result in designs that exhibit maximum cohesion and minimum coupling to reduce interface complexity
  - Reduce the impact of requirements and technology changes
  - Address key requirements such as performance, reliability, maintainability, and security
  - Leverage ‘Design for X’ techniques, (such as Design for Test) to improve design quality

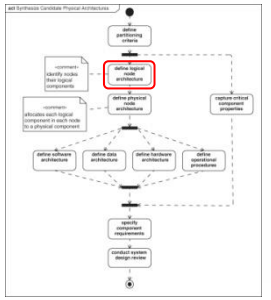
Partitioning Examples

- ESS Example
  - Event Detection Manager and System Controller partition common functionality for detecting events and controlling actions in response to inputs from multiple sensors
    - Can add sensors (e.g., carbon monoxide) with minimal impact
- Other example
  - Higher failure rate parts should be physically partitioned so that they can readily be accessed and replaced

Partitioning Criteria

- Partitioning criteria may address:
  - Modularity (maximize cohesion, minimize coupling)
  - Component isolation (encapsulation, hiding)
  - Reuse, COTS, and design constraints
  - Common functions, inputs and outputs
  - Data (dynamic vs. static, shared)
  - Performance (accuracy, response time, throughput)
  - Temporal (concurrency, update rates, phasing, events)
  - Reliability/maintainability (failure rate, time to repair)
  - Producability (e.g. design for assembly)
  - Safety and security (access control)
  - Physical/environmental (e.g. location, power)
  - Design for change (dependencies)
  - Subcontractor/development responsibility
  - Life cycle phase or development increment

Define Logical Node Architecture



- Nodes used to partition components based on location or other criteria
- ESS nodes include the Site Installations and the Central Monitoring Station

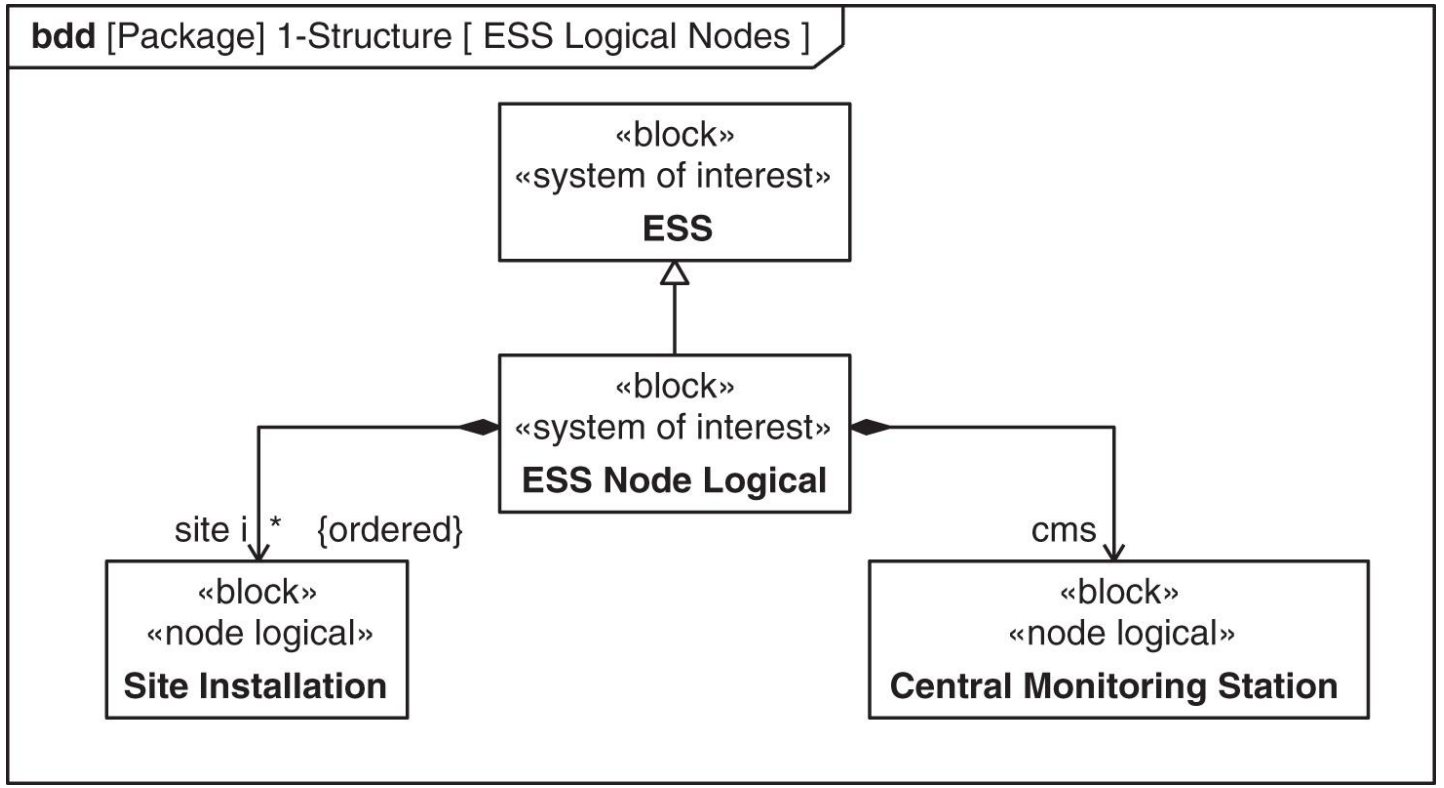
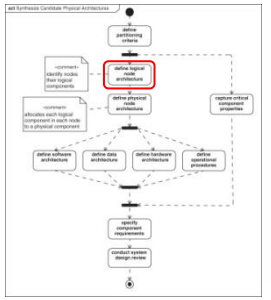
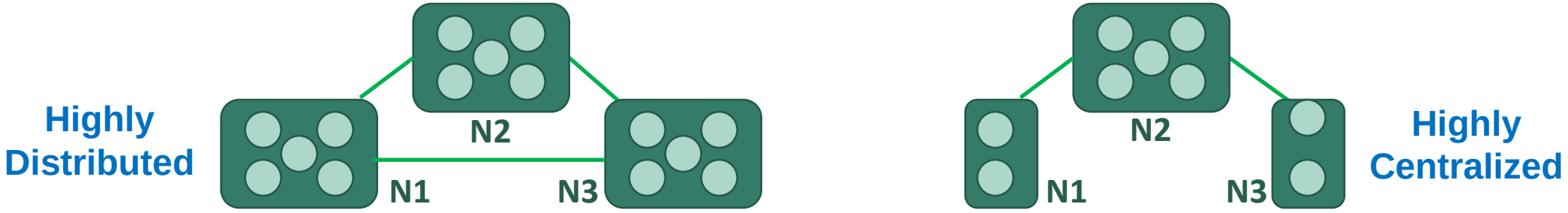


Figure 17.25

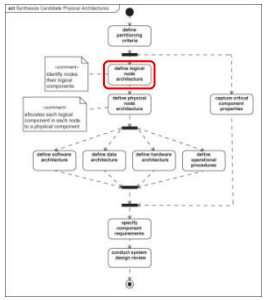
Define Logical Node Architecture (cont.)



- Options for the distribution of functionality, control, and data
  - Highly distributed, such that each node can operate autonomously
  - Highly centralized, where most functionality, control, and data reside at a central node, and local nodes just provide an interface at a particular location (e.g., central ac unit for a hotel with console i/f in each room)
  - Partially distributed, where each node performs a subset of the total functionality, control, and data
- Trade studies are performed to optimize the distribution based on performance, availability, security, and cost



Define Logical Node Architecture (cont.)



- A logical node aggregates logical components at a location
  - The selection of components at a node determine the functionality, control, and data at that location

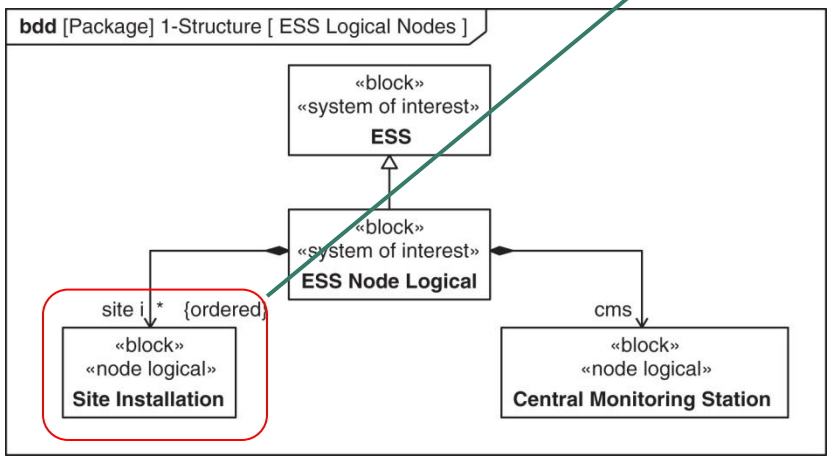


Figure 17.25

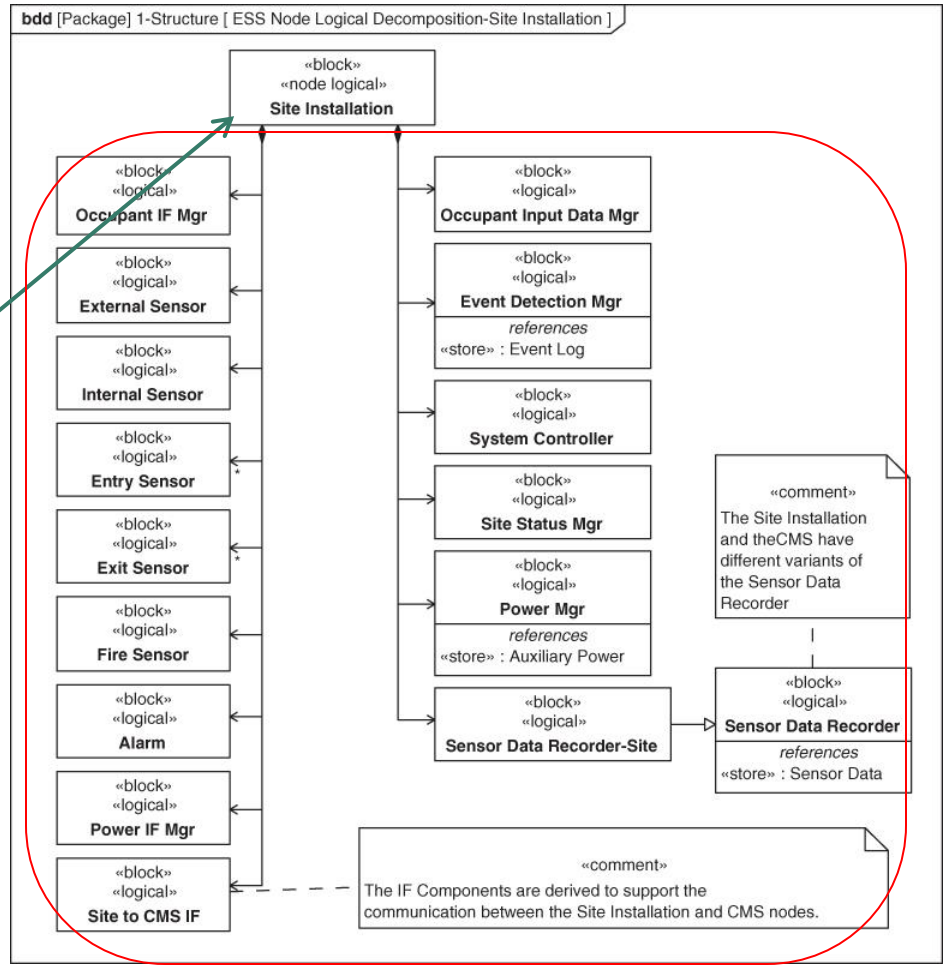
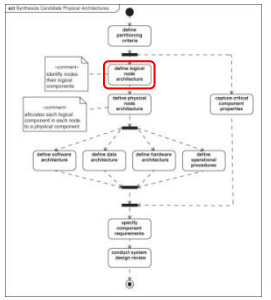


Figure 17.26

Define Logical Node Architecture (cont.)



- Interaction of logical components distributed across the logical nodes

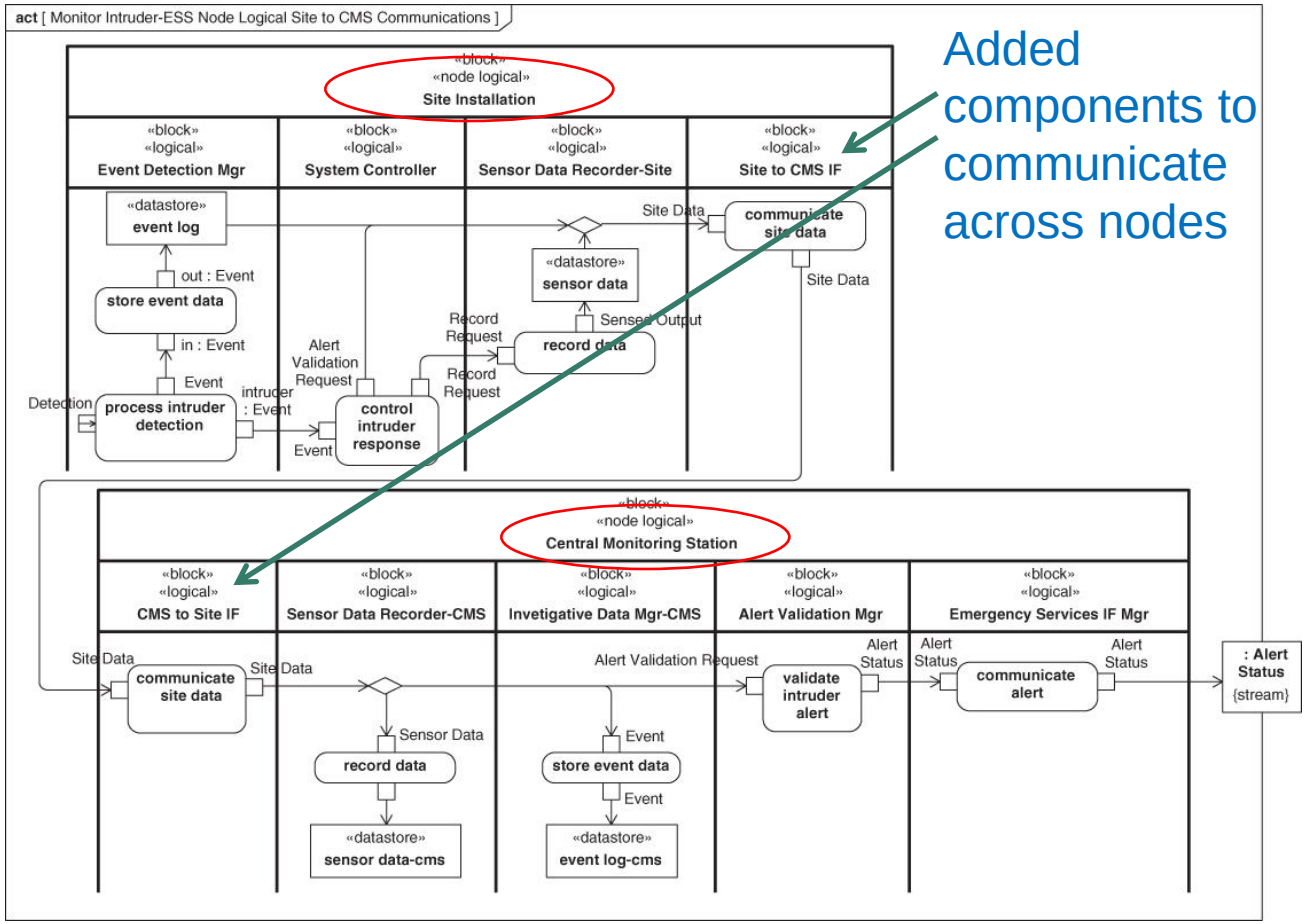
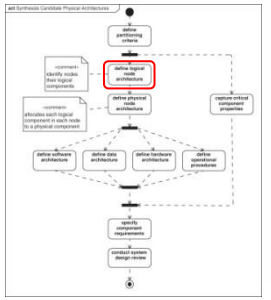


Figure 17.28

Define Logical Node Architecture (cont.)



- Interconnection among logical components at a node

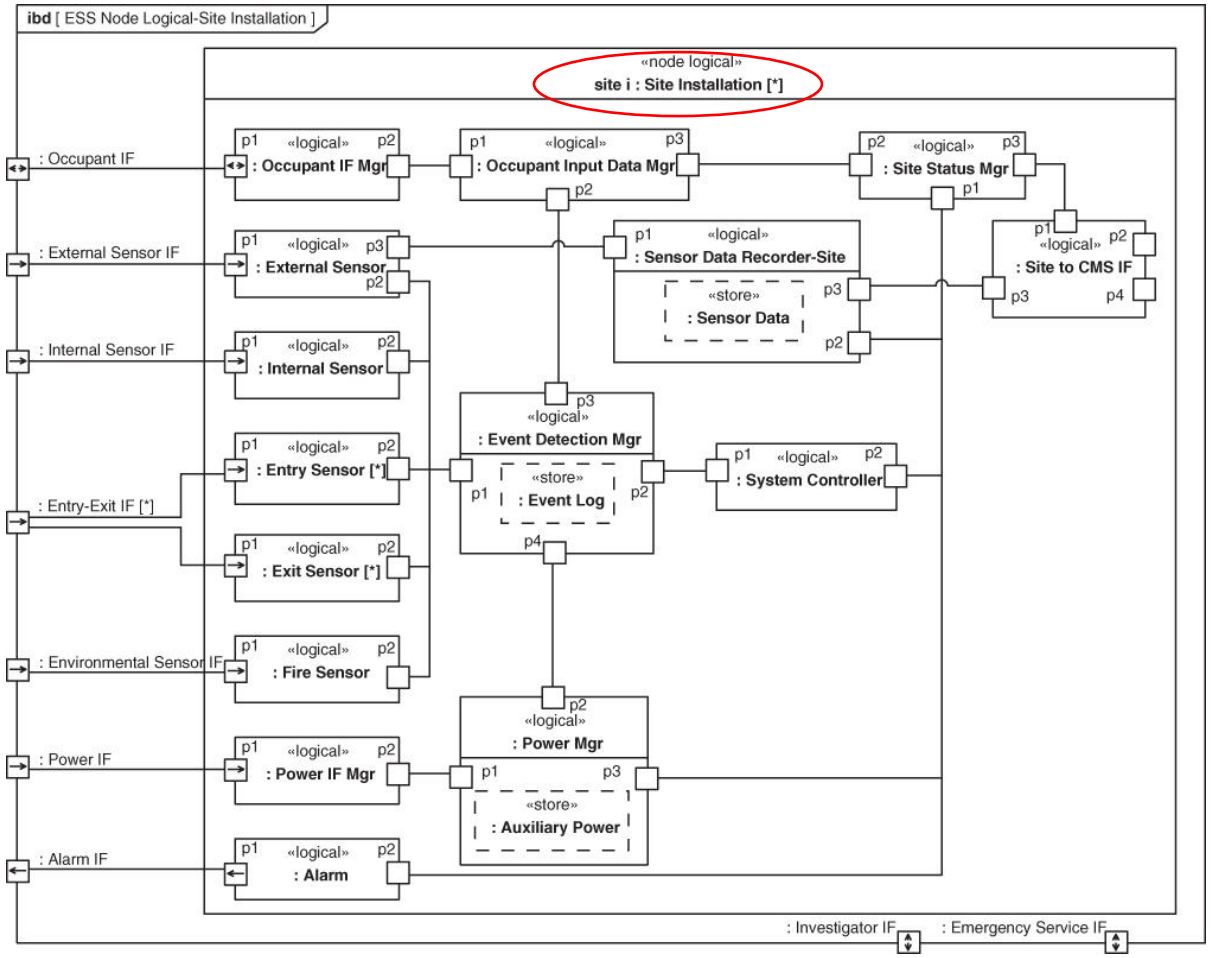


Figure 17.29

Define Physical Node Architecture

- Physical nodes
  - composed of physical components instead of logical components
  - may have a specific location
  - may include additional sizing constraints, interfaces, and other requirements specific to the node

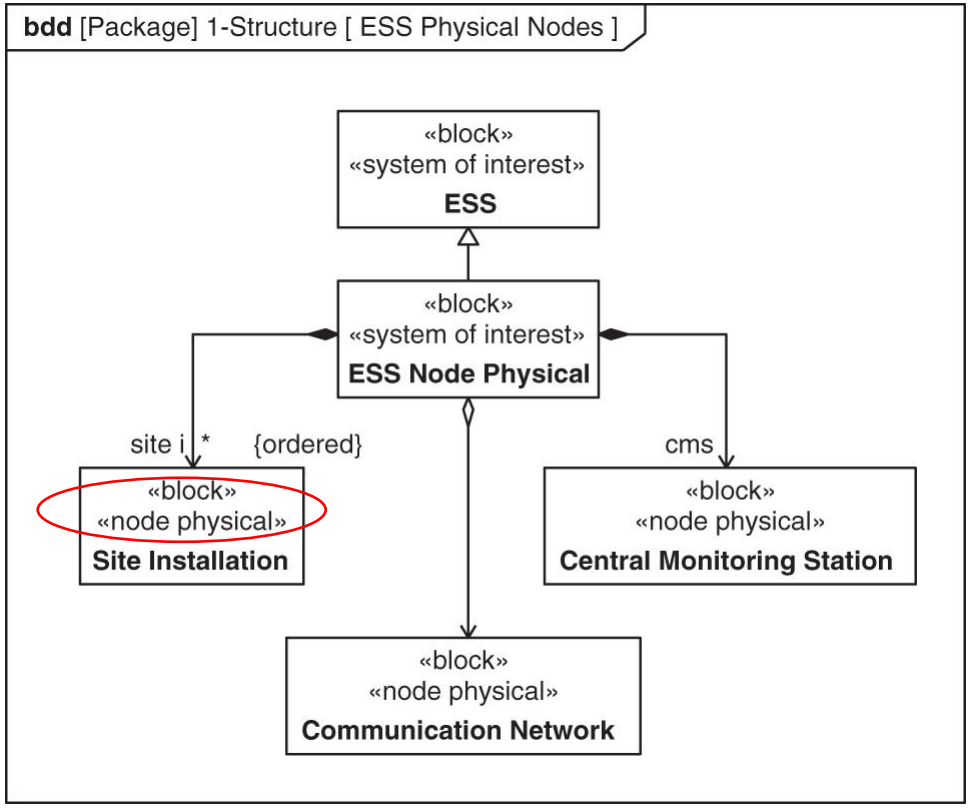


Figure 17.33

Define Physical Node Architecture (cont.)

- Logical components from each logical node are allocated to hardware, software, data, people, and facilities (e.g., physical components) at the corresponding physical node
- Allocation is based on trade-off analysis addressing technology and implementation concerns related to performance, reliability, sizing, security, cost, and other considerations
  - Capture rationale for allocate relationship
- Design constraints identified from ‘Analyze System Requirements’ are imposed on the physical design
- A logical component allocated to software must then be allocated to a hardware component to execute it

Define Physical Node Architecture (cont.)

- Logical components at each node are allocated to physical components

Physical Components

Logical

Logical to Physical Allocation (Hardware)	Auxiliary Battery	Console	Contact Sensor	Fire Detector	Hard Drive	IR Motion Detector	LAN-Site	Memory Card	Modem	Multi-Mode Alarm	Power Stri p	Power Switch	Site Hardware	Site Processor	Video Camcorder	Wiring-Site	Application and?	Cabling-CMS	CMS Hardware	Disk Drive	LAN-CMS	Modem-CMS	Video Server	Video Storage D?	Workstation
Alarm										↗															
Emergency Services IF Mgr																					↗				
Entry Sensor			↗																						
Exit Sensor			↗											↗											
External Sensor															↗										
Fire Sensor				↗																					
Internal Sensor						↗																			
Investigator IF Mgr																					↗				
Occupant IF Mgr		↗																							
Power IF Mgr											↗														
Power Mgr	↗											↗													
Sensor Data Recorder																									
CMS to Site IF																						↗			
Connection Infrastructure?																		↗			↗				
Connection Inrrastructure?							↗									↗									
Invetigative Data Mgr-CMS																				↗					
Sensor Data Recorder-CMS																							↗	↗	
Sensor Data Recorder-Site															↗										
Site to CMS IF							↗		↗																

Figure 17.31

Define Physical Node Architecture (cont.)

- Each physical node aggregates physical components

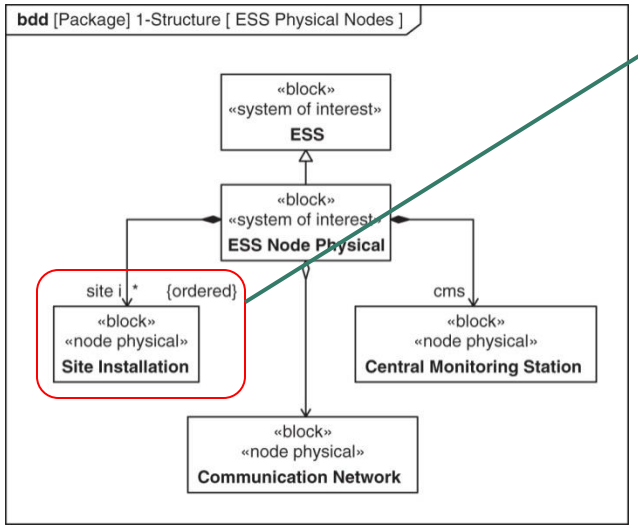


Figure 17.33

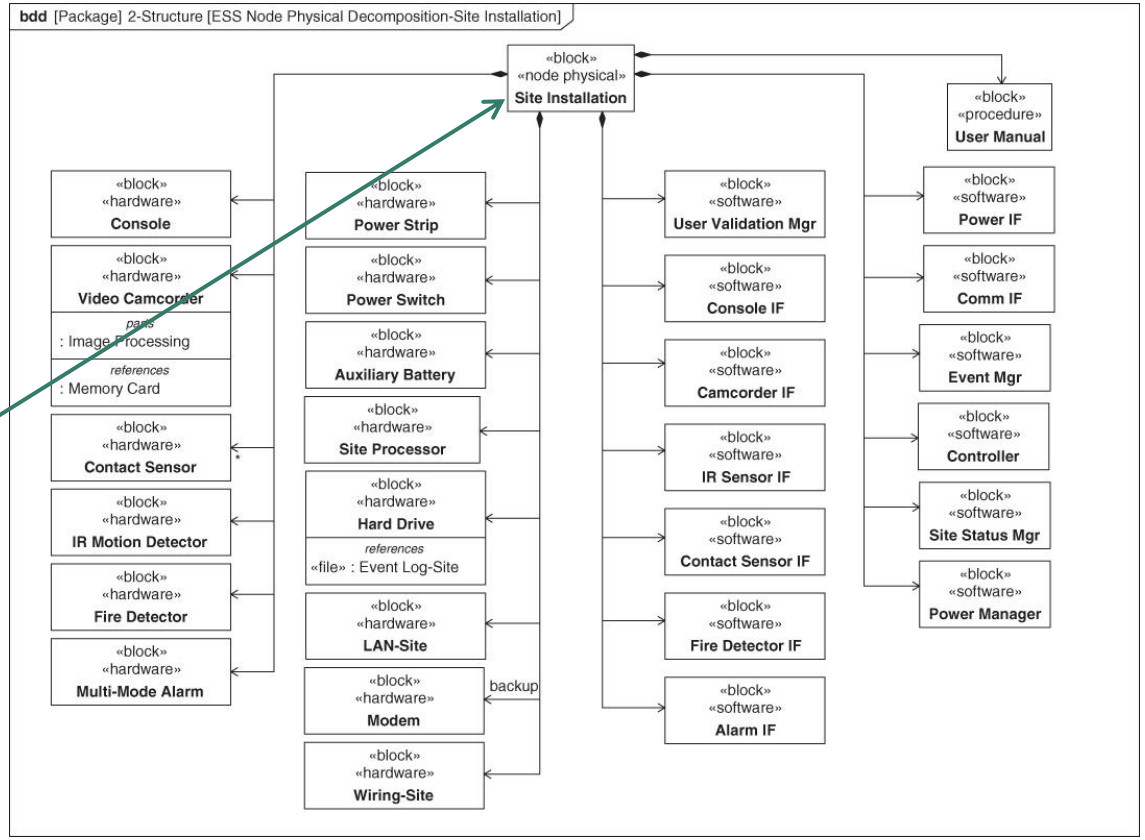


Figure 17.34

Define Physical Node Architecture (cont.)

- Interaction of physical components across nodes

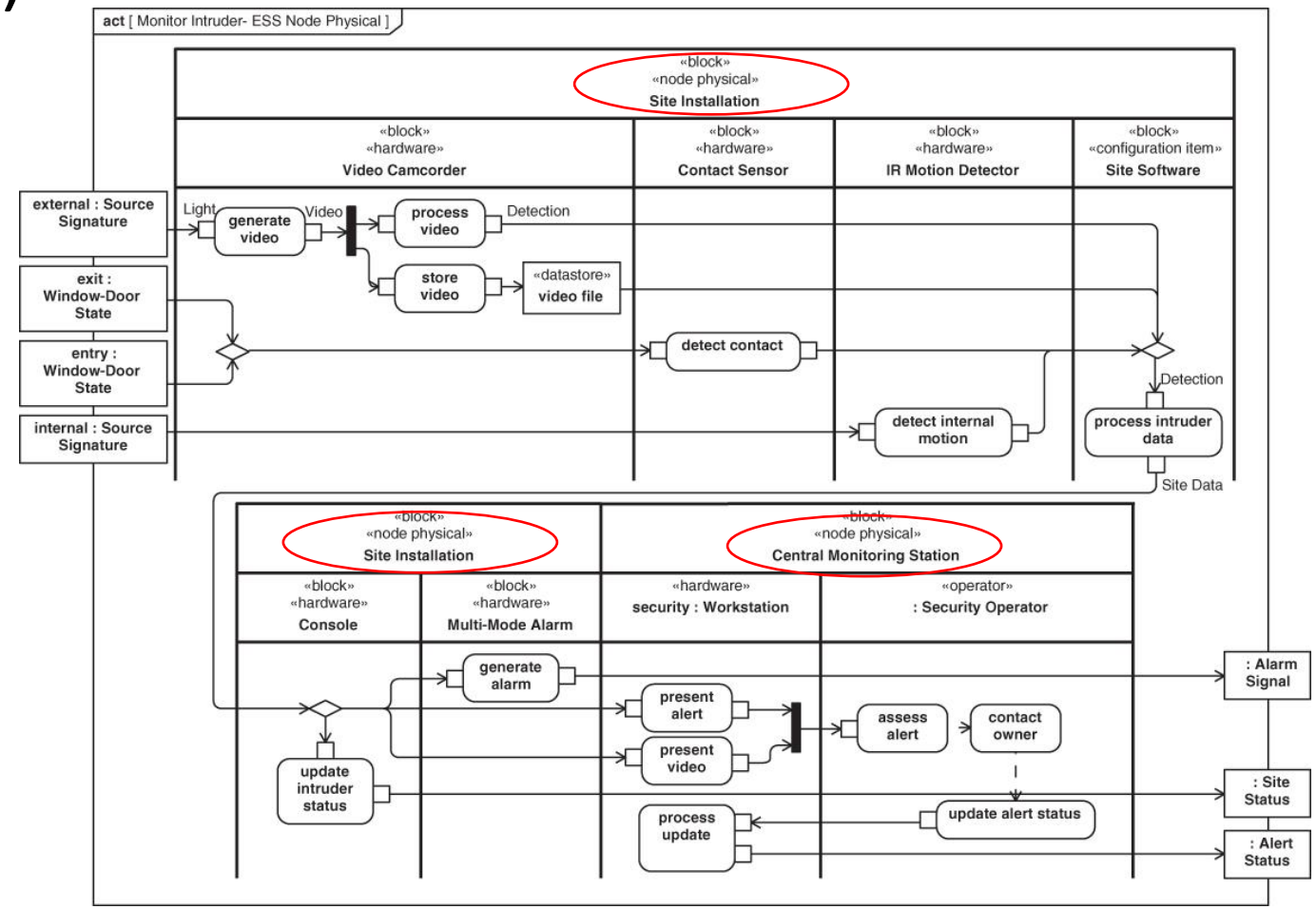
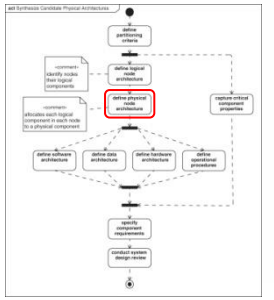


Figure 17.36

Define Physical Node Architecture (cont.)



- Interconnection among physical components at a node

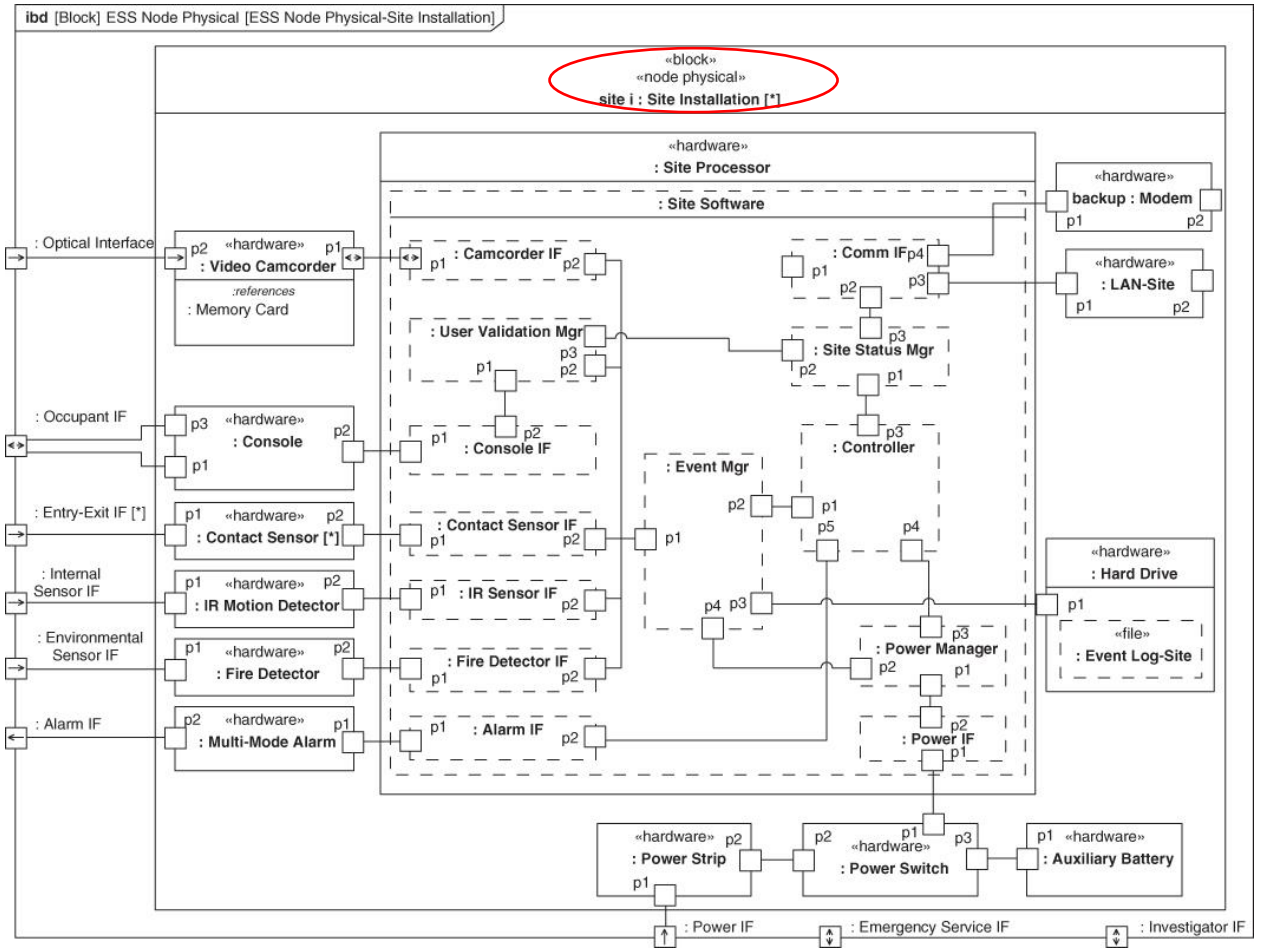


Figure 17.37

Define Software Architecture

- A view of the system architecture that includes software components and their relationships
- The software components at a particular node are aggregated into software configuration items

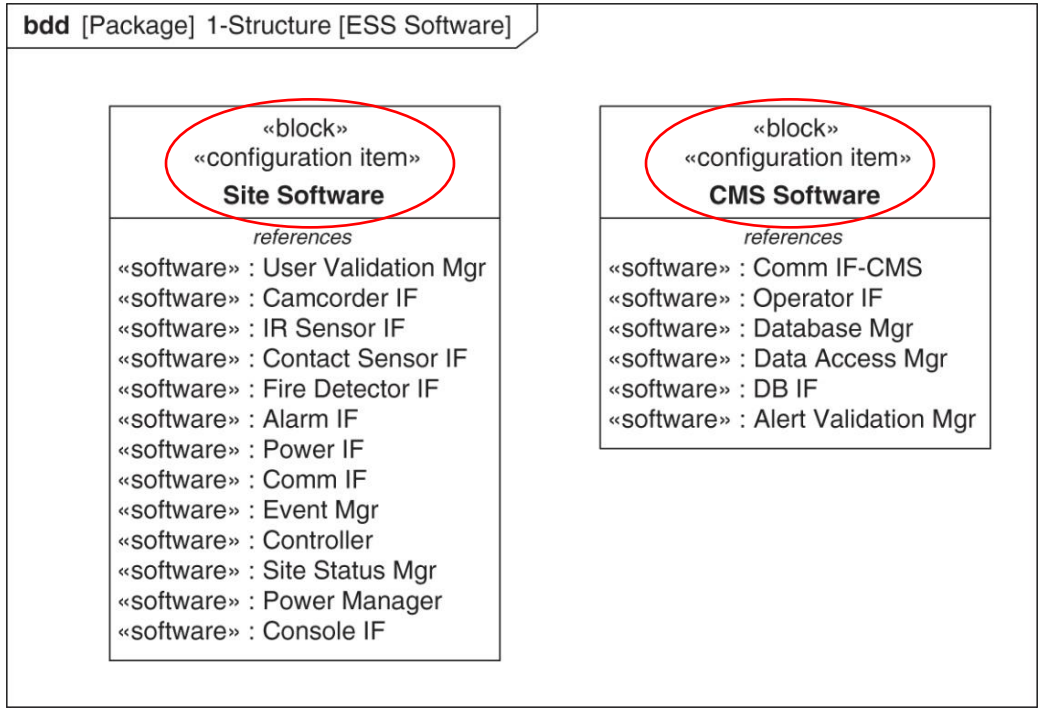
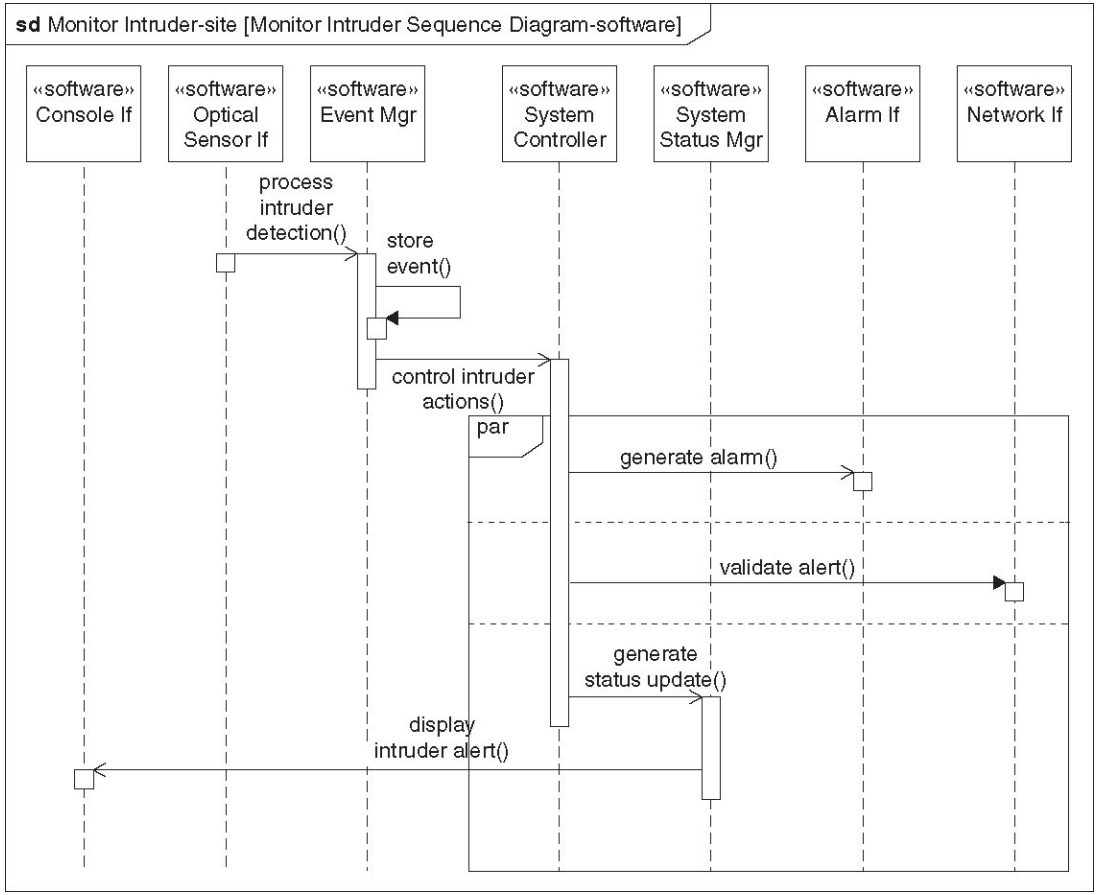


Figure 17.39

Define Software Architecture (cont.)

- Sequence Diagrams can be used to specify the interaction between the software components to derive additional software requirements



Define Software Architecture (cont.)

- Domain specific architecting considerations are applied for enterprise & embedded real-time applications
- UML and the associated software development methodology may be used to further elaborate the software architecture and design
- Traceability is maintained between the black box software component specifications in the system architecture and the software design components

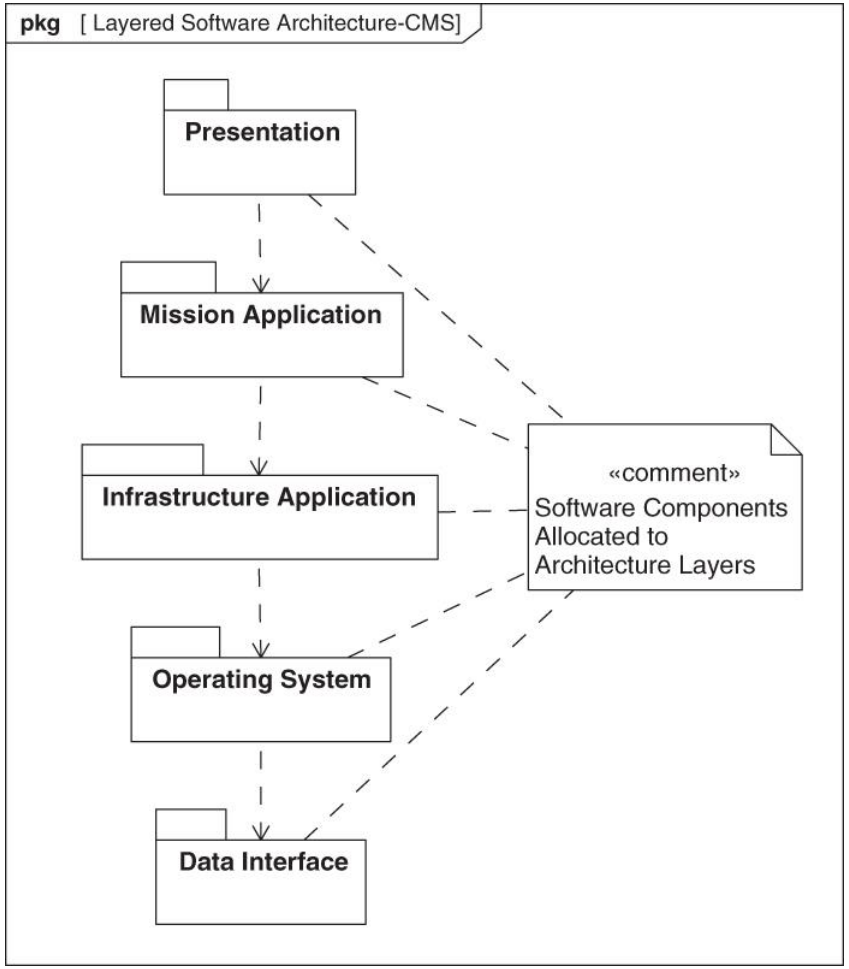
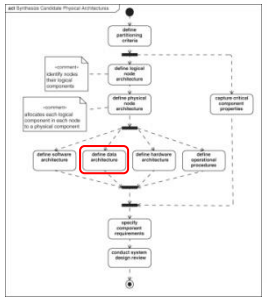


Figure 17.40

Define Data Architecture



- A view of the system architecture that includes the persistent data components
  - Persistent data derived from the scenario analysis and input/output definitions
  - Data stored by a physical component
  - Domain specific trade-offs and considerations determine how data is stored, synchronized, accessed, etc.

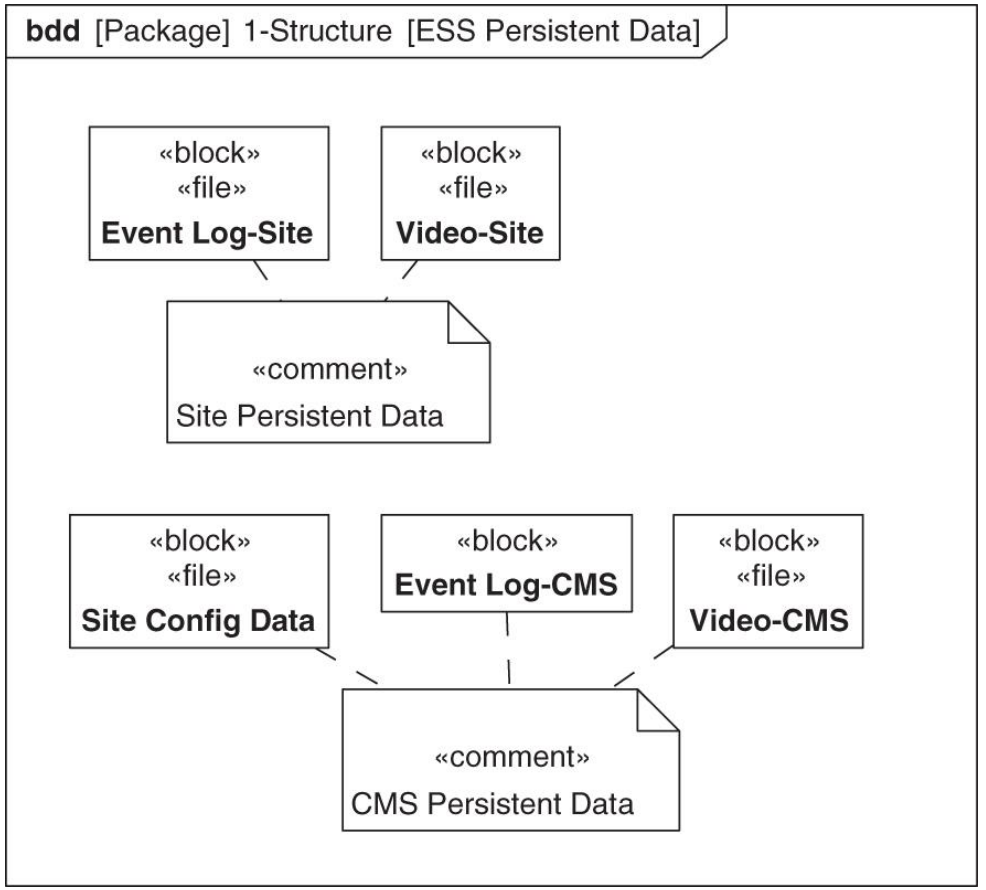
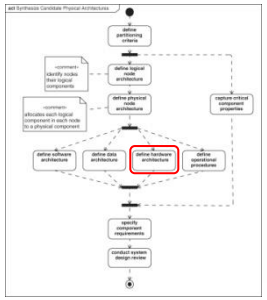


Figure 17.41

Define Hardware Architecture



- A view of the system architecture that includes the hardware components and their relationships
- The hardware components at a particular node are aggregated into hardware configuration items or assemblies
- Must address impact on sizing, layout, power, reliability, producibility
- Map components to BOM/CAD

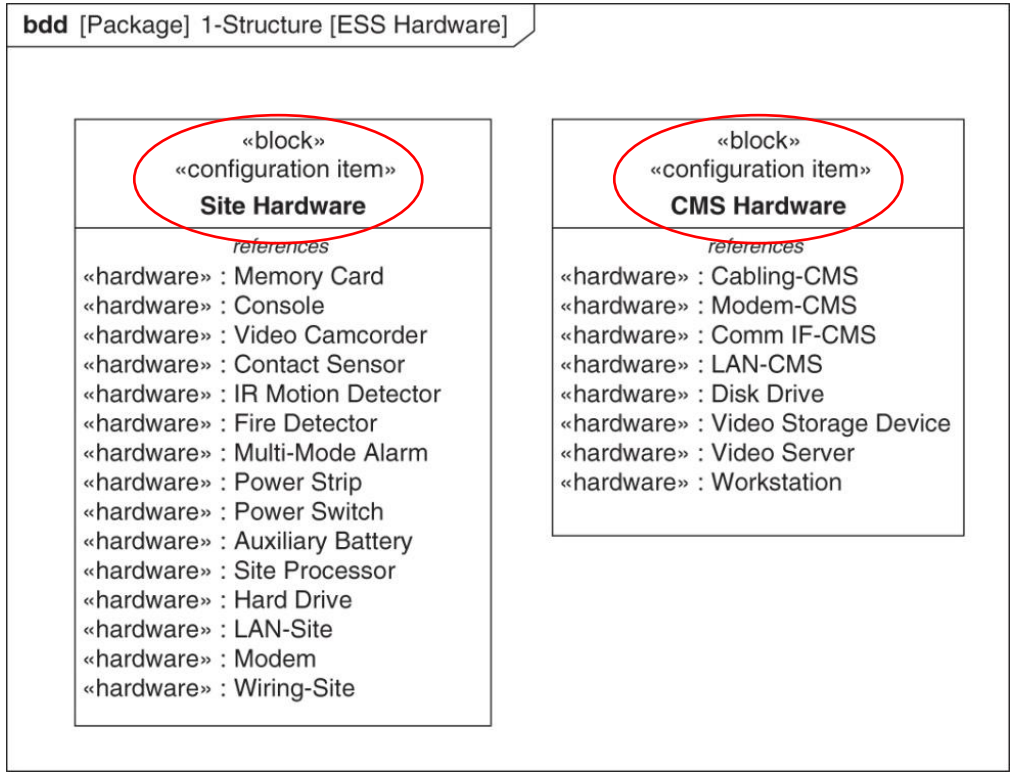
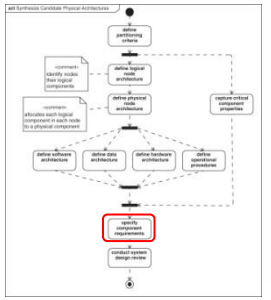


Figure 17.42

Specify Component Requirements



- The Physical Architecture results in the black box specification of hardware, software, data, and procedural components

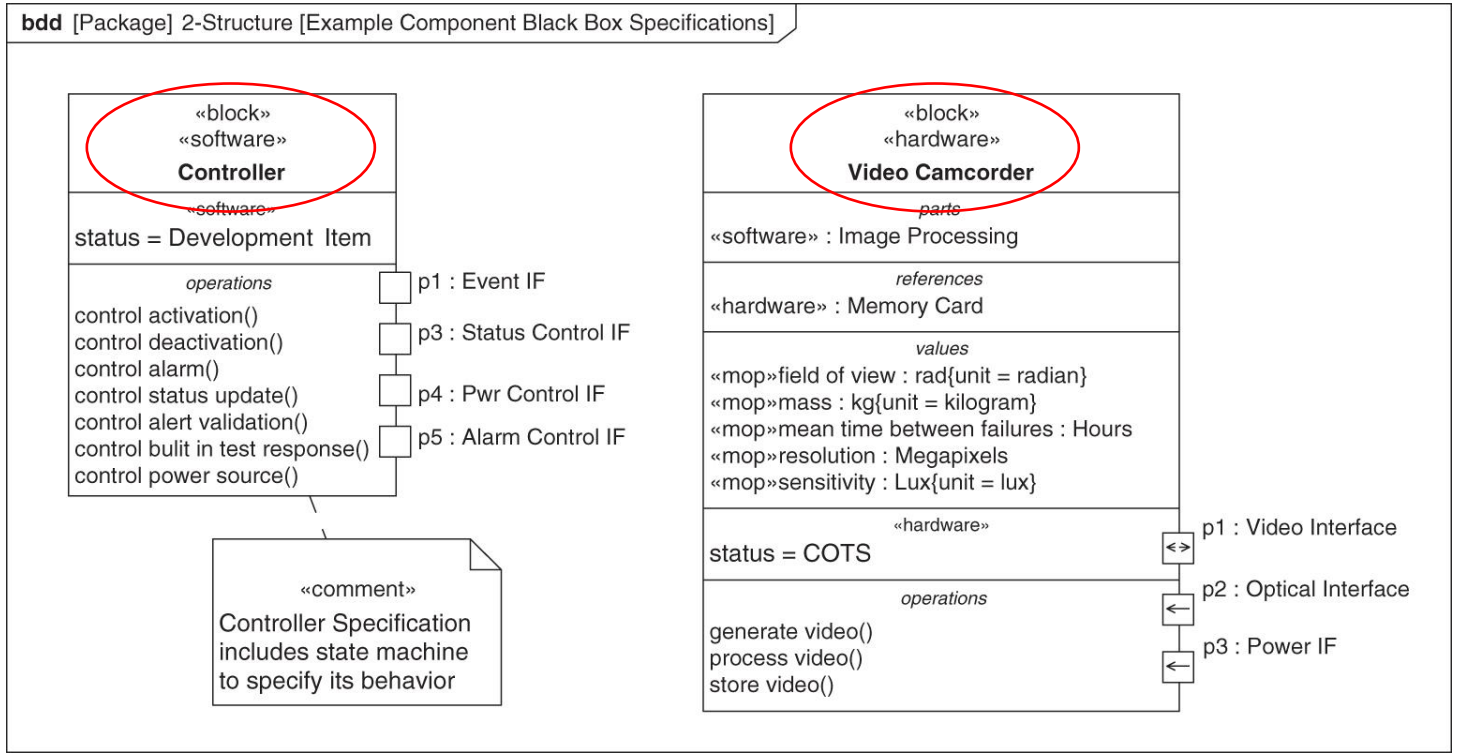


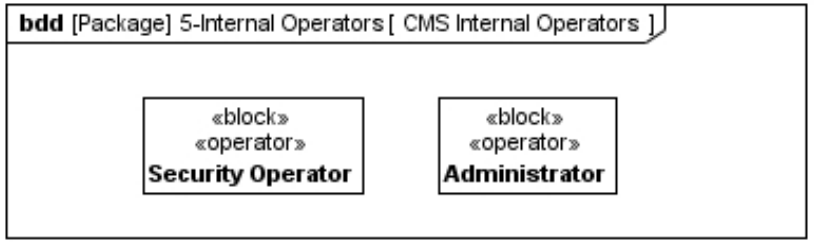
Figure 17.44

OOSEM Support for Hardware/Software Requirements

- Improved Hardware/Software Requirements
  - Precision
  - Completeness
  - Consistency
  - Traceability
- Hardware/Software Requirements Specification
  - System Context
  - Functional/behavioral requirements
  - Interface requirements
  - Performance, physical and other non-functional requirements
  - Test cases

Define Operational Procedures

- Operators can be external or internal to the system, depending on how the system boundary is established
  - ESS Operators
    - Internal – Security Operator, Administrator
    - External – Resident/User
- The requirements for what an Operator must do to operate the system are specified in terms of operational procedures



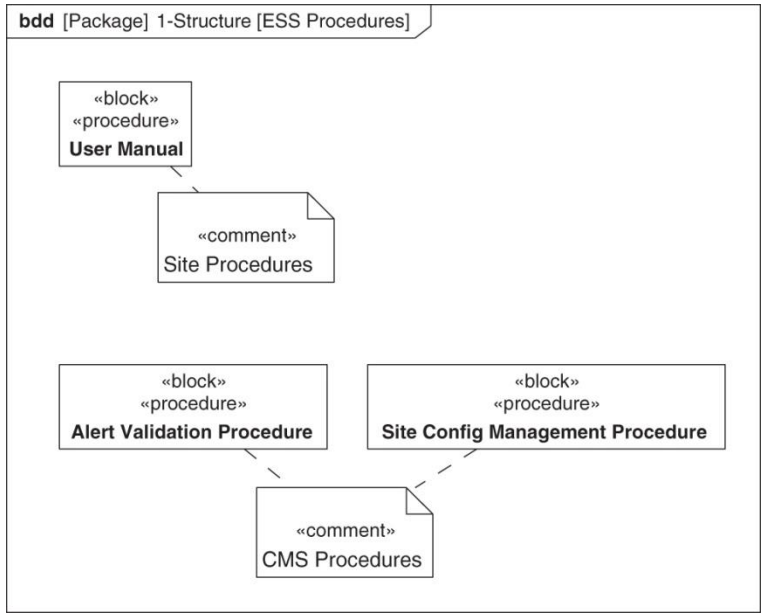
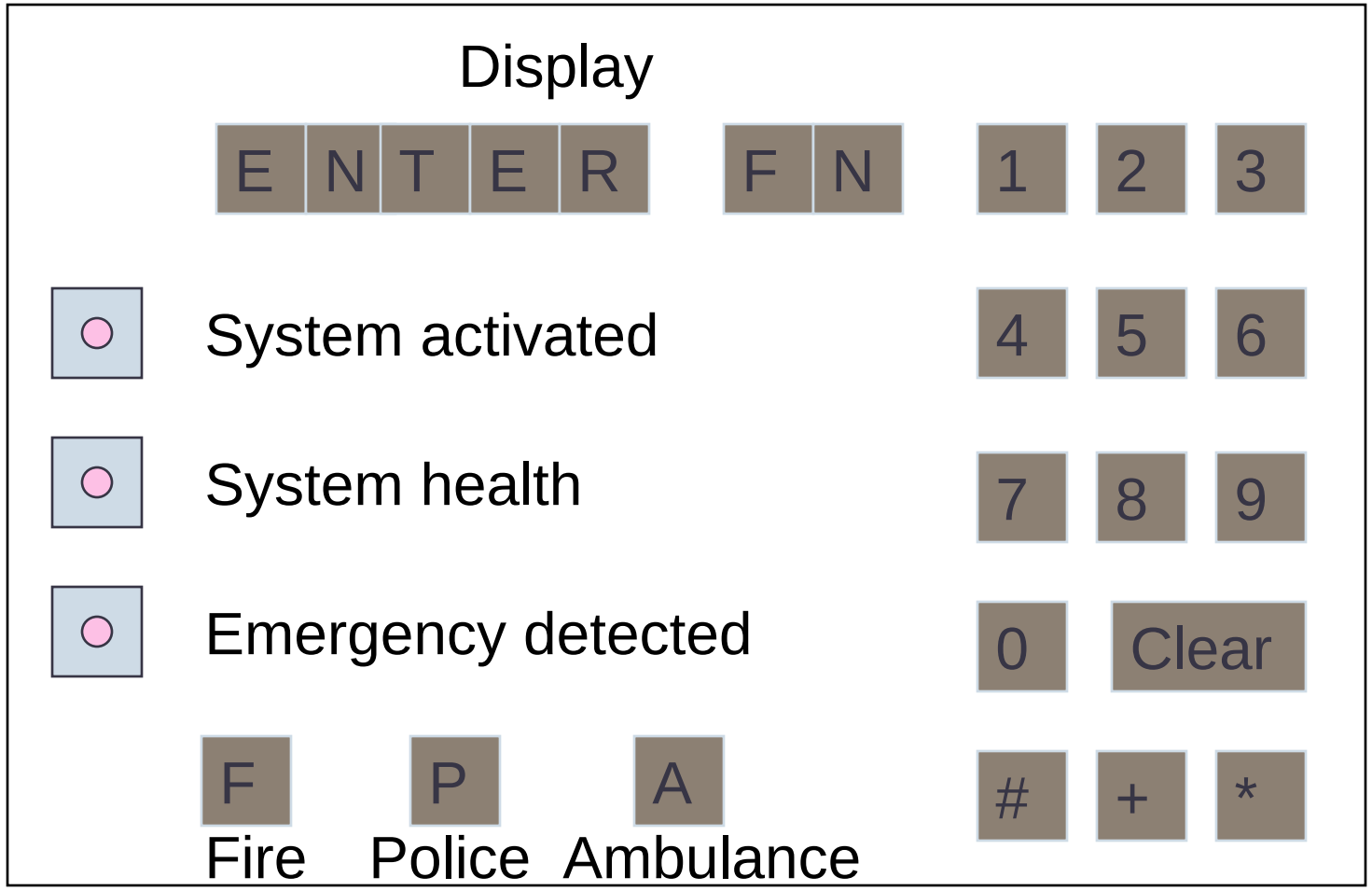


Figure 17.43

ESS Occupant Display Interface



Defining Other Architecture Views

- Views and viewpoints capture specific stakeholder perspectives
- A viewpoint specifies a subset of the model that is of interest to the stakeholder
- A view provides a filtered portion of the model that conforms to the viewpoint
- A document or report can represent a view

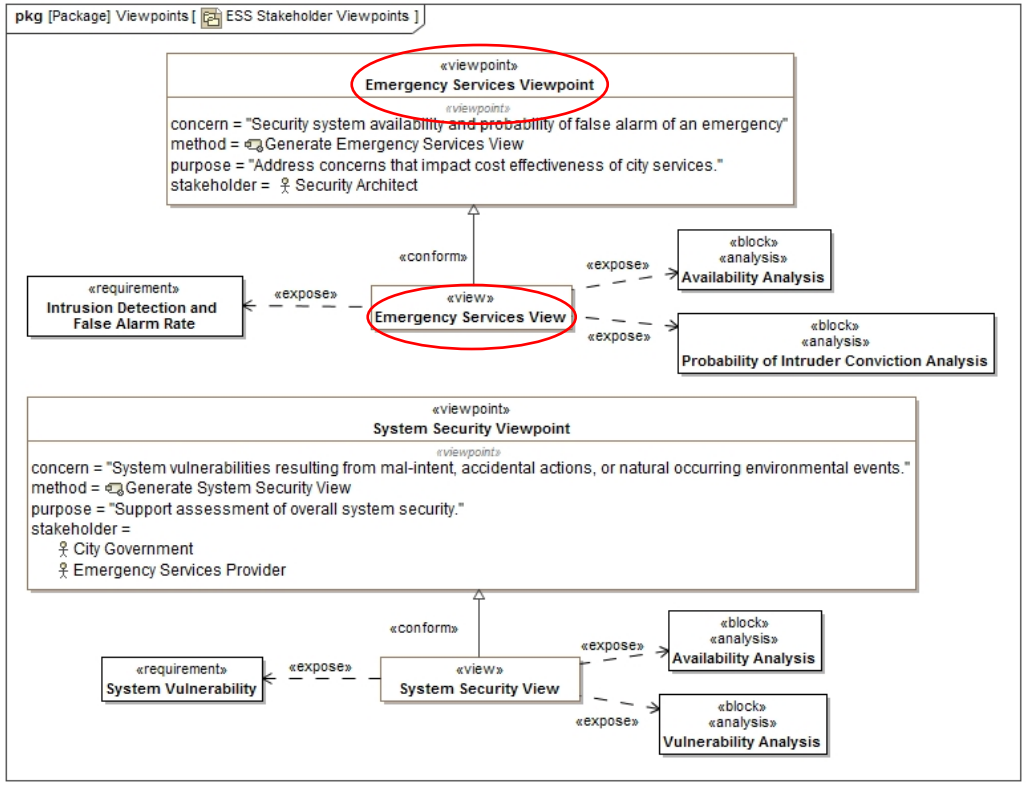
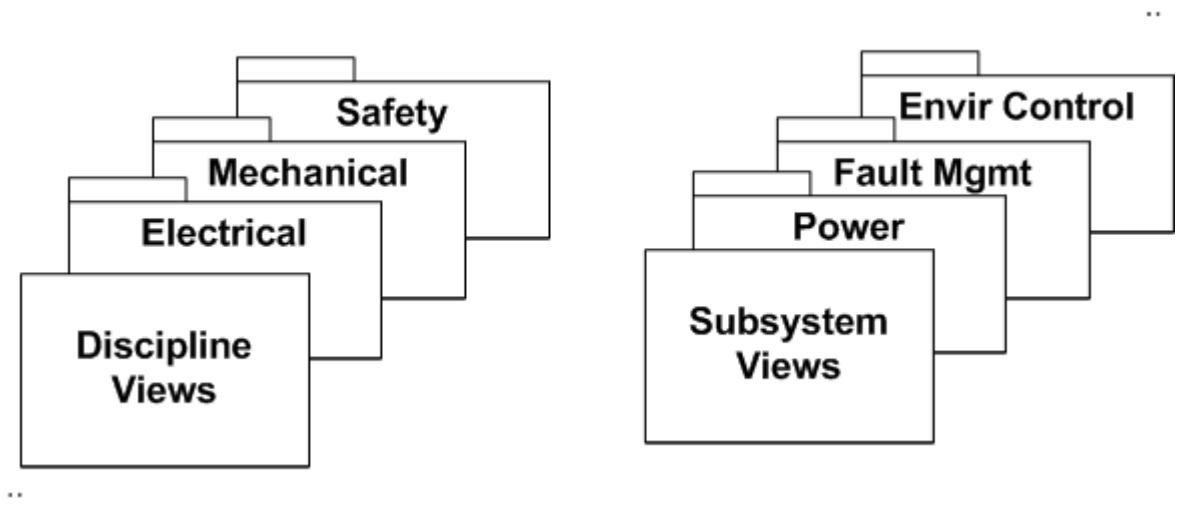


Figure 17.47

Architecture Views (cont.)

- Views can be an effective way to communicate with other members of the engineering team
  - Internal block diagrams and requirements can provide effective summary level views



Summary

- Establishing explicit partitioning criteria enables good architecting practice
- A logical node aggregates logical components at a location
- The logical components at each logical node are allocated to physical components at the corresponding physical node
- The hardware, software, and data architectures are views of the system physical architecture
- The hardware and software components are specified as a black box (e.g., functional, interface, performance)
- Other cross cutting views of the architecture can be defined to support specific concerns (e.g., subsystems, security, ..)