CISC 322

Software Architecture

Lecture 12:

4+1 Views

Emad Shihab

Paper by: Philippe Kruchten

Recap of Last Class

- Reference Architecture
  - Webservers
    - Apache
    - AOLServer

Motivation for Architecture Views

- Many stakeholders (e.g. customers, developers, project managers, etc.), who care about different aspects of the system
- Cannot provide one representation that will satisfy all stakeholders without becoming too complex
- Stakeholders generally want to interact with parts of the system that are most important to them

Architecture Views

- Various parts of the architecture have to be modeled using different approaches
- A *view* is a set of design decisions related to a common concern (or set of concerns)
- A *concern* is an aspect of the system that a stakeholder is concerned about

The Different Views

- **Logical**: captures the logical (often software) entities in a system and their interconnections
- **Process:** captures the concurrency and synchronization aspects of a design

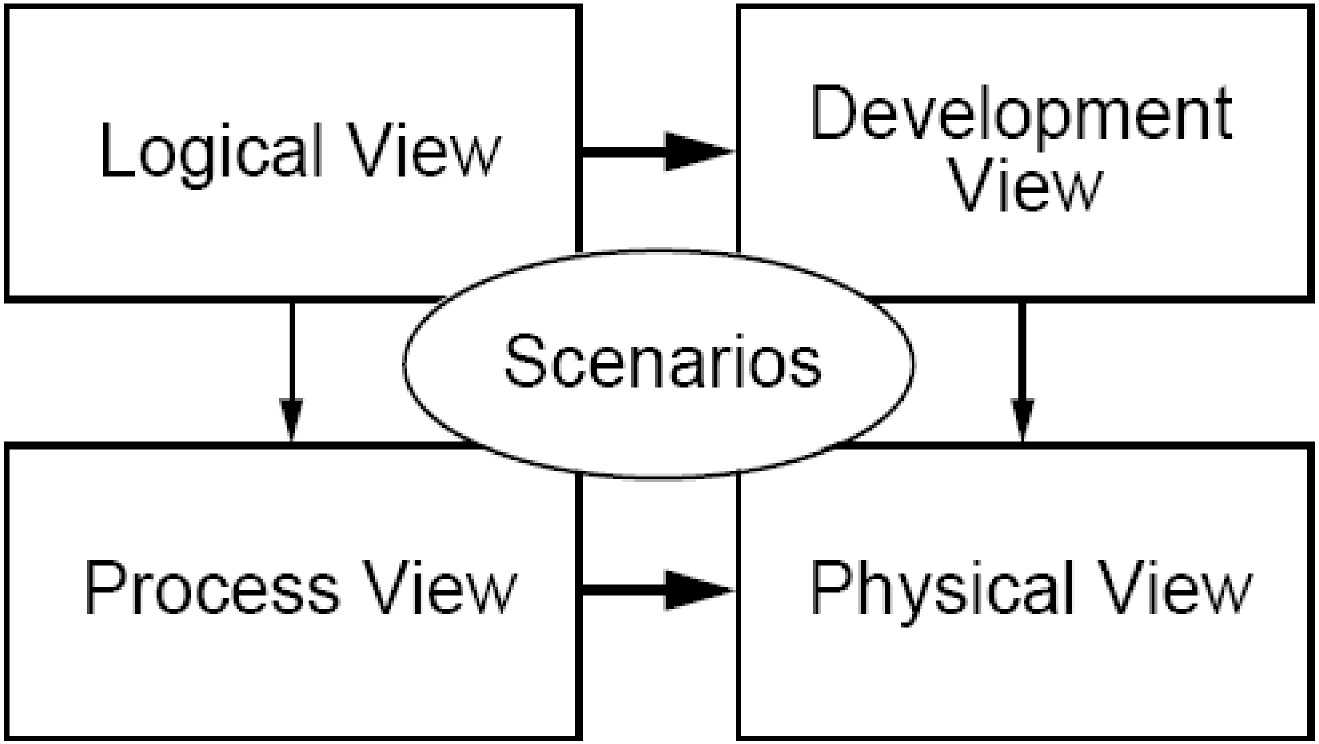
The Different Views

- **Development**: describes the static organization of the software in its development environment
- **Physical**: captures the physical (often HW) entities in a system and their interconnections
- **+1:** Use cases and scenarios to illustrate these views

Architectural Views

Stakeholder: End-user
Concern: Functionality

Programmers
Software management



Logical View

Development View

Scenarios

Process View

Physical View

Integrators
Performance
Scalability

System engineers
Topology
Communications

Logical View

- Components: Classes
- Connectors: Associations, containment, inheritance
- Stakeholders: End-users
- Concerns: Functionality (i.e. functional requirements)

Why Logical View ?

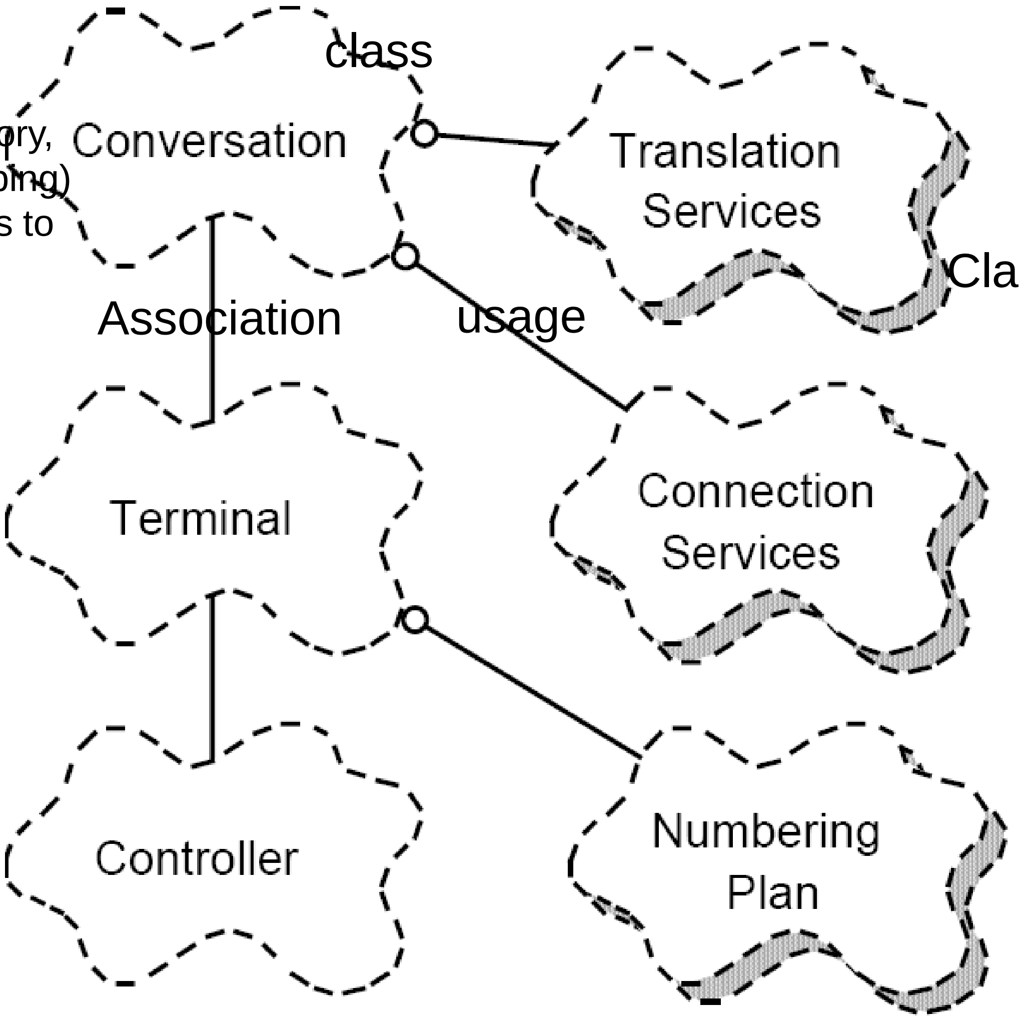
- Decomposition used for
  - functional analysis and,
  - to identify common elements in the system

Logical View Example

Represents terminal in conversation
Uses translation (directory, logical to physical mapping) and connection services to establish voice paths

Maintain state
Use numbering to interpret dialing

Decode and inject signals on line interface card (start, stop, digit)



class

Class utilities

Association

usage

Process View

- Components: Tasks/threads, processes
- Connectors: Messages, RPC
- Stakeholders: Integrator
- Concerns: Performance, availability, fault tolerance

Process View - definitions

- **Process**: a grouping of tasks that form an executable unit
- **Task**: a separate thread of control that can be scheduled individually

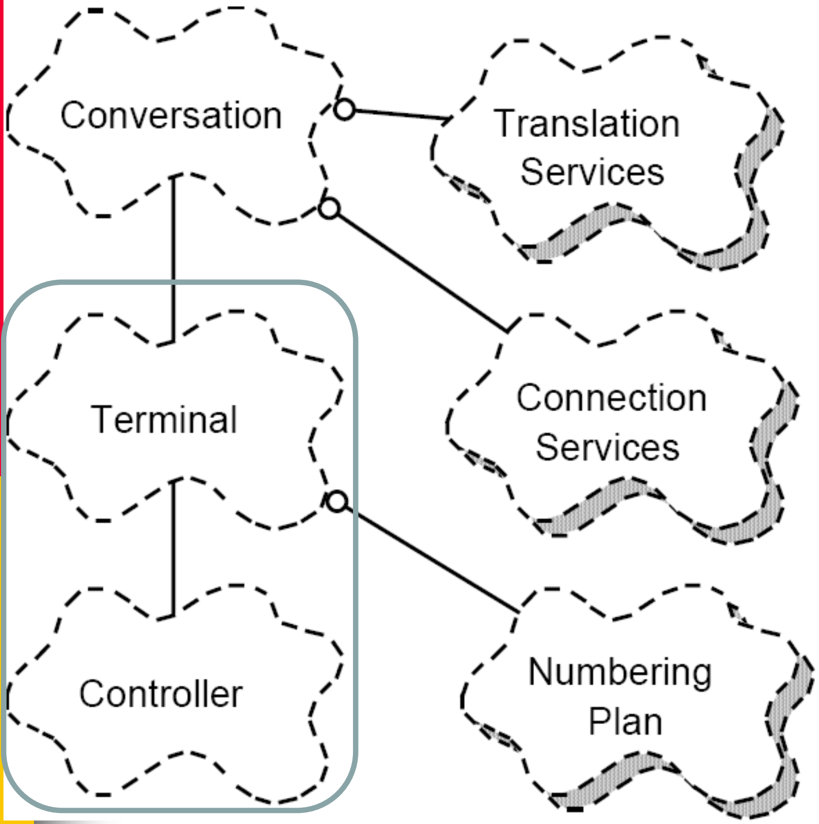
- Can be replicated for load distribution or improved availability

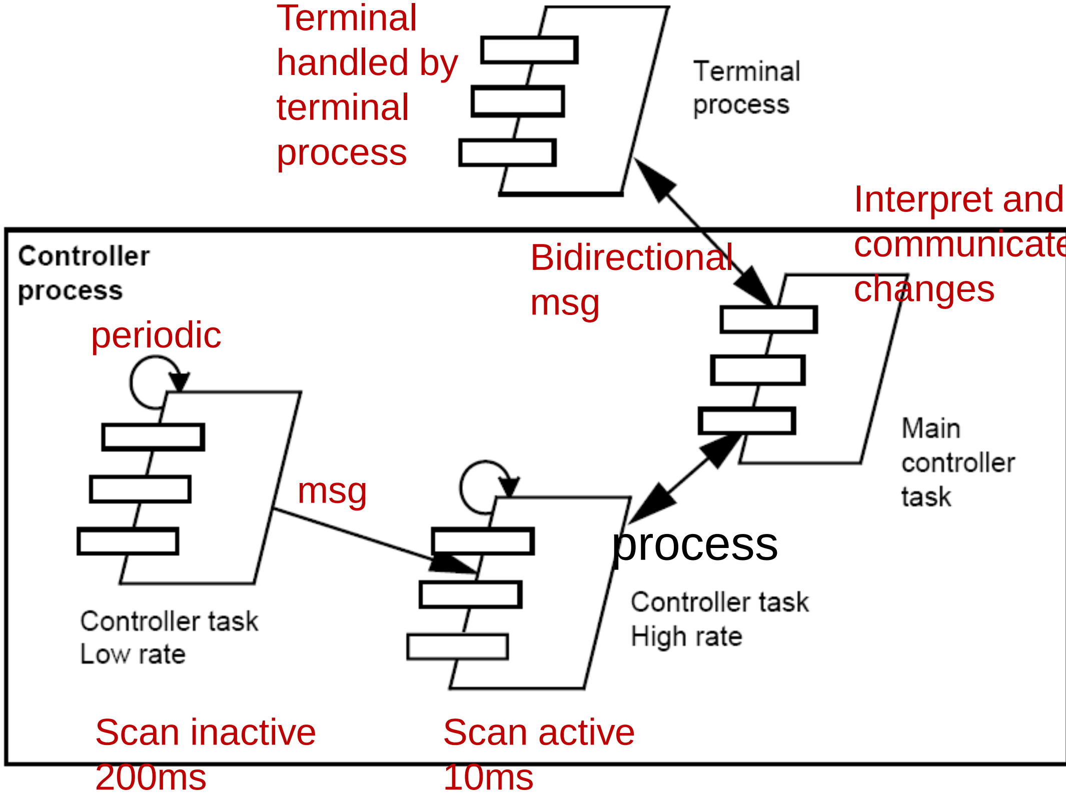
Process View

- *Major tasks* – architectural elements that can be uniquely addressed
- *Minor tasks* – introduced for implementation reasons (e.g. buffering)

- Flow of messages and process loads can be estimated and used to gauge performance

Process View





Terminal handled by terminal process

Interpret and communicate changes

Bidirectional msg

periodic

msg

process

Scan inactive 200ms

Scan active 10ms

Development View

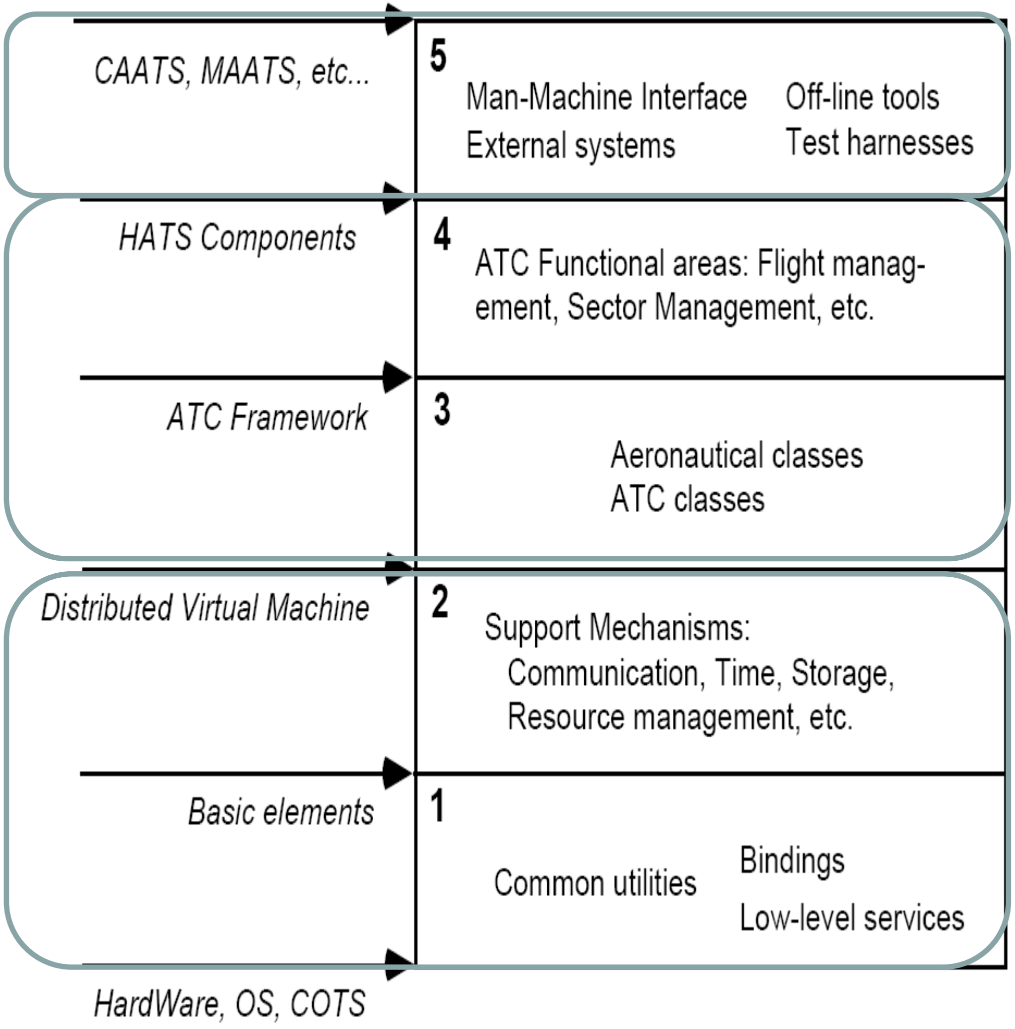
- Components: Module/Subsystem
- Connectors: Dependency (e.g. include)
- Stakeholders: Developers
- Concerns: Organization, reuse

Why Development View?

- Takes into account internal requirements related to ease of development, software management, reuse
- Serves as basis for work allocation

Development View

72 subsystems



CAATS, MAATS, etc...

5

Man-Machine Interface
External systems

Off-line tools
Test harnesses

HATS Components

4

ATC Functional areas: Flight management, Sector Management, etc.

ATC Framework

3

Aeronautical classes
ATC classes

Distributed Virtual Machine

2

Support Mechanisms:
Communication, Time, Storage,
Resource management, etc.

Basic elements

1

Common utilities

Bindings
Low-level services

HardWare, OS, COTS

Customer specific

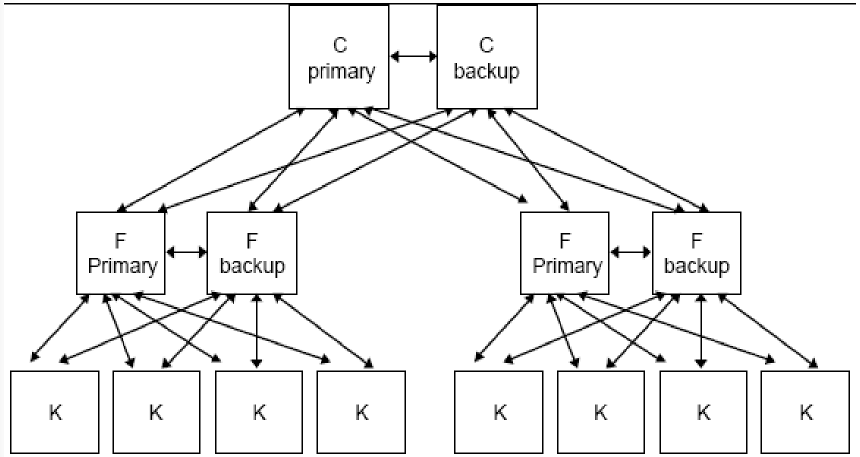
Domain specific

Domain independent

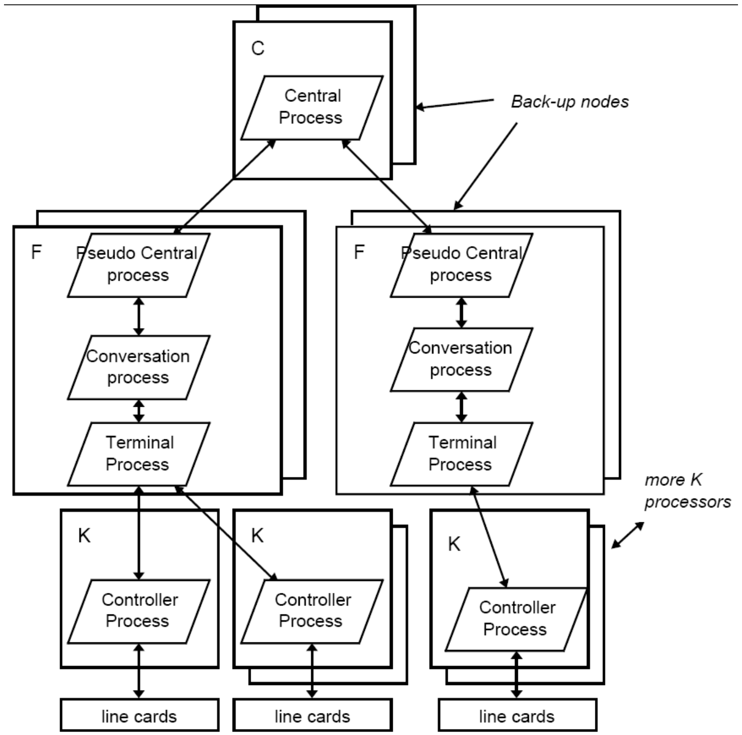
Physical View

- Components: Node
- Connectors: Network (LAN, WAN)
- Stakeholders: System designer
- Concerns: Nonfunctional requirements (e.g. Scalability, performance, availability)

Physical View



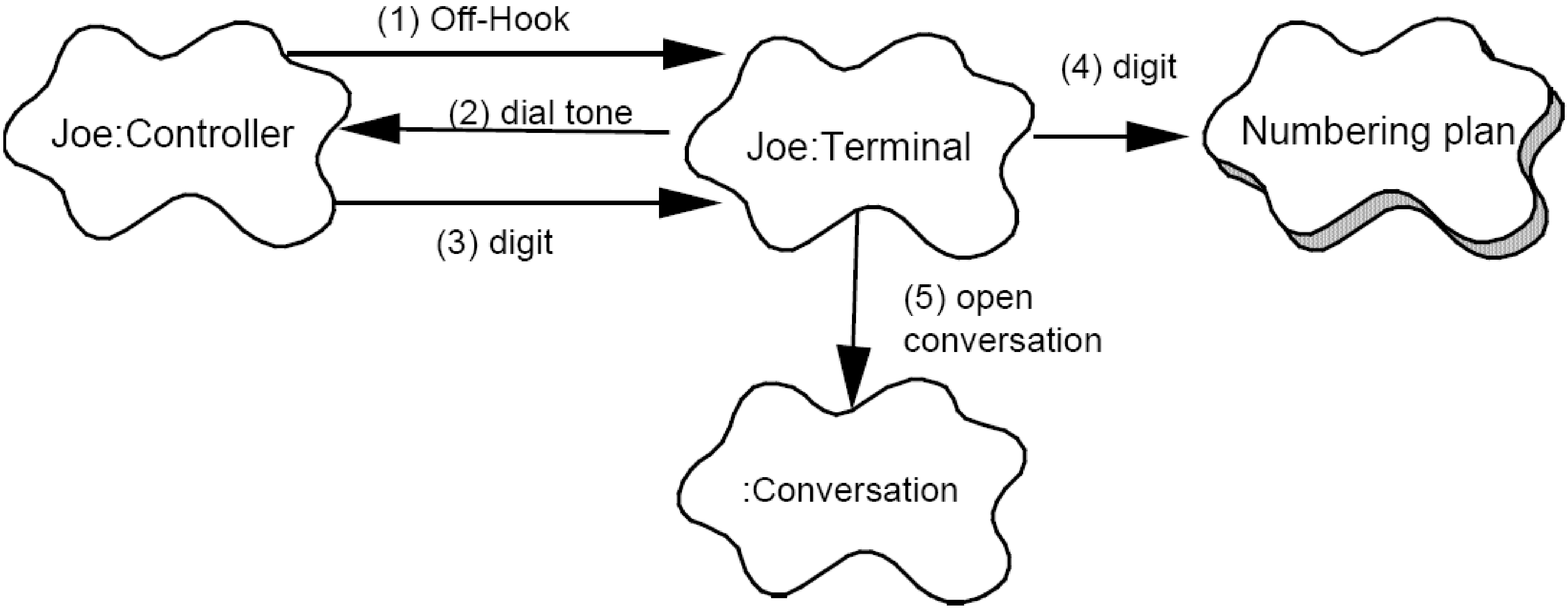
C,F, and K are different types of computers



Scenarios “+1” view

- Shows how the four views work together seamlessly
- Redundant with other views, hence “+1”
  - Drives the discovery of architectural elements during architecture design
  - Validates and illustrates role after architecture design in complete
- Representation
  - Similar to logical view

Scenarios “+1” View Example



Summary

- Different views address different concerns
- Not all views are necessary
- Lots of efforts needed to maintain these concurrent views, especially as the software system evolves
  - inconsistency, inaccurate