Architectural Blueprints – The "4+1" View Model of Software Architecture

Paper By: Philippe Kruchten, IEEE Software 1995.

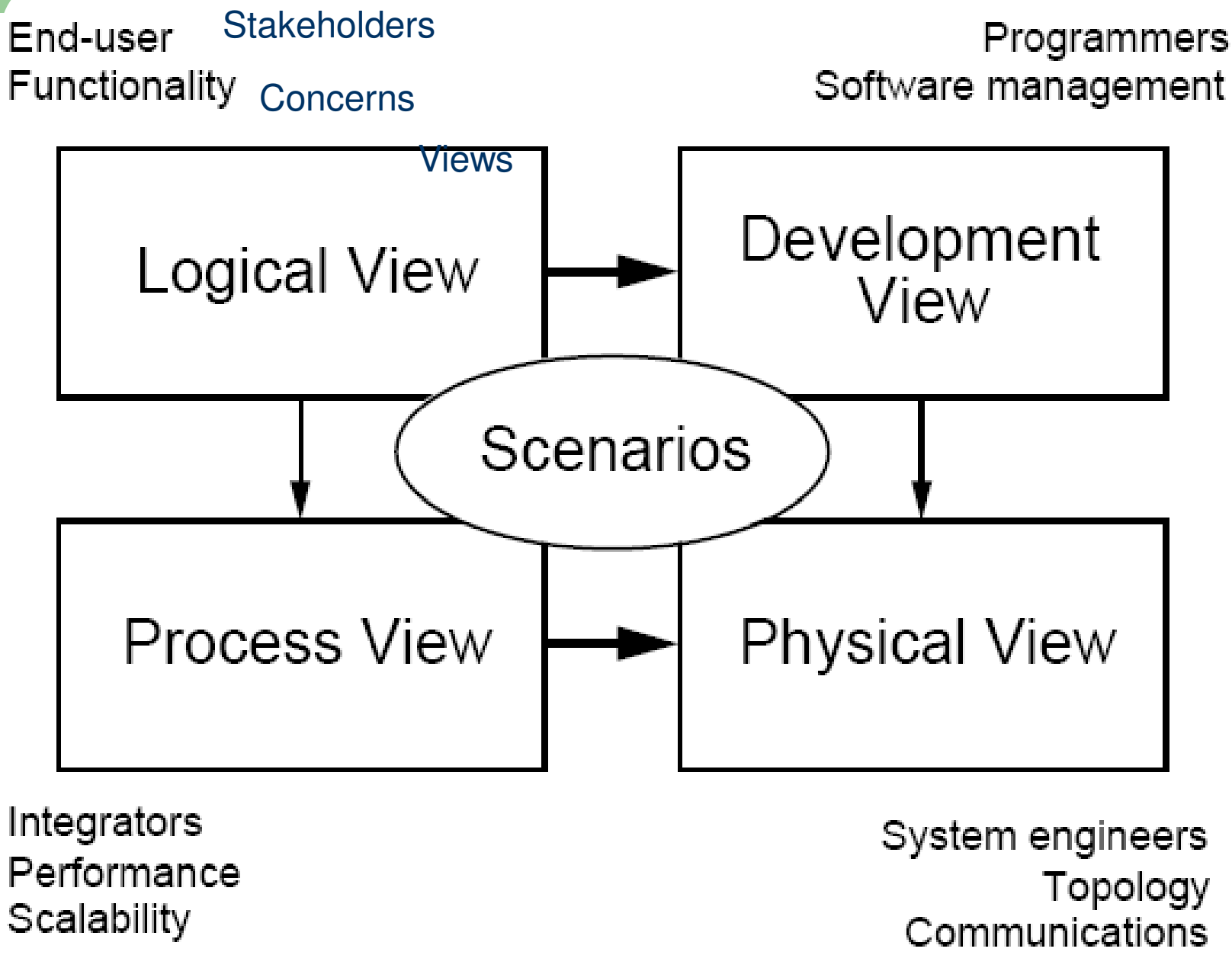
Slides By: Jack ZhenMing Jiang

Outline

- Problems and Solutions
- “4 + 1” Views
  - The Logical Views
  - The Process Views
  - The Development Views
  - The Physical Views
  - The Use Case Views
- Conclusion

Problems

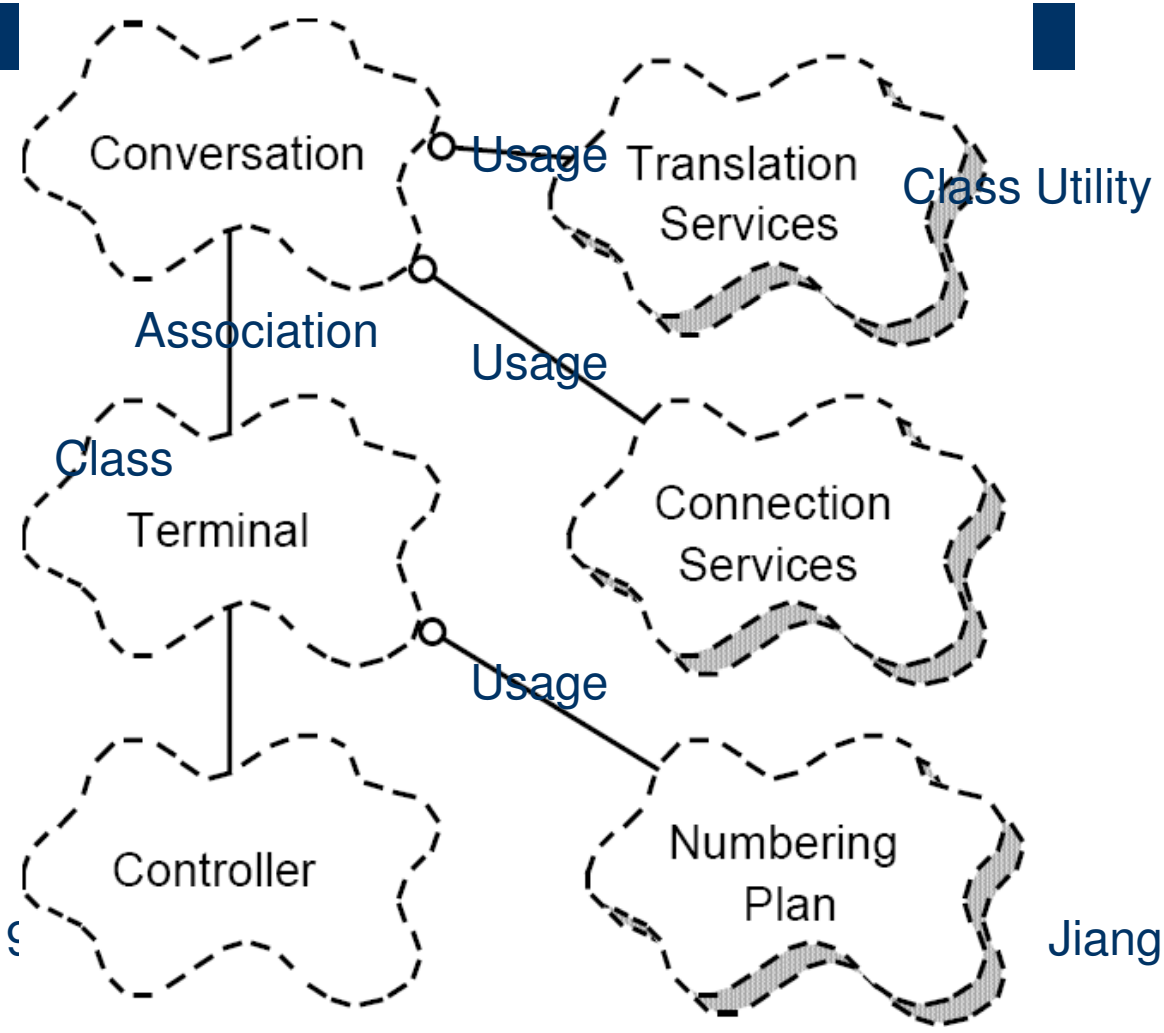
- Ambiguities in Boxes-and-Arrows Diagrams
  - Boxes can be programs, chunks of source code, physical computers, logical groupings of functionalities, …
  - Arrows can be data flow, control flow, or both.
- Architecture documents over-emphasize one aspect of software development or
- Architecture documents do not address the concerns of all stakeholders


End-user
Functionality
Stakeholders
Concerns
Views
Programmers
Software management
Logical View
Development View
Scenarios
Process View
Physical View
Integrators
Performance
Scalability
System engineers
Topology
Communications

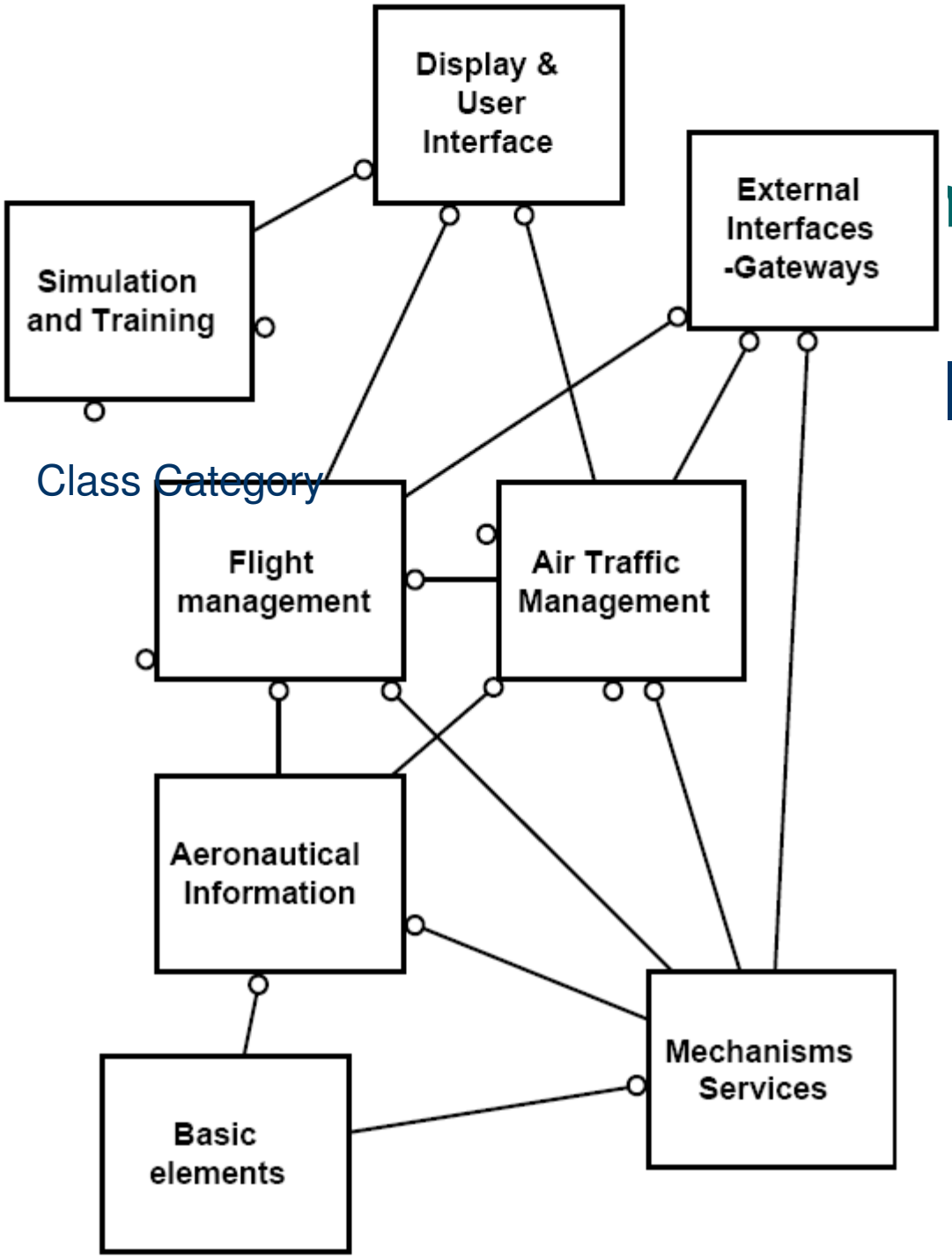
Logical Views – The Object-Oriented Decomposition

- Concerns
  - Primarily supports the functional requirements (services to users)
- Representation
  - Class diagrams (classes and logical relationships)
  - Class categories
  - Class utilities

An Example of Logical Architecture - PABX



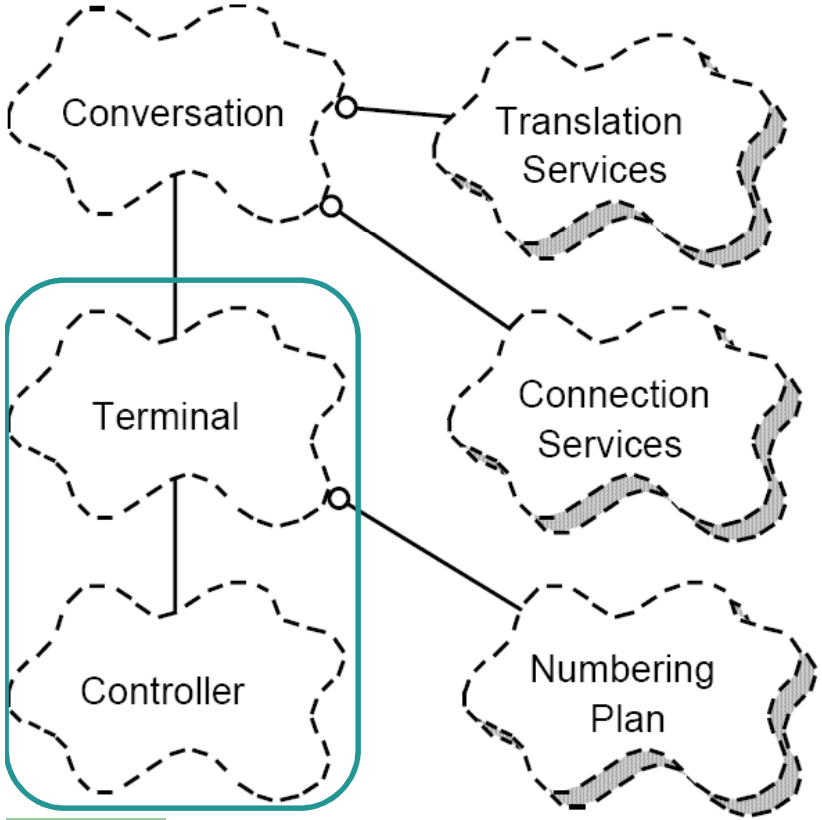
A Big
Archi
Syste
ntrol

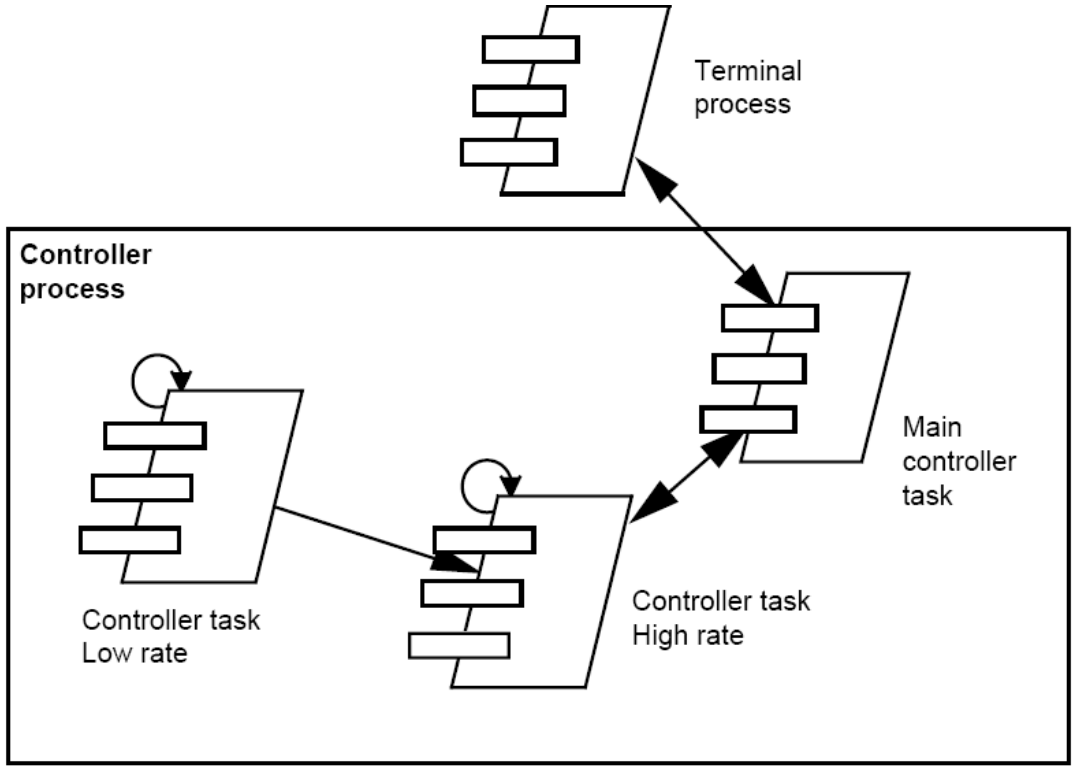


The Process Architecture – The Process Decomposition

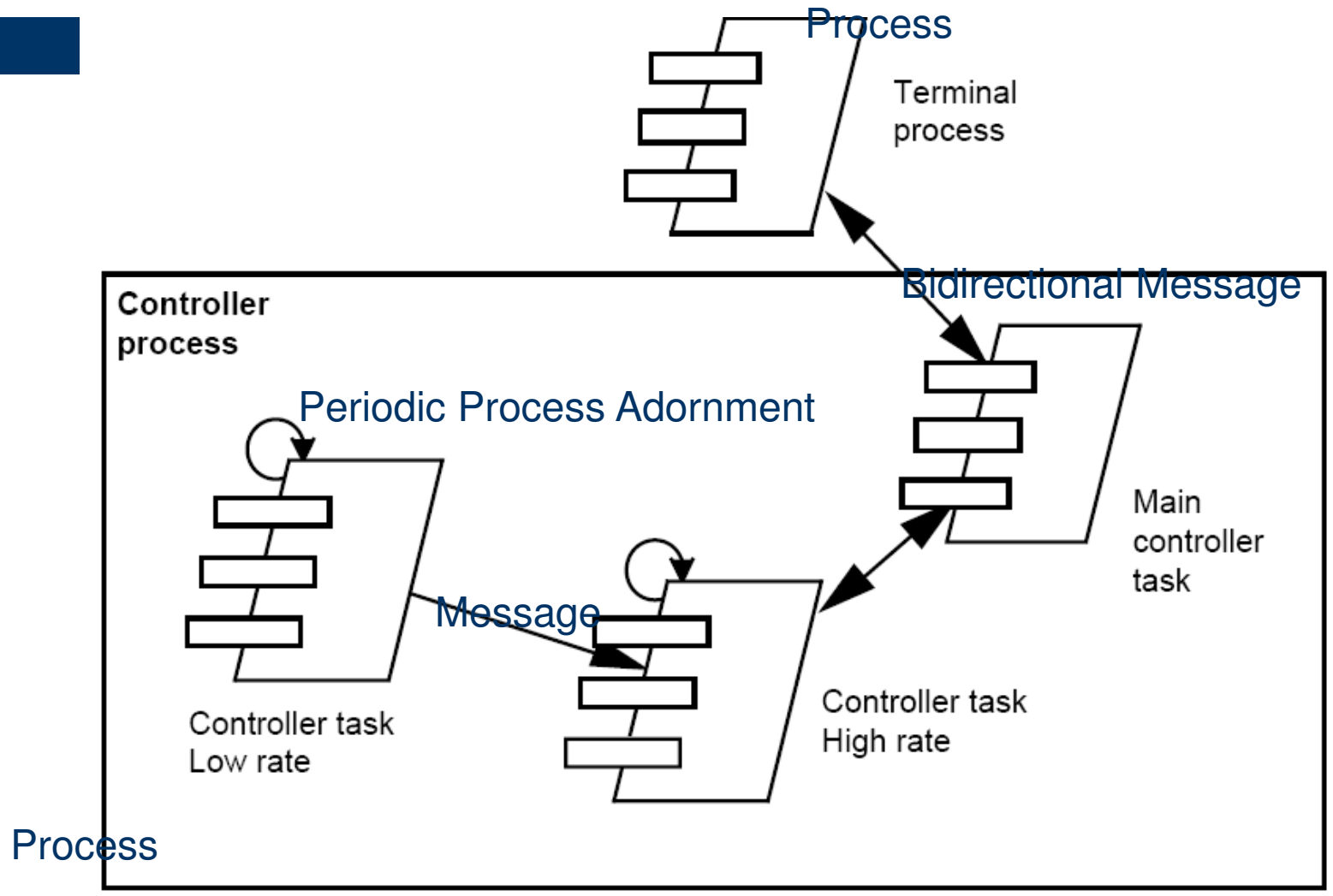
- Concerns
  - Nonfunctional requirements (concurrency, performance, availability, etc.)
- Representation
  - Different levels of abstractions
    - Processes and Threads
    - Major Tasks, Minor Tasks
    - Communication Mechanisms
      - Major tasks uses synchronous and asynchronous message communications, RPC, and event broadcasts, etc.
      - Minor tasks uses rendezvous or shared memory

An Example of the Process Architecture - PABX





An Example of the Process Architecture (Continued)

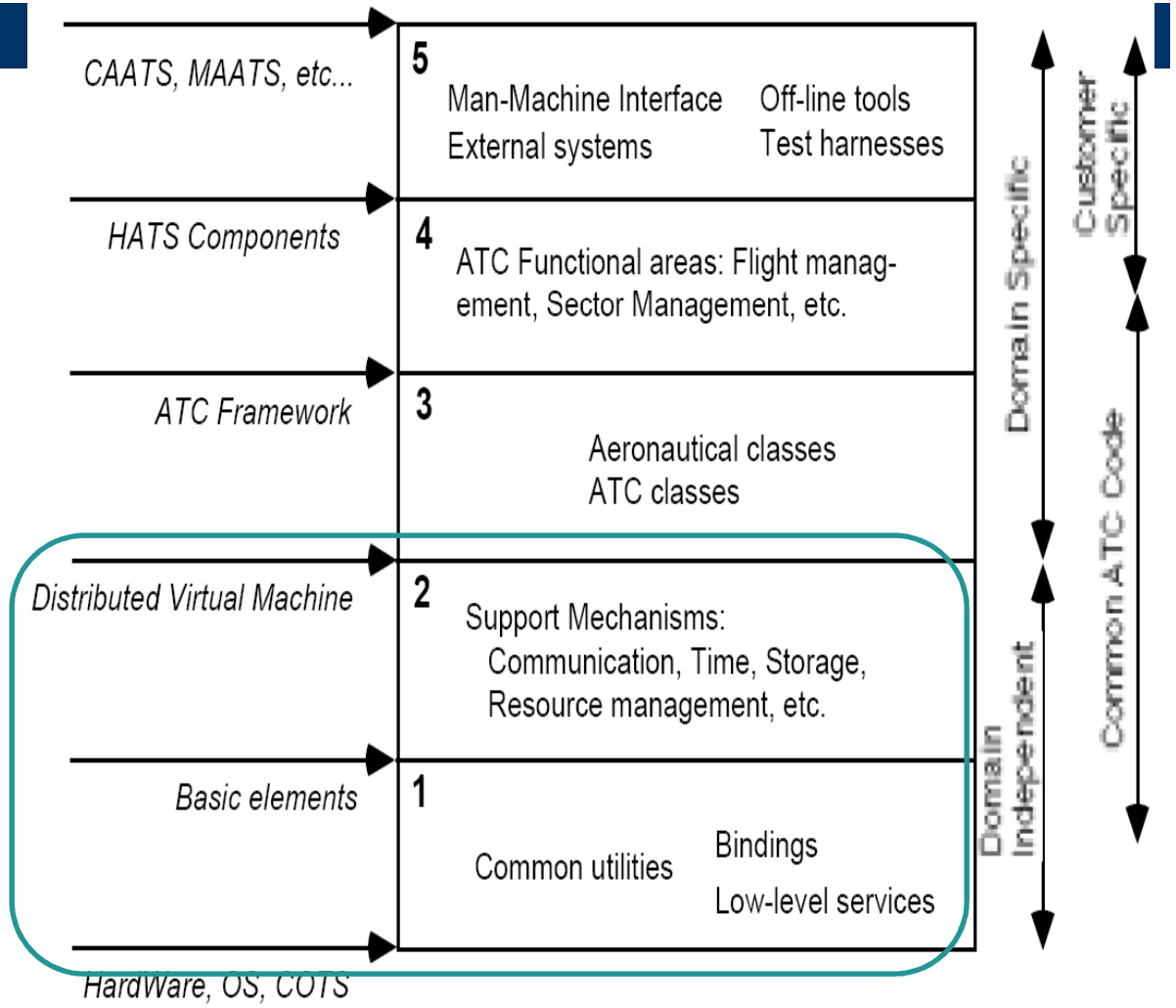


The Development Architecture – Subsystem Decomposition

- Concerns
  - Actual software module organization on the software development environment
- Representation
  - Layered Style (depends on same levels or layers below)

An Example of the Development Architecture

- 72 subsystems across 5 layers
- each layer about 10 to 50 modules

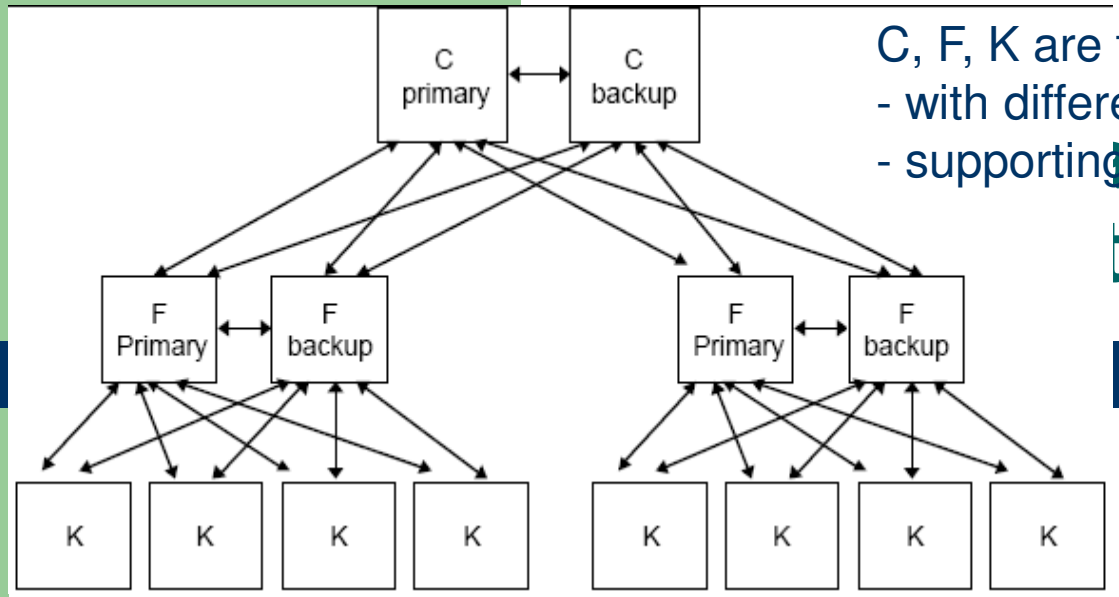


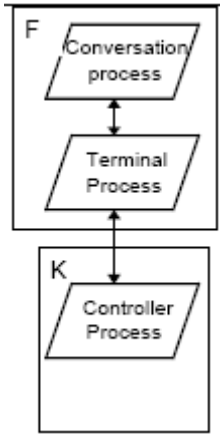
The Physical Architecture – Mapping the Software to Hardware

- Concerns
  - Primarily the nonfunctional requirements of the systems (like availability, reliability, scalability)
- Representation
  - Various forms (words, notations) over the process view

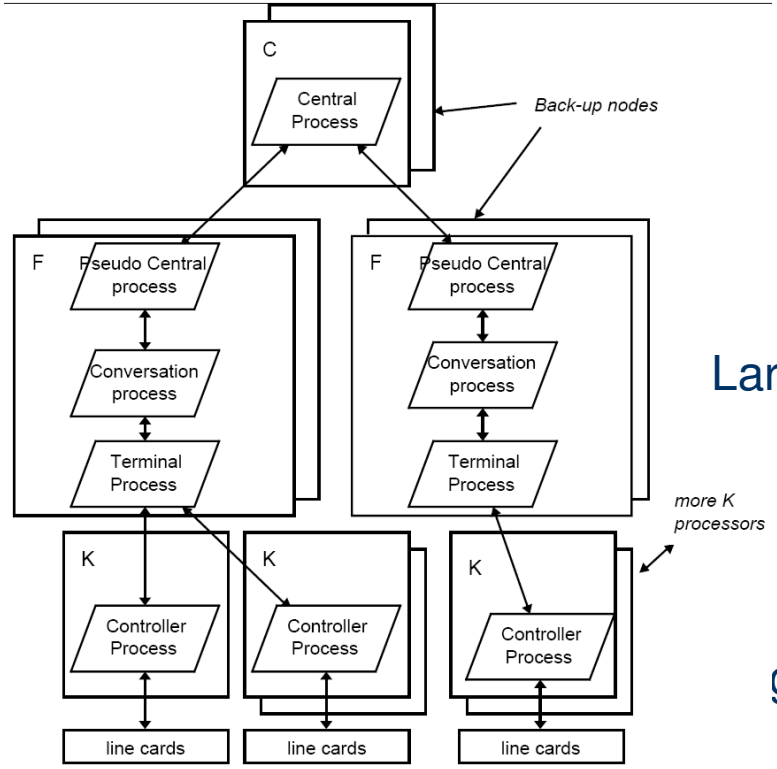
C, F, K are three types of computers
- with different capacity
- supporting different executables

tinued)





Small PABX



Large PABX

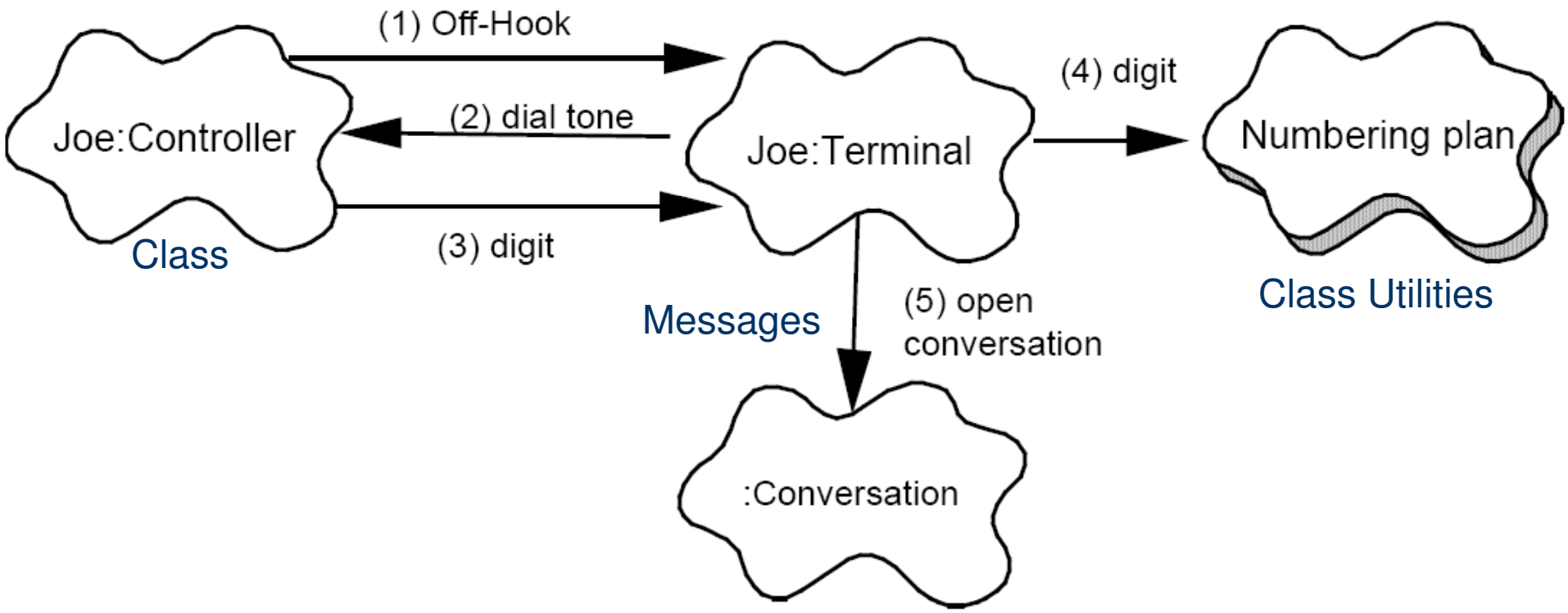
9/7/2008

g Jiang

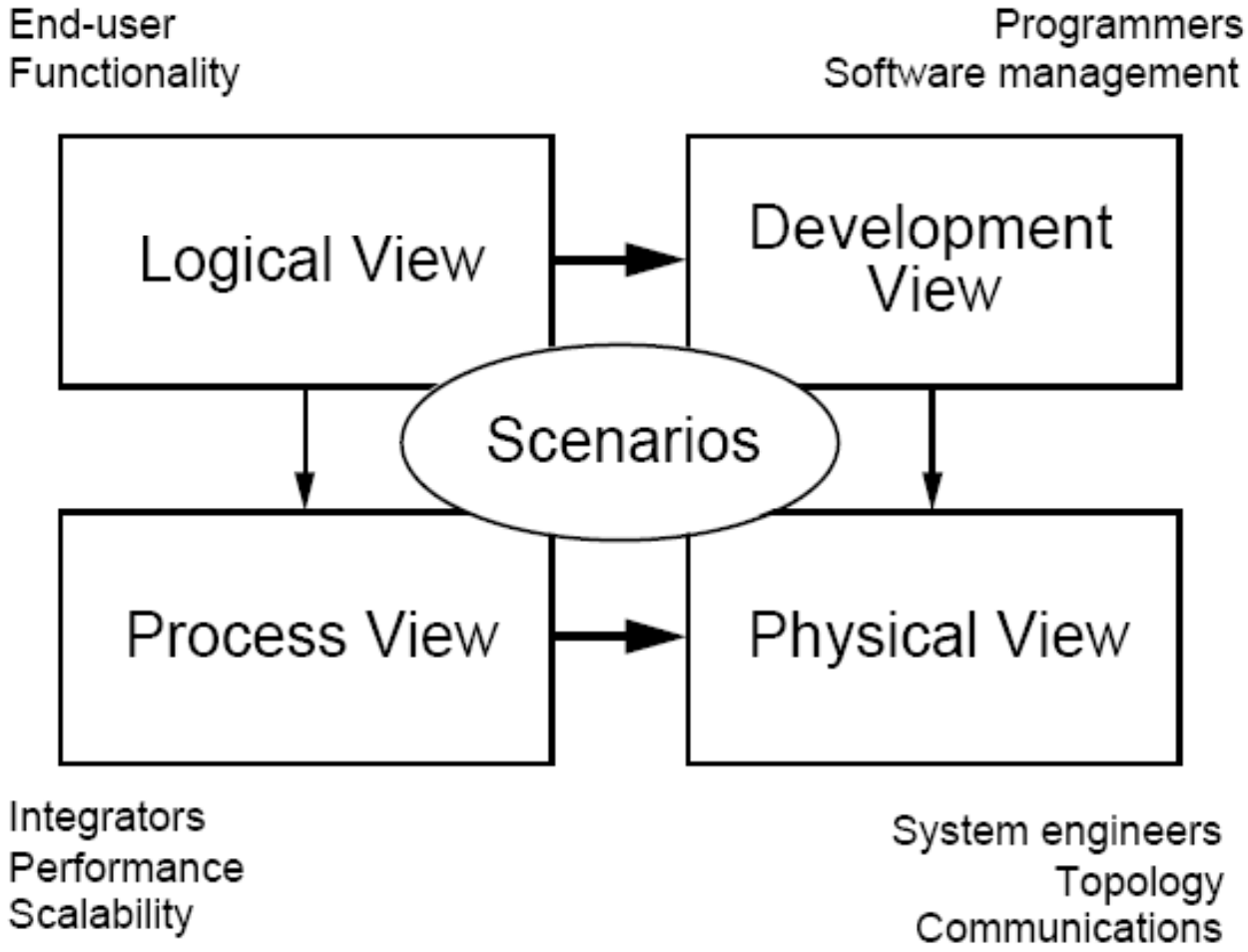
The Use Case View – Putting It All Together

- Concerns
  - Redundant with other views (thus “+1”)
  - Drivers to discover architectural elements
  - Validation and illustration to show the design is complete
- Representation
  - Similar to the logical view but a few variations

An Example of the Scenarios - PABX



Correspondence Between the Views

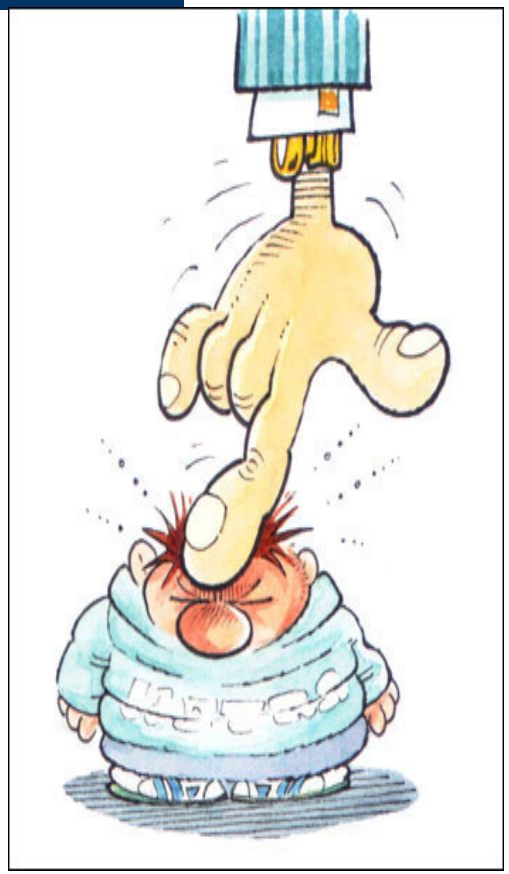


Conclusions

View	*Logical*	*Process*	*Development*	*Physical*	*Scenarios*
Components	Class	Task	Module, Subsystem	Node	Step, Scripts
Connectors	association, inheritance, containment	Rendez-vous, Message, broadcast, RPC, etc.	compilation dependency, "with" clause, "include"	Communica-tion medium, LAN, WAN, bus, etc.	
Containers	Class category	Process	Subsystem (library)	Physical subsystem	Web
Stakeholders	End-user	System designer, integrator	Developer, manager	System designer	End-user, developer
Concerns	Functionality	Performance, availability, S/W fault-tolerance, integrity	Organization, reuse, portability, line-of-product	Scalability, performance,av ailability	Understand-ability
Tool support	Rose	UNAS/SALE DADS	Apex, SoDA	UNAS, Openview DADS	Rose

Conclusions

- 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
- An Nice Introduction using UML: http://www-128.ibm.com/developerworks/wireless/library/wi-arch11/



Extra – IBM Introduction to “4+1 Views

Views	Notations
The Logical View	Class Diagrams, Sequence Diagrams, Collaboration Diagrams
The Development View	Package Diagram
Process View	
Physical View	Deployment Diagram
Use Case View	Case Diagram and Use Case Specifications