Reflecting on Architectural Evolution:
- Questions from Change Management & Conceptual Modeling

- Architectural Evolution – System Discontinuity
- What SCM relies on
- The Modeling/View Crisis
- Back to Basics: Conceptual Modeling
- What to model? Separation of Concerns
- How to apply model? Unit Operations
- Conclusions and Challenges
- Architecture is dead. Viva the organisation.

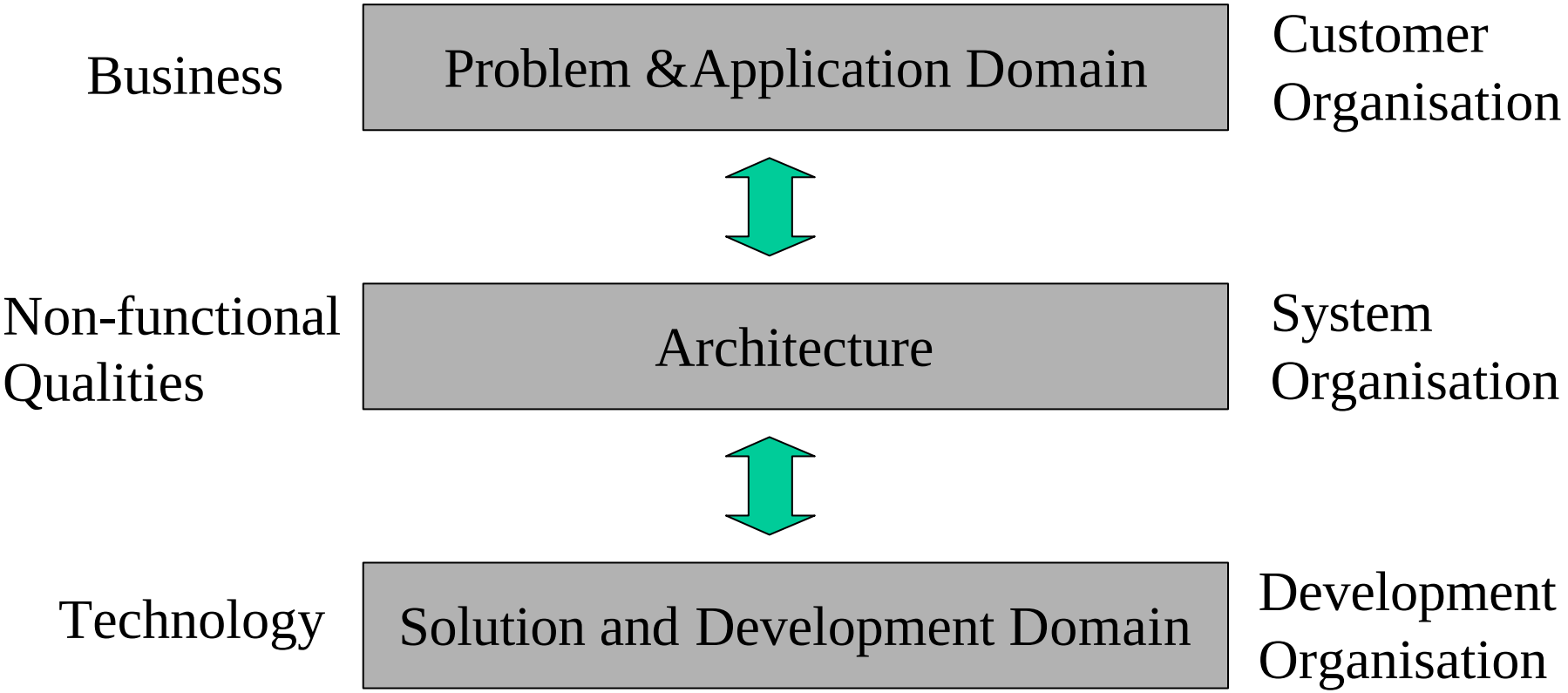
Lars Bendix & Palle Nowack*,

Dept. of CS, Aalborg University

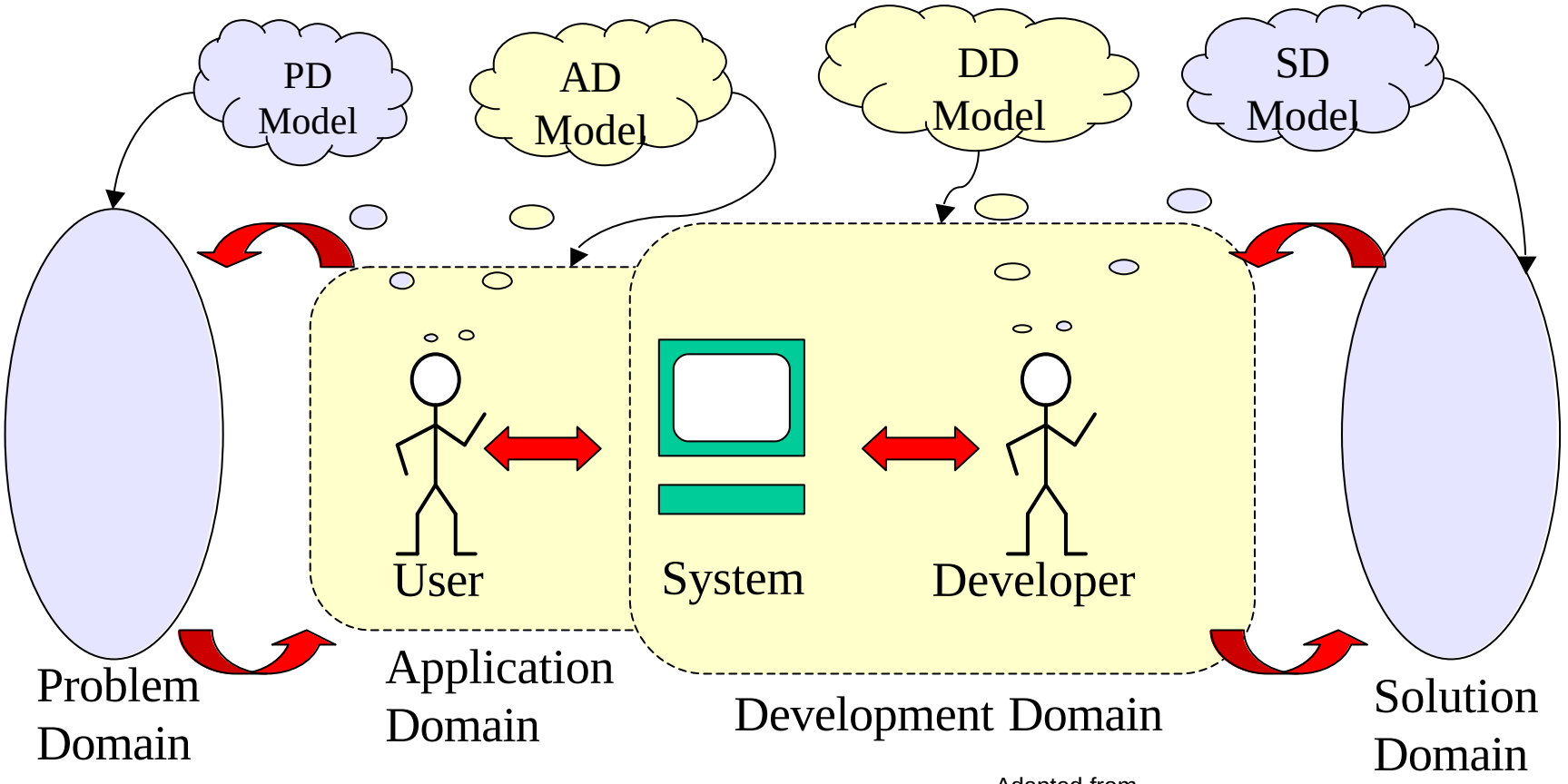
nowack@mip.sdu.dk

(*Maersk Institute, University of Southern Denmark)

Architecture & Evolution
- Continuity versus Discontinuity



Software Domains and Models



Adapted from
Jacobsen et al.: “Architecture = Abstractions over Software”, Proceedings of TOOLS Pacific’99.

Software Configuration Management

What SCM needs:

- Configuration Items
  - Source code
  - Documentation
  - Test cases
  - …
- Operations
  - Delete
  - Insert
  - Diff
  - Propagation
  - Merging
  - …

Can SWA deliver?

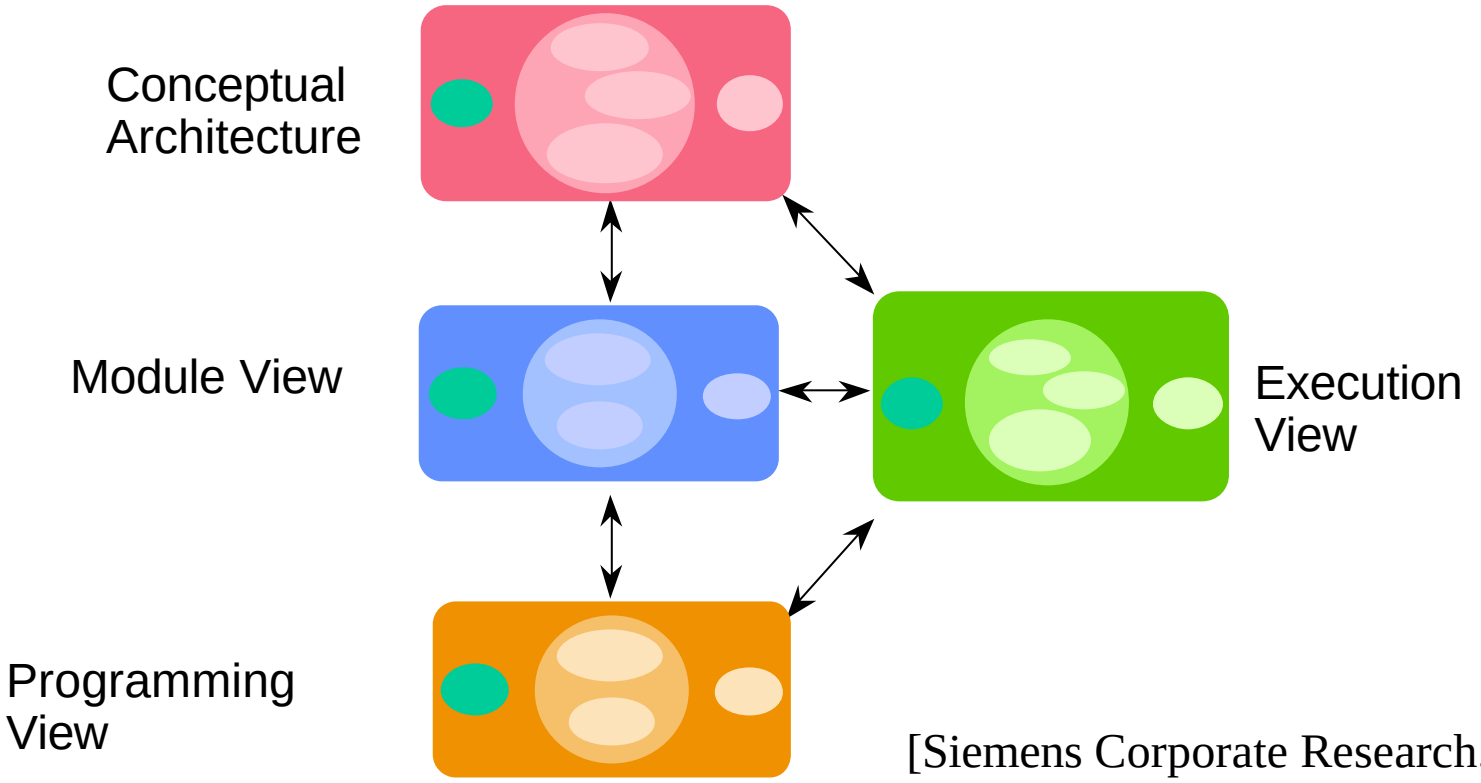
Modeling – Loosing Faith?

- A model is a simplification of reality
- We build models so that we can better understand the system we are developing
- We build models of complex systems because we cannot comprehend such a system in its entirety
- The choice of what models to create has a profound influence on how a problem is attacked and how a solution is shaped
- Every model may be expressed at different levels of precision
- The best models are connected to reality
- No single model is sufficient.

Booch et al.: UML user guide. Addison-Wesley, 1999.

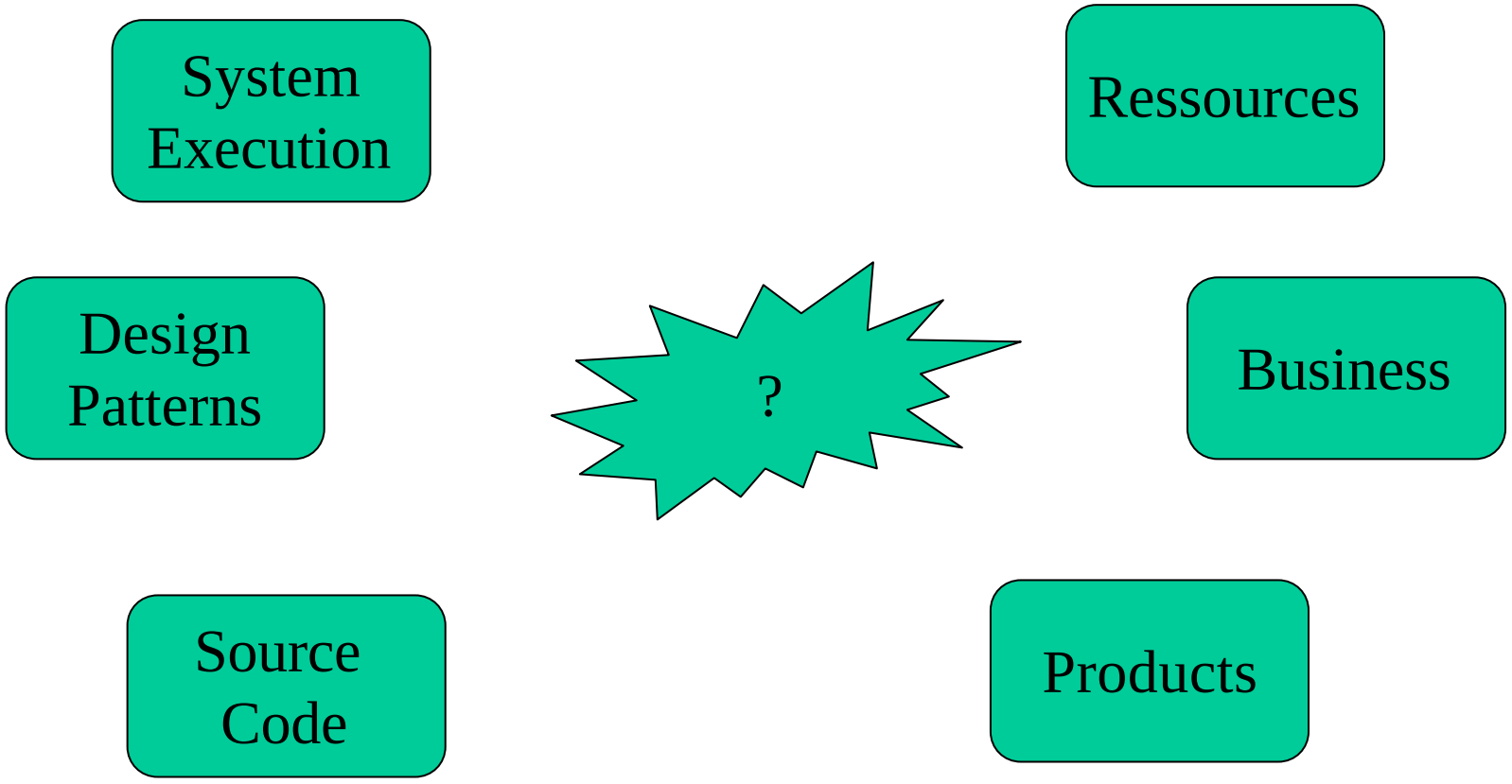
- Very precise formal modeling -> Programming
- Very abstract informal modeling -> Strategic Management
- Todays modeling focus on analysis and design -> Too much paper, too many details

5-1: The Siemens Views

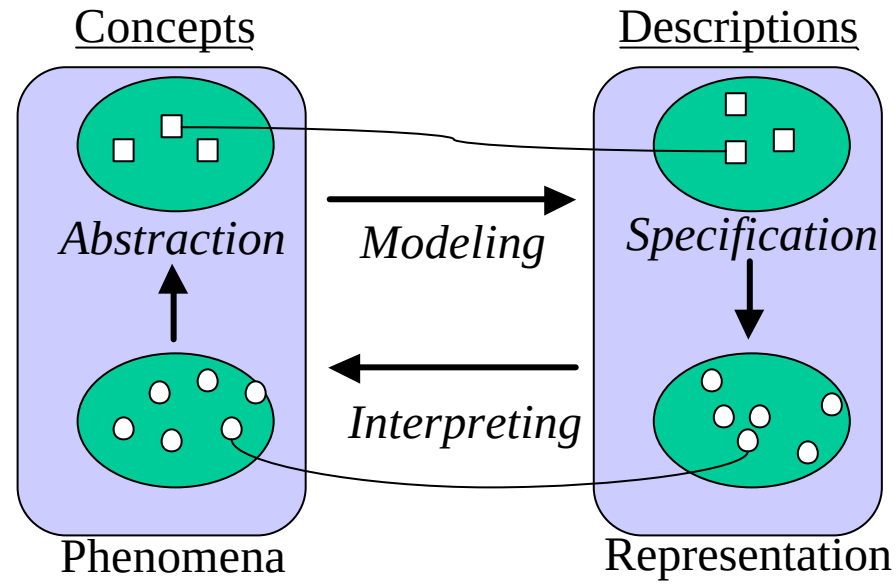


[Siemens Corporate Research., 2000]

Many and Different Views
- Do we need the system anymore?



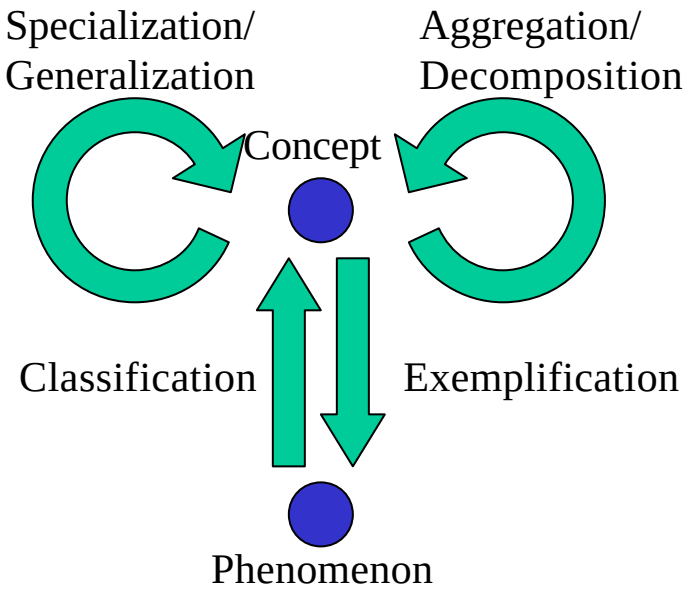
Conceptual Modeling:
Traditional OO versus Architectural Modeling



Jacobsen et al.:
"Architecture = Abstractions over Software",
Proceedings of TOOLS Pacific'99.

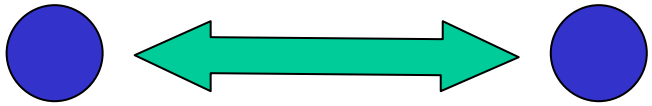
- Trad.:
  - Naive programming
- Recent:
  - Architectural Abstractions
- Variants
  - UML
  - Reflective/Meta-approaches
  - Semiotics

Conceptual Abstraction

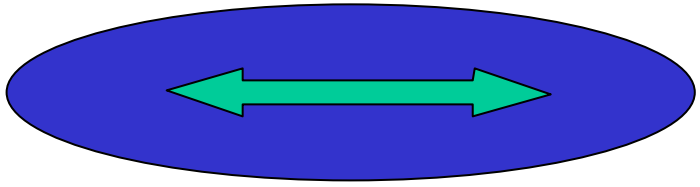


[Bent Bruun Kristensen]

Coupling & Cohesion



Coupling: A measure of how closely two entities are connected



Cohesion: A measure of how well an entity is tied together

[Bent Bruun Kristensen]

- What entities?
  - Components
  - Structures
  - Interactions
- Where to apply?
  - In the software?
  - In the models?
  - Between views?
  - Between view and software?

Unit Operations

- Seperation
  - Uniform decomposition
    - Part-whole: concept aggregation
    - Is-a: concept specialization
  - Replication
- Abstraction
- Compression
- Resource Sharing

- What is the unit?
  - Components
  - Structures
  - Interactions
- Where is the unit?
  - Software
  - Model
- What is missing?
  - A clear relation between model and referent system

[Bass et al.: Software Architecture in Practice. Addison-Wesley, 1997]

Conclusion
- & Challenges from SCM

- No fixed set of architectural views
- Each view describes Architectural Abstractions.
  - Apply Concept formation processes
- Replace Layering with Separation of Concerns
- Traceability between views
  - Apply SCM ideas

- But then we need to find:
  - Configuration Items:
    - Components
    - Structures
    - Interactions
  - Configuration Operations
    - Analysis: Change tracking
    - Synthesis: Unit operations
    - Autopoiesis: Propagation

Research: Architecture vs. Organization

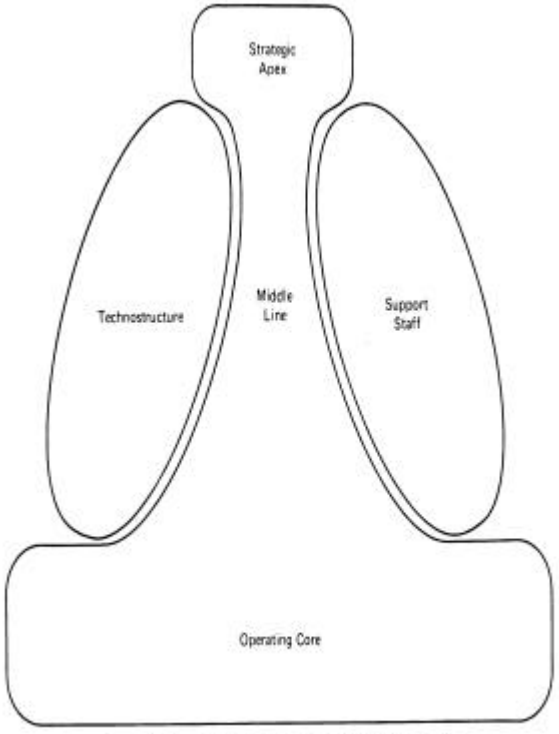
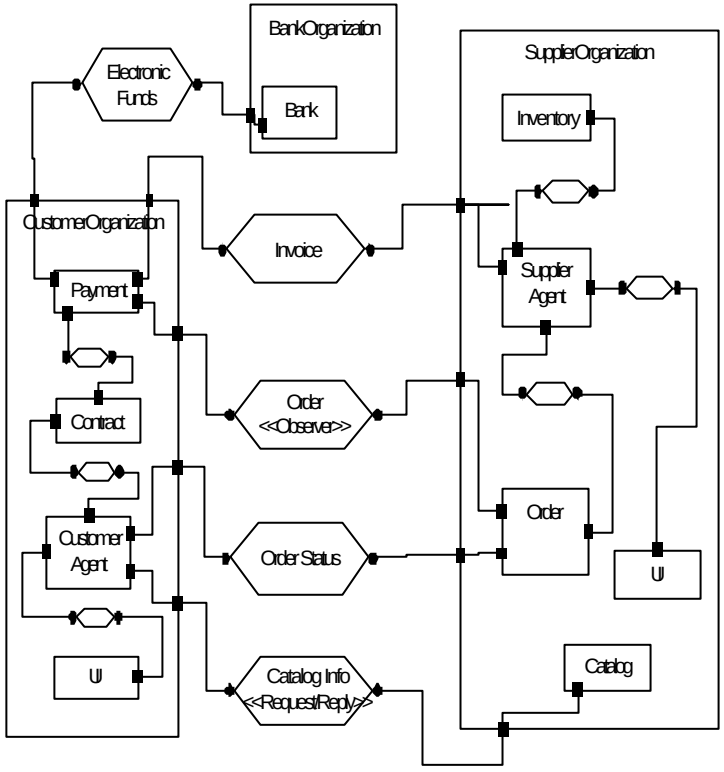


Figure 1-2. *The five basic parts of the organization*



Mintzberg: Structures in Fives.
Prentice Hall, 1983.

Views on Software: Architecture or Organization?

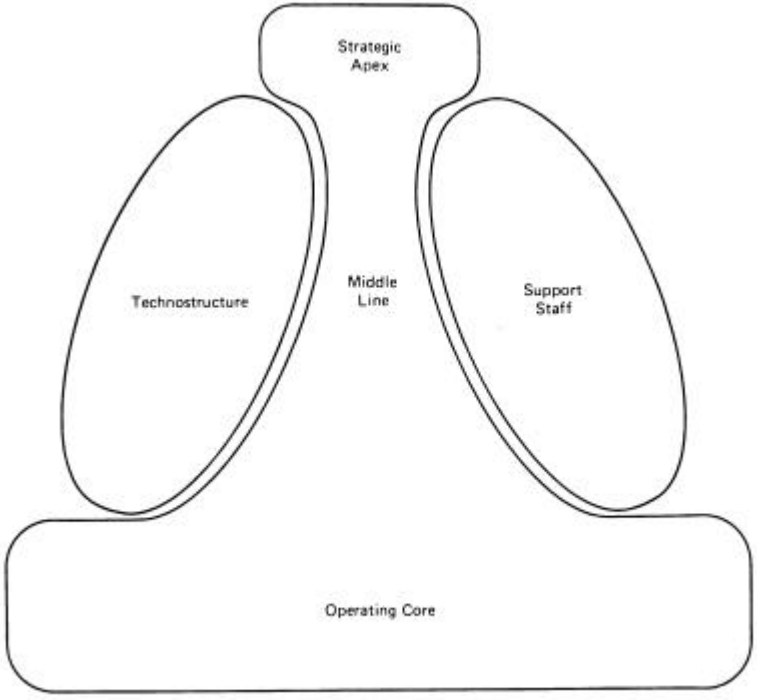


Figure 1-2. *The five basic parts of the organization*

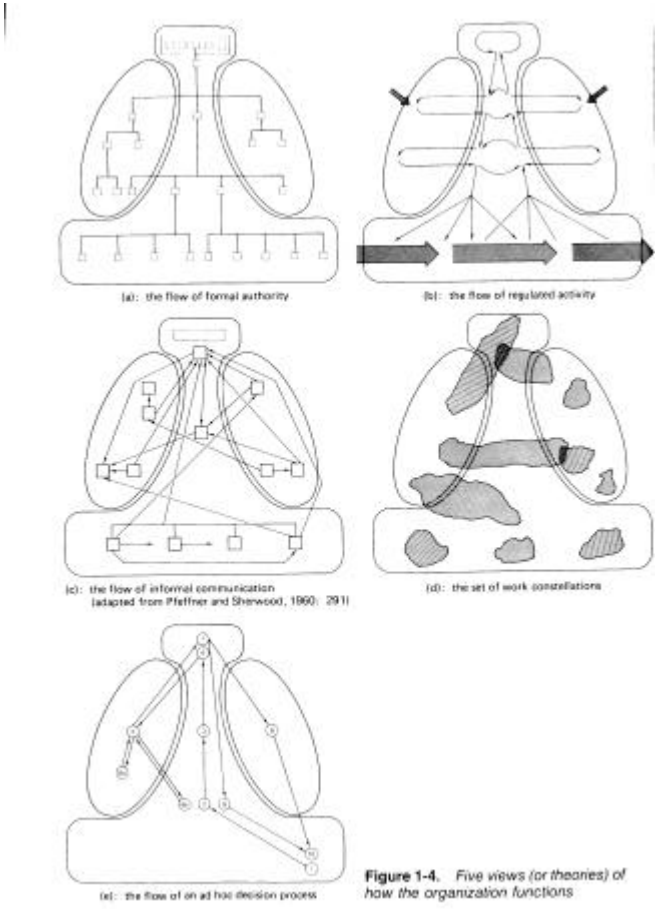


Figure 1-4. *Five views (or theories) of how the organization functions*

Mintzberg: Structures in Fives. Prentice Hall, 1983.