NC STATE UNIVERSITY

ENGINEERING ONLINE

Lecture Notes

Course Number:	CSC 513
Instructor:	Dr. Singh
Lecture Number:	4

Outline

Challenges of Electronic Business
- Business Environments
- Service Engagements

Architecture in IT

Contracts and Governance

XML Concepts and Techniques

XML Modeling and Storage

Summary and Directions

The Evolution of IT

- **Applications:** Control of computations hidden in code; integration a nightmare
- **Workflows:** Control abstracted out; integration still difficult
- **Standards-driven orchestration:** Integration improved; limited support for autonomy
- **Messaging:** Integration simplified by MoM and transformations; limited support for autonomy
- **Choreography:** Model conversations over messages; limited support for autonomy
- **Governance:** Administer resources via interactions among autonomous parties

Technical Service

- Generally, an abstraction of a computational object
  - Traditional, as in web or grid services
  - Improved: Abstraction of a "capability"
- Well encapsulated, i.e., a black box
- Interface defined at the level of methods or messages

Service Engagement

An aggregation of business relationships

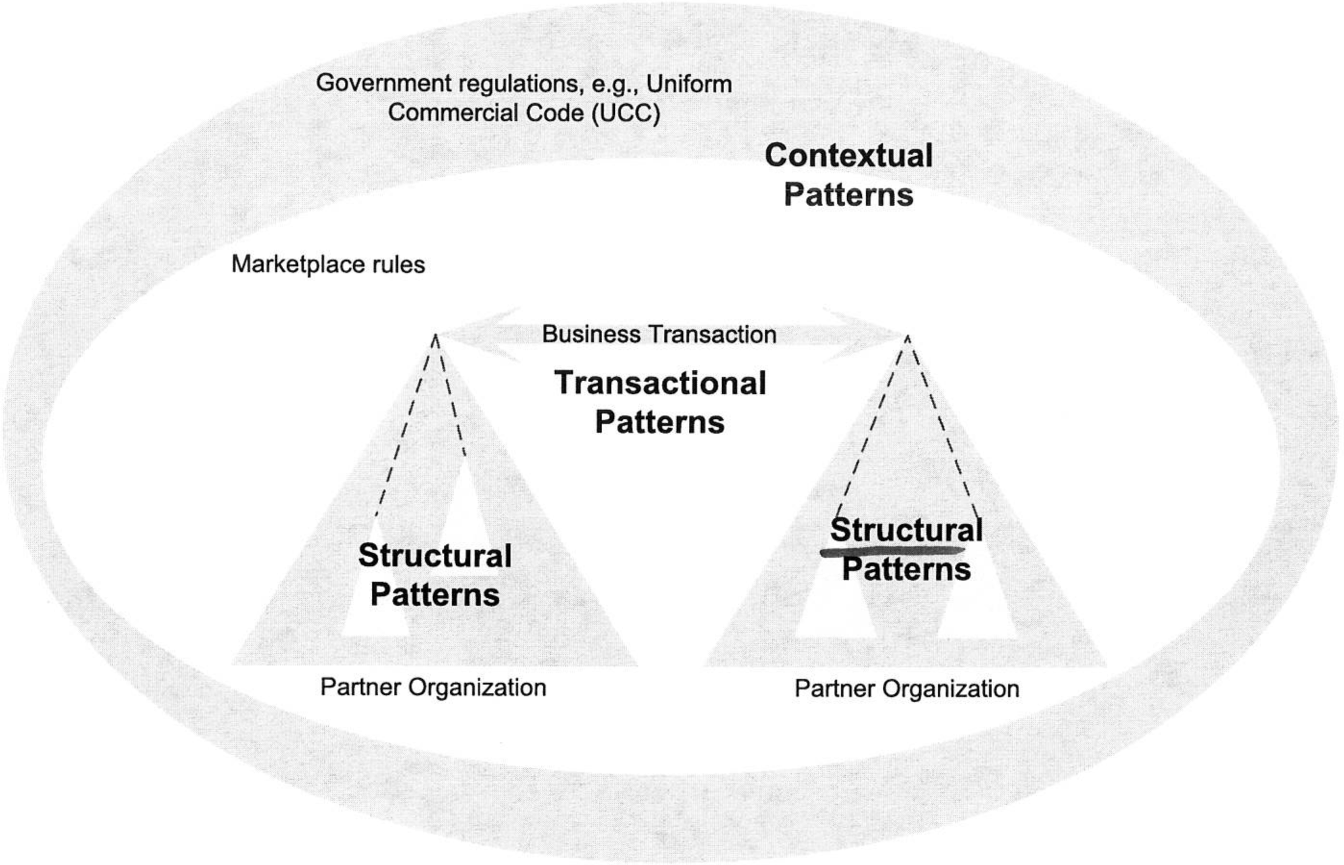
- Trillions of dollars worth of commerce conducted every year
- Characterized by
  - Independence of business partners
  - Coproduction
    - Participation by all, though not at the same level
    - Symmetric relationships: complementary capabilities and goals
  - Complex contracts among the partners
  - Participants are not black boxes

Business Service

Participant in a service engagement

- Characterized by transfer of value, not bits
- Typically long-lived with on demand enactments
- Instantiated on the fly (coproduction)
    - Unlike a product
    - Though may be constructed using products or about products

Conceptual Elements of a Service Engagement



- Transactional: main purpose and enactment, specifying value exchanged
- Structural: partnerships and contracts
- Contextual: setting of the engagement

Traditional Technical Approaches

Quite unlike a real-life service engagement

internal control

- Take participants flows (e.g., in BPEL, BPMN) as units of abstraction
  - Mix private policies and public interactions
  - Proprietary: may not be available for reuse
  - Context-laden: even when available, cannot be readily reused
- Focus on low-level (e.g., WS-CDL) or data-level meanings (e.g., OWL)
  - Ignore business-level significance of messages
  - Ambiguous; not verifiable

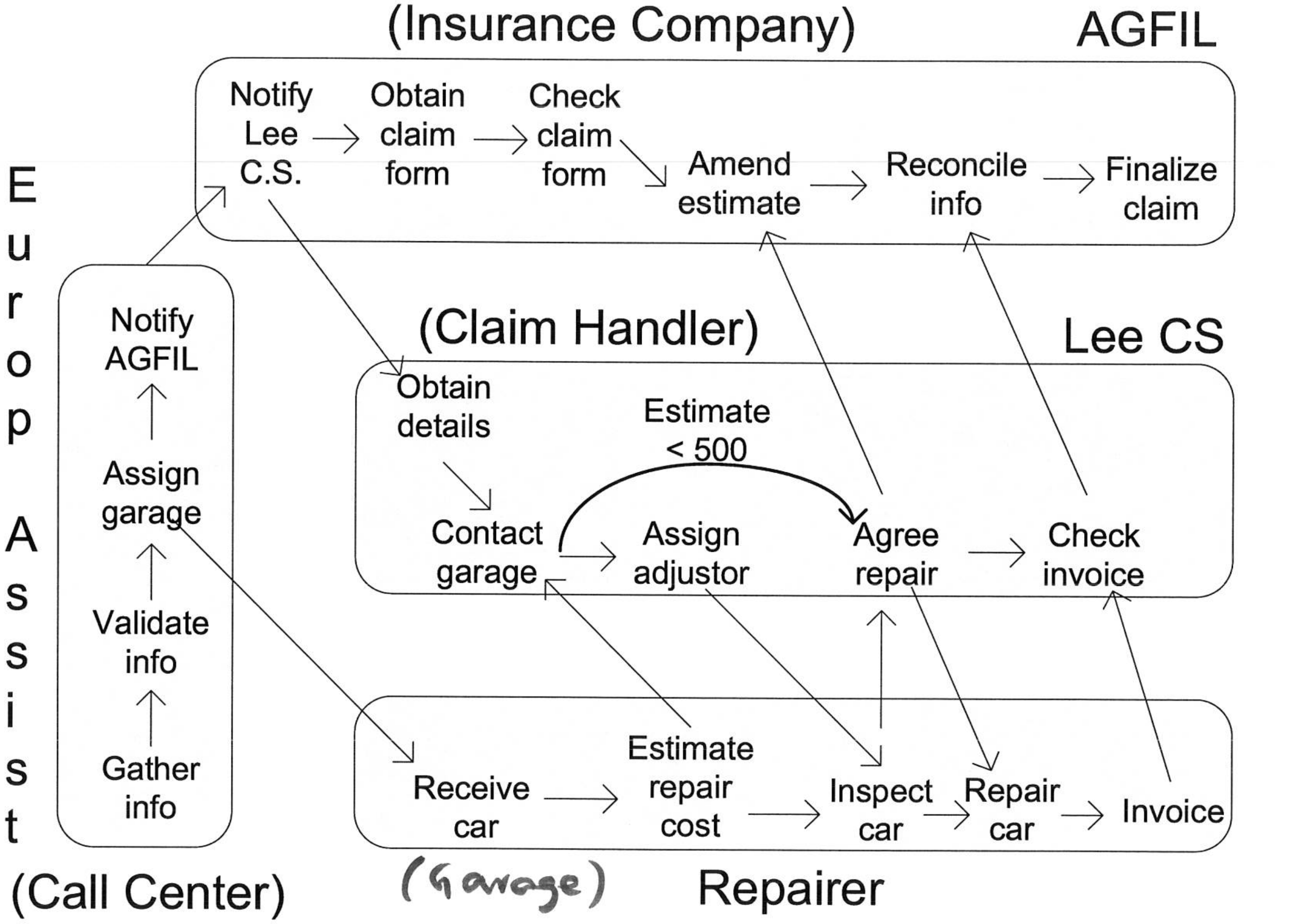
BPEL, BPMN, WS-CDL, OWL are well-known standards

A Real-Life Service Engagement

Allianz

Operationally over-specified as interacting flows

(Ireland)



(Insurance Company) AGFIL

Notify Lee C.S. → Obtain claim form → Check claim form → Amend estimate → Reconcile info → Finalize claim

(Claim Handler) Lee CS

Obtain details → Contact garage → Assign adjustor → Agree repair → Check invoice

Estimate < 500

Europ Assist

Gather info → Validate info → Assign garage → Notify AGFIL

(Call Center)

Receive car → Estimate repair cost → Inspect car → Repair car → Invoice

(Garage) Repairer

BUSINESS - IT DIVIDE

value	bits
(contracts	flows

SEEK:

- HIGHER-~~E~~LEVEL ABSTRACTIONS
- COMPUTATIONALLY EFFECTIVE

Outline

Challenges of Electronic Business

Architecture in IT

Contracts and Governance

XML Concepts and Techniques

XML Modeling and Storage

Summary and Directions

Architecture in IT

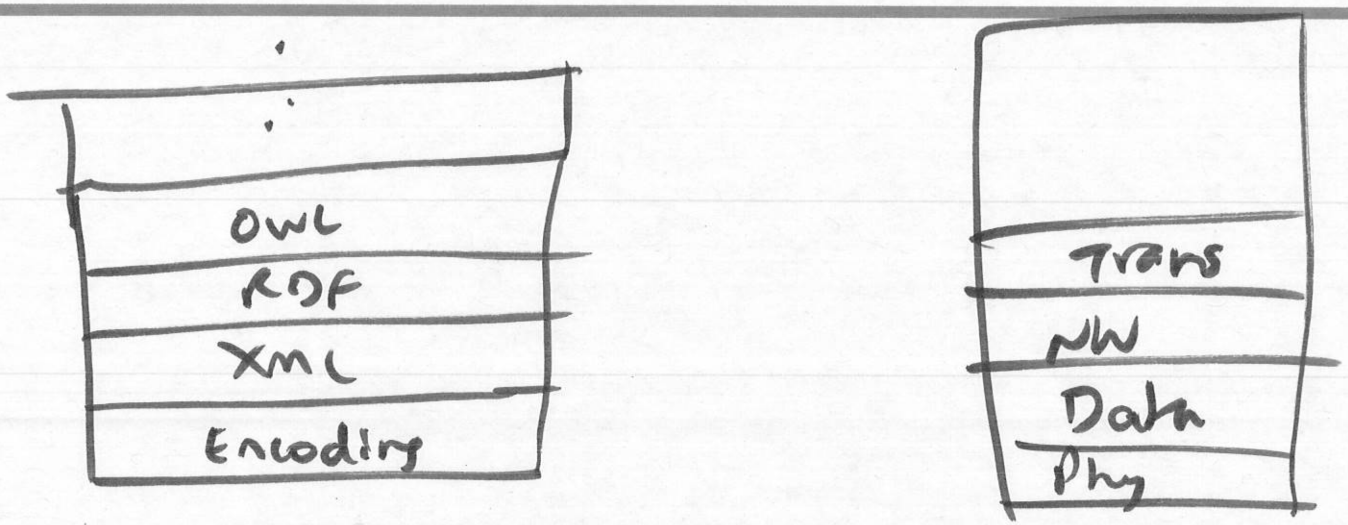
In the sense of information systems

- Important themes
  - Conceptualizing architecture
  - Enterprise architectures
  - Tiered architectures
  - Architecture as a basis for governance (next section)
- Not quite the same as conventional software architecture, though the topics are converging

Architecture Conceptually

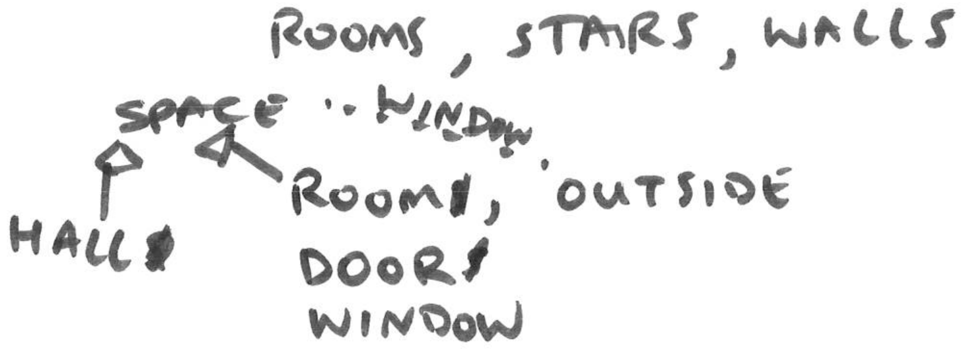
As opposed to the description of a system via a blueprint

- How a system is organized
- An over-used, vaguely defined term
    - Software architecture
    - Standards, e.g., Berners-Lee's "layer cake" and the networking standards
    - May include processes
        - That exercise the system
        - By which the system is built and maintained
        - By which the system is administered
    - May include human organizations

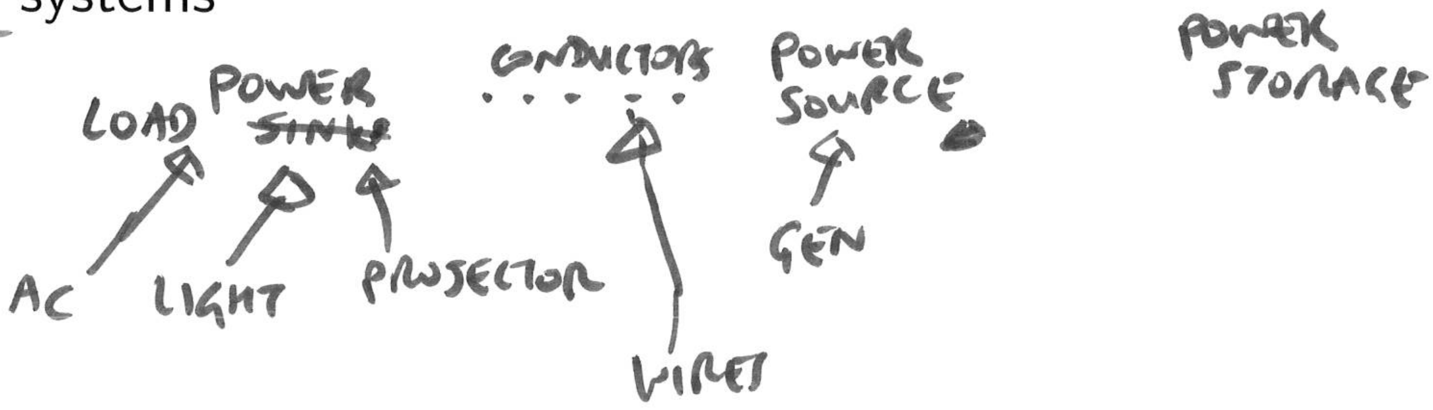

OWL
RDF
XML
Encoding
Trans
NW
Data
Phy

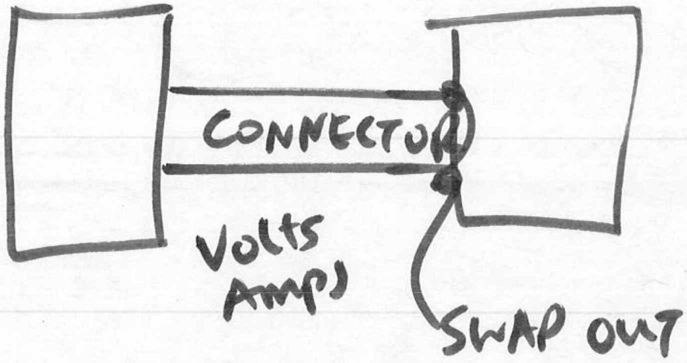
Exercise: Examples of Architecture

Identify the main components and interconnections



- Buildings
- Plumbing
- Power systems





Volts
Amps
Info

Understanding Architecture: 1

(Description) of a system in terms of modules
Conception

⊻

Components

what

how put together

- Two main ingredients of a system
  - Components
  - Interconnections
- *Openness* entails specifying the interconnections cleanly
  - Physical components disappear
  - Their logical traces remain
- *Information environments* mean that the interconnections are protocols

Understanding Architecture: 2

- Components and interconnections are not sufficient to characterize an architecture
- Two additional ingredients of an architecture

  HARD REQ

  - Constraints on the components and interconnnections
  - Patterns involving the components and interconnnections
- *Openness* entails the constraints
  - Do not apply on the physical components directly

Exercise: Examples of Architecture

Identify the main constraints and key patterns

REQ
- # People
- Height
- Material
- ELEVATORS

Room

CONSTRAINT

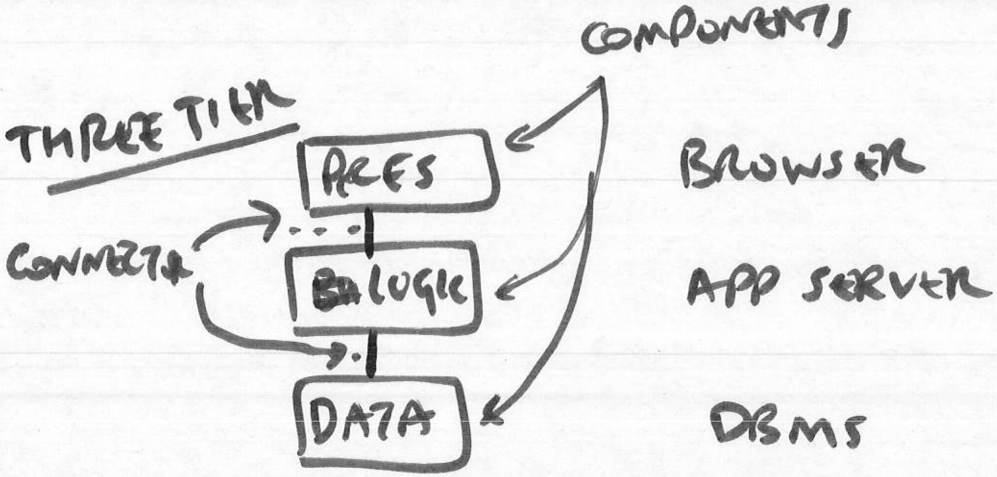
A ROOM MUST HAVE AT LEAST ONE DOOR

- Buildings
- Plumbing
- Power systems

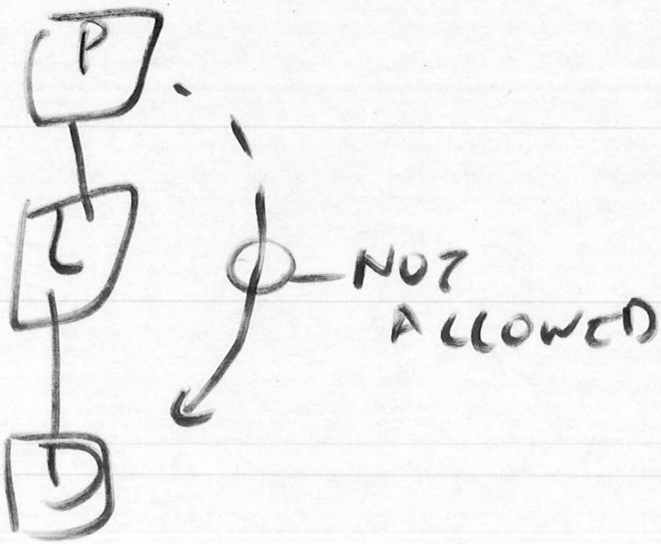
PATTERN

DOORS CONNECT ROOMS TO HALL BUT NOT ROOMS TO ROOMS

THREE TIER



CONSTRAINTS


P
L
D
NOT ALLOWED