Large-Scale System Partitioning
System Partitioning

- The logical view of the architecture emphasizes large-scale elements (LSEs):
  - Layers, subsystems, modules, frameworks.

- Before considering detailed object design, it is recommended to start with a draft design partitioning the system into LSEs.
  - 1. The elements
  - 2. Their interfaces
  - 3. The inter-element collaborations
  - These will change and evolve through the Elaboration phase iterations.
Large-Scale Element Collaboration

- It is Good Thing to achieve early clarification of the interfaces of LSEs, and their collaborations.
  - Of course, mutable in the spirit of iterative development.
  - This effort helps to partition parallel design and implementation by workers.
    “Stubs” can be defined for incomplete LSEs.
- Define the operations and parameters in detail.
- Be mindful of coupling and cohesion.
Inter-LSE Coupling

- Follow the *Don’t Talk to Strangers* (Law of Demeter, Whole-Part) pattern.
  - Avoid coupling into the internals of an LSE, or across several LSEs.
Large-Scale Element Collaboration

- It is a common experience to “reasonably” design the static partitioning into LSEs without “too much” trouble or error, during early design.

- However, the early design of the LSE interfaces and collaborations (dynamics) is usually quite speculative, and unstable.
  - These will change dramatically during the elaboration phase.
  - They will become more detailed and correct during the detailed object design of individual LSEs.
  - Since this is important, the Elaboration phase should focus on their clarification, via iterations that focus on “connecting the dots” between LSEs and on stress/load/evolution evaluation.
Logical Architecture Design

- Part of the Analysis & Design workflow.
- Identify the macro-level logical partitioning into layers and subsystems.

Small team

- Inception
  - 4 weeks Arch Des & Impl
- Elaboration
  - 4 weeks Arch Des & Impl
  - 2 weeks Req's & Biz case POC & Proto.
- Construction
  - 4 weeks Arch Des & Impl
  - 2 weeks Features
  - 2 weeks Features
  - ... 
  - 2 weeks Features
  - ... 
  - 2 weeks Iter N

Larger team

- Organization along time
- Organization along content
- Core Process Workflows
  - Business Modeling
  - Requirements
  - Analysis & Design
  - Implementation
  - Test
  - Deployment
- Core Supporting Workflows
  - Configuration & Change Mgmt
  - Project Management
  - Environment
Advice: Multi-Tier Layered Architectures

- Separate presentation and application logic, and other areas of concern.

![Multi-Tier Layered Architecture Diagram]

UI Layer

“Domain” or “Application Logic” Layer

Services Layer
- Persistence Subsystem
- Logging Subsystem
- ...
A Simple Logical Architecture

- In the UML, the logical partitioning is illustrated with package diagrams.
Advice: An Application Coordination Layer

- Consider an “application coordination layer” whose objects represent use cases. They may also hold session state.

```
UI
  Summaries  Editors

AppCoordination
  RentingVideo  PayingFines
    UCHandler    UCHandler

Domain
  Core  Evaluation Policies
```
It is useful to show the coupling with UML dependency lines.

A CASE tool can reverse engineer these diagrams.
Ordering Work

- What can we start with?
- What can we do in parallel?
  - How?
Advice: Early Load Testing

- In the Elaboration phase, “connect the dots” between the LSEs, application servers, ORBs, databases, and so forth, and do early realistic load testing against this skeleton architecture.
  - i.e., Load test the core architecture early, before filling in functionality.

- The UP philosophy of the Elaboration phase:
  - Get the correct core architectural design early, before investing in operational-oriented functionality.