

# Distributed Software Development

## Summary

Chris Brooks

Department of Computer Science  
University of San Francisco

# Themes

- We've covered a lot of territory this semester, but there are several themes that have shown up repeatedly
  - Communication
  - Heterogeneity
  - Coordination and Consensus
  - Scalability and Failure

# Communication

- Communication is an essential problem in a distributed system.
  - Synchronous/asynchronous, reliable/unreliable, in-order/not in-order
  - How to sequence messages without a global clock?
  - How to request services and reply to requests?

# Heterogeneity

- Distributed system components can be heterogeneous in many ways:
  - Software
  - Operating System
  - Hardware
  - Network
  - Capabilities
  - Requirements

# Heterogeneity

- Ramifications:
  - XML, RDF, XSLT are language/platform independent
  - Service-oriented computing
  - P2P architectures account for network differential
  - Distributed problem solving
  - Integration and interoperation

# Coordination and Consensus

- This is a fundamental problem in distributed systems
- Occurs at many different levels.
  - Causality
  - Group Membership
  - Leader Election
  - Mutual Exclusion
  - Task Assignment
  - Transactions
  - Replication
  - Problem Solving

# Coordination and Consensus

- Different sorts of problems demand different solutions
- By characterizing the problem, we know whether solving this will be trivial, difficult, or impossible.
- In general, we can tolerate either failing processes or a failing network, but not both.

# Scalability and Failure

- Many of the topics we've discussed are responses to failure or scalability:
  - Replication
  - Peer-to-peer
  - Leader Election
  - Group Formation

# Scalability and Failure

- We've also talked about how failure and scalability can affect an algorithm's implementation.
  - Centralized bottlenecks
  - Types of failure
  - Failure detection and recovery
  - Multilateral negotiation
  - Search in large-scale networks