**Graceful Degradation:**
Component failures reduce functionality rather than cause system failure

**Research Questions**
- How might system architecture contribute to achieving graceful degradation?
- Can an architectural style make graceful degradation an inherent property?
- Can we methodically generate an architecture for graceful degradation?

**Exploratory Approach (a first step):**
Examine current architectural styles and categorize their properties; for example:

### Intelligent Hierarchical Control (IHC)
- Functions organized into hierarchy of layers
  - Failures in one layer not propagated through system
  - Modularization
- Modeling tactical and intelligence layers is difficult
- Hierarchy can cause data flow bottlenecks

### Software Functional Hierarchy:
- Intelligence
- Tactical
- Servo

(Mesarovic89)

### Distributed Intelligent Control and Communications (DICC)
- Distributed system control in “smart” sensors/actuators
  - Reduces single-point failures
  - Flexible allocation
- Communication intensive
- Control on network may cause real-time misses
- Harder to detect failures in distributed functions

(Upender93)

### Device-Specific Agents (DSA)
- System decomposed into small functions strongly coupled to specific hardware devices
  - Parallelization
  - Designing for growth (“forward compatibility”)
  - Strong function-to-device correlation
- Parallelizing serial functions is difficult

(Carnegie Mellon)

GM

**System Architectures for Graceful Degradation**

Charles P. Shelton