# Network Centric Systems Subgroup

- Ian Foster
- Jim Fowler
- Doug Lea
- Adam Porter
- Steve Ray
- Rick Schantz
- Jim Waldo

# Outline

- New Forces
- Example Applications
- Technical Problems
- Research Directions and Approaches
- Other Required Research
- Benefits
- Observations on Approach

# Forces

- Everything is a computer
- Everything is a networked computer
- Everything is potentially interdependent
- Things connect to the real world
- Increasing heterogeneity
- Increasing dynamic range and volatility of the dispersed interconnection

# Example Applications

- Traffic control
  - Sensor data from 1000s of vehicles
- Theater battle management
  - varying granularities of coordination/missions in a hostile environment
- Supply chain management
- Community analysis of scientific data
  - Soft-real-time response and query optimization from 1000s of users, via coordinated management of 1000s of resources
- Home power management

# Technical Problems

- Resource conscious programming: QoS, time/dependability/energy/footprint
  - Lack of a computational model that allows for engineering tradeoffs
  - Lack of end-to-end properties in composite systems
- Dynamic resource behaviors (in time and space): failure, heterogeneity, etc.
  - Again, lack of a computational model and end-to-end properties
- Legacy: things that were not designed to work together now need to

# Technical Problems (continued)

- Risk, trust and control management
  - Policy/security/admin domains
  - Safety and validation of very dynamic systems
  - privacy
- Scale
  - Number of entities
  - Size of entities
  - Number of entities composed in a single computation
  - Timescale over which network centric systems exist and non-stop behavior

# Compelling Research Directions and Approaches

- Four complementary thrusts that need to be addressed at all levels; one crosscutting/ coupling issue
- (1) Programming models that deal with “the problems”
  - New engineering tradeoffs
  - Operation at massive scales
  - Dynamic resource behaviors
  - Risk, trust and safety management

# Directions and Approaches (continued)

- (2) Work on basic mechanisms that underlie "the problems"
  - Resource tradeoffs: QoS mechanisms, RT, etc
  - Adaptive behavior
  - Scaling in various dimensions
  - Distributed control
- (3) Multi-level resource management that deal with "the problems" in the aggregate
- (4) Construction of large systems by dynamic composition of (small scale) network centric components; interoperability

# Other Required Research

- Tools that allow users to use new computational models
- Validation, simulation, verification: understanding a new class of highly complex systems
- (Automated) configuration and management of large-scale distributed applications

# Benefits

- Societal
  - Build things that we just can't build now
  - Increase reliability/control of future networked systems etc.: design by engineering not debugging
  - reduce exploding software development costs for the complex requirements of network centric systems in the real world
- Training
  - Seed industry (and academia) with people able to write new software & engineer new systems

# Observations on Approach

- Teamwork is fundamental
- Need for large-scale projects to help us discover the real problems and validate partial results
- International collaboration