Network Centric System Building
A Direction Forward

Rick Schantz

BBN Technologies

SDP Workshop

April 18, 2001

Taking Stock

- Connecting parts is easy, building end to end systems is hard
- Volatility in the computer environment as the rule, not the exception
- emphasis on “managed properties” of the applications and their tradeoffs (real time behavior/predictable performance, dependability, security/safety)
- trends
  - scaling way up, without reprogramming the pieces
  - scaling way down, without reinventing the concepts
  - Larger footprint software, smaller components, higher expectation for a more controllable and consistent “user” experience
- need for lots of network-centric applications each with their own right thing vs. standardized one size fits all infrastructure, by ordinary system engineers
- interesting behavior is between the nodes -> emphasis on evolution of multi-level middleware and how it fits with basic programming concepts and environments

Integrated Adaptive System Concept

System-wide QoS

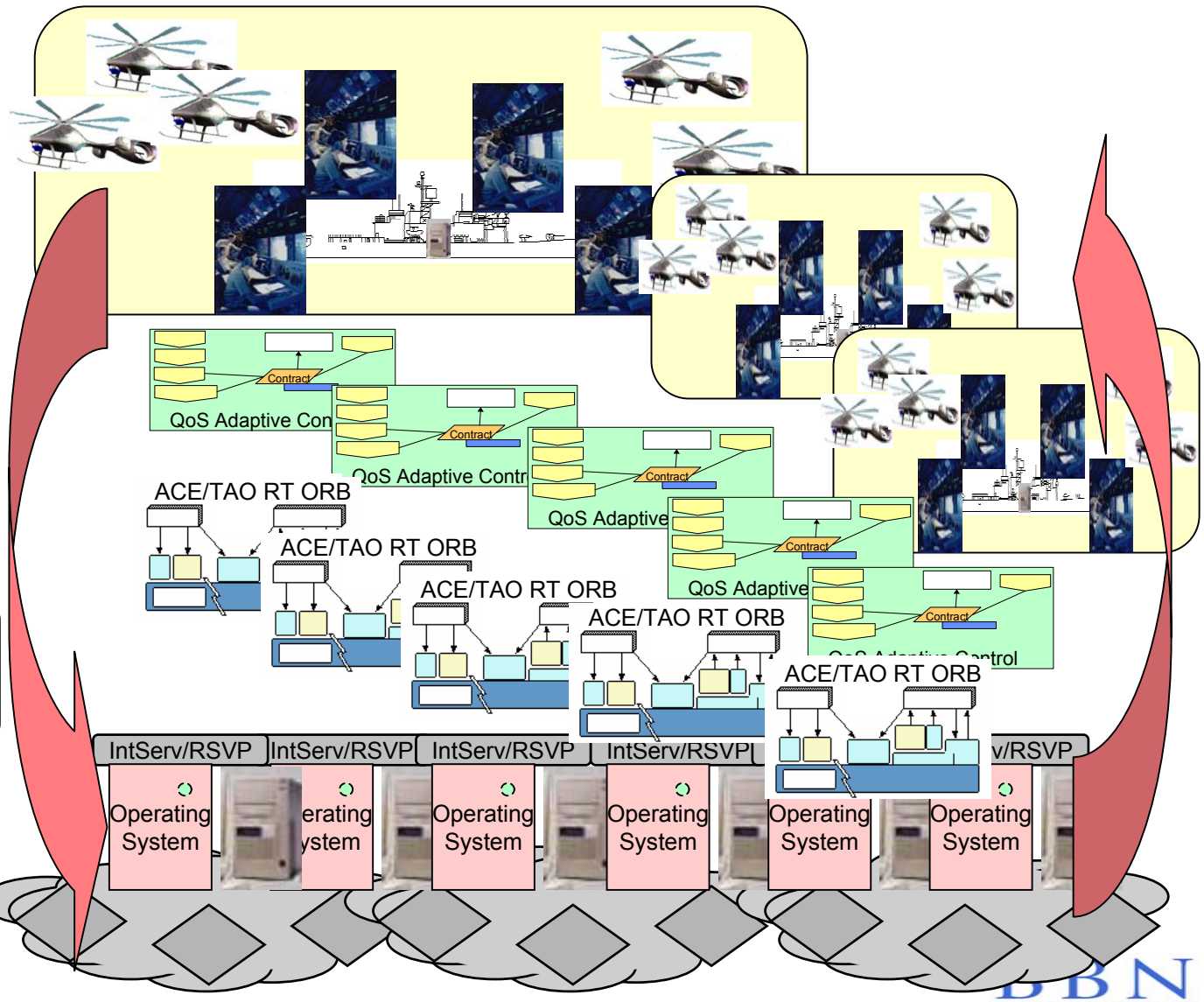
Application or Domain-specific QoS

Common Middleware Services QoS

Distribution Middleware QoS

Operating System QoS

Network QoS



Moving forward

- "Property" and load invariant behavior over changing environments and events becomes fundamental to development methodology and part of the normal application runtime
- Integrated properties rule, in scaled up and down environments along with formal repeatable software engineering disciplines to design, construct, validate and safely operate such systems
- moving away from
  - current computer centric culture of "best effort" or "when you need it most, it works the least"
  - formally linking parts and then fixing the shortfall
- moving toward
  - human centric & unattended embedding culture with systems responsible for adapting to provide consistent experience
  - system constraints first, with parts filled in accordingly and varying with time

Current Paths to Build On and New Directions

- Build on formalized connectivity abstractions and services orientation
- focus on adding concepts for platform independent “property” management tradeoffs (end to end composability with increasing and decreasing scale)
- hint at a direction in aspect oriented programming
  - add to current software engineering directions or new path?
- Some Specific Directions
  - making integrated, multiple (property) points of view a dominant programming orientation (alongside functionality)
  - embedded programming support for validation, testing and safety over variable/adaptable conditions
  - organizing and controlling self-configuration and self repair
  - defining appropriate scoping and granularity for integrated multi-layer resource management
  - organizing strategies for using throwaway low cost elements in large redundant numbers