



# Fine-grained Middleware Composition for the Boeing NEST OEP

Venkita Subramonian, Chris Gill,  
Huang-Ming Huang, Stephen Torri  
Washington University, St. Louis  
{venkita,cdgill,hh1,storri}  
@cs.wustl.edu

Jeanna Gossett, Tom Corcoran, Douglas Stuart  
The Boeing Company, St. Louis  
{jeanna.m.gossett, tom.corcoran, douglas.a.stuart}  
@boeing.com

**Research Supported under the NEST program  
DARPA contract F33615-01-C-1898 with AFRL**



# Topics

- NEST overview
- Boeing OEP overview and architecture
- Motivation for nORB
- Relationship of CORBA to nORB
- nORB components
- Future work



# NEST overview

- Networked Embedded Software Technology
- Distributed Real-Time system with 100 to 100,000 networked nodes
- Resource constrained hardware components
- Requires fine-grained fusion of hardware and software components
- Applications in advanced avionics and space systems, weapon systems, wireless devices



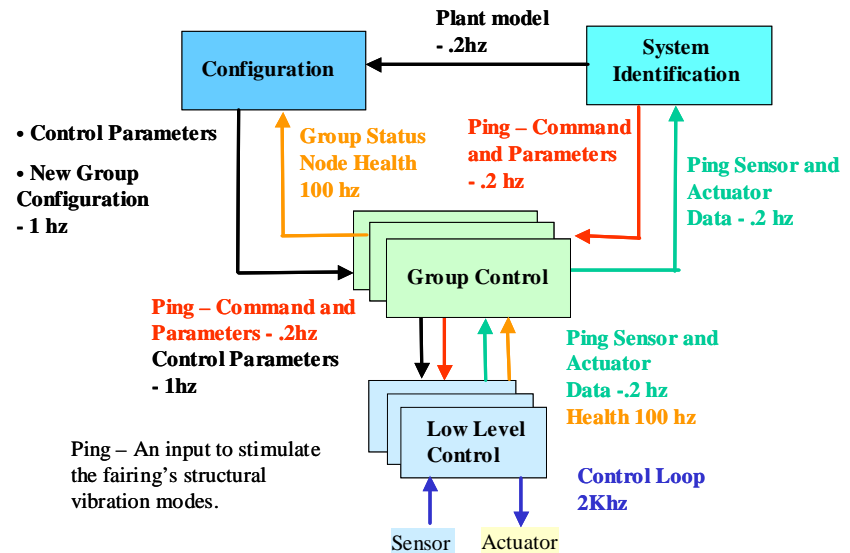
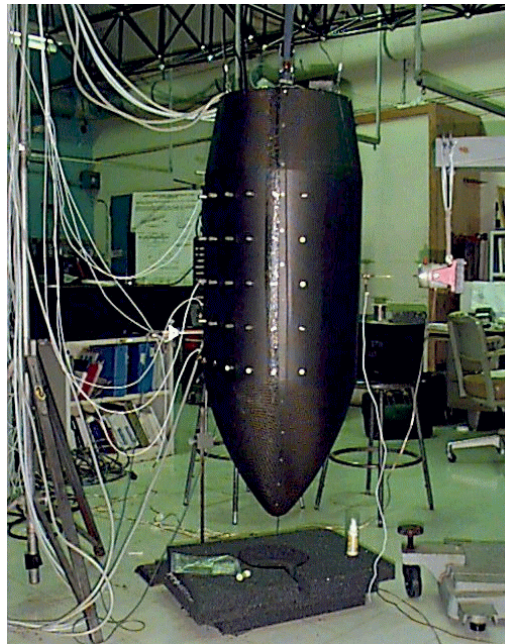
# NEST middleware services

- Real-time Coordination Services
  - Fault tolerance
  - Data exchange
  - Synchronization
  - Self-stabilizing protocols
  - Replication
- Automated synthesis of services



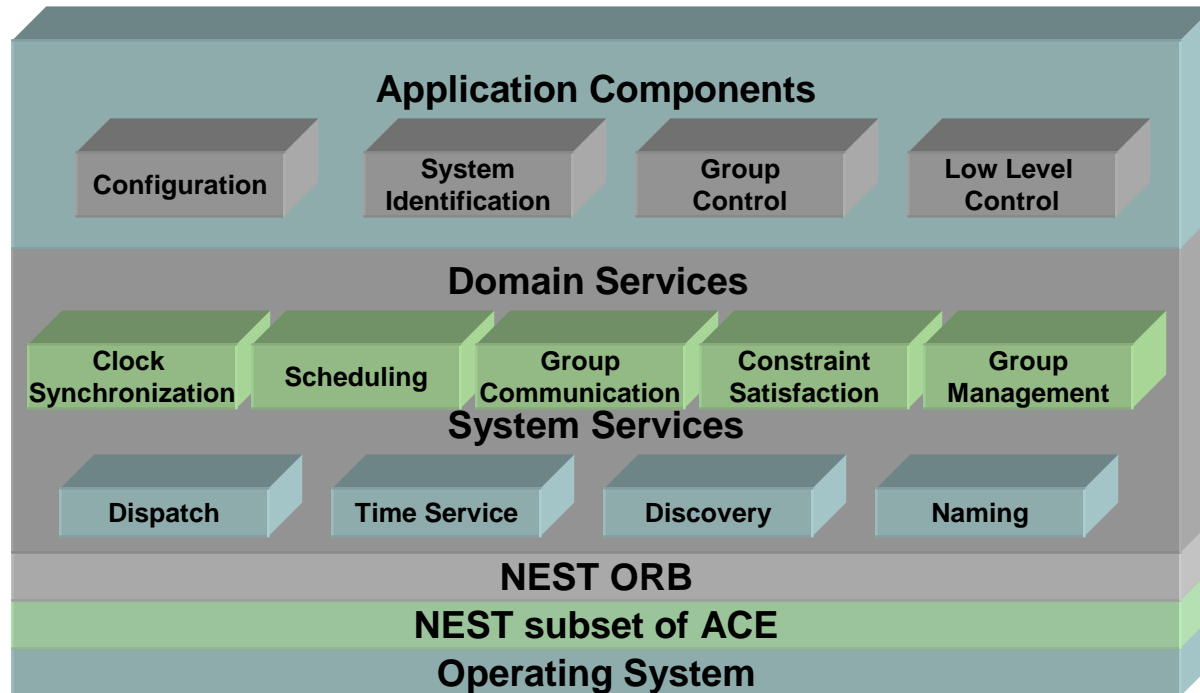
# Boeing NEST OEP

## Acoustic and Structural Damping





# Boeing OEP architecture





# Why Not a Full CORBA ORB?

- Mis-match with the NEST Domain
  - Most ORB features not necessary in NEST
  - Application specificity is a key issue
  - Limited number of “objects” in NEST
    - Scalability in terms of number of nodes
      - not number of objects per node
  - Footprint constraints



# Motivation for nORB

- Provide primitives for NEST middleware and services
- Limit communication middleware features to application requirements
- Reduce footprint
- Assure Timeliness, Support Adaptation
- Top-down approach very difficult, prolonged, and error-prone to achieve
- More minimal than minimum-CORBA



# Related work

- Ubicore
  - Framework to hide different middleware protocols like IIOP, COM, Java RMI
  - Generic request broker for hand-held devices
- MicroQoSCORBA
  - CASE-tool based approach to middleware customization/composition
  - Prototyped in Java



# CORBA-ness of nORB

- GIOP messages –
  - Request, Reply, Locate request, Locate reply
- Object Adapter, albeit a very simple one
- Remote Object References
- IDL compiler generated stubs and skeletons
- Limited form of Exception support



# nORB interface

- registerObject()
- resolveReference()
- objectToString()
- stringToObject()
- init()
- run()
- shutdown()

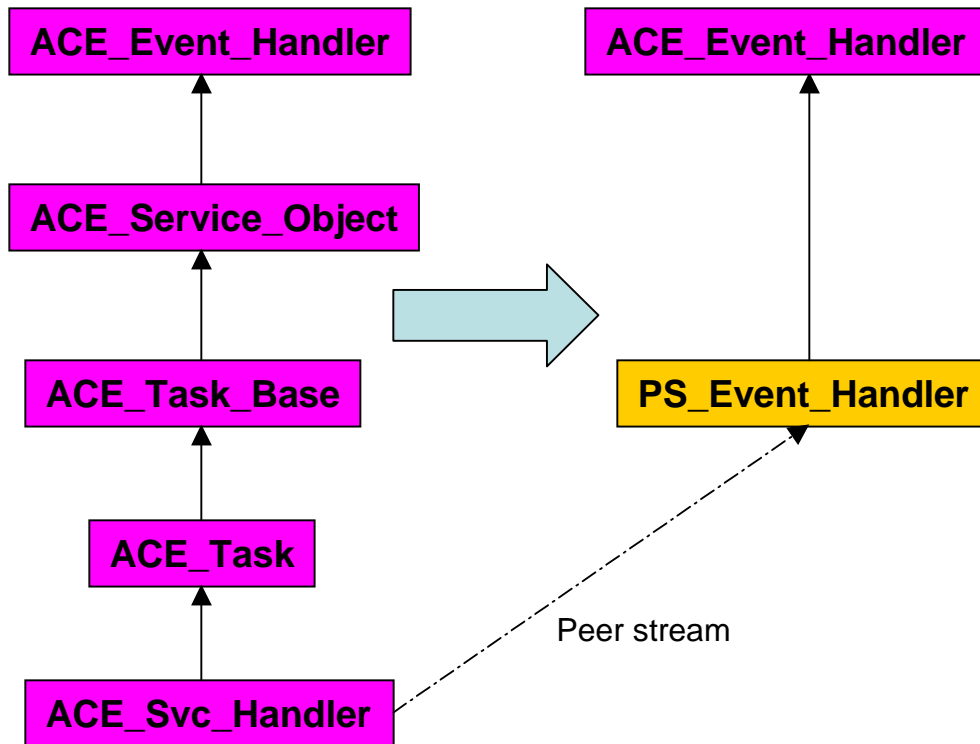


# nORB - Under the hood

- ACE - Adaptive Communication Environment
  - Highly efficient, portable primitive components
- ACE components used in NEST
  - Reactor, Timer, Threads, Acceptor, Connector, CDRStream
- Further decomposition of ACE primitives on an as-needed basis.



# Towards a Fine-Grained Substrate



- Decoupling concerns
  - Reactor
  - Acceptor
  - Connector
  - Event Handler
  - Svc Handler
  - Task



# nORB Dispatcher

- Based on Kokyu Scheduling Framework
- Adaptive Scheduling in NEST
- Scheduler as a middleware service in NEST
- Dispatching infrastructure to support the scheduler
- Support for FIFO (RMS)
- Future support for different scheduling strategies – EDF, MLF



# Dispatcher interface

- `initQueues()`
- `clearQueues()`
- `registerDispatchable()`
- `mergeSchedule()`
- `clearSchedule()`
- `dispatch()`
- `start()`
- `stop()`



# Future work

- Infrastructure Composition techniques for scheduling/dispatching
- Further decomposition of ACE classes to achieve additional footprint reductions
- Apply Generative/Generic/AOP techniques for customizable middleware composition
- nORB performance metrics/optimizations