Distributed Control and the Machine Bus Architecture

OMG Robotics Working Group

Introduction

- Chris Cooper, Software Architect at the Machine Bus Corporation.
- Machine Bus is a Chicago based company that specializes in distributed control systems for robotic and automated machinery applications.
- http://www.machinebus.com

Presentation Summary

- Distributed control architectures (DCA) for robotic systems
- Distributed control architecture and the product life-cycle
- What to expect from DCA in the near future

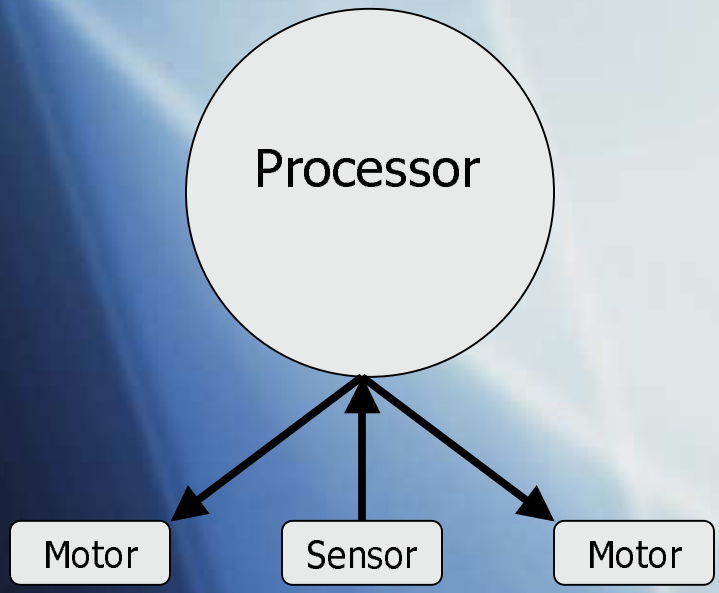
Distributed Control Architectures for Robotic Systems

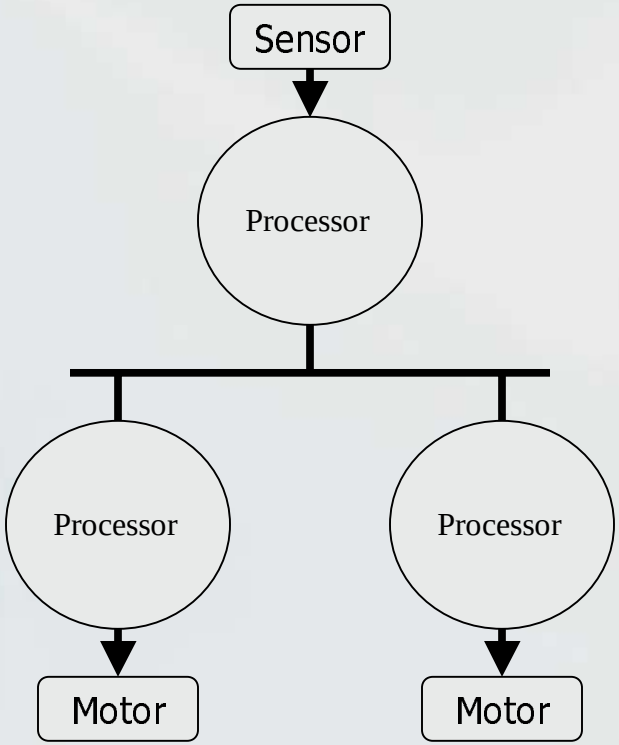
Definition, Benefits, and Selection

What is a distributed control architecture

- A distributed control architecture is a method of designing a control system in a way that spreads control functionality among many processing units
- This is in contrast to a centralized control architecture where all control functionality resides in a single processing unit
- Sensory input and actuation are coupled→

Distributed Control vs. Centralized Control





The Benefits of a Distributed Control Architecture

- Modular
- Reusable
- Scalable
- Extensible
- Reliable

DCA Benefit: Modularity

- Decompose complexity into smaller, more logical modules
- Object-oriented software maps well onto modular hardware
- Interfaces: prototyped modules can be swapped in/out with production optimized modules

DCA Benefit: Reusability

- Modular designs lead to reusability
- Functionally decomposed, fine grained modules can be reused within as well as across multiple applications
- Opens up a market for robotic hardware and software components

DCA Benefit: Scalability

- True parallel processing
- Increased real-time capacity
- Support for heterogeneous designs
- Avoid simulated processes that use context switching

DCA Benefit: Extensibility

- Add functionality by adding modules
- Devices with various I/O can be normalized.
- Integration of disparate systems
- Diagnostic and debugging equipment can be added
- Simulation possible

DCA Benefit: Reliability

- Modules are easier to test and verify correctness
- Reduced wiring complexity
- Redundancy
  - Processing nodes
  - Communication paths
- Partial failure

Polybot Example

QuickTime™ and a
YUV420 codec decompressor
are needed to see this picture.

Courtesy of Palo Alto Research Center Inc.

- Modular
- Reusable
- Scalable
- Extensible
- Reliable

Selecting the Proper Platform

- Deterministic: Real-time capable
- Fault-tolerant
- Topology/Synchronization
- Development Environment

Platform Selection: Determinism

- Ability to be certain that an operation will complete within its allotted time frame
- Factors affecting determinism:
  - Medium access control
  - Data frame size
  - Data frame prioritization

Platform Selection: Fault Tolerance

- The ability to identify and recover from errors
- Hardware vs. software implementation
- Various type of Error detection
- Fault confinement
- Immunity to electrical noise

Platform Selection: Communication Structures

- How processing nodes are physically and logically connected affects performance and reliability
- Avoid single points of failure
- Extensibility and flexibility
- Activity coordination

Platform Selection: Development Environment

- Rapid Application Development
- In-system programming
- Debugging and diagnostics
- Tool chain
- Language support

Distributed Control Architecture And the Product Life-Cycle

Scenario: A Robotic Arm

Scenario: A Robotic Arm

- A typical robotic arm
- Three degrees of freedom.
- One motor and one rotational encoder per joint.

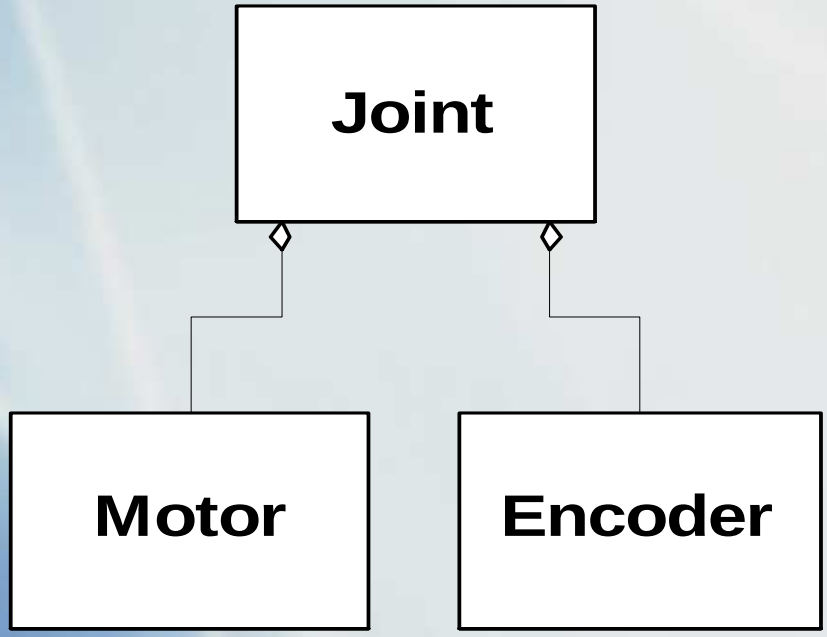
The Product Life-Cycle

- Design Phase
- Production Phase
- Service in the field

Design Phase

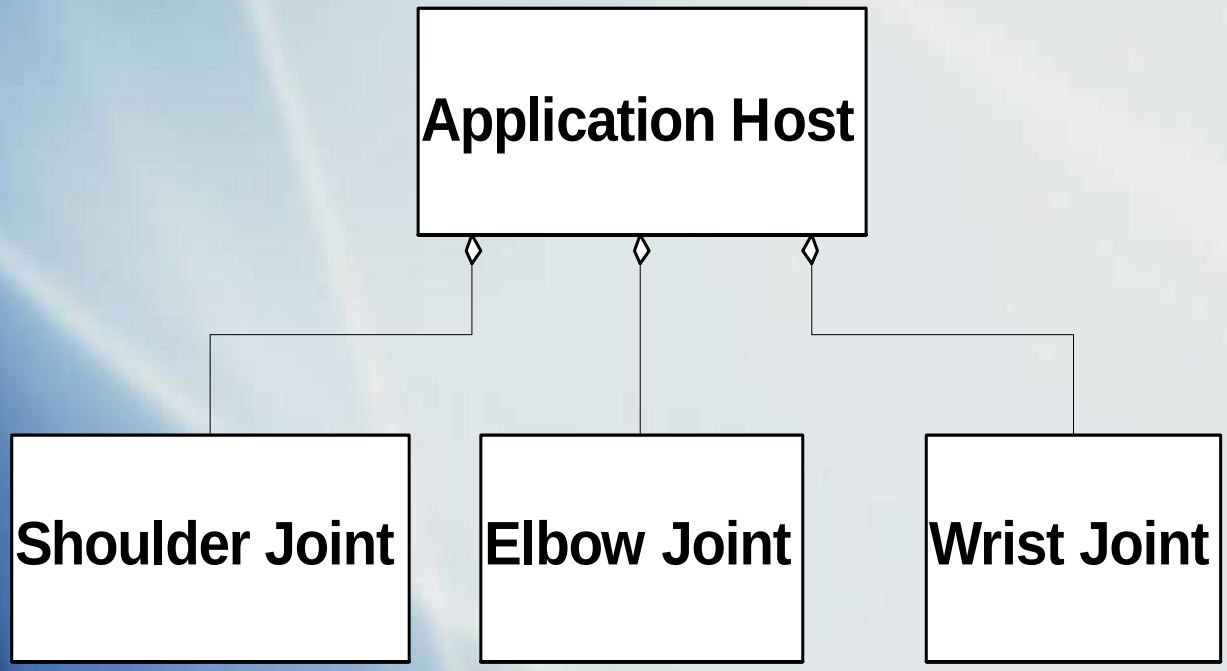
- Control module design
- Application design
- Prototyping and testing

Control Module Design



Joint Control Module

Application Design



Arm Application

Prototyping and Testing

- Simulation
- Testing frameworks
- Flexible redesign

Production Phase

- Optimization to reduce cost and size
- Quality control

Service in the Field

- Diagnostics
- Module replacement

Technology Affecting DCA in the near future

Application of Model Driven Architecture

- Open vendor neutral approach to system specification and interoperability
- MDA is a way to separate the architecture of an application from its implementation.
- In MDA, platform-independent models (PIM's) are initially expressed in a platform-independent modeling language, such as UML

Super Distributed Objects -

- A PIM for Distributed Control Architectures and Robotics?
- Attempts to model real world entities (e.g. devices) as objects, deploys them in a highly distributed environment
- Needs RT and a robotic domain profile (RFP work has just begun)

minimumCORBA

- a subset of CORBA designed for systems with limited resources.
- Are robotics devices too limited for minimumCORBA?
- Proxy approach (similar to JTRS and SCA's proxying of FPGA's)

Distributed Control and the Machine Bus Architecture

OMG Robotics Working Group

chrisCooper@machinebus.com
info@machinebus.com