Control System S/W Architecture R & D Proposal

Robert C. Sass

25-Aug-99

Agenda

- Topics
  - Infrastructure
  - Object/Relational Database
  - NLC EDD/DM
  - CORBA/DCOM
  - Selected Component Implementation
- Conclusion

Infrastructure

- Education
  - OO Requirements class with UML
  - Java Class
  - OO analysis class with UML and the selected analysis tool
  - OO design class with UML and the selected design tool
  - OO implementation class in Java
  - Oracle 8i and Designer/Developer changes
  - Hire/contract a Mentor
  - Self Study time

Program Development Tools

- Investigate UML based design tools for building distributed OO systems
- Evaluate and select perhaps 2 Java and C++ compile and test tools
- Survey and make a recommendation for a code management system
- Software release control
- HyperNews

System Management Tools

- Evaluate the current market for distributed system management tools
- Write a requirements spec for our management tools needs
- Implement some subset of system management functions for use during the R & D phase

Test Hardware

- NT Server
- UNIX server
- LINUX Server
- One Oracle 8i instance just for our use
- Several test workstations
- Several IOCs
- Local isolated network connecting everything
- Large RAID data storage system

Object/Relational Database

- Get a local instance of 8i just for our use
- Thoroughly evaluate 8i
- Test table partitioning and indexing improvements for large tables.
- Evaluate and define how we can use the File System based on Oracle Tables
- Evaluate how we can use the extensive effort made by other labs
- Define how we can build complete sophisticated applications around 8i and our display capabilities
- Performance test 8i for a very large database
- Evaluate OO databases if necessary
- Design and implement test data to support the rest of the R & D effort
- Continue support of the modeling and device database for the NLC

NLC EDD/DM

- Write a full-scale requirements document
- Obtain the code and documentation of the Java DM done by Gary Cooper
- Define and design the initial implementation
- Define performance tests
- Make a series of “production” releases for use by other R & D efforts

CORBA/DCOM

- Implement one of each CORBA; freeware and commercial
- Evaluate DCOM/CORBA bridges
- Evaluate Microsoft DCOM tools
- Evaluate and define which CORBA services we need
- Evaluate high-level application builders like Iona's Orbix
- Define how we'll use Enterprise Java Beans i.e. server-side components
- Construct an operating framework in the test system that can be used to run NLCDM and its interactions with the Database, EPICS and other components

Component Implementation

- Using the selected tools, do a system design for the NLC control system in ultra miniature
- Using the selected tools, do a detailed component design for a small set of representative functions
- Define system level performance and throughput tests
- Implement at least Java & C++ components; NT, UNIX & LINUX
- Implement a legacy component such as modeling
- Implement several database components
- Configure selected applications to use SLC data via PCAS
- Combine the whole shebang into a set of displays representing the NLC Control System in ultra miniature

Conclusion

We can expect the whole business of the Object Web to evolve rapidly over the next few years, just during the time period for our R & D. We will have a Golden Opportunity to see what works and what doesn't. By the time we cast our Architecture in concrete, we'll have hopefully made all of the mistakes and can wisely design a framework that we can build on for the next 20 years or so!

That's the Theory.