Aida; Accelerator Integrated Data Access

Middleware Continued

Episode 3; Verdi’s Revenge

In the Last Episode...

The Un-classic not-so-thin client with a JVM & XML

<table>
<thead>
<tr>
<th>CDEV</th>
<th>CORBA compliant ORB</th>
<th>Channel Access</th>
</tr>
</thead>
</table>

Non-I OC Application Cloud

CDEV | CORBA compliant ORB | Channel Access | I OC Applications
The Current Episode

Support get/set/monitor any data from/to any source

- Application Layer
- Middleware
- Heterogeneous NLC Database Cloud

- Application
- Configuration Data
- Online Device Data
- Enterprise
- Archive
- EPICS
- Attribute Dictionary
- Name Service
Supporting Ideas

- Name and schema mapping
- Data location discovered at runtime.
- CORBA middleware & services
- Server resilience (auto reconfigure on server down)
- API and OPI
- Language and OS independent
- Collections, Aliases, Range Names
Each name.attribute{.attribute}$_{0+}$ maps to a database query
Salient Features

- Given a name.attribute, AlDA finds the source and appropriate query
- Uses its own name services
- Data Location and Query persist/cached (not rediscovered)
- C++ & Java Bindings
- Can code as API or OPI
Simple OO Example

Class QUAD {
    private AIDA d;
    QUAD( string name ) { d = new AIDA(name); }

    float get( string attribute ) {
        return d.get(‘B_desired’);
    }
    float[2][6] getTwiss() {
        org.omg.CORBA.Any t= d.getAny(‘Model.Twiss’);
        ...
    }
}

QUAD fred(‘quad:pr04144’);
float bfieldwanted = fred.get(‘B_desired’);
float twiss[2][6] = fred.getTwiss();
Where We’re At

- Have a complete set of requirements.
- Accessed EPICS & old SLC data via Corba.
- Java Corba -> JDBC -> Oracle
- C++ Corba -> ODBC -> Oracle
- Sample Name & attribute databases with query mapping to get data
- Have a Java Orb on VMS
- Testing Notify Service for Monitors
- Ready to implement something real
Cosmic CUD

Alpha

SCP

SLC DB

Aida

IOC

MANAGE CUD DISPLAYS

CUD DISPLAY

CUD DISPLAY

Oracle CUD Configuration