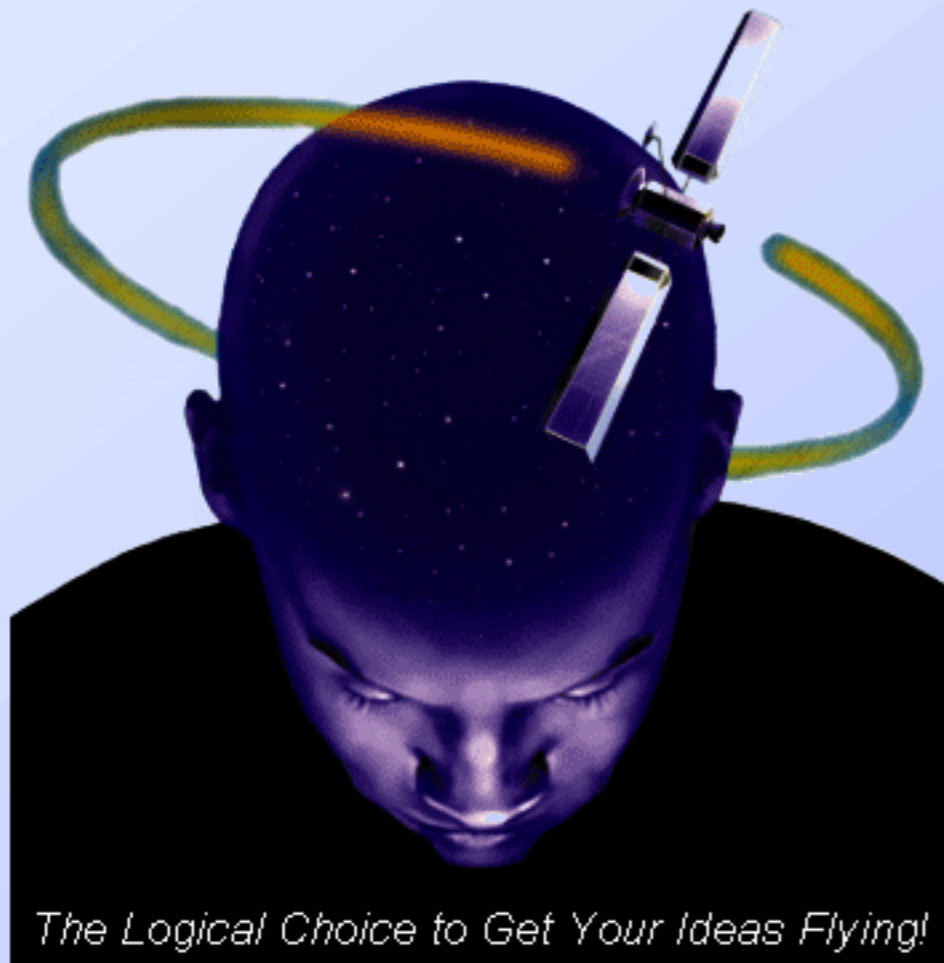


Deploying a Common Software Architecture for Real Time Launch, Test and Satellite Ground Systems

GSAW 2003



Rob Andzik
andzik@rtlogic.com

RT Logic!

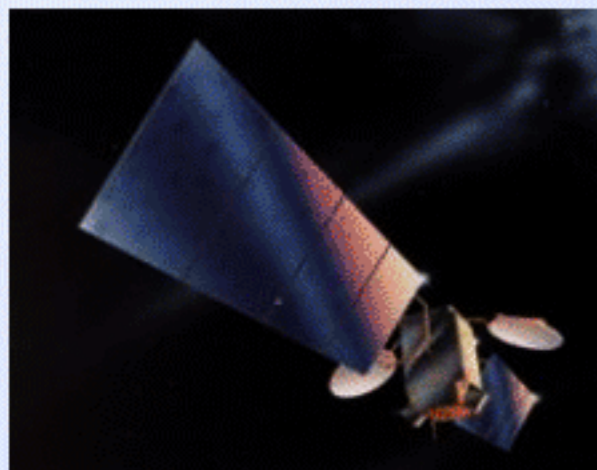
1042 Elkton Drive
Colorado Springs, CO 80907

(719) 598-2801

www.rtlogic.com



- Scope Of The Telemetrix Software Architecture
- Considerations & Approach
- Component Design
- Middleware And Framework
- How the Architecture is Deployed
- Where the Architecture Is Used



- ▶ **Satellite Ground**
 - ▶ Control Centers
 - ▶ Ground Antenna

- ▶ **Satellite Test**
 - ▶ Vehicle Tests
 - ▶ Payload Tests
 - ▶ Launch Site Tests

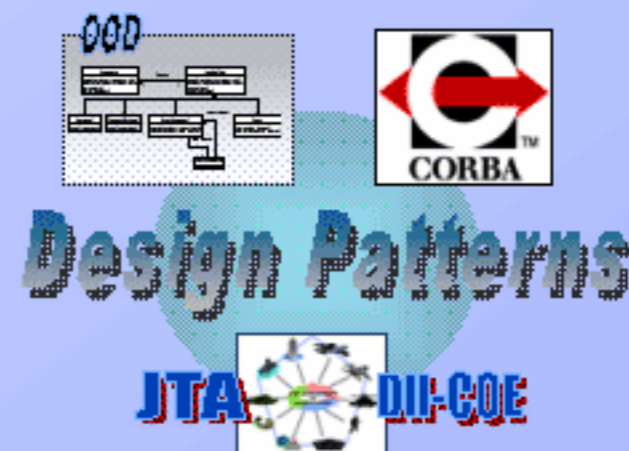
- ▶ **Launch Vehicle/Range**
 - ▶ Telemetry Acquisition
 - ▶ Resource Management
 - ▶ Radar



**Common Software
Architecture Across All
Three Business Areas**

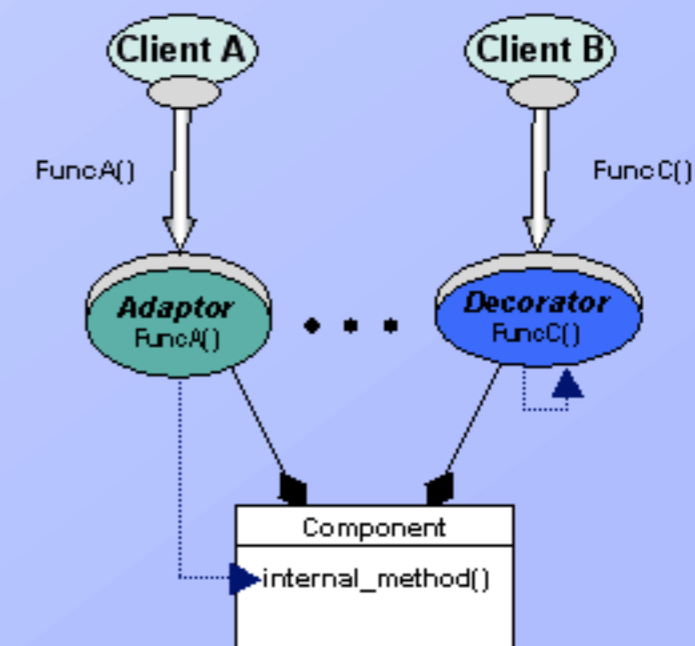
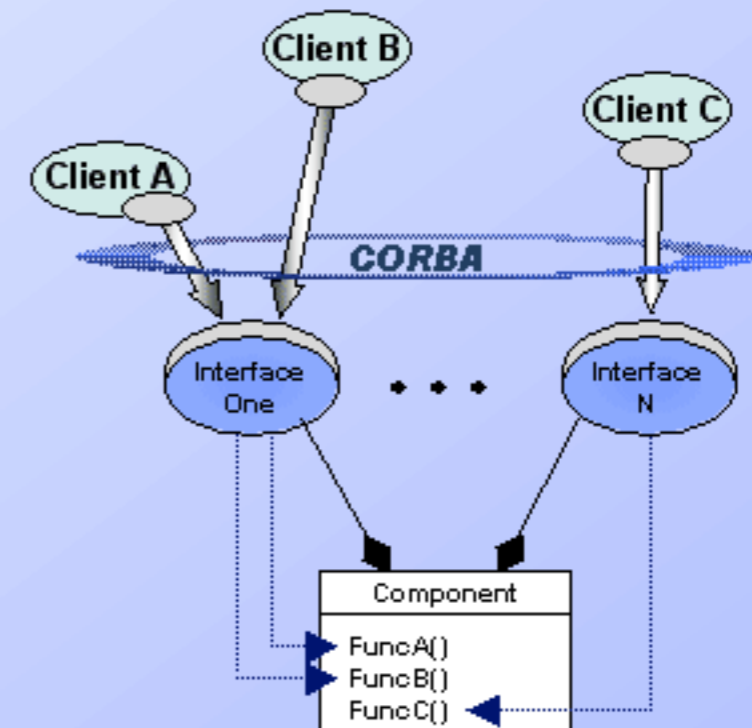


- Approach
 - Distinguish Between Software Architecture and System Architecture
 - Software Architecture
 - Utilizes Generic Concepts Such As Design Patterns To Define a Framework
 - Supports Software Development And Promotes Reuse
 - Provides a Common “Pattern Language” Between Systems
 - *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*
Douglas Schmidt et. al.
 - System Architecture
 - Extends The Software Architecture As Needed To Accommodate Unique Requirements
 - Incorporate Industry/Government Standards
 - DII/COE, JTA, TENA
 - CORBA, CCSDS SLE, SNMP, XML



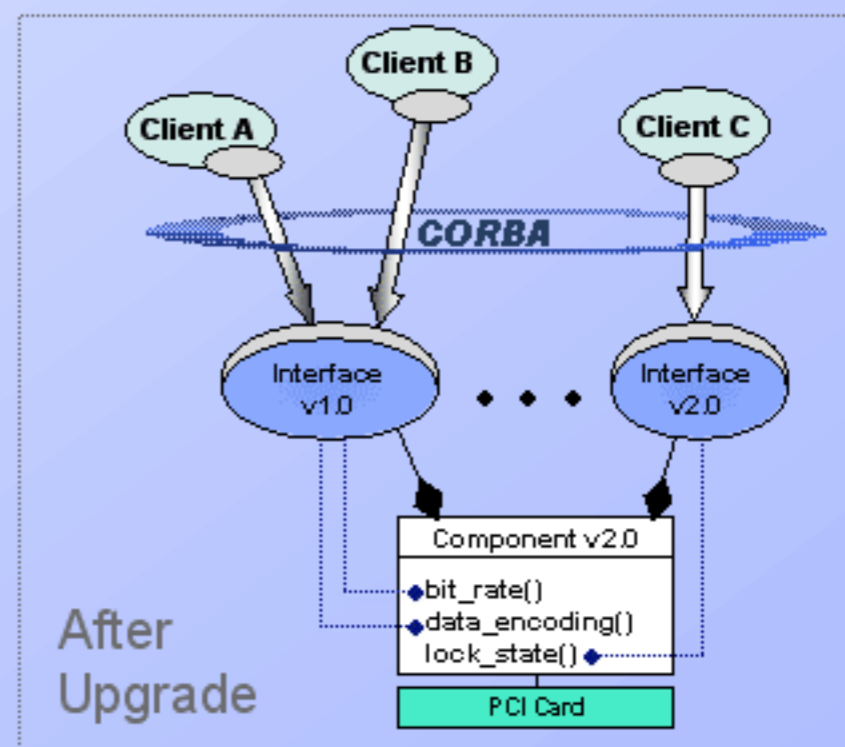
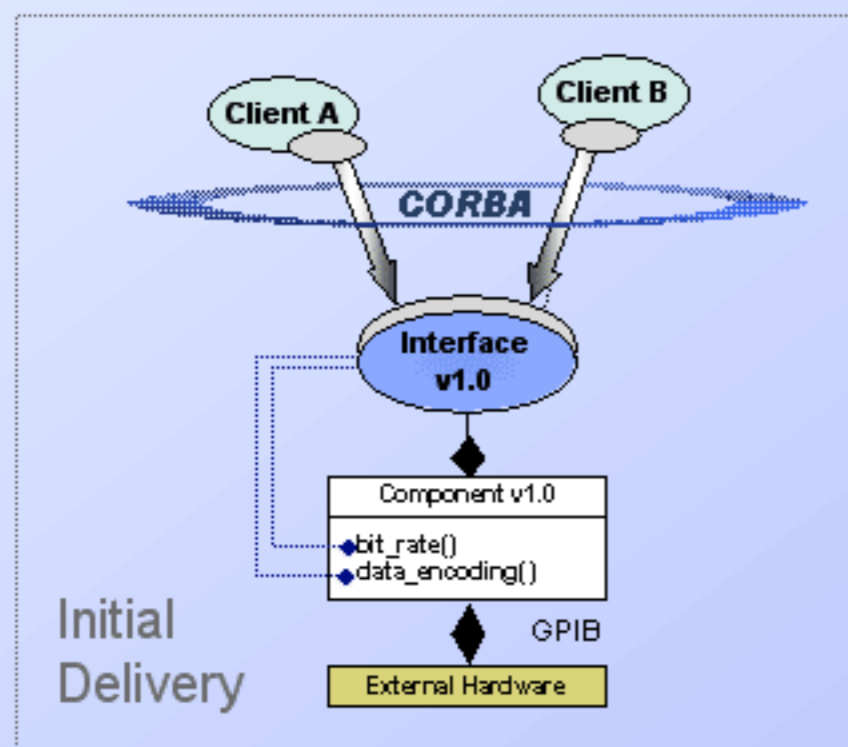


- Component Based Architecture
 - Supports Multiple Interfaces
 - API Defined Using CORBA IDL
 - Easily Maps To Both Software And Hardware Modules
 - Abstracts Client Applications From Hardware Semantics and COTS implementations
- Interface Aggregation
 - Provides Multiple “Views”
 - Adaptors & Decorators
 - Adapts Component To New Interfaces
 - Extends Functionality With Changing Component Implementation



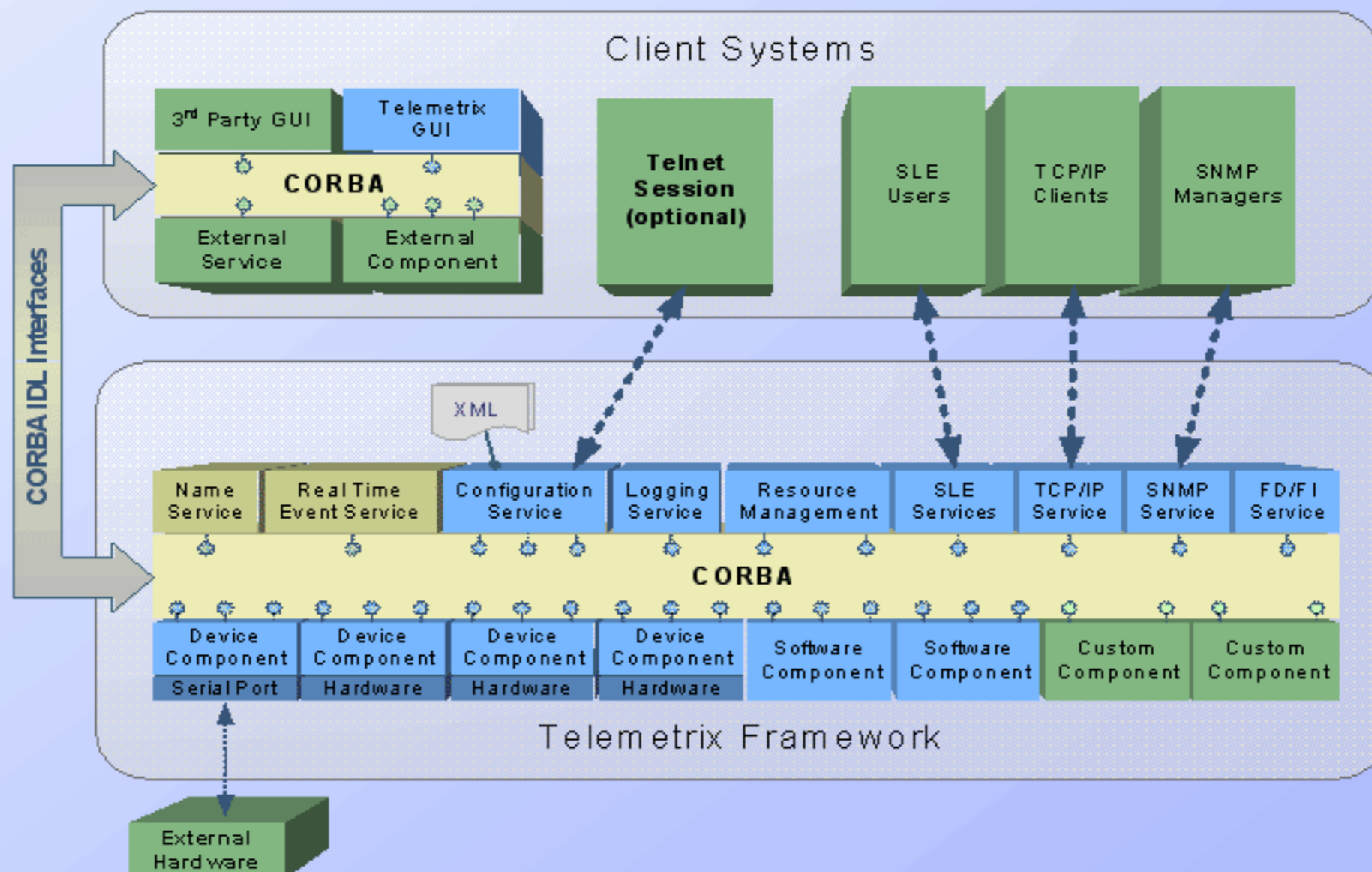


- Address System Evolution At the Architecture Level
 - Abstraction and Componentization Are Critical
 - Define A Clear and Concise Purpose For Each Component
 - Simultaneously Support Multiple Versions Of A Given Interface





- The Framework Provides Full System Functionality
 - Supports Multiple Standards-based Interface Options
 - Easily Adapts To Meet New Requirements



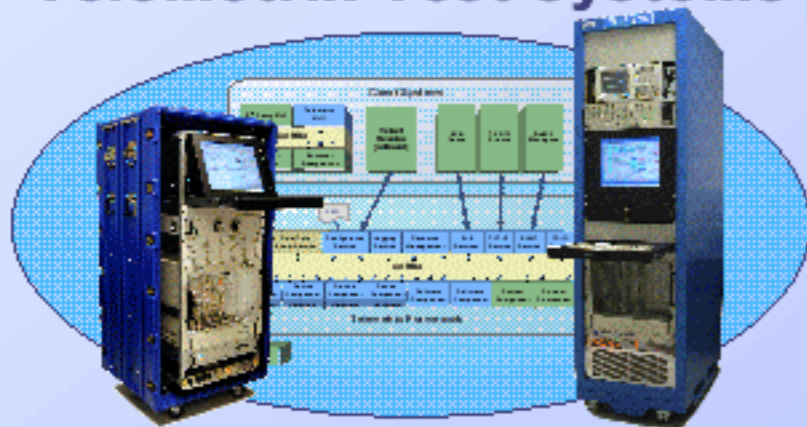


The Architecture Is The Basis For System Implementation

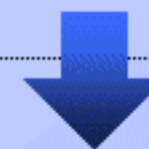
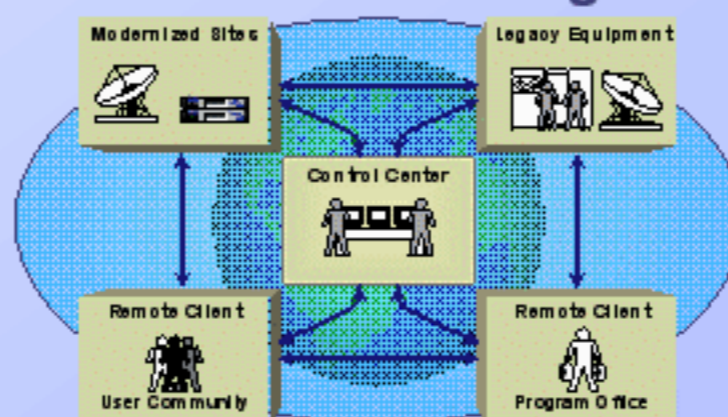
- Common Software Baseline
 - Blueprint for all Telemetry Systems
 - Simplifies Integration
 - Supports System Evolution
 - Managed Release Process
- Reusable Building Blocks
 - Standard Services
 - Well Defined Interfaces
 - Multiple Interface Options



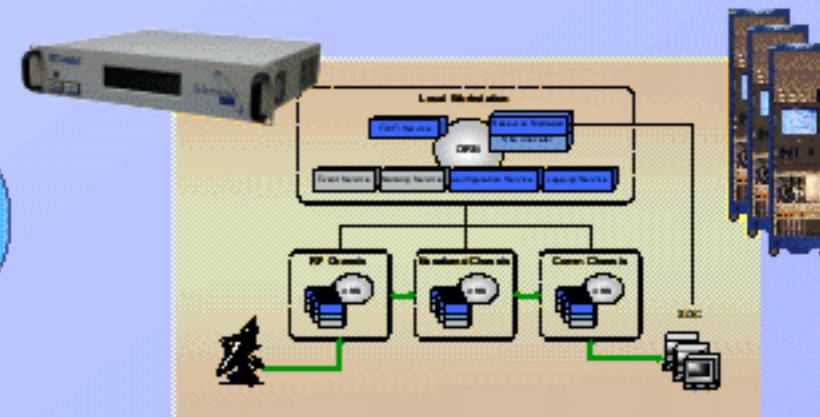
Telemetry Test Systems



Telemetry Range

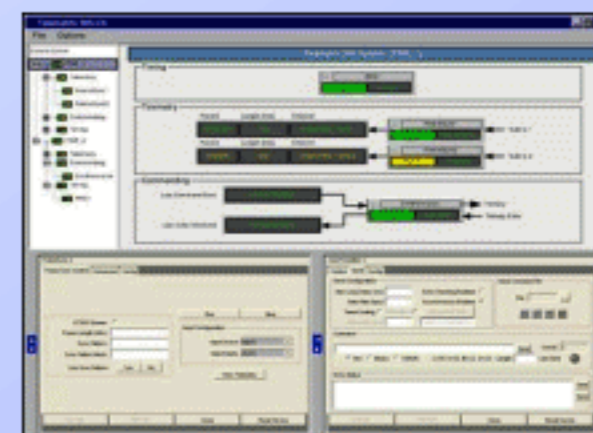


Telemetry Control Systems





- **This Approach Works**
 - 100+ Systems Delivered Per Year
 - Accomplished with ~40 Engineers
- **Notable Projects**
 - **Satellite Control Systems**
 - SBIRS-Low
 - CCS-C
 - Wideband Gapfiller
 - NAVSOC
 - RSC
 - SCNC (In Progress)
 - PanamSat (In Progress)
 - CSEG (In Progress)
 - **Satellite Test**
 - Ball RF Test Consoles
 - SBIRS EGSE
 - **Space Lift Range**
 - Post-Detect Telemetry Subsystem
 - SLRSC
 - *Primary Software Architecture*





- Define A Component Based Architecture
 - Simplifies System Evolution
 - Encourages Reuse
 - Supports Forward/Backward Compatibility
- Important To Define a Software Framework
 - Permits Rapid Development Of Complex Systems
 - Services/Components May Be Interchanged To Meet System Requirements
- Built In Interoperability Across Multiple Programs
 - Telemetrix Has Been Deployed In A Wide Range Of Systems
 - Standards Based Interfaces
- Breaks The Traditional Proprietary COTS System Approach
 - Open Architecture, CORBA IDL Interfaces
 - Offers Customers The Ability To Replace or Extend Functionality Within The Framework