AUTOMATIC RANGE EQUIPMENT SETUP AND CONTROL

Rob Andzik
Charles (Chuck) N. Brans
RT Logic Inc.
8591 Prairie Trail Drive, Suite 500
Englewood, CO 80112

ABSTRACT

Today Ranges are faced with the typical dilemma of doing more with less—less money, less time, and less experienced range personnel. Meanwhile, Ranges are being forced to make their operations more efficient in use of time, money, and functionality. As a result, Ranges are looking for automated ways to remotely configure and operate their tracking stations and to provide interoperability between ranges, sites, and equipment.

RT Logic worked with numerous range operators and equipment vendors to create an open software architecture that provides rapid device configuration, equipment status at a glance, and automatic fault detection and isolation. RT Logic’s architecture utilizes the CORBA specification to achieve extensibility and scalability for future range requirements. Adoption of this architecture and approach will reduce costs, time, and mistakes.

KEY WORDS
Telemetry, Remote Control, CORBA, TENA, Automation

INTRODUCTION

Range operators are faced with doing more with less; less money, less experienced personnel, and less time. In addition, the missions are getting more complex by requiring faster data rates, multiple targets and simultaneous missions. One way to respond to these demands is to use software that remotely configures and operates entire telemetry sites using common interfaces. A critical aspect of this is selecting a software architecture that supports current needs while allowing future scalability, flexibility, and enhancements. RT Logic has created a system based on the Common Object Request Broker Architecture (CORBA) specification that allows range operators to easily configure, monitor, and control their range equipment. In addition, by incorporating the CORBA specification, gateways to Test and Training Enabling Architecture (TENA) middleware and applications are easily achieved.
OVERVIEW OF APPROACH

When selecting a software architecture, an organization faces several challenges. In addition to technical challenges, it is important to consider the architecture’s ability to accommodate project realities. System requirements can and do change over a project’s life cycle. This is particularly true in the area of high-tech range equipment. When change occurs, it often has a dramatic impact on internal and external interfaces to the system. Project success is then dependent on how well the chosen architecture handles these types of changes.

It is therefore important for the architecture to address the following fundamental concepts:

- **Support system evolution**
  - Standardize interfaces
  - Enable adaptation of legacy interfaces
  - Isolate client applications from changes within the server

- **Design interfaces to be platform and language independent**
  - Use abstract interface definitions
  - Utilize industry standards whenever possible

- **Focus on reliability, scalability, reusability and quality of service**
  - Centralize common functionality into well tested modules
  - Leverage existing, heavily used & tested communication mechanisms
  - Use UML & object oriented techniques from the ground up

The approach RT Logic used was to first define a component-based framework for system development that utilizes design patterns, promotes reuse and provides a common “Pattern Language” (Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects -- Douglas Schmidt et. al.) between systems. The approach incorporates industry and Government standards such as DII/COE, JTA, TENA, CCSDS SLE, SNMP, and XML.

The resulting Telemetrix software architecture utilizes the COBRA specification to create a distributed, component-based framework for configuration, status, and control of a wide variety of range oriented devices. To date, the Telemetrix architecture has been used to setup and control over 1800 systems from satellite control stations, satellite test systems, and Range Telemetry processing stations. Figure 1 depicts the Telemetrix framework.
Telemetrix provides a common software baseline that becomes the blueprint for system level development. Using a common software framework simplifies integration and supports system evolution. Over time, a toolbox of reusable building blocks, standard services and device components has evolved.

For range applications and devices, RT Logic leveraged the Telemetrix framework and created a software application called Automated Range Control Suite (ARCS). In addition to the basic services provided within the Telemetrix architecture, ARCS offers system-level fault detection and isolation, as well as automated signal generation and simulation.

The operational concept behind ARCS focuses on the configuration of large suites of range equipment (e.g., receivers, bit synchronizers, telemetry decommutators, multiplexers, and demultiplexers). Historically, range configuration activities have been a significant manual undertaking requiring complex processes and configuration management. Using ARCS, this process occurs automatically and significantly reduces the labor and time required.

Figure 2 depicts a typical ARCS ground station.
ARCS utilizes a “signal-centric” approach to setup a suite of equipment for a specific operation. To define an operation, the operator describes the characteristic and attributes of the signal such as the modulation type, data encoding, bit rates, and TDM sync pattern in the Configuration Editor. During this process, the Configuration Editor validates the user inputs and offers the user suggestions for parameter selection, such as signal inputs and outputs, based on the physical system configuration.

The defined operation is stored in an XML file that can be later recalled for use or copied for a similar operation. The same XML file can be used by other ARCS enabled range telemetry stations that support the same vehicle or mission. These telemetry stations do not need to deploy the same hardware suite or configuration. The ARCS system accounts for these differences automatically and configures the local hardware appropriately. The ARCS software maintains configuration control over the operation file throughout the range and verifies proper system configuration each time the operation is loaded.

Figure 3 shows the ARCS Configuration Editor user interface.
Local or remote operators select defined operations from a drop-down list. The ARCS system automatically verifies the devices required to perform the operation are available and allocates, or reserves, them for the operation. After the allocation process completes successfully, the system configures the allocated suite of equipment to process the defined signal. In addition, a simulated signal (or signals) is created to verify that all of the equipment is processing correctly. If a device is not operating correctly, the device and string is color coded with the appropriate warning color (red or yellow) showing the operator that the string or device has a problem. This entire process takes approximately 15 seconds. In contrast, a typical manual configuration using
current range systems can take hours or even days of manual configuration to setup and test a suite of equipment for a single operation.

The ARCS system includes an easy-to-use graphical user interface. Figure 4 represents an operation containing three allocated and configured strings of equipment. The upper left quadrant shows the status of all of the equipment strings. The lower left quadrant shows the specific devices allocated for each string. In this example, the system is showing problems with the FM discriminator in the first string and with the frame synchronizer in the second string. The lower right quadrants contain specific device panels that show the detailed configuration.

![Figure 4 - Example Operator Console](image)

If changes are required to the system after the devices have been allocated and configured by the ARCS software, the operator can go to the associated device panel and make whatever changes are necessary. Figure 5 shows the soft panel for a bit synchronizer. Commonality between device panels reduces training and confusion resulting from multiple vendor interfaces.
In addition to the operational view, ARCS provides the Maintenance Editor window. This window provides a system-level view of all equipment in the telemetry station. It indicates what devices are allocated to specific operations and allows the user to manually tweak the configuration if necessary. Figure 6 provides a view of this window.
The Maintenance Editor interface also allows the user to take equipment offline, thus preventing its use when a new operation is selected.

Two other important features required by range operators are fault isolation and logging of system status and changes. ARCS handles this using a “butterfly” self-test pattern that drives a simulated signal through all of the possible combinations of equipment down to the LRU level. Any problem that is found is logged so that the operator can take the appropriate action to fix the device or mark the device offline.
CONCLUSION

Local and remote automated setup, monitoring, and changing of range equipment is available today, and can save Range operators time and money while increasing capabilities. The important aspect is having a flexible architecture based on extensible, platform-independent middleware such as CORBA. RT Logic’s ARCS is an example of such an application that is currently being used to control numerous range devices. ARCS provides operators the ability to quickly load configurations and verify the status of their operation at a glance.

ACRONYMS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>ARCS</th>
<th>Automated Range Control Suite</th>
</tr>
</thead>
<tbody>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>EJB</td>
<td>Enterprise Java Beans</td>
</tr>
<tr>
<td>TENA</td>
<td>Test and Training Enabling Architecture</td>
</tr>
</tbody>
</table>