Command and Control using World Wide Web Technology

Patricia J. Dousette(POC)
Litton Data Systems
29851 Agoura Rd.
Agoura Hills, CA 91376-6008
818 597-5388
pdousett@vines.littondsd.com

Ari Danesh
Litton Data Systems
29851 Agoura Rd.
Agoura Hills, CA 91376-6008

Matthew Jones
Litton Data Systems
29851 Agoura Rd.
Agoura Hills, CA 91376-6008

1. ABSTRACT

This experience paper describes the approach used in preparing Command and Control (C2) data for display by a Web browser. The C2 data to be displayed is resident in an ODBC compliant database and is accessed by a server program using commercial middleware. The C2 objects are then sent to the client program and displayed via a Java applet/Web Browser.

1.1 Keywords

Command and Control, WWW, Java, Architecture

2. INTRODUCTION

The purpose of the Advanced Command, Control, Communications, Computers and Intelligence (AC4I) project at Litton Data Systems is to make air defense domain Command and Control (C2) data accessible across the military internet (e.g. SIPRNET). Since most of the appropriate C2 information resides in databases developed with COTS products such as Oracle, the approach to this task became to first access the data from the database, package it into a set of domain specific objects, and then display the objects.

The objective for this Advanced C2 system was to develop a modern 3-tier client/server Windows NT based architecture that would retrieve radar track and plot data from a commercial data base and display a mini-Air Situation Display via a Web Browser e.g. Netscape or Internet Explorer. The 3-tier client/server architecture for the system known as C2NT is shown in Figure 1.

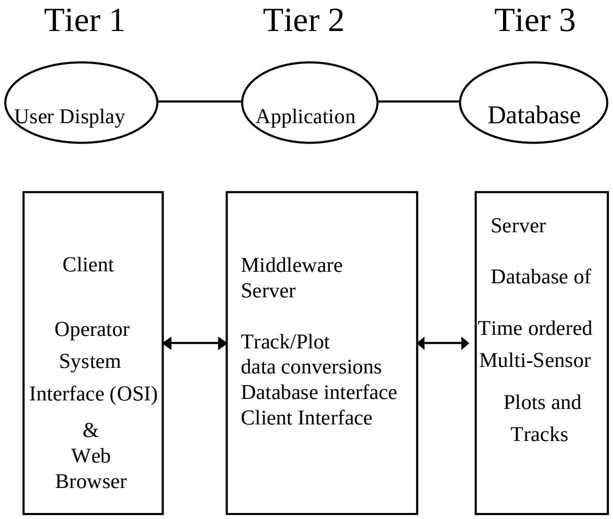

Figure 1 3-Tier Architecture

3. APPROACH

In keeping with using Windows NT as the host and target platform, it was also decided to develop the C2NT architecture using state-of-the-art commercial development tools and component technology.

The C2NT software architecture developed is shown in Figure 2.

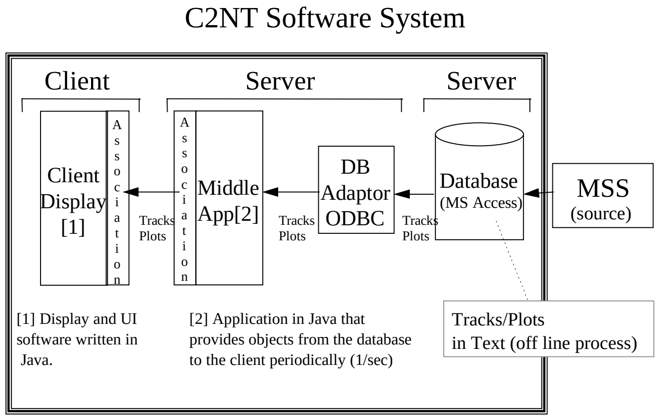
C2NT Software System
Client
Server
Server
Client Display [1]
Association
Association
Middle App[2]
DB Adaptor ODBC
Database (MS Access)
MSS (source)
Tracks Plots
Tracks Plots
Tracks Plots
Tracks/Plots in Text (off line process)
[1] Display and UI software written in Java.
[2] Application in Java that provides objects from the database to the client periodically (1/sec)

Figure 2 C2NT Software System

The server tier consists of the Microsoft Access database containing the scenario simulation track and plot data generated by the Multi-Sensor Simulator (MSS). MSS produces simulated tracks/plots, based on a scenario of radar sensors and air targets input by the operator, that are imported into a Microsoft Access Database (ODBC compliant). The middle server consists of a software layer that controls the database access and an application that processes the data for display. The middleware product selected to provide the database connectivity and the associated web based objects is WebObjects. The middle tier of the architecture shown in Figure 1 is a WebObjects (WO) application in Java that provides input/output methods to/from the database fields, and packages the information into objects to be sent to the client display process. The WebObjects application consists of code generated by WebObjects and of custom Java classes developed to meet specific application requirements. WebObjects generates class access, constructor, and environment methods. The custom classes fit the database objects to a particular client program. The client program consists of a Java applet embedded in an HTML page accessible from a web browser. The client receives track and plot data periodically via an HTTP link and refreshes the C2 geographic surveillance display with the new data. It also provides menus and controls for manipulating the display, such as scale and zoom controls.

The sequencing of operations provided by the WO Application is shown in Figure 3.

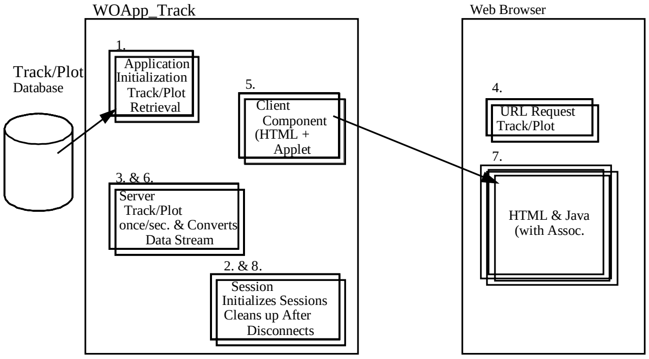

Figure 3 Sequence of Operations

4. TOOLS AND METHODOLOGY

Software development tools and products used to develop and checkout the C2NT prototype are as follows:

- WebObjects - Project Builder - builds (compiles, makes, links) a WO Application with custom functionality and database connection.
- WebObjects Builder - generates Web pages and corresponding code for a WO Application based on user selection of Web page static(text box) and dynamic (buttons, lists etc.) objects.
- VisualCafe - Java code development toolkit, also generates Web pages from a menu of predefined icons (e.g. images, buttons etc.) It supports both Java applets and applications.
- MS Peer Web Server - provides Web server services.
- MS Access - ODBC compliant database
- Formatting Utility - reformats the output of MSS to be MS Access importable.
- MSS - Multi-Sensor Simulator

5. CONCLUSIONS

5.1 Experiences

- Middleware - Scoping middleware capabilities took more time than expected, even with vendor provided training. Some advertised capabilities were not there. Vendor support was reasonable.
- Code generation capabilities were good, both in Web Objects and Visual Cafe
- Java/middleware performance appears acceptable - HTTP bus track updates are close to real time, but subject to Network loading - performance will continue to be tracked.
- Connection to the database using a prefiltered query provided by Web Objects was much easier than the connection to the client.

5.2 Basic Metrics

- Server Statistics

  Total source lines of Java - 271

  Auto generated source (Web Objects) - 137
- Client Side Statistics

  Total source lines of Java - 2172

  Auto generated source (Visual Cafe) - 1130
- Middleware connection - 45

5.3 Lessons Learned

- Middleware has a steep learning curve without respect for the complexity of the native application ==> don’t tackle too many new technologies at once e.g. Windows NT, Java, Middleware, Databases (we had to give up CORBA and Oracle for the first iteration of the prototype).
- We only used a fraction of the capabilities provided by the tools ==> if time to market is important then a consultant is probably a good idea!
- It is easy to get sidetracked by what the tool is capable of ==> don’t go into “play” mode if you have a specific objective to meet.
- Source lines of code is not the best metric for estimating and reporting productivity ==> this type of development needs some new/different metrics.
- Scalability ==> this is a small project (3 developers), there may be issues of scale e.g. training for larger efforts.

The best part of the development:

This is a flexible, easily extensible architecture. In spite of the COTS learning curve it came together fairly quickly. We expect to see a significant ROI as we go forward with enhancements and increased functionality. We also believe that there is significant development advantage in matching languages, tools, methodologies, and COTS components to the application to be developed.

6. FUTURE

The next part of the project will be to port some of our fully functional Air Defense segments written in Ada 95 to the Windows NT platform and then integrate them into the middle tier of the architecture. Adapting the server to communicate with different databases (Oracle, Sybase, custom container classes/files) will also be added. This capability is already part of the WebObjects middleware package.