GIS In the U.S. Army's Advanced Battle Command System - MCS

Dr. E. V. Shrum,
The MITRE Corp., Eatontown, NJ
eshrum@mitre.org

June 23, 2006

1. Introduction

This paper describes performance measurements by the U.S. Army's Maneuver Control System (MCS) development team on their use of the C/JMTK mapping software, and their plans for providing GIS Services in a Service Oriented Architecture (SOA) environment. Section 2 describes the background for this work; Section 3 provides the testing methodology; Section 4 gives the test results, and Section 5 gives an analysis of those results. Section 6 summarizes the results, and Section 7 describes future plans for more advanced GIS services.

2. Background

MCS is the primary Command and Control system used by the Army for Planning, Preparation and Execution of battle command operations from Corps to Battalion. It works with other Battlefield Functional Area (BFA) systems to provide comprehensive command and control functions in the Advanced Battle Command System (ABCS) "system of systems". Today it is deployed in active combat with troops in Iraq and Afghanistan.

MCS is built for the Microsoft Windows operating system and currently runs on laptop computers for both clients and servers. It was developed in .NET, C#, and Java. Before 2006, MCS used their own custom software for mapping and rendering symbols that was built with Microsoft tools. In particular, they used the Windows Graphical Device Interface (GDI) standard for representing and manipulating graphical objects. MCS Version 6.4 was built with Windows .NET, Visual Studio and C#; the corresponding graphics tools are referred to as GDI+.

A decision was made for MCS Version 6.4 to use NGA's Commercial Joint Mapping ToolKit (C/JMTK) software which the Government intends to become the future graphics standard for all their developed products. C/JMTK consists of the commercially-available ArcGIS software from the Environmental Systems Research Institute (ESRI) combined with Northrop Grumman's TASC-developed

components to meet further graphical and imagery needs. Many of the other ABCS BFA systems either already use C/JMTK or are planning to move to it[1].

MCS 6.4 was designed to use the following ESRI ArcGIS tools for manipulating graphical entities:

- ArcEditor
- ArcMAP
- ArcToolbox
- ArcCatalog
- Geodatabase
- ArcObjects

Actually, MCS did not build their application on the framework of ArcEditor and the other ArcGIS tools; if they had, MCS would have the ESRI graphical interface look and feel. Rather, they accessed the ArcGIS functionality through the ArcObjects programming interface, and MCS continues to have the same graphical interface its users are familiar with.

MCS 6.4 was originally planned to use the C/JMTK Military Overlay Editor (MOLE) which is part of the NGA's Military Analyst to render graphical symbols. However, the MOLE proved to have performance shortcomings, and the MCS developers decided in early releases to replace it with their own software built with the Microsoft GDI+ tools. They continued to use the ArcGIS tools for map loading and manipulating. Much of the motivation for this testing was to evaluate the possibility of moving to MOLE.

3. GIS Testing Methodology

3.1. Test Goal

The goal of the work described in this memo was to quantitatively measure the performance of ArcGIS with MOLE, and to compare those results with the same measurements on the MCS symbol rendering software using GDI+.

---

[1] **COPF** – custom mapping, considering C/JMTK
JTCW – currently using Atlas software written by NG, switching to C/JMTK
ASAS Light – ESRI framework
AFATDS (Unix) – JMTK
AMDWS – JMTK + GSD symbol software
FBCD2 – TMTK (NG) + GSD symbol software
GCCS-A – C/JMTK
C2PC – Atlas by NG, switching to CJMTK
MCS – older prototype used their own software named Magellan, now using CJMTK.
DTSS – C/JMTK + MOLE
MCS Heavy - JMTK
FalconView/AMPS – custom, use GSD symbol software, considering C/JMTK

3.2. Simple Shapes Tests

A first series of tests drew simple shapes: points, lines, areas, and ellipses; also used were simple text, and text with background color. (Note that a circle is just an ellipse with eccentricity of zero.) Fifty thousand, one hundred thousand, and two hundred thousand of each of these shapes were drawn and measured, both with the ESRI ArcGIS tools and with the GDI+ based tools.

All complex symbols, such as Boundaries, Areas, Control Points, and Units are built up from these primitive shapes, and so the results from these tests give insight into the source of any experienced performance problems.

3.3. Unit Symbols Tests

In a second series of tests, more complex shapes were drawn. Symbols of a military unit were drawn with varying numbers of text labels (e.g., no label, Echelon label, Time label, Parent Unit label). Tests for one thousand, three thousand, five thousand, and ten thousand unit symbols were written, each with zero to three labels. In all cases, the symbols were drawn from memory and not from any database.

These symbols are more typical of what will be encountered in a real military application, and so the results from these tests give insight into system performance such as how long it will take to draw a battlefield Common Operational Picture.

3.4. MOLE Symbol Conformance with Standards Tests

Next, testing was done at a higher, systems level in an environment more equivalent to a field situation. A subset of symbols were drawn consisting of 93 commonly used symbols (selected several years ago as being the most important symbols used in military ground operations and adding air control symbols), plus 36 Military Operations Other than War (MOOTW) symbols. The results were compared with the "*Uniform Standard for ABCS Symbology (USAS,)*" which is based on MIL-STD 2525-B.

3.5. ArcSDE Network Bandwidth and Delay Tests

The goal of this category of testing was to quantitatively measure the performance of ArcGIS with symbols stored in a remote ArcSDE Server, as a function of network delays and reduced bandwidths similar to those that can be expected in a typical tactical satellite network. For this testing, rather than the previously used hand-generated symbols in a development environment, we desired a more realistic system configuration, and so we simulated a satellite based network with

NIST Net, a delay and bandwidth simulator written by the National Institute of Standards and Technology that runs under Linux[2]

3.6. Test Procedures

Overlays were generated manually containing the desired number of shapes or unit symbols; input variables that specified how many items to draw were changed for each test.

For each test, measured was the *Prep Time*, the time in seconds required to render (create and store) the symbols into memory; *Drawing Time*, the time in seconds to draw the symbols from memory to the screen, and *Memory Usage*, the computer memory in megabytes used in that process. Drawing Time was measured manually, and Prep Time was measured by software instrumentation. Derived from these measurements was the *Total Drawing Time*, the sum of Prep Time and Drawing Time. Computer memory usage was collected by the Windows Task Manager tool.

3.7. Test Environment

All the Simple Shape and Unit Symbol tests were executed with ArcGIS 9.0, Service Pack 2, with MOLE 9.0, and then with MCS Renderer 2.0 software. Later, a subset of those tests and the MOLE Conformance and Satellite tests were executed with ArcGIS 9.1; all the tests are planned to be rerun onArcGIS 9.2 by 3Q06.

The Simple Shapes tests were executed on a Dell 2.66 GHz desktop machine with 512 MB of RAM under Windows XP with Service Pack 1.

The Unit Symbols tests were executed on a Dell 2.0 GHz desktop machine with 512 MB of RAM under Windows 2000.

3.8. Pass/Fail Criteria

A test was considered to pass if the desired number of shapes or symbols were correctly drawn on the screen and the system was still operational. If the system was not operational and had to be restarted, or did not display the correct number of symbols properly, the test was considered to have failed.

---

[2] The *NIST Net* network emulator is a general-purpose tool for emulating performance dynamics in IP networks. The tool is designed to allow controlled, reproducible experiments with network performance sensitive/adaptive applications and control protocols in a simple laboratory setting. By operating at the IP level, NIST Net can emulate end-to-end performance characteristics imposed by various wide area network situations (e.g., congestion loss and transmission delays due to satellite nodes).

4. Test Results

4.1. Simple Shapes Tests

In summary, Figure 1 compares the Total Draw Times results for GDI+ and ESRI for lines, ellipses, areas, plain text, and test with background color. Points could not be drawn with GDI+ because that software did not offer an interface for drawing points; otherwise, all tests passed for 50,000, 100,000 and 200,000 shapes.

Figure 1: Total Draw Times for Simple Shapes Tests

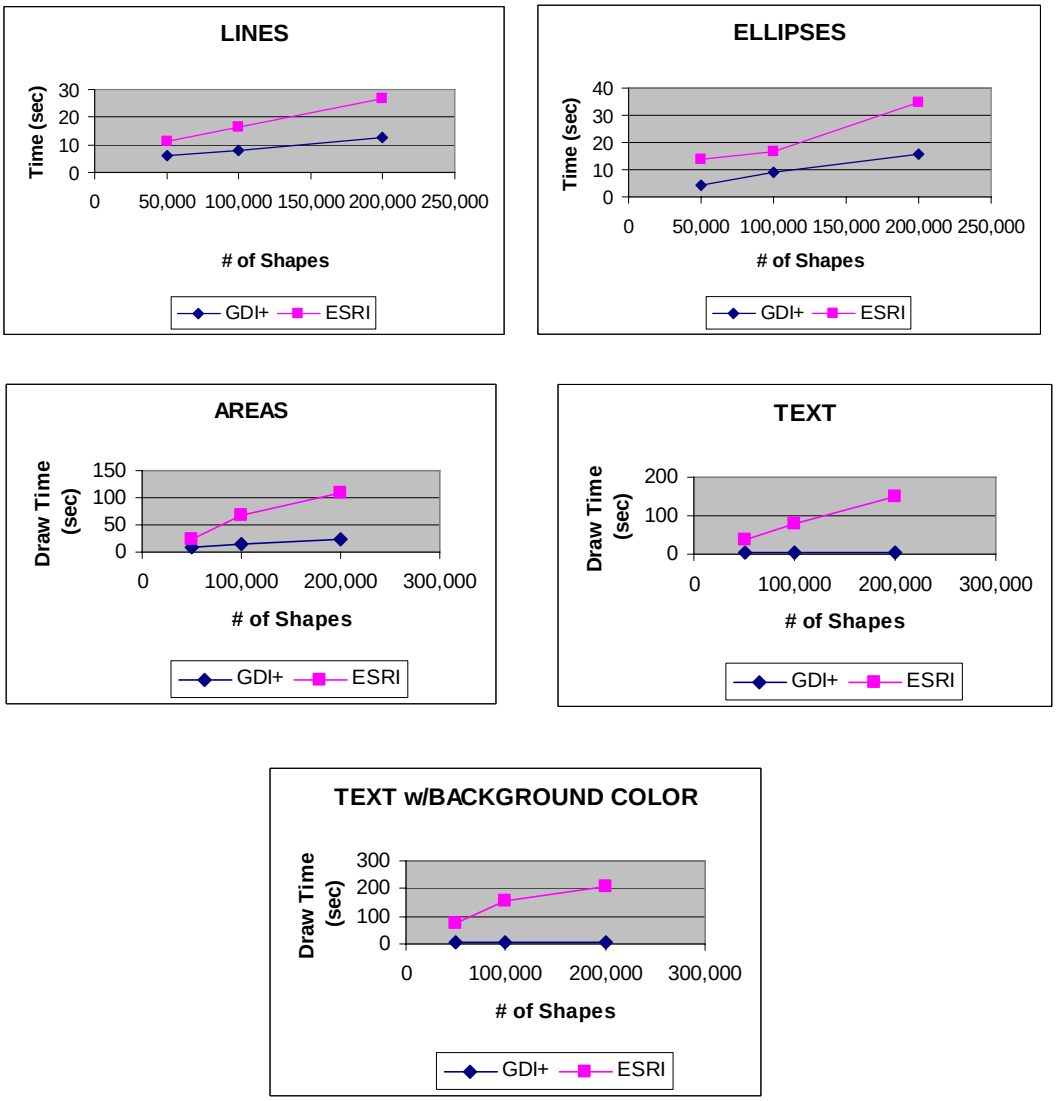
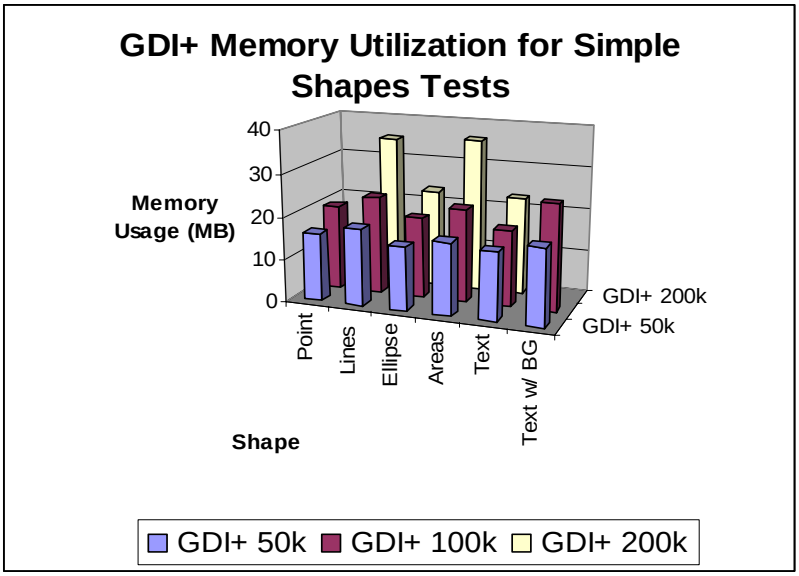
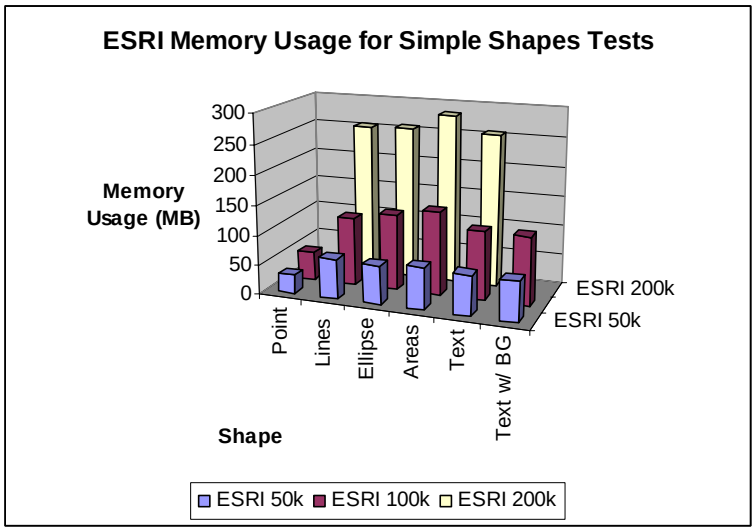


Figure 2 summarizes the Memory Utilization for the Simple Shapes Tests, plotting the test data for 50,000, 100,000 and 200,000 of each of the shapes for GDI+ and ESRI.

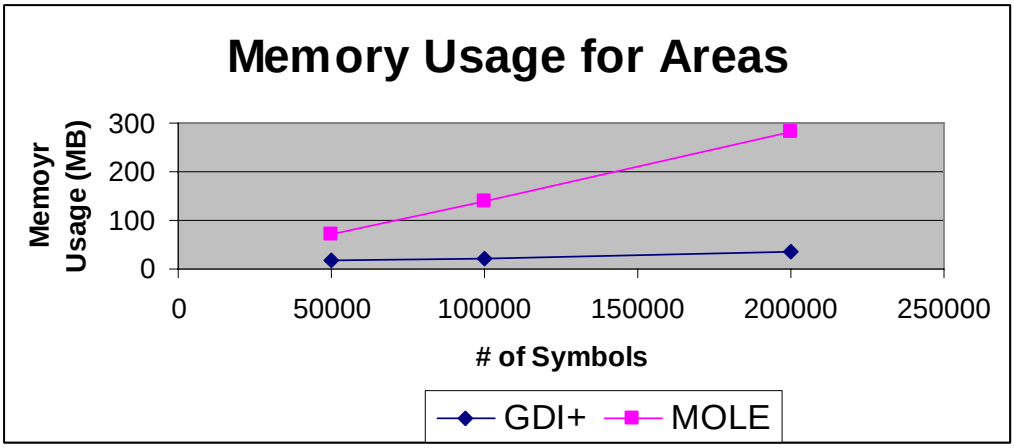
Figure 2: Memory Utilization For Simple Shapes Tests





In order to more easily see a comparison of GDI+ and ESRI, consider just the measurement results for the shape 'Area' as depicted in Figure 3.

Figure 3: GDI+ and ESRI Memory Utilization for Areas



4.2. Unit Symbols Tests

Figure 4. graphs a simplified subset of the Total Drawing Time (includes Prep Time), plotting only the times for no text labels. All GDI+ tests passed. All MOLE tests for 1,000, 3,000 and 5,000 unit symbols passed, but when drawing more than 5,000 unit symbols, the software under test became unstable and the computer sometimes had to be rebooted. Thus the MOLE test for 10,000 units failed and the data was not reported.

Figure 4: Unit Symbol Total Draw Times, No Labels

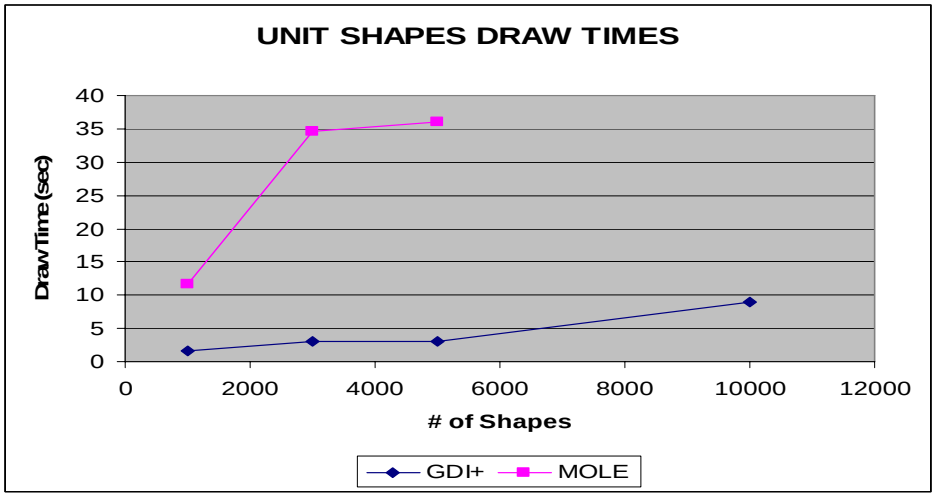


Figure 5 gives the Unit Test Total Draw times for different numbers of Text Labels.The GDI+ Draw times were nearly independent of the number of Text Labels, and so that data was combined for this graph. The tests were run for 1000, 2000, 5000 and 10,000 units; the MOLE tests for 10,000 units failed.

Figure 5: Unit Test Total Draw Times, Multiple Labels

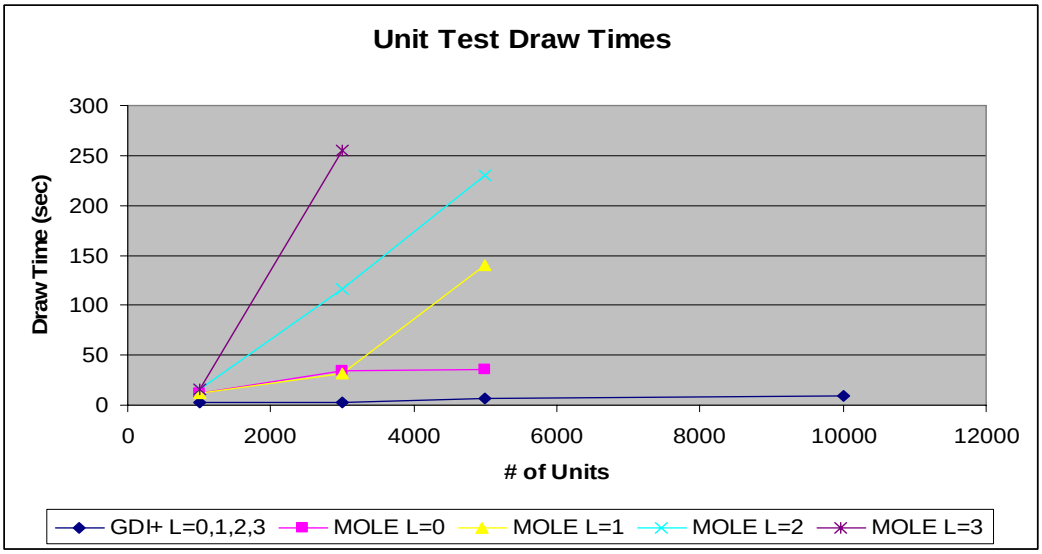
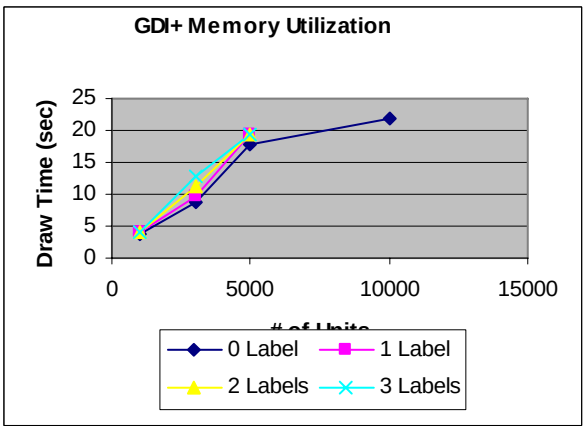
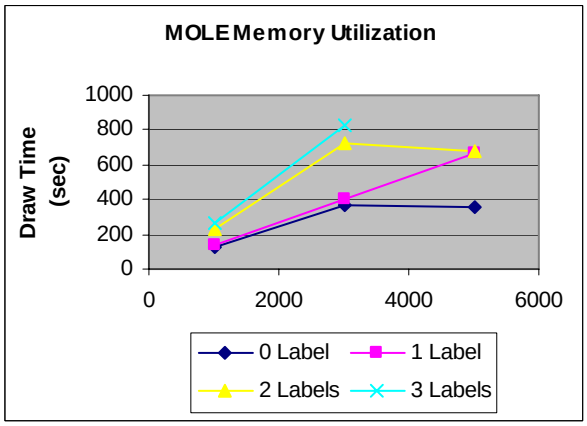


Figure 6 gives the dependency of Memory Utilization on the number of Text Labels for both GDI+ and MOLE. Tests were run for 1000, 2000, 5000 and 10,000 units; the MOLE tests for 10,000 units failed; the GDI+ tests for more than zero labels were not tested.

Figure 6: Unit Test Memory Utilization





4.3.MOLE Symbol Conformance with Standards Tests

For the MOLE Conformance Tests, 129 symbols were drawn individually and then rendered by both the MCS software and by MOLE. Data was entered for all the possible text labels, and the resulting display compared with the USAS depiction of the same symbol. A summary of the results is given in Figure 7.

Figure 7: MOLE Symbol Conformance with Standards

# of Symbols	MCS Conformance	MOLE Conformance
139	4%	8%

4.4.Network Bandwidth and Delay Test Results

To test the sensitivity of the ArcSDE configuration to network delays, the C/JMTK Draw Wizard was executed on a Brigade MCS client that was configured to connect to the ArcSDE Server on the Division Battle Command Server. Wizard setup times were measured with varying NIST Net simulated bandwidths from 400 to 60k bits per second, and with simulated delays from 0 to 1200 mSec. The 1200 mSec. test was equivalent to 2 satellite hops at the maximum delay (600 mSec.), or almost 4 hops at the median delay (350 mSec.). The results are given in Figures 8. and 9.

Figure 8: Network Bandwidth Test Results

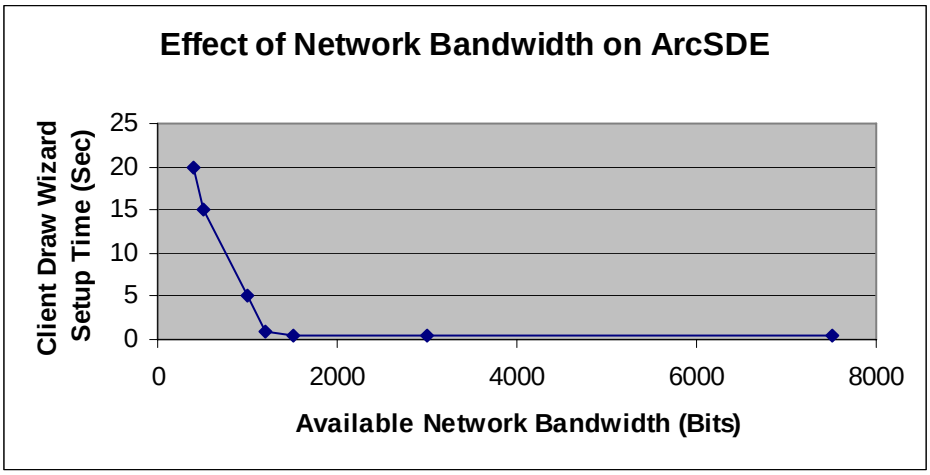
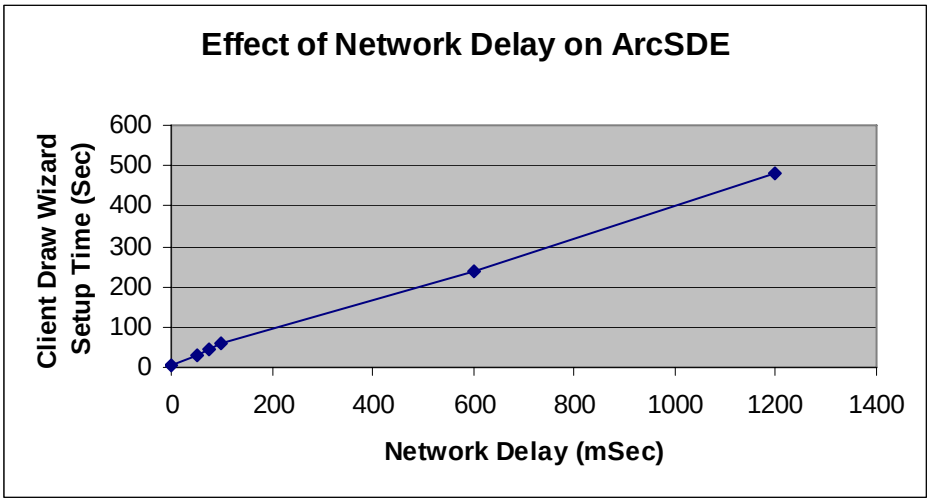


Figure 9: Network Delay Test Results



5. Analysis

Several points can be discovered upon careful study of the test data:

1. Prep Time has little impact on the overall drawing time for GDI+; for ESRI it is 30 - 50 % of the Draw Time. The ESRI Prep Time is 10-30 times longer than for GDI+.

2. Memory Utilization is 3-5 times more for ESRI. This is not enough to account for sometimes as much as 40x longer drawing times. Memory usage may be the cause for MOLE to become unstable for the largest number of unit symbols; this *must* be improved, since it is expected in tactical field situations that battlefield Common Operational Picture will contain 5,000 to 10,000 symbols.

3. Text labels are very expensive for ESRI even though the underlying primitives are not. This implies that positioning the labels is difficult. Text labels are not very expensive for GDI+.

4. For the Simple Shapes, the times to paint the screen from memory are similar for GDI+ and ESRI except for text and text plus background which are much longer for ESRI. This impacts the MOLE Unit Symbols even more; for GDI+ there are only moderate increases in Draw Time as the number of labels is increased, but for MOLE the time doubles as one goes from 0 to 3 labels.

5. Conformance to USAS standards of the MOLE symbols is acceptable. The discrepancies must be fixed, but given the complexity of the symbols and the

fact that this is the first time the MOLE symbols have been tested, this is a rather favorable result. It should be noted that even though the project had spent considerable effort on their own rendering software to ensure conformance to standards, they were still non-conformant in some cases.

6. For ArcSDE, bandwidths greater than 1200 bits/sec will work satisfactorily, but the delay associated with even a single hop satellite link will provide unacceptable performance. This is probably because there may be many database accesses for a single graphics event; the packet delays are cumulative and database timeouts can occur.

Several aspects of the data were not understood and warrant further testing:

- For Simple Shape GDI+ testing, why were Ellipses faster than Lines for 50,000 shapes, yet slower for 100,000 and 200,000 shapes?
- For Simple Shape GDI+ testing, why did Ellipses take less memory for 200,000 shapes than Lines or Areas took?
- For Simple Shape ESRI testing, why for 200,000 shapes was memory utilization lower for Text w/ Background Color than for Text only, yet higher for 50,000 and 100,000 shapes?
- For Unit Test MOLE testing, why did Memory Utilization drop for 5000 units with zero and two Text Labels? Why did it not likewise drop for one Text Label?

This test program did not measure performance of database access as opposed to memory access, which, of course, will be substantially slower and is extremely important. Re-running these tests with shapes/symbols stored in the ArcSDE or SQL database would be very valuable. Note that many of the important benefits to the military of using a GIS such as spatial and terrain analysis require the use of a database.

6. Testing Summary and Conclusions

In all cases, the ESRI and MOLE software was significantly slower in both Prep Time and in Drawing Time than the underlying GDI+ software and took much more memory. The more symbols were drawn, the bigger the differences. In general, ESRI/MOLE was about 5 to 10 times slower and took roughly 30 to 40 times more memory.

Improvements are needed by ESRI in ArcGIS for:

- Prep Times,
- Drawing Text labels,
- Drawing Areas,
- Memory usage.

Improvements are needed by TASC in MOLE for:

- Prep Times,
- Drawing Labels,
- Memory usage.

Note that improvements in ArcGIS will improve the MOLE performance, but it is expected that these improvements alone will not be enough. It can be noted however, that a moderate 20% improvement in Prep Times and a 20% improvement in Text Draw Times might give an improvement of 40% - 60% for the MOLE Unit Draw times.

The software instability for more than 5,000 units due possibly to memory usage, is a “show stopper” and must be fixed if MOLE is to be used for systems that support the Common Operational Picture with up to 10,000 and more symbols.

Since it is known that accessing symbols/shapes from a database is significantly slower than drawing them from memory as the tests did, the database access software must be studied and improved as well.

The sensitivity to Satellite delay times is unacceptable and must be fixed. Increasing the SQL Server buffer size to 64K as suggested by ESRI did improve performance somewhat, but not nearly enough. The advertised improvements in ArcSDE 9.2 are eagerly being awaited for retesting.

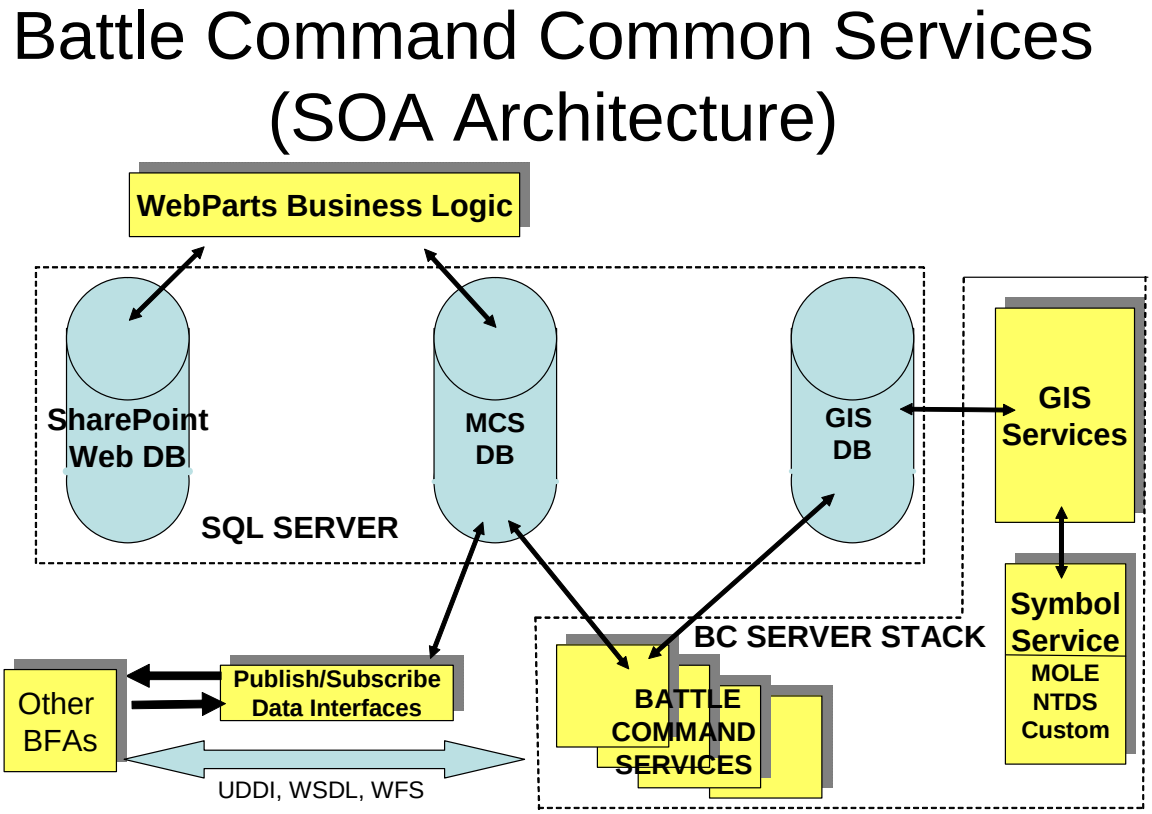
7. Future GIS Directions

Based on these test results, the decision was made to move MCS away from their own rendering software to MOLE. More importantly, the ABCS system is transitioning to a Service Oriented Architecture. They plan to provide GIS Services via transactional Web Features Services (WFS) and Analysis Services (Search, Spatial Query, Content Finder, Place Finder). ArcGIS Network, Spatial and 3D extensions will be used to provide results via ArcGIS Connector and WFS.

Later plans are to provide a User-defined Symbology Service to supply symbols to users based on pre-defined preferences, providing support for MOLE, some NTDS (Navy), and custom symbols such as SVG and others).

Figure 10 provides an overview of how the SOA architecture will utilize GIS services.

Figure 10: Battle Command Common Services



8. Acknowledgements

Much of the testing described in this paper was performed by Osie David, Mitchell Burger, Mike Walsh, John Brighton, Robert Geradi, and Simon Zebro of the MCS development team at the Software Engineering Center, Ft. Monmouth, NJ. Special thanks are particularly due to their supervisor Mr. Joao Brandao for his support of this effort and analysis of the results.