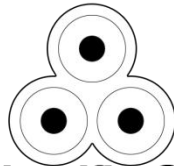


Data Management Systems
ENGR-3700U-001
Project Phase 3:
Military Logistics Management System



TriOptimum



Military Scientific Commercial

Created by:

Daniel Smullen

Rayhaan Shakeel

Khalil Fazal

Table of Contents

Project Abstract.....	3
Introduction.....	3
Project Goals.....	3
Design and Implementation	3
Relation to Similar Systems.....	5
System Overview	6
Dynamic Query Generation	6
Google Maps API SOA Integration.....	6
XML Output.....	6
Future Iterations	6
Results and Analysis	7
Lessons Learned.....	7
Usability.....	7
Appendix A – Additional System Screenshots	9
Figure 1 - Personnel Readout.....	4
Figure 2 - Corps and Divisions Filter Controls.....	4
Figure 3 - Consumables Readout.....	4
Figure 4 - ORBAT View (Map Mode) with Corps Selected	5
Figure 5 - ORBAT View (Satellite Imagery) with Brigades/Regiments Selected.....	5
Figure 6 - LMS Sample Query Overview.....	9
Figure 7 - ORBAT View with Corps Selected	9
Figure 8 - ORBAT View with Divisions Selected	10
Figure 9 - ORBAT View with Brigades Selected.....	10
Figure 10 - ORBAT View with Regiments Selected	11
Figure 11 - Command Roster Sample Output.....	11
Figure 12 - Soldier Roster Sample Output	12

Project Abstract

The Military Logistics Management System – Department of Defence Subsystem (MLMS – DOD SYS) is a system required for management and visualization of military assets in the field, designed for use by the US Army and Department of Defense. The system is designed to connect with a large database of personnel, equipment items, vehicles, and consumables. These entities are organized by military hierarchal structure which can be traversed at a glance, giving commanders a visual summary of statuses through an ASP.NET driven web interface.

Introduction

Military affairs are chaotic in nature. Statistics, occurrences, and the organization of materials and equipment are difficult to manage. Battlefield commanders require the ability to view large amounts of information in a simplistic manner, particularly as ongoing situations become more complex and more units, personnel, and equipment enter the battle space. The Army requires that battle space materiel logistics be displayed by hierarchy. The Department of Defence requires that the same information be visualized and plotted globally. Both organizations require that personnel rosters be created for both commanders and units as a whole for introspection by other government bodies.

Project Goals

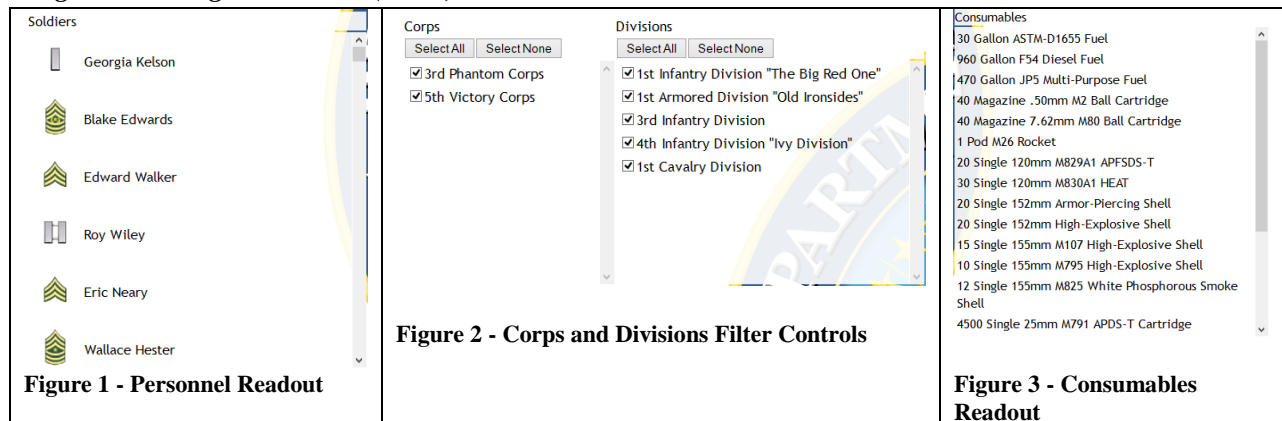
Primarily, the goal of the MLMS system is to allow for quick data access in a highly visual manner. The basis for the design was to create a system which can make a tremendous volume of information seem simple. This was accomplished by segmenting the system into several different views which would detail a specific aspect of the information provided. One view would show the breakdown of the Army hierarchy, allowing commanders to select which units to focus on and retrieve information for. Another view would graphically plot the location of these units using some kind of geo-informatics. Since the Army and Department of Defence would both need to interact with the system and have it display different outputs, XML would represent a universal layer of interaction for outputting the necessary rosters, since XML can be used anywhere.

Design and Implementation

Database operators would perform behind the scenes queries in order to input data en-masse into the database. The large database comprehensively captures the hierarchy of the Army units, and has a variegated sampling of equipment and soldiers. In the case of the demonstration product, the database has been manually implemented with this information based on a previously designed schema from Project Phase 2.

In the interest of avoiding repetition, this document will focus heavily on the design features applicable to this project phase, as opposed to the underlying database architecture which remains unchanged.

Logistics Management View (LMV) with Force Breakdown



The screenshot displays the Logistics Management View (LMV) interface. It is divided into three main sections:

- Figure 1 - Personnel Readout:** A list of soldiers with their names and rank insignias. The names listed are Georgia Kelson, Blake Edwards, Edward Walker, Roy Wiley, Eric Neary, and Wallace Hester.
- Figure 2 - Corps and Divisions Filter Controls:** Two filter panels. The 'Corps' panel has 'Select All' and 'Select None' buttons, with checkboxes for '3rd Phantom Corps' and '5th Victory Corps'. The 'Divisions' panel also has 'Select All' and 'Select None' buttons, with checkboxes for '1st Infantry Division "The Big Red One"', '1st Armored Division "Old Ironsides"', '3rd Infantry Division', '4th Infantry Division "Ivy Division"', and '1st Cavalry Division'.
- Figure 3 - Consumables Readout:** A list of consumables with their quantities and names. The items include: 30 Gallon ASTM-D1655 Fuel, 960 Gallon F54 Diesel Fuel, 1470 Gallon JP5 Multi-Purpose Fuel, 40 Magazine .50mm M2 Ball Cartridge, 40 Magazine 7.62mm M80 Ball Cartridge, 1 Pod M26 Rocket, 20 Single 120mm M829A1 APFSDS-T, 30 Single 120mm M830A1 HEAT, 20 Single 152mm Armor-Piercing Shell, 20 Single 152mm High-Explosive Shell, 15 Single 155mm M107 High-Explosive Shell, 10 Single 155mm M795 High-Explosive Shell, 12 Single 155mm M825 White Phosphorous Smoke Shell, and 4500 Single 25mm M791 APDS-T Cartridge.

The LMV incorporates dynamically generated SQL queries in order to generate information readout on the bottom half of the page, using a specially designed filter control on the top half of the page. Vehicles are displayed with an icon that corresponds to their locomotion type, personnel are displayed with rank insignia, and consumables are displayed by quantity. These values are all stored in the database in separate tables which are joined. Based on examining spanning subsets of the entire database which are retrieved by selecting the parameters in the filter control, the type of result is filtered further and separated into four distinct columns – personnel, vehicles, equipment, and consumables.

In order to filter the resultant dataset using military hierarchy, 5 columns are dynamically populated with the corps, divisions, brigades, regiments and platoons within the database. The names and designations are retrieved from 5 respectively named tables. These table also store attributes pertinent to the ORBAT subsystem, such as the designation for the unit (used to display the unit logo) and latitude and longitude.

In the LMV system, each selection of a hierarchal level displays only the units underneath that unit in the hierarchy. That is to say, selecting one corps will display only the divisions associated with it, and each division only the brigades associated with it, et cetera, until the platoon level (the smallest organizational unit). Once a platoon is selected, the other half of the readout is populated. Each change to a hierarchal level forces the levels beneath to be updated again asynchronously.

The bottom half of the page contains the actual logistical information. To display the rank insignias for each entry in the soldiers table, the rank attribute of each soldier is matched with a rank insignia graphic. For each entry in the vehicles table, the vehicle locomotion type attribute is similarly matched. After further processing and filtering, consumables are aggregated using a SUM function which determines the overall quantity of each tuple and these quantities are concatenated with the unit quantity name, and the consumable name. The equipment items are displayed as-is.

The views for each of these GUI elements are dynamically generated, and therefore do not conform to any one static view. However, the filtering process narrows this interface to showing nine separate views, combined into a single view. More information on dynamically generating the SQL queries is in the System Overview section.

Order of Battle (ORBAT) Imagery using Geo-Informatics

ORBAT is displayed by connecting the underlying database with service-oriented-architecture requests, implemented via calls to the Google Maps API. These are done using a special custom control which was

developed by an external provider and credited in the source code. This control allowed for the display of imagery data, and the project database provides the tie-in with the imagery enhancements that are required. In this view, controls are provided to manipulate which units are requested to be rendered by the remote service. The MLMS server holds instances of the graphics which are sent to the Maps API based on the tuples retrieved from the database for each unit in the hierarchy. Corps and divisions have a distinctive logo associated with them, with the filename stored in an attribute within their two respective tables. Brigades and regiments are associated similarly with separate filenames. The images associated with each are their NATO standard unit type insignia. Platoons are too small to be represented by an icon, and are encapsulated by their regiments. All units are joined with a locations table – if a unit has a deployment location attribute associated with them, they are displayed at the location of deployment. If a unit's deployment location is null, they are displayed at their headquarters location. This is found by joining these tuples with the locations table, which provides the geographic coordinates for each location. All of the underlying processing for these operations take place on the database/web server, and then within Google's internal infrastructure. None of the processing takes place on the client's side except for rendering the final output to the screen.

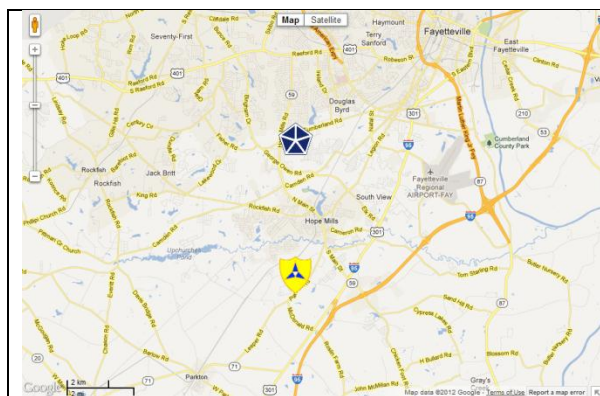


Figure 4 - ORBAT View (Map Mode) with Corps Selected

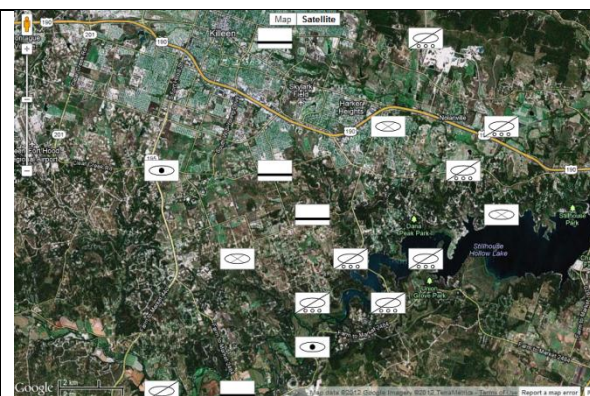


Figure 5 - ORBAT View (Satellite Imagery) with Brigades/Regiments Selected

Relation to Similar Systems

MLMS is unique in that it aggregates several disconnected processes and systems which are handled by paper based systems, or disconnected databases spread across the Department of Defence secure networks. Traditionally these processes would use pen and paper filing systems to store records on premade forms. Using a centralized database eliminates the need for traditional processes and allows for extensible application development, adding tools for new purposes as they are needed. The US Army currently uses the ARIMS system to aggregate several separate digital information repositories and provide tools for accessing them. The system is completely separate from the battle space management systems and logistics systems used by individual organizational units, however. No unified system currently exists which can track all of the entities that MLMS tracks or provides a single front end.

It is likely that the US Army and Department of Defence do not publish information concerning their internal management systems that would match the intended usage of MLMS, for security reasons. It could easily be speculated upon, however, that a system or group of systems similar in functionality to MLMS do exist.

System Overview

This section will detail some of the key design aspects of MLMS, including innovative features which were successfully implemented.

Dynamic Query Generation

The LMV component of MLMS incorporates a custom control that dynamically generates and executes SQL queries in order to filter the database contents. The concept behind the usage of these queries is that large sections of the database are designed to be joined, but doing so represents an unmanageably large set of data, with attributes that may only be applicable to the tables they originated from. By filtering this into subsets of data and displaying each subset separately, a large amount of data can be visualized simultaneously. Using static views did not allow for the streamlined filtering that was afforded by generating a query based on what was selected in the main interface, and utilizing static views for each individual interface component would disconnect the controls from the data, preventing updates to the unit hierarchy from being reflected in the data that can be selected.

The result of the dynamic query generation is that the query string which retrieves data from all the required locations can be appended to or shortened based on what the desired units are. Each selected item that participates in the query widens or narrows the selection by expanding the number tokens to be matched against. These are directly appended to the end of the query using string manipulation, and the final string is sent as a query.

Google Maps API SOA Integration

The ORBAT component of MLMS utilizes service oriented architecture, by offloading the processing for map imagery to the Google Maps service through their API. When a user selects a unit for display on the map, this request is forwarded to the web/database server which acts as the service provider. In turn, another request is made between the server and Google Maps to retrieve the imagery. This is processed by the web server and the connection between the client's browser and the map imagery is brokered. This allows the client to consume the imagery data stream while interactively adjusting the parameters of the display.

XML Output

In order to meet the design requirement of having the system be able to have interoperability between US Army and Department of Defence systems concerning accounting and statistics of personnel, the XML output feature of the software utilizes a predefined schema for outputting two different types of unit rosters in a universal format. The first roster outputs all personnel who are commanders of a specific unit, with data including the command responsibility information, rank, and name of the commander. The second roster outputs all personnel who are in the database, with data including their unique identification number, rank, and name. These views are both attached to a style sheet for easy viewing on the web interface, and separate the structure of the XML from the style of the presentation.

Future Iterations

Newer versions of MLMS should undergo extensive performance improvements, as some elements of the system may easily get bogged down by numerous service requests. This is particularly evident with the LMS system, as the dynamic SQL query generation is very taxing on the web server, although the database appears more than capable of handling the volume of queries. This is likely due to the looping

behavior of the query analysis algorithm which loops through every tuple from the resultant dataset as the results of the queries are tabulated.

Another feature which would be useful is the ability to modify the database data through the web interface directly. This would require the creation of a special view inside the system which requires extra authentication credentials to access. In this view, individual tables within the database could be selected and their contents displayed. Users would then be able to adjust the values stored in the tuples directly. Columns would not be modifiable – the schema is frozen now that the system is in production.

Soldiers or commanders in the field could also use a mobile device oriented version of the system in order to retrieve information while away from a permanent base. Special views oriented specifically for mobile devices could be generated, although the existing iteration already automatically scales page elements to conform to whatever display resolution is being used. This feature could be used in conjunction with scaled-down images, fonts, and simplified controls for mobile devices.

Results and Analysis

The resultant MLMS product achieves the design goals – the two separate LMS and ORBAT views provide a simple and interactive visualization of the required data. The database which the system connects to is extensive, providing a reasonable facsimile of the real world utilization of the product. The performance in the LMS view is somewhat slower than expected. This component would benefit from performance optimizations; however, given the size of the database and the processing of the large resultant datasets, the performance is acceptable for demonstration if not for real world use.

Lessons Learned

Productivity of systems development increased significantly when the project database was centralized, which is one of the main design principles of the program. Developing the system with individual users holding an instance of the database on their disconnected PCs resulted in divergences between the databases which limited our testing and validation capabilities because the data being displayed was not the same. However, upon completion of the code for the system, it was exceedingly difficult to decentralize and instantiate the database for demonstration purposes – this was the result of a version mismatch between the SQL server instance which centralized the database, and the version located on the individual PCs. Future projects would benefit greatly from an internet connected database server that could operate independently regardless of developers' locations, and had separate instances for development mode and production mode.

Usability

Overall, the usability of this project seems to suggest that it would be useful for implementation. While undoubtedly the US Army and Department of Defence use classified internal systems for logistics operations data, it is likely that the systems in place feature many similar components to what MLMS showcases.

One potential usability issue which exists with MLMS is in the ORBAT view – if the Google Maps plugin is missing or unsupported on the platform which the system is being used on, the ORBAT imagery may not be displayed. This requires that the plugin exists for the platform. If the plugin does not exist, this limits the usability on the platform. This is related to the rendering aspect of the SOA used within this

system – although all the processing and information retrieval occurs separately from the client, eventually the imagery data must be returned to the user.

With respect to the future plans for a mobile version of the application, the existing interface elements and the large graphical format provides good usability with touchscreen interfaces. Desktops and tablet PCs with touch screens can take advantage of this feature at the current stage.

Appendix A – Additional System Screenshots

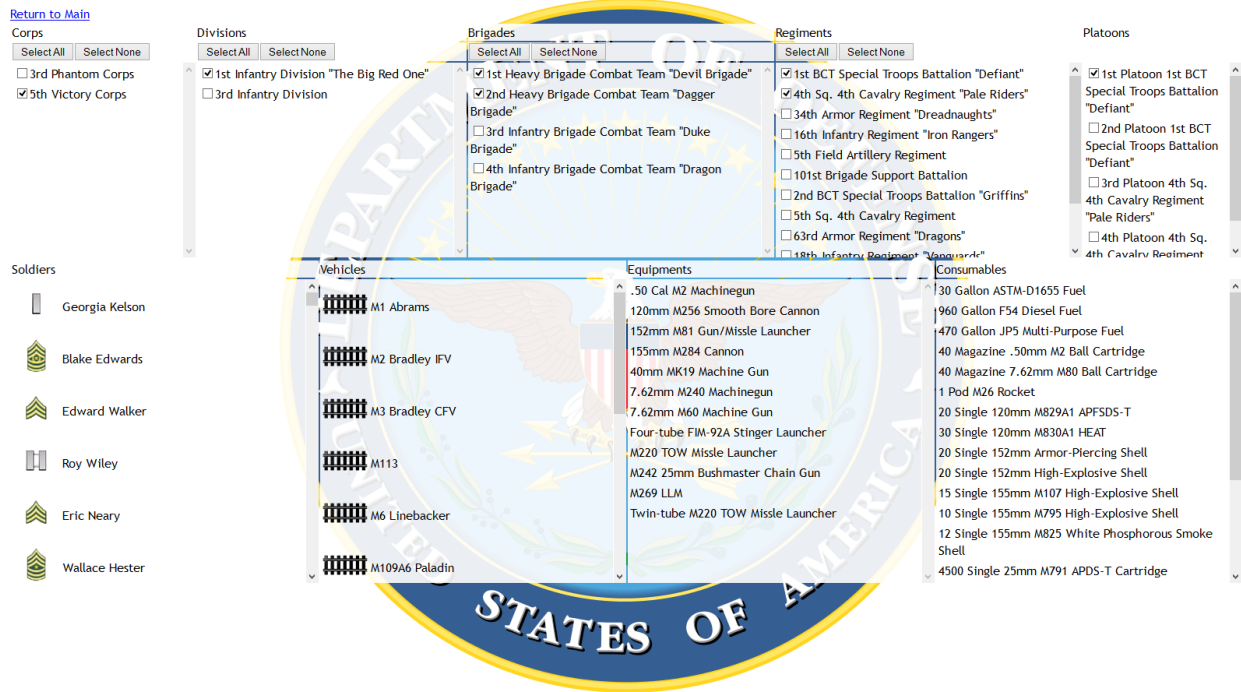


Figure 6 - LMS Sample Query Overview



Figure 7 - ORBAT View with Corps Selected



Figure 8 - ORBAT View with Divisions Selected



Figure 9 - ORBAT View with Brigades Selected



Figure 10 - ORBAT View with Regiments Selected

Colonel: AaronBarclay6th Infantry Regiment
 Colonel: AlfredBallard5th Infantry Regiment
 Colonel: AllenBlair2nd BCT Special Troops Battalion "Griffins"
 Colonel: AllenWigfall27th Field Artillery Regiment
 Colonel: AnthonyPang1st Sq. 13th Cavalry Regiment
 Colonel: BillyFierros1st Field Artillery Regiment
 Colonel: BrettJohnson4th Sq. 9th Cavalry Regiment "Darkhorse"
 Colonel: BrianMcDade703rd Brigade Support Battalion "Maintain"
 Colonel: CedricFuller34th Armor Regiment "Dreadnaughts"
 Colonel: CharlesLane2nd Bat. 5th Cavalry Regiment "Lancers"
 Colonel: CharlesSmotherman36th Infantry Regiment
 Colonel: ClarenceCastanon5th Sq. 7th Cavalry Regiment "Warpaint"
 Colonel: CoyBrown5th Sq. 4th Cavalry Regiment
 Colonel: CynthiaKramer101st Brigade Support Battalion
 Colonel: DanielWeekley1st Sq. 9th Cavalry Regiment "Head Hunters"
 Colonel: DarrickMerino2nd Bat. 12th Cavalry Regiment "Thunderhorse"
 Colonel: DarrickMerino5th Bat. 82nd Field Artillery "Black Dragons"
 Colonel: DavidCouch203rd Brigade Support Battalion "Eagle"
 Colonel: DavidPayne3rd Sq. 7th Cavalry Regiment "Saber"
 Colonel: DavidPlummer18th Infantry Regiment "Vanguards"
 Colonel: DavidRichardson123rd Brigade Support Battalion
 Colonel: DennisPulliam3rd BCT Special Troops Battalion
 Colonel: DerekCrutchfield215th Brigade Support Battalion "Blacksmiths"
 Colonel: DessieAllen3rd Bat. 8th Cavalry Regiment "Warhorse"
 Colonel: DonaldShultz26th Brigade Support Battalion "Challenger"
 Colonel: DonaldSimmons5th Field Artillery Regiment
 Colonel: DustinRutter1st Bat. 5th Cavalry Regiment "Black Knights"
 Colonel: EdwinGreenfield29th Field Artillery Regiment "Pacesetters"
 Colonel: ElizabethBelanger6th Sq. 8th Cavalry Regiment "Mustang"
 Colonel: EvelynHughes2nd Infantry Regiment
 Colonel: GaryHall51st Infantry Regiment

Figure 11 - Command Roster Sample Output

1: Corporal/Specialist: RichardLehmann
2: Colonel: PatrickKing
3: Sergeant: DerrickHirst
4: Command Sergeant Major: EdwardGarcia
5: Chief Warrant Officer CW5: MartyGamez
6: Corporal/Specialist: MerrillArmstrong
7: Private First Class: StephanieMartin
8: First Lieutenant: MatthewJones
9: Chief Warrant Officer CW5: LiviaBroussard
10: Colonel: MaryGoddard
11: Staff Sergeant: CalvinCooley
12: Command Sergeant Major: GeorgePagan
13: Major: BradGrogan
14: Staff Sergeant: JosephQuick
15: Command Sergeant Major: EddieCunningham
16: Chief Warrant Officer CW2: WilliamPrince
17: Major General: CoryJordan
18: General: OliverGreene
19: Master Sergeant: AaronBird
20: Captain: DennisGibbon
21: Sergeant First Class: AlanFields
22: Chief Warrant Officer CW2: ReynaldoGoff
23: Brigadier General: MilesHatcher
24: Colonel: CedricFuller
25: Corporal/Specialist: RaymondMiddleton
26: Sergeant First Class: JamesMartinez
27: Brigadier General: ChristopherRudd
28: Sergeant Major: DavisBrody
29: Private First Class: RandolphFowler
30: Private: RogerJones
31: Brigadier General: WilliamCalderon

Figure 12 - Soldier Roster Sample Output