Transportation Systems Condition-Based Maintenance Plus – Phase II

Final Report

Prepared under:

NCMS Project No. 142072-B and
Cooperative Agreement HQ0034-20-2-0007
for the
Commercial Technologies for Maintenance Activities (CTMA) Program

September 2023

National Center for Manufacturing Sciences
3025 Boardwalk
Ann Arbor, Michigan 48108-3230

Table of Contents

Section **Page**

List of Figures

List of Tables

Table **Page**

Acronyms and Abbreviations

Term	Definition
ABCD	Army Bulk CBM Data
ABS	Antilock Braking System
API	Application Programming Interface
BOD	Business Object Documents
CAN	Controller Area Network
CBM	Condition-Based Maintenance
CBM+	Condition-Based Maintenance Plus
CDF	Common Data Format
CTMA	Commercial Technologies for Maintenance Activities
DM1	Diagnostic Message
DTC	Diagnostic Trouble Code
DOD	Department of Defense
ETL	Extract, Transform, and Load
FMTV	Family of Medium Tactical Vehicles
FMI	Failure Mode Identifier
FSR	Field Service Representative
GDLB	Ground Digital Logbook
GFI	Government Furnished Information
HDFS	Hadoop Distributed File System
IRS	Interface Requirements Specifications
JDMS	JTDI Delivery Management System
JTDI	Joint Technical Data Integration
NASA	National Aeronautics and Space Administration
NCMS	National Center for Manufacturing Sciences
ODASD-MR	Office of the Deputy Assistant Secretary of Defense, Materiel Readiness
OEM	Original Equipment Manufacturer
PLS	Palletized Load System
PMCS	Preventive Maintenance Checks and Services
PM TS	Program Manager Transportation Systems
PPMx	Prognostics and Predictive Maintenance
RDI	Ricardo Defense, Inc.
S&T	Supply and Transportation
SPN	Suspect Parameter Number
SQL	Structured Query Language
TACOM	Tank-Automotive and Armaments Command
TCU	Telematics Control Unit
TWV	Tactical Wheeled Vehicle
U.S.	United States

1. Executive Summary

When vital assets are unavailable due to unplanned maintenance or system failures and operations cannot be performed, the result is reduced productivity, degraded profits, and dissatisfied customers. For the Department of Defense (DOD), this problem could threaten mission accomplishment. Currently, organizations in both private industry and within the DOD face obstacles in collecting, migrating, integrating, analyzing, and displaying the data generated on a single platform, as well as aggregated data at the various Fleet Management echelons, which limits an organization's ability to implement Prognostics and Predictive Maintenance (PPMx).

Program Executive Office Combat Support and Combat Service Support platforms, and specifically Project Manager Transportation Systems (PM TS) Tactical Wheeled Vehicles (TWVs), extended the pilot project that demonstrated Condition-Based Maintenance Plus (CBM+) capabilities on 29 Family of Medium Tactical Vehicles (FMTV) A1P2s at Fort Irwin, California. This extension added 10 Palletized Load Systems (PLSs) at Fort Irwin in the Supply and Transportation (S&T) motor pool to demonstrate the interoperability of both the Condition-Based Maintenance (CBM) hardware and the Analytics Algorithms. The objective during this extension was to establish a PLS CBM kit, continue collecting vehicle data to demonstrate algorithm detection of oil degradation status change, and further mature predictive algorithms to justify maintenance and repair activity changes.

As with Phase I, the project was divided into two different paths: Organic and Original Equipment Manufacturer (OEM). The Organic path can be found in Ricardo Defense, Inc.'s (RDI) CTMA Final Report (142072-A). The OEM path's high-level tasks identified for Phase II were:

- Task 1 – PLS CBM Kit Design
- Task 2 – FMTV A1P2 Advancements
- Task 3 – Data Collection
- Task 4 – Analyze and Visualize Data
- Task 5 – PLS CBM Retrofit
- Task 6 – FMTV A1P2 Advancements
- Task 7 – Uninstallation
- Task 8 – Systems Integration Lab Telematics Support

Funding for the collaborative effort was secured through the National Center for Manufacturing Sciences (NCMS) Commercial Technologies for Maintenance Activities (CTMA) Program and the Office of the Deputy Assistant Secretary of Defense, Materiel Readiness (ODASD-MR).

1.1 Results

Integrating CBM requires not only a formal process for moving data, but also hardware integration on vehicle platforms. Any number of additional components increases the government's logistical footprint. Execution of Phase II allowed for the demonstration of interoperability of CBM vehicle components as well as commonality throughout the entire CBM+ architecture and processes.

The Oshkosh Defense, LLC (Oshkosh) approach to Phase II was to capitalize on the benefits demonstrated in Phase I. Data collected during Phase I allowed for a confident selection of the necessary sensors and data transfer method. The CBM kit for Phase II was tailored and did not include the same targeted systems as Phase I, but where the systems were the same, the sensors used were the same.

Although the monitored systems were different, Oshkosh commonized the data ingestions to allow for two different data sets to flow through the same pipelines. With this approach, the data could be analyzed and displayed in the same manner.

By collaborating with RDI, Oshkosh demonstrated the potential connections between the Organic path and the OEM path. Oshkosh demonstrated a proof of concept during the final months of the pilot that leveraged OEM knowledge and data linking, RDI's Joint Technical Data Integration (JTDI) Delivery Management System (JDMS) and Ground Digital Logbook (GDLB) application, and government formatting and protocol standards. This concept allows for flexibility, commonization, and consistency for accurate and updated data.

1.2 Benefits

Common Solution

Oshkosh applied the same hardware (e.g., oil degradation sensors) that was used on other defense vehicle platforms enabled with CBM capabilities. Applying common components across multiple platforms allowed the team to leverage purchasing from vendors. Additionally, the use of common hardware reduced the variability in the data received and aid in the quick development and application of condition-based algorithms that use the same data.

Software developed to facilitate data transfer on other CBM-enabled platforms can be applied to platforms upgraded with CBM capabilities. The commonality within the CBM data transfer software ensures that the data is moved securely and with little to no data loss. The data between CBM-enabled systems will be consistent and allows for cross-platform analytics.

Commonality between analytical techniques is also a benefit when comparing to other CBM-enabled vehicle platforms. Algorithms developed from similar systems can be applied with little adjustments as opposed to having to develop new analytics from scratch. For example, the basic calculations of the engine oil algorithm can be applied to a newly CBM-enabled system without having to change the entire mechanics of the code behind it, saving time and leveraging hours of testing and development with no overlap.

Oshkosh commonized the application of analytical visualizations between CBM-enabled platforms, allowing users to see a consistent output from the collected and analyzed data. Multiple vehicles' systems can be displayed on the same visualization with consistent measures and data points displayed side by side, allowing users to make data-driven decisions about their entire fleet of vehicles, not just one vehicle platform.

OEM Data Connection

As the OEM, Oshkosh leveraged engineering expertise from vehicle design and performance factors to optimally place the necessary elements. Additionally, Oshkosh ensured performance specifications were not unintentionally impacted. Oshkosh's engineering expertise was critical in establishing thresholds for measures and understanding key drivers for the results of the data collected.

Oshkosh connected the telematics vehicle data to the Logistics Support Analysis data, allowing for a direct application to maintenance and service. This data connection is critical to creating insights and recommendations in a more automated and repeatable process that adequately adjusts with configuration changes.

1.3 Invention Disclosure

Invention Disclosure Report(s):

DD882 Sent to NCMS ☐
No Inventions (Negative Report) ☒

1.4 Project Partners

- United States (U.S.) Army Program Manager Transportation Systems (PM TS)
- U.S. Army Integrated Logistics Support Center

- U.S. Navy Program Management Office Joint Technical Data Integration (JTDI)
- U.S. Army Materiel Systems Analysis Activity *(observer)*
- U.S. Army Combat Capability Development Command Ground Vehicle Systems Center *(observer)*
- U.S. Army Aviation and Missile Command Logistics Center *(observer)*
- Oshkosh Defense, LLC
- Ricardo Defense, Inc. (RDI)
- National Center for Manufacturing Sciences (NCMS)

2. Introduction

2.1 Background

The concepts of CBM, CBM+, Predictive Maintenance, and Prognostic Maintenance have been around for many years; however, the application and execution of these concepts to facilitate improved readiness and reduced costs is difficult. Commercially, there is an abundance of data and limited conditions regarding the ability to collect and transmit data.

2.2 Purpose

In 2018, Oshkosh entered into a Collaborative Agreement with PM TS and NCMS to demonstrate the capabilities and benefits of commercial telematics and CBM+ technology within the TWV community. Oshkosh strengthened its partnership by leveraging machine learning and data analytics to bring real-time solutions to the fleet that ultimately improved availability and lowered lifecycle costs. The target for Phase I was the FMTV A1P2.

For Phase II, Oshkosh targeted the PLS A1 to apply a similar solution to establish commonality in terms of CBM design as well as analytics and results.

2.3 Scope/Approach

Oshkosh established and demonstrated the foundations of an end-to-end PPMx during Phase I of the PM TS PPMx pilot project. Phase II is intended to build upon the success of Phase I and address some of the key obstacles identified using a combination of commercial and government capabilities that will demonstrate the value of PPMx in improving readiness of the TWV fleet.

3. Project Narrative

A virtual project kickoff occurred in early 2021 due to COVID-19 protocols. The objective of the kickoff was to solidify the intent of the pilot, and all parties agreed upon the project plan and milestones. This pilot focused heavily on support and collaboration with Fort Irwin motor pools and, as such, this initial meeting identified key points of contact and communication requests.

Upon completion of the Start of Work meeting, Oshkosh immediately began work. Figure 1 shows the high-level operational model used to facilitate the plan. Note that Oshkosh applied the same concept from Phase I to the integration of PLS to the pilot.

3.1 Design

The integration of condition sensors and telematics on the FMTV A1P2 during Phase I proved valuable in many ways. To illustrate commonality and integration across platforms, Phase II added PLS as a platform to integrate both sensors and telematics. Oshkosh identified the most valuable pieces from the FMTV A1P2 design as the oil degradation, delta pressure sensors, and telematics. These elements produce the data necessary to impact the Preventive Maintenance Checks and Services (PMCS) oil change schedules.

To begin analysis of the design, Oshkosh assessed whether the main hardware components from Phase I could be used in this design as well (the actual sensors and telematics). The FMTV A1P2 and PLS A1 have different engines; however, this only impacted the placement of the sensor and not the actual sensor itself so it was determined that the same oil degradation sensor could be used. The only deviation Oshkosh made was to use an antenna that was more common with commercial use.

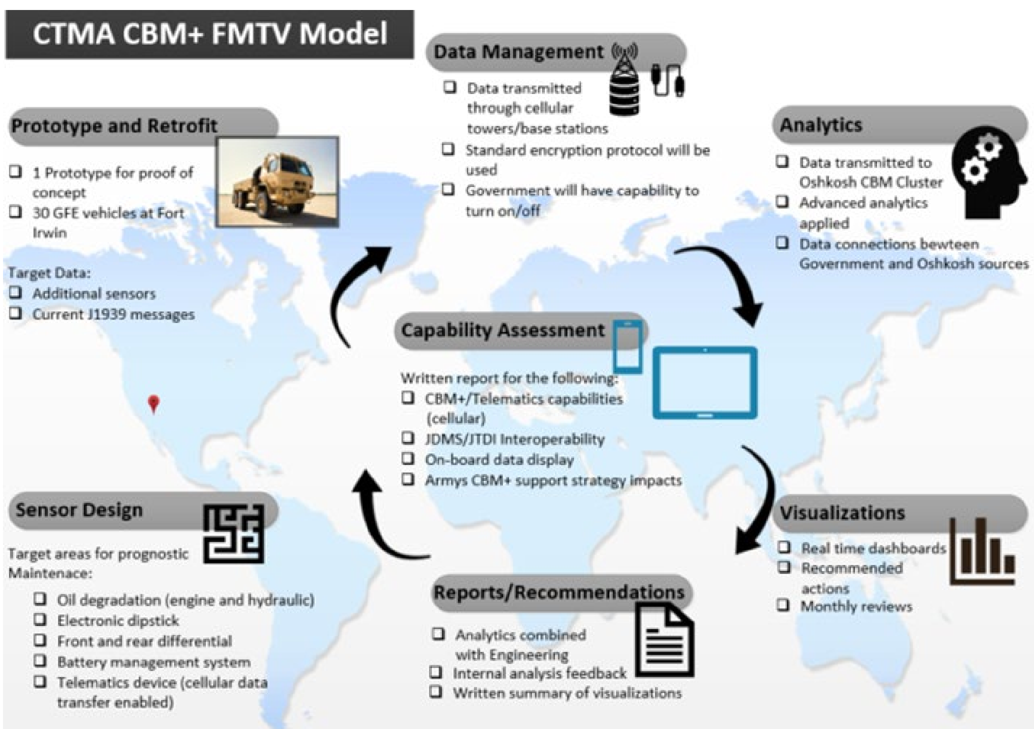

Figure 1. High-Level Operational Model
Oshkosh applied the same initial concept of operations for the CTMA OEM path to the integration of PLS to the pilot.

Figure 2 shows the components included in the PLS A1 CBM kit for this pilot:

- Engine and hydraulic oil degradation sensors
- Engine oil filter and fuel filter pressure sensors (two each) (fuel filter pressure sensors ultimately not installed due to incorrect harness connectors)
- Data logger/telematics module and cellular antenna
- Input/output module and harnesses
- Mounting hardware and fittings

3.2 Prototype

Oshkosh used its own asset to physically fit the sensors and validate the design. The assets participating in the pilot were a different configuration and did not allow for the fuel sensor to be added.

3.3 Retrofit

Over a period of two weeks, Oshkosh integrated the CBM design on a quantity of 10 PLS A1s at Fort Irwin. Oshkosh Field Service Representatives (FSRs) performed the installation with on-site support from Oshkosh engineers and the S&T and Maintenance Troop motor pools. FSRs and Oshkosh personnel reviewed work instructions prior to the installation activity and the entire team performed installation on the first vehicle to ensure proper documentation and processes were followed.

Team members performed the following steps to complete the retrofit activity on each vehicle:

1. **Engine oil and steering hydraulic oil samples taken.** Oil quality information from the vehicles before kit installation aided in data analysis and algorithm application.
2. **Vehicle inspection.** FSRs documented the current condition of the vehicles and any issues needing correction prior to kit installation and recorded serial numbers and Telematics Control Unit (TCU) numbers to complete the necessary data connections.
3. **CBM kit hardware installed**. FSRs completed the installs with on-site support from Oshkosh engineers.
4. **Engine oil and hydraulic oil filled.** FSRs drained and replaced all hydraulic oil during the installation of the steering hydraulic oil degradation sensor. Filters were not changed, nor was any other maintenance performed during the retrofit.
5. **Vehicles powered on and run in the yard.** Vehicles were started and ran to confirm data transmission from the TCU to Oshkosh servers. Oshkosh created a dashboard to verify and audit all signals within five seconds of vehicle start. Confirmation of the data transmittal was nearly instantaneous due to this real-time validation.

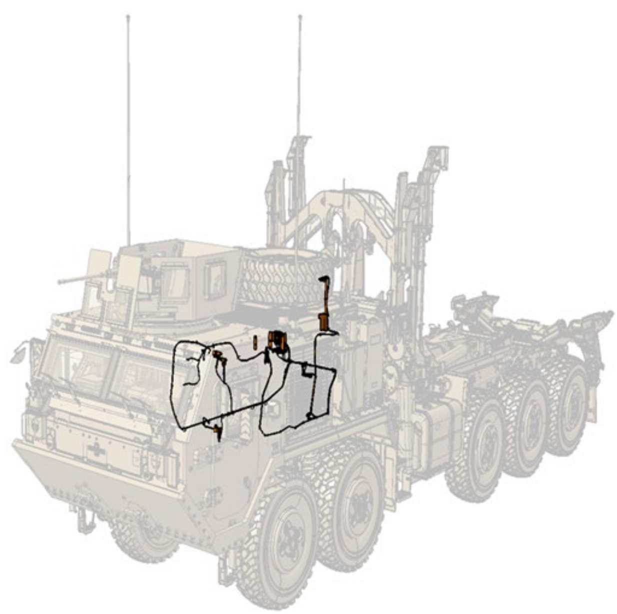

Figure 2. CBM+ Pilot Program Kit Components for PLS A1
Oshkosh identified locations and methods for integrating additional sensors and devices into existing circuits and vehicle systems.

– **PLS Installation Dashboard: Real-Time Installation Validation.** Ensuring that all sensors, wiring harnesses, and antennas are installed and properly functioning is critical to ingesting good, reliable data. This ensures not only that data was transmitting after a vehicle had a CBM kit installed, but also that Oshkosh could validate the signals being transmitted in near real time.

  Oshkosh successfully implemented a validation dashboard of CBM kits that worked in near real time. This dashboard automatically evaluated and displayed results within minutes of the vehicle being run. Oshkosh accomplished this not only by tracking signals being broadcast and received, but also by building in a testing plan that included hitting certain engine rpms and vehicle speeds as well as an array of other designed values. This validation step helped prevent rework and identify any issues with loose wires or sensors.

  The validation included tests for each Controller Area Network (CAN) node and installed sensor, including the alternator, Engine Control Module, engine oil sensor, engine oil filter (the sensor had schematic issues), fuel filter (the filter also had schematic issues), hydraulic oil, Antilock Braking System (ABS), and transmission systems. The dashboard would fill up semi-circle gauges as valid signals were ingested, providing a clear indicator of which systems were performing as expected and which systems were not (Figure 3). Oshkosh data scientists and the installation technicians were able to validate each truck installation in real time (Figure 4).

After completion of the retrofit effort, Oshkosh returned the vehicles to the units for immediate use. Oshkosh provided New Equipment Training to inform the units of the new kit components. No impacts to regular operation or maintenance procedures were expected as a result of the kit installation.

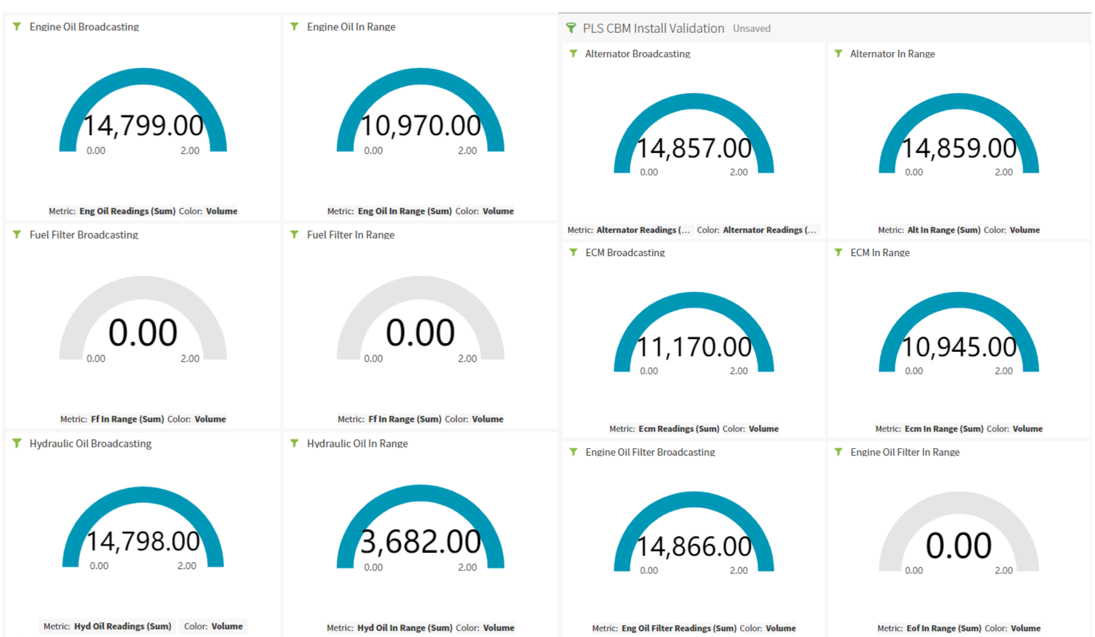

Figure 3. PLS Installation Dashboard
Dashboard gauges indicated which systems were performing as expected and which were not.

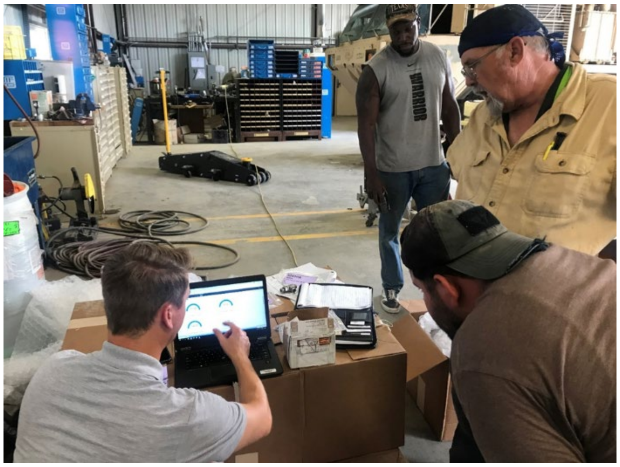

Figure 4. PLS Installation Validation
The dashboard allowed real-time installation validation.

Similar to the FMTV A1P2, the PLS A1 did not have an onboard display, so Oshkosh modified the CTMA Web Portal that was established during Phase I to include the PLS A1 data and results. The data was different between the FMTV A1P2 and the PLS A1, so the necessary modifications were made to merge the data and display it correctly. During Phase I, Oshkosh allowed secure access to the analytics through tablets, which were again available throughout Phase II. These tablets provided an option to bring the data as close to the end user as possible. Oshkosh trained and registered end users to access the secure CTMA Web Portal on the tablet.

3.4 Data Management

Oshkosh used the same data flow architecture that was established during Phase I of the pilot for Phase II (Figure 5). The PLS A1 CBM kit included the same data transfer protocol and TCU with cellular data transfer enabled as the FMTV A1P2. This validated the CBM kit and data management practice could be applied to different platforms.

Oshkosh collected raw vehicle operational data and condition-based data from the installed CBM kit, transferred it from the vehicle, and stored it in the Oshkosh data repository. Once the data was in the repository, transformation of the raw signals occurred, preparing the data for analytics and reporting. The transfer of data from the vehicle to the data repository was done securely. Data at rest and data in motion were subject to encryption. After analysis and transformation of the data was complete, the extracted information was securely passed back to the government in the form of visualizations and recommendations via a web-based portal. Figure 5 shows a high-level overview of the data flow architecture.

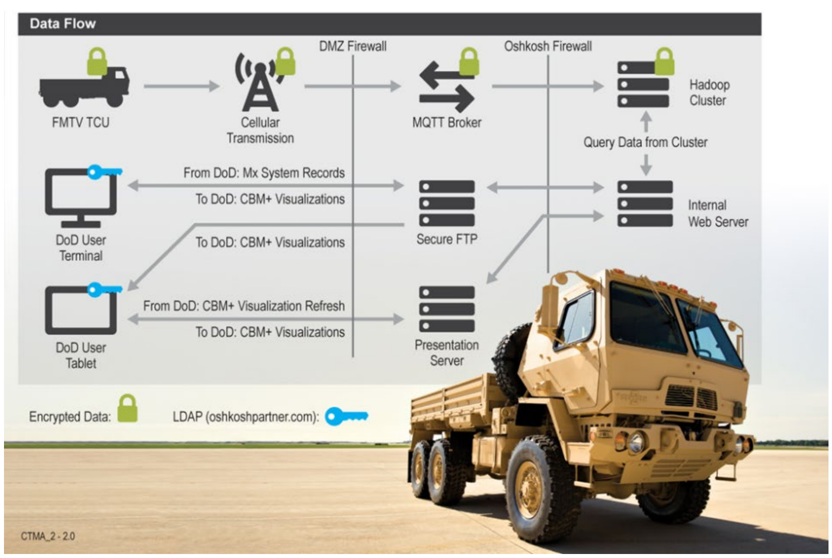

Figure 5. Data Flow Architecture
Oshkosh's data flow architecture efficiently allowed for near instantaneous data transmission.

3.4.1 Data Transfer Architecture

In Phase I, data transfer off-platform was proven through means of a cellular enabled TCU. Oshkosh replicated the fidelity and confidence in this data transfer for Phase II with the integration of PLS A1. Below is a detailed overview of the data transfer method established during Phase I and carried over for use in Phase II.
In this pilot, Oshkosh used a telematics module fitted to each vehicle to collect and transmit the data from the vehicle. The modules that Oshkosh used for this pilot had the capability to transmit data from the vehicle by means of cellular communication. The telematics module had the ability to transform raw signal data into manageable data packets that could be transmitted by cellular communication. The data packets were given unique identification and time stamp information to the dataset, which aided in transformation and analytics performed in Oshkosh's on-premise cloud. Oshkosh also applied Transport Layer Security 1.2 encryption to the data while on the telematics module before it was transmitted from the vehicle.

The TCU collected data from the J1939 CAN, as well as from the new fitted CBM sensors. When the vehicle's master power was turned on, each TCU sampled, or collected the data, at a rate of one reading per second. The TCUs carry their own Transport Layer Security 1.2 encryption and are pre-programmed to package the collected data into manageable data packets. Each data packet contains TCU identification time stamps of the sampled data and the data collected from the various systems.

The TCU software currently has no messaging prioritization. Messages leave the TCU via a first-in-first-out queuing system. However, data is transmitted to and stored on Oshkosh servers and can be pieced out or prioritized to accommodate for variability in bandwidth. The data stored in the Oshkosh servers is then translated and transformed into analysis ready tabular data.

As noted earlier, the telematics module transfers the data from the device by way of a cellular connection through a commercial wireless carrier's 4G network. If no cellular service is available, the TCU stores the information onboard until the TCU connects to a cellular network. Once the data is successfully transferred from the TCU, the data is removed from the TCU, freeing up memory on the device. The data is streamed through a secure Virtual Private Network data tunnel to a receiver that collects the data packets and transfers them to the Oshkosh on-premise Internet of Things platform where the data is prepared for analytics. The data transfer method off-platform was similar to Phase I, but during Phase II, Oshkosh focused on making the pipeline more efficient and reliable and added in quality measures.

During Phase II, Oshkosh used various methods to Extract, Transform, and Load (ETL) the vehicle data from the vehicles to the CTMA Web Portal. Once the TCU message moves though the wireless network and the data staged at the queuing service, it is then moved into the database, transformed for analytics usage, and staged for the Application Programming Interface (API) calls on the portal. Once the data from the J1939 passes through the TCU and is stored in the short-term queuing service, the data flow management tool is responsible for data enrichment and movement of the data into a long-term message queuing service. Data quality is completed in the data flow management tool where duplicate detection takes place. Here lineage data is added to the original message to determine if the message was received more than once by the short-term queuing service. Once this data lineage is added to the message it is then stored in a in-memory database for duplicate detection. If a duplicate is detected, that message is removed from the flow management tool. Once the data has moved through the data quality checks, it is then stored into the distributed file system. High level process flow is depicted in Figure 6.

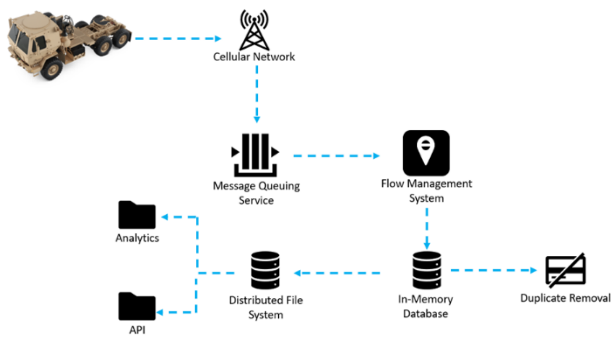

Figure 6. Pipeline Process Flow
Phase II improved efficiencies in the data pipeline and streamlined data accuracy.

HDFS

From the NiFi pipeline, all the messages are stored in one table (def_fmtv_cbm_raw). All records are stored in a string data type. From here, Oshkosh branched out to create all the views needed for the data analytics. Oshkosh created seven views from the raw table to include def_fmtv_abcd, def_fmtv_daily, def_fmtv_fleet_sensor_err, def_fmtv_historical, def_fmtv_last_sensor_reading, def_fmtv_pilot_status, and def_fmtv_sensor_status, which are described in Table 1.

The second part of the Hadoop Distributed File System (HDFS) is the faults side of the NiFi pipeline. The NiFi pipeline takes all the Diagnostic Message (DM1) messages and parses out all the fault codes that are in the actives column from the vehicle. From here, all faults are stored in HDFS as dm1p event types in the def_fmtv_cbm_fault table. From this table Oshkosh created two views, def_fmtv_faults_combined_nifi and def_fmtv_faults_active, which are described in Table 2.

Table 1. Data Analytics Views
Views are created to specifically satisfy data science and analytics projects.

View	Description
Def_fmtv_abcd	This view combines all the data needed to create the Army Bulk CBM Data (ABCD) files to send to U.S. Army Tank-Automotive and Armaments Command (TACOM). This data is used in the Python script that transforms the data to be accepted by the TACOM database.
Def_fmtv_daily	This view takes the vehicle data from the last day the vehicle ran. Data used in this view is composed of start/end time, start/end hours, start/end miles, total miles, total hours, average moving speed, and sensor status.
Def_fmtv_fleet_sensor_err	This view looks at the CBM sensors installed on the vehicle. If a sensor is returning null values or 0s this indicates the sensors are not providing any data through the J1939 CAN bus.
Def_fmtv_historical	This view does the same thing as the def_fmtv_daily view but it contains all days that the vehicle ran.
Def_fmtv_last_sensor_reading	This view takes the last reading from the engine and fuel filter sensors, calculates the pressure difference between the two, then does a calculation on the pressure difference to determine the life percentage of the engine and fuel filters.
Def_fmtv_pilot_status	This view takes daily data from each vehicle and calculates miles traveled, top speed, average moving speed, max rpm, number of records, and number of idle records. This is then combined to get the overall numbers per vehicle.
Def_fmtv_sensor_status	This view determines whether the sensors are sending data back from the vehicle and marks them as active or inactive. It is strictly used for portal usage.

Table 2. Faults Views
Faults need additional attention and data engineering applied to create views for data science and analytics.

View	Description
Def_fmtv_faults_combined_nifi	This view takes the faults from the vehicles and combines them with two other tables to get fault descriptions, the system each fault belongs to, and task name (if applicable).
Def_fmtv_faults_active	This view only shows current active faults on the vehicles. Once a fault is no longer active it will drop off this view.

Portal Presentation Database

After Oshkosh created all the views needed to do analysis on the vehicle data, it was moved into the Portal Presentation Database to be sent to the portal pages. Here, 15 separate Structured Query Language (SQL) queries were made to HDFS to pull the data into Portal Presentation Database for the portal APIs to call. Table 3 shows the 14 queries that were used to populate this data.

It is a common misconception that the raw data from the vehicle can automatically be transferred into meaningful insights. The process outlined above is necessary to accurately convert the results into meaningful information. All of the queries described above allow the CTMA Web Portal to function accurately and efficiently. These are automatic, repeatable, and transferable and allow for more efficient processes to consume and analyze the data. Figure 7 shows how the data appears in the CTMA Web Portal.

3.5 Analytics

With the data views above created and automatically processed, Oshkosh applied advanced analytics techniques to the data to address the issues surrounding readiness and support costs. Oshkosh collaborated on targeted areas for analysis during biweekly Integrated Product Teams calls. Oshkosh applied the Cross-Industry Standard Process for Data Mining process model for data science projects. The following section describes the analytics projects and approaches.

Table 3. Staged Portal Views
Data is staged for efficient use in the web-based portal using queries.

Query	Description
Pilot_hours	This query combines all hours the vehicle has run and breaks them down by model, branch, and site.
Pilot_miles	This query combines all miles driven and breaks them down by model, branch, and site.
Trucks_reporting	This query lists how many vehicles are sending data back to Message Queuing Telemetry Transport broker and breaks them down by model, branch, and site.
Startups	This query counts how many times a vehicle in the program has started, broken down by model, site, and branch.
Bad_sensor	This query looks at the sensor data and checks whether null values are being returned. If there are null values, it will flag the vehicle and which sensor group is not sending data back.
Fleet_last_sensor_reading	This query displays the results from the def_fmtv_last_sensor_reading for the oil and fuel filter graph on the portal page.
Eng_oil_deg	This query displays the engine oil status (good, caution, or bad).
Svc_schedule	This query shows when the next service is predicted to be completed.
System_fault_count	This query counts how many active faults there are currently and breaks them down by model, site, and branch.
Fleet_sensor_status	This query displays the results from the def_fmtv_sensor_status to show which sensors are active and which sensors are inactive.
System_fault	This query displays all active faults for each vehicle and possible troubleshooting action.
Cbm_alert	This query displays CBM sensor issues for each vehicle and required maintenance task (if applicable).
Daily_fleet	This query displays data from each vehicle for the last time that vehicle was in operation (e.g., engine hours, average speed, and idle engine hours).
Fleet_daily_all	This query displays the same data as the daily_fleet but sums together every time a vehicle was in operation to get overall totals for each vehicle.

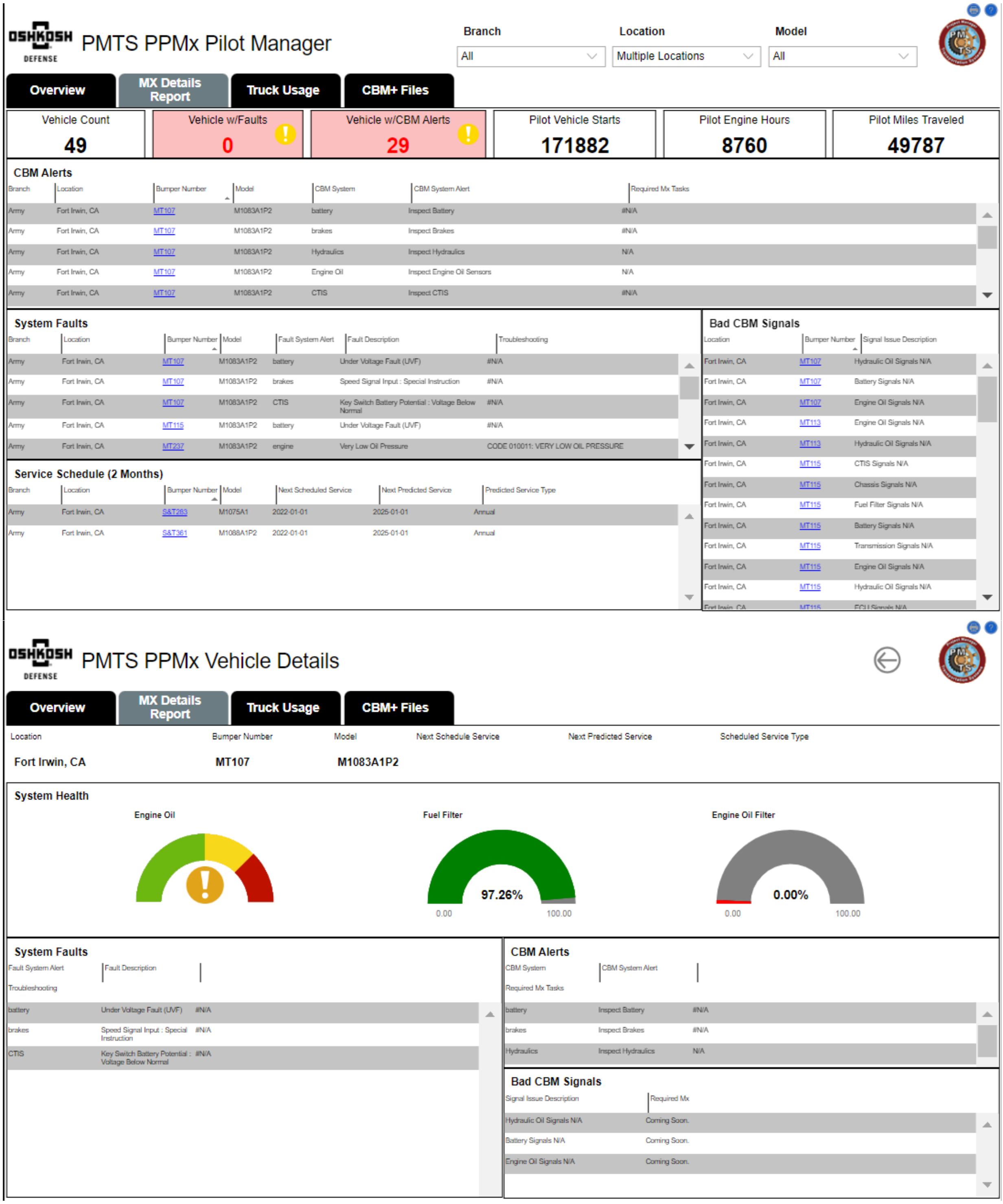

Figure 7. Customer-Facing View of CTMA Web Portal
The dashboard home page depicts an overview of pilot fleet status.

Vehicle Faults: Making Connections

Most vehicles have built-in warning and fault indicators. These indicators can report anything from low battery to low tire pressure. Some of these indicators have a warning light on the dashboard such as “ABS active.” Most faults, however, do not have a clear indicator light and require special tools to read the diagnostic fault code. Oshkosh used the telematics system to broadcast these diagnostic fault codes. From there, Oshkosh translated the encoded faults into meaningful text by connecting the codes to their J1939 definitions, including vendor-specific J1939 codes. In addition to decoding the fault codes, Oshkosh linked certain fault codes to troubleshooting tasks listed in the Interactive Electronic Technical Manual. All this information was displayed on the CTMA Web Portal, which resulted in a streamlined process of fault triggered to troubleshooting task identified (Figure 8).

Driver Behavior: Providing Informative Scores

Vehicle use insight is critical information that affects the health of a vehicle. Incorrect driving habits and patterns (i.e., when the vehicle is not used as designed and recommended) can increase the risk of accidents or safety incidents, increase wear on components, and cause component failure. Oshkosh created a comparative framework to provide insight into vehicle driving behavior, which identified several features for each vehicle trip, such as harsh braking, maximum speed achieved, and average speed. Oshkosh created a probabilistic distribution for each feature and compared this distribution to an ideal distribution. This led to similarity scores that provide insight and feedback based on the features used to monitor each trip.

The first step for this process was to take the raw one-second-level telematics data and transform it into individual vehicle trips by measuring the time gaps between consecutive data points. If there were more than 10 minutes between data points, Oshkosh considered those two data points to be the end/start of a trip. Once these trip indicators were created, the data was aggregated into individual trips and calculated the selected trip features from the telematics data. After the data was transformed into individual vehicle trips, the dataset was further reduced by only keeping trips that met a minimum distance threshold. With this dataset, probabilistic distributions were created.

The ideal probabilistic distribution can be achieved in several ways. One method is to use trips from known good or experienced drivers to create the distribution. Another method is to statistically generate the distribution based on known ideal conditions. The method Oshkosh used was a random sampling of data from all drivers, which creates a distribution of how the vehicles are typically used. The method for determining the ideal distribution can be dependent on implementation of the driver behavior algorithm.

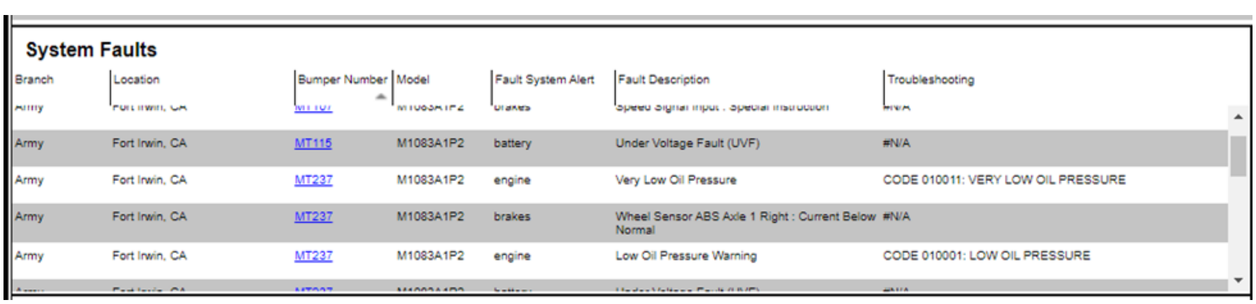

*Figure 8. **Web Portal: System Faults Connected to Troubleshooting***
End users can quickly link current vehicle faults to troubleshooting paths in the Technical Manuals.

One implementation is analyzing trips belonging only to a test course. In this case, Oshkosh likely would not want a random sampling of trips from all drivers as many of the drivers would be driving the vehicles for the first time and thus are unlikely to be ideal drivers. In this scenario, Oshkosh would want an expert driver or trainer to perform several trips on the course to build up the ideal distribution. Then the newly trained drivers could be compared to the expert driver and could provide feedback during their training (Figures 9-11).

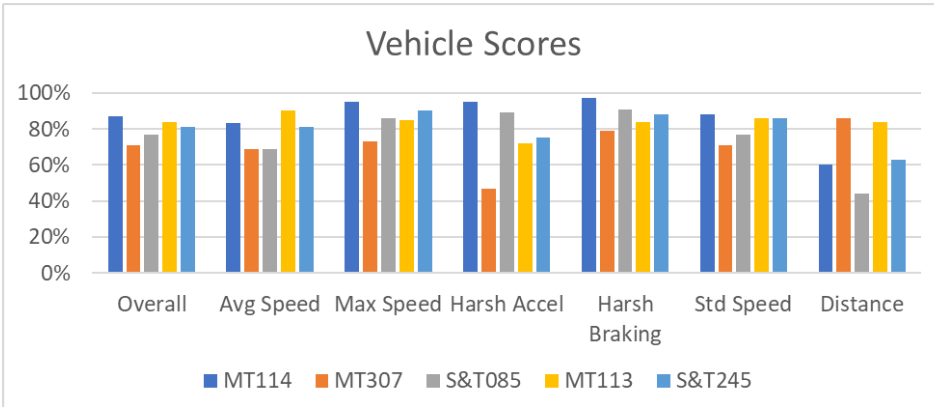

Figure 9. Vehicle Scores Compared to the Standard Driver
Visual of the vehicle scores allows for quick comparisons and potential feedback areas.

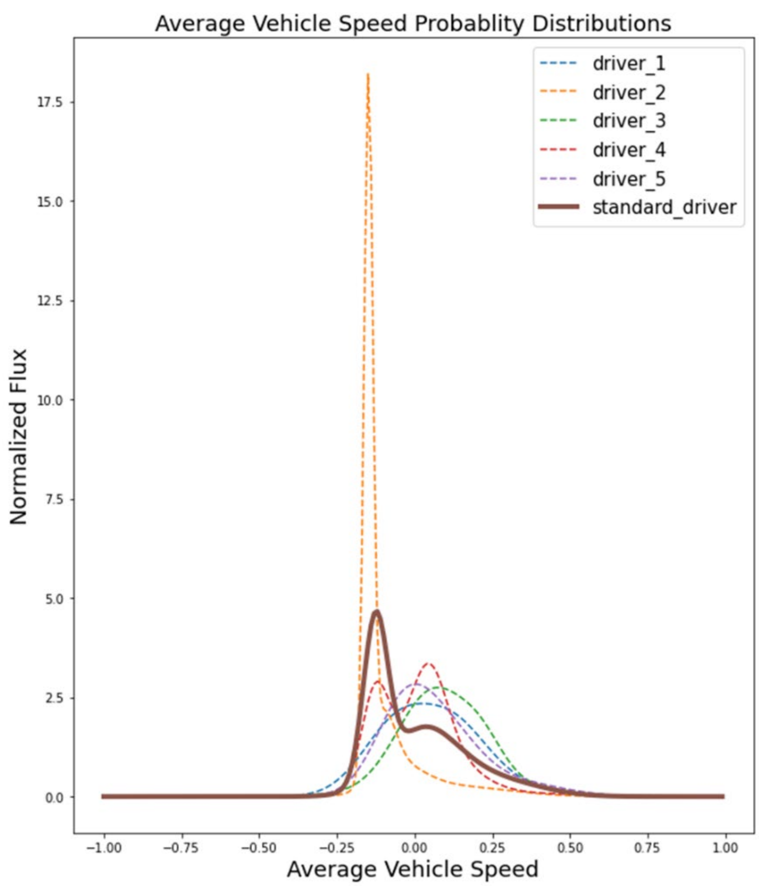

Figure 10. Probabilistic Average Speed Distribution
Detailed view from one of the major feature elements of the driver score.

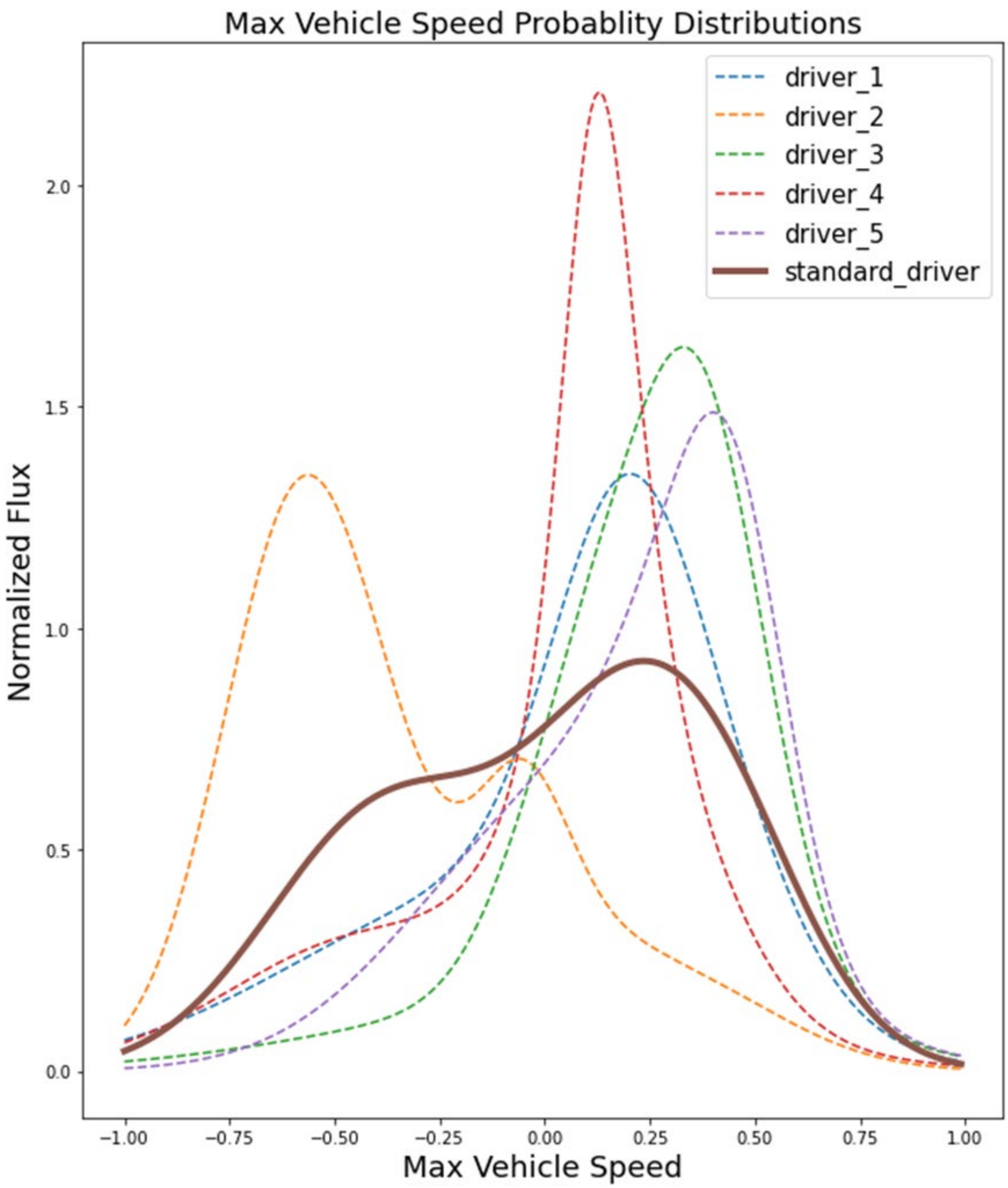

Figure 11. Probabilistic Max Speed Distribution
Detailed view from one of the major feature elements of the driver score.

Engine/Hydraulic Oil

The ability to prolong periods between oil changes leads to increased fleet readiness and reduced waste in the form of man-hours, down-time, cost, and materials. Currently, FMTV engine oil is changed every six months or 6,000 miles, whichever occurs first.

Typical usage in the pilot fleet of FMTVs was not in the range of 6,000 miles after six months. The vehicle with the highest usage after 10 months of data collection travelled less than 3,000 miles, and oil sample analysis results led the team to believe that the oil quality is not degrading prior to oil changes. After 29 months of data collection, the highest usage vehicle, ST245, travelled just over 6,100 miles and ran for just over 500 engine hours (Figures 12 and 13).

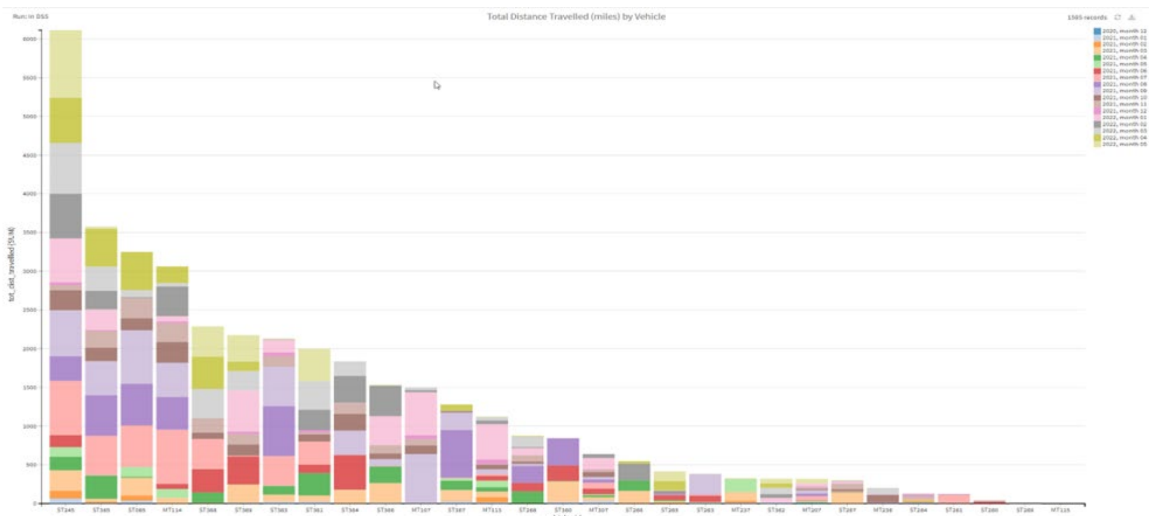

Figure 12. Distance Traveled by Vehicle
Total miles driven is a key component to interval maintenance, which was not achieved during the observed period.

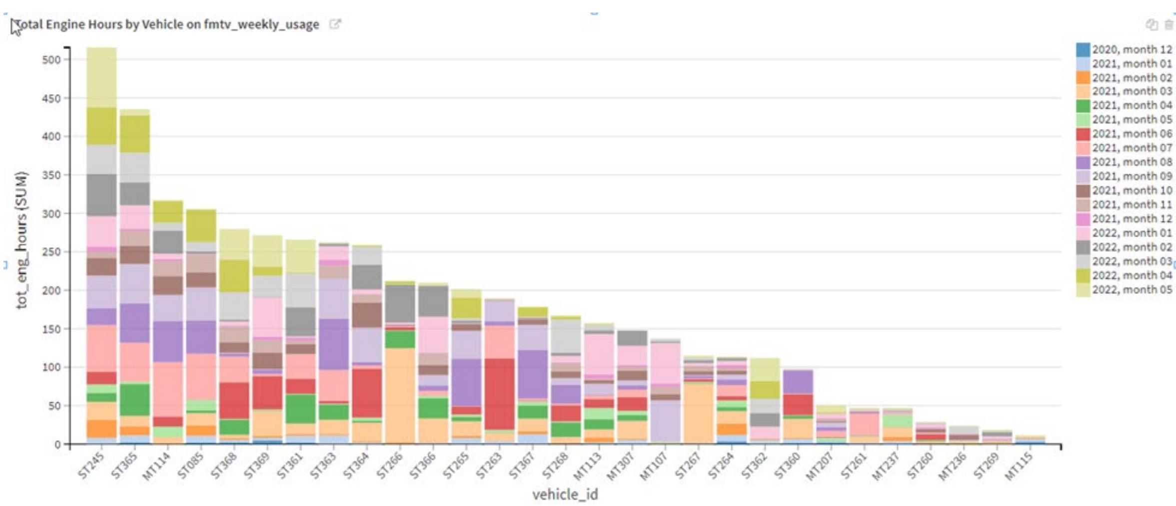

Figure 13. Engine Hours by Vehicle
Engine hours is another component to interval maintenance, which again, was not achieved during the observed period.

Oshkosh is seeking to use sensor technology to measure the quality of oil in each individual vehicle. Once there is accurately determined oil quality, a number of things can be done:

- Use historical usage data to create predictive algorithms of oil quality based on actual engine operating conditions. These conditions might include vehicle driving cycle, engine type, and environmental conditions. Severe driving cycles are known to accelerate the rate of engine oil degradation. These include short trips, particularly in cold weather, high-speed high-load driving, stop-and-go driving, and driving in dusty/dirty environments.
- Combine future usage information with sensor data to predict oil quality over a time period to provide Remaining Useful Life estimates.
- Recommend an optimal oil change schedule for a fleet utilizing anticipated mission information.

Data Understanding

To measure the change in oil quality, Oshkosh installed a fluid property sensor that measures density (g/cm³), viscosity (cSt), temperature (°C), and the dielectric constant of the oil. The values are measured by the sensor once every 30 seconds.

By observing the parameters of the oil, it is possible to detect the presence of the following contaminants: soot, water/coolant, fuel, and metal. Soot can be detected at the lower limit of 0.2% and the upper limit of 5%. Water can be detected between 2,000 and 20,000 ppm. Coolant can be detected between 4,000 and 40,000 ppm. Fuel can be detected between 1% and 10%. Metal can be detected between 20,000 and 100,000 ppm. The levels of contaminants listed correspond to levels that accelerate the oxidation of the oil leading to faster oil degradation. The sensor does not detect metals at the low threshold of single-digit ppm that is possible using oil sample analysis to detect wear metals, but rather at the high concentration known to accelerate oil oxidation.

Each of the oil parameters changes over the life of the oil. It is crucial to look at the oil age in engine hours from a known point of oil change and observe the change in property as a function of engine hours. Since the engine oil was changed during the installation of the sensors in December 2019, Oshkosh could begin to observe the change in oil properties from the new oil phase into the normal usage phase. Oshkosh expected to see three to four phases of oil in a normal usage cycle: reak-in phase, normal usage phase, oil change phase, and past due phase. Ideally, once the oil reaches the oil change phase, an oil change is performed, and the oil never reaches the past due phase. Then, the goal of an oil condition prediction algorithm is to estimate when the vehicle will reach that oil change phase, giving the operators and maintainers time to react with appropriate PMCS actions.

The oil change interval is typically six months to one year in normal 30- to 50-hour weekly driving. That corresponds to 1,000 to 2,000 hours of engine hours. Oshkosh expects to see each phase change in hundreds of hours of engine usage. After two and a half years of monitoring, the CTMA pilot fleet had several vehicles cross the threshold of usage needed to see degradation of oil. The weekly usage chart in Figure 14 shows that the vehicles in the pilot fleet were rarely used for 50 hours per week.

Oshkosh conducted periodic oil samples to monitor the oil while extending the oil past the recommended six-month oil change time frame. Oshkosh took approximately 150 oil samples across the fleet during the 30-month period. Higher usage vehicles were sampled more frequently.

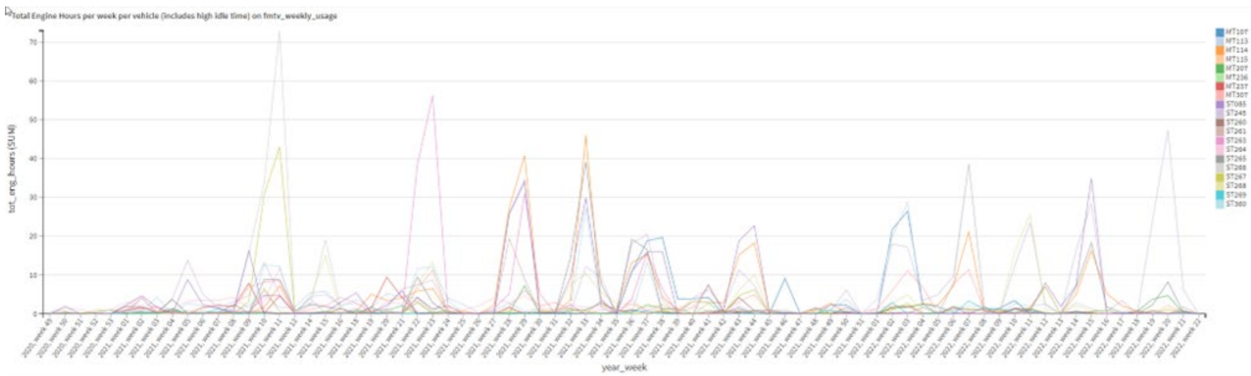

Figure 14. Weekly Vehicle Hours
Vehicle usage trends indicate typically less than 50 hours per week.

Data Preparation

Before creating a reliable oil degradation algorithm, Oshkosh needed to clean and prepare the data for analysis and modelling. The data directly from the sensor had a combination of noise, error values, and initialization values that were removed. Oshkosh also observed that the stability of the sensor readings increased as the temperature of the oil increased.

Oshkosh employed statistical techniques to determine the optimum thresholds within which to sample the sensor values to build a clean dataset. Thresholds were based on temperature and engine speed.

Finally, Oshkosh utilized a Loess filter to smooth the data and further reduce the noise in the data before building temperature models for each oil parameter.

Data Analysis

Oshkosh used oil sample analyses to help validate the data cleaning and modeling effort.

As a result of the periodic oil samples, some signs of change in oil condition were observed, but no indication that the oil needed to be changed on any vehicle until early 2022, more than 24 months into the life of the oil. In early 2022, two vehicles showed signs of needing an oil change: one had an extremely low viscosity level of 8.0 cSt, not consistent with MIL-STD 15W40 oil, indicating that vehicle may have had an oil change with incorrect oil used; the other showed signs of silicon contamination that may indicate dirt in the oil. Oshkosh identified both vehicles for immediate oil changes.

Oshkosh saw the same low viscosity in our sensor readings, which provided supporting evidence that the methods used to process and analyze the data were correct (Figure 15).

A correlation analysis between the sensor readings, the oil sample analysis results, and the processed oil degradation sensor data indicates high correlation between oil degradation, high viscosity, high total acid number, and a high amount of low-speed usage (Figure 16).

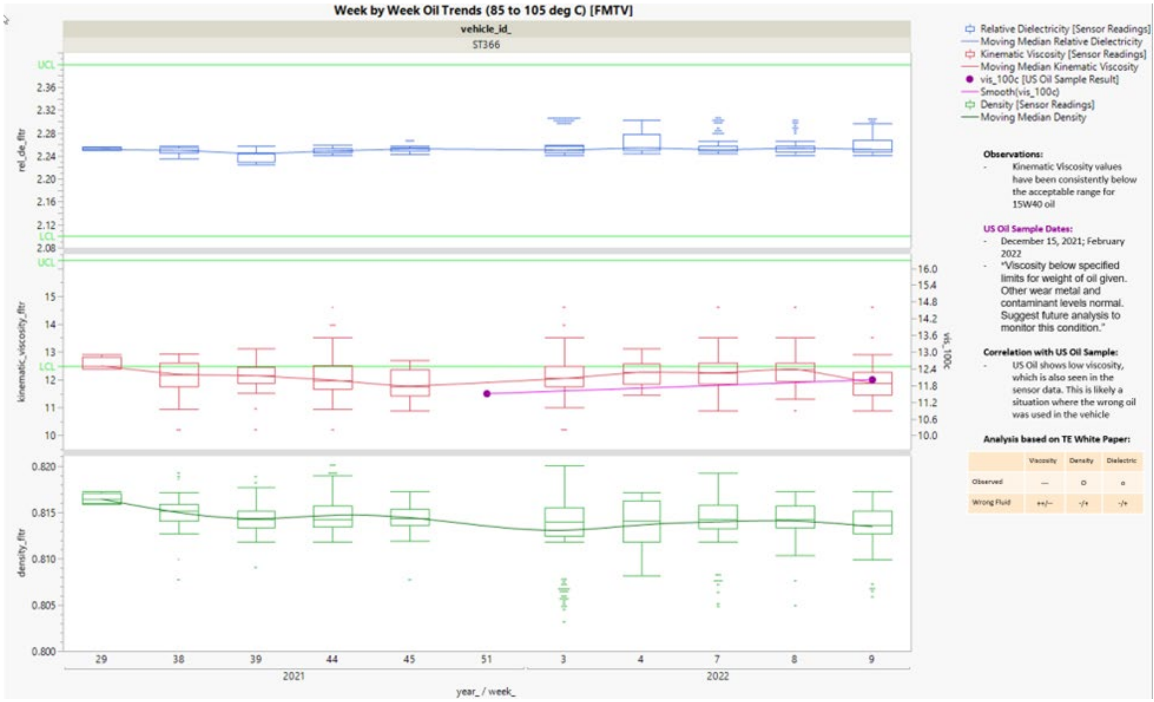

Figure 15. Week by Week Oil Trends
Observed sensor readings and oil spectrometer results indicated validation of applied algorithm.

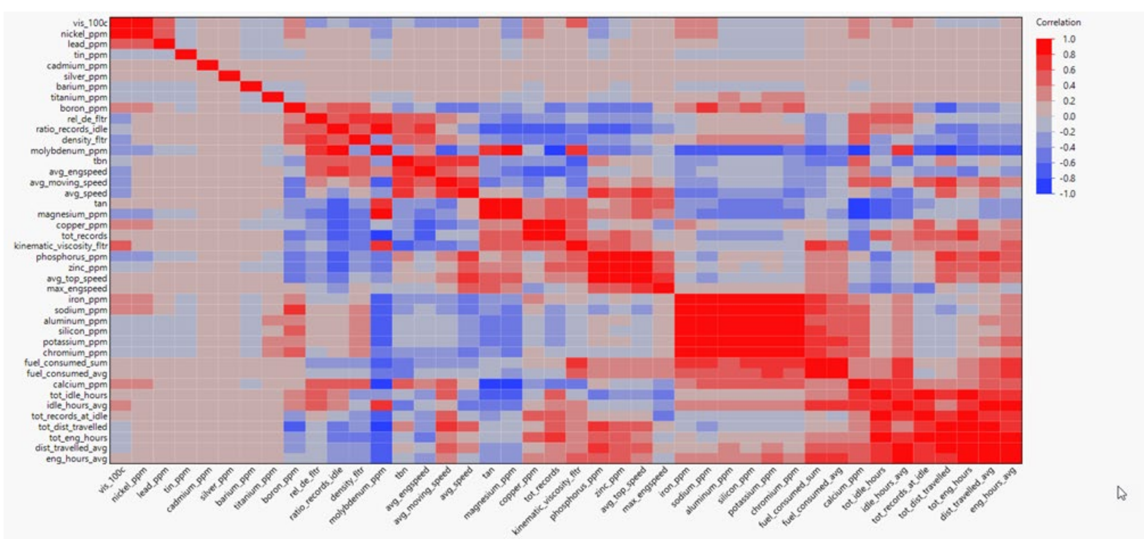

Figure 16. Correlation Analysis
Key feature elements identified contributing to the degradation of oil.

Table 4 shows the observed correlation between wear metals and usage. From the table, it can be seen that there is a strong positive correlation between the total number of records for a vehicle and the amount of copper in the oil. The total number of records is related to the time a vehicle was powered on and sending data over the course of the pilot. Additionally, lower average engine speed, lower average vehicle speeds, and a lower ratio of time spent at idle can be seen to correlate with copper particles. This may indicate that low-speed, non-highway driving is correlated with wear metals. The average and total daily fuel consumed by a vehicle is also correlated with the presence of wear metals and contaminant metals (Table 5).

On the other hand, a correlation between usage and the presence of additive metals can also be seen. Additive metals would be expected to decrease over the life of the oil, but Oshkosh hypothesized that the high-usage vehicles have oil added more frequently as the usage causes some to burn off. This could lead to the additive metal concentration remaining high. From the figure, low idle time, and thusly high average top speed can be seen to correlate with a higher amount of additive metals remaining in the oil, while distance travelled, high idle time, and higher viscosity is negatively correlated with additive metal concentration (Table 6).

Table 4. Wear Metals
Feature comparison allows for a look at the range of correlation.

Variable	By Variable	Correlation	Abs [Correlation]	Lower 95%	Upper 95%
tot_records	copper_ppm	0.92	0.92	0.91	0.93
avg_engspeed	copper_ppm	-0.73	0.73	-0.76	-0.70
ratio_records_idle	copper_ppm	-0.66	0.66	-0.70	-0.63
fuel_consumed_sum	chromium_ppm	0.63	0.63	0.60	0.67
fuel_consumed_sum	iron_ppm	0.62	0.62	0.58	0.66
fuel_consumed_sum	aluminum_ppm	0.59	0.59	0.55	0.63
fuel_consumed_avg	chromium_ppm	0.56	0.56	0.52	0.60
fuel_consumed_avg	iron_ppm	0.54	0.54	0.49	0.58
fuel_consumed_avg	aluminum_ppm	0.53	0.53	0.48	0.57
kinematic_viscosity_fltr	copper_ppm	0.51	0.51	0.37	0.63
avg_speed	chromium_ppm	-0.47	0.47	-0.51	-0.42
avg_speed	iron_ppm	-0.44	0.44	-0.49	-0.39
avg_moving_speed	copper_ppm	-0.42	0.42	-0.47	-0.37

Table 5. Contaminant Metals
Validated theory for wear metals and contaminant metals correlation.

Variable	By Variable	Correlation	Abs[Correlation]	Lower 95%	Upper 95%
fuel_consumed_sum	potassium_ppm	0.682	0.682	0.648	0.713
fuel_consumed_sum	silicon_ppm	0.621	0.621	0.581	0.657
fuel_consumed_avg	potassium_ppm	0.608	0.608	0.567	0.645
fuel_consumed_sum	sodium_ppm	0.602	0.602	0.562	0.640
fuel_consumed_avg	silicon_ppm	0.554	0.554	0.510	0.595
fuel_consumed_avg	sodium_ppm	0.521	0.521	0.475	0.565
avg_speed	sodium_ppm	-0.452	0.452	-0.499	-0.401

Table 6. Additive Metals
Additional correlation ranges for additive metals round out the experiment.

Variable	By Variable	Correlation	Abs[Correlation]	Lower 95%	Upper 95%
avg_top_speed	phosphorus_ppm	0.90	0.90	0.89	0.91
avg_top_speed	zinc_ppm	0.89	0.89	0.87	0.90
ratio_records_idle	zinc_ppm	-0.77	0.77	-0.79	-0.74
tot_idle_hours	calcium_ppm	0.75	0.75	0.71	0.78
tot_idle_hours	magnesium_ppm	-0.74	0.74	-0.77	-0.70
ratio_records_idle	phosphorus_ppm	-0.69	0.69	-0.72	-0.66
tot_dist_travelled	phosphorus_ppm	0.68	0.68	0.64	0.71
avg_speed	phosphorus_ppm	0.68	0.68	0.64	0.71
ratio_records_idle	magnesium_ppm	-0.67	0.67	-0.70	-0.63
dist_travelled_avg	phosphorus_ppm	0.62	0.62	0.58	0.66
idle_hours_avg	calcium_ppm	0.62	0.62	0.57	0.66
dist_travelled_avg	zinc_ppm	0.60	0.60	0.56	0.64
tot_dist_travelled	zinc_ppm	0.60	0.60	0.56	0.64
tot_records	magnesium_ppm	0.58	0.58	0.54	0.62
idle_hours_avg	magnesium_ppm	-0.57	0.57	-0.62	-0.52
avg_moving_speed	calcium_ppm	0.54	0.54	0.50	0.58
avg_speed	zinc_ppm	0.54	0.54	0.49	0.58
kinematic_viscosity_fltr	zinc_ppm	0.52	0.52	0.38	0.63
kinematic_viscosity_fltr	magnesium_ppm	0.51	0.51	0.37	0.63
kinematic_viscosity_fltr	calcium_ppm	-0.51	0.51	-0.63	-0.37

Oshkosh established an in-house oil testing experiment to create a controlled environment in which to assess the oil quality using onboard sensors. Oshkosh constructed the oil test stand in the Test and Development Center with the capability of running 24 hours a day to artificially age oil. At the conclusion of the CTMA pilot, Oshkosh aged a sample of oil to 500 engine hours under load and was in the process of ageing another batch to 500 hours with an updated sensor from the sensor vendor. Oshkosh took oil samples every 150 engine hours and sent them to the lab. After 500 engine hours, the lab results showed that the oil was still within usable thresholds, consistent with the oil sample data gathered from the CTMA fleet. The experiment will continue with both batches of oil being combined and aged to 1,000 hours.

Conclusion

The data gathered in the CTMA pilot is leading to an understanding of indicators and trends of oil degradation. The correlations seen between the sensor values, vehicle usage, and oil sample data provide an indication that Oshkosh can estimate oil life using a combination of oil sensor data and vehicle usage data. Two key factors have prevented a fully automated solution from being developed at this time:

- The low usage of vehicles in the CTMA pilot and the high idle time did not provide enough consistent quality data to see strong degradation trends.
- The controlled experiment that is in progress is needed to gain confidence in the oil degradation algorithm output before productionizing a model.

3.6 ABCD File

One of the objectives of the CTMA CBM+ pilot at Fort Irwin was to take the streamed vehicle data and create a data product that could be transmitted to the U.S. Army so they could store the data and use it for future analytics. The Army has developed a file format specification named ABCD that leverages the National Aeronautics and Space Administration (NASA) Common Data Format (CDF) to create CBM data files for all their assets that are collecting data. The ABCD files not only contain the time series data collected from the asset but maintain the metadata about the asset and the measurements collected within the data file. The Army collects ABCD files and stores them within their own data lake where they can be retrieved at any time for analysis.

The current implementation for creation of ABCD files is to reference the guidance from Interface Requirements Specifications (IRS) for the generation of ABCD files set by Data and Information Standards Center of Excellence, Army Aviation and Missile Research, Development and Engineering Center, and Software Engineering Directorate at Redstone Arsenal. Additionally, file creators must use the NASA CDF libraries and API for creating and reading ABCD or CDF files. The ABCD IRS documentation outlines required metadata elements and optional data elements to be included in the files; this gives creators instruction as to what data to include in each data file, any file or data element naming conventions to be used, and descriptions of units of measure to be included in each file. As part of this CBM+ pilot, Oshkosh developed ABCD files in accordance with the guidance from the sources mentioned. The ABCD files are to be created within the Oshkosh Cloud platform and then transmitted to the Army by means of Secure File Transfer Protocol.

The data used for creating the ABCD files largely uses the same process that is outlined in Section 3.5 of this paper. The generalized data flow in Figure 17 illustrates how the data is collected from the vehicle, transmitted to the data lake and prepared for ABCD file creation, and finally delivered to the end user. Additional processes had to be performed to get the data ready for delivery. The green parts of the diagram are the areas where Oshkosh had to deviate from our normal ETL and data product creation practices to create the ABCD files.

Oshkosh's data engineers added custom tables to the data lake that contained metadata about the vehicle systems and data measurements to be used in the ABCD file creation applications. The table of vehicles or vehicle listing contains metadata about the platform identification or assets described in the ABCD files and contains required information for processing into the ABCD data lake. Oshkosh also had to add a listing of sensor or streamed data that would be transmitted in the ABCD files described as measurement locations. Measurement locations uniquely identify the data captured on the vehicles and add metadata about the sensor information, giving it a definition, assigning it a data type (integer, string, float, etc.), and providing default values to enter when data is missing from time series. The measurement location is a crucial element of the process to create ABCD files and needs to be determined prior to files being created.

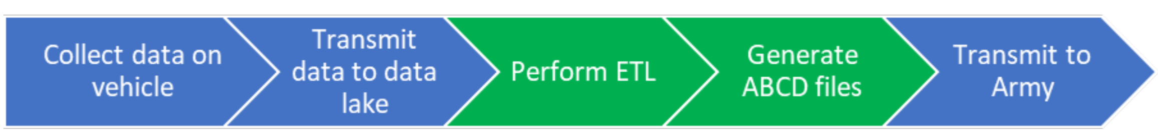

Figure 17. Generalized Data Flow
Data is collected, transmitted to the data lake and prepared for ABCD file creation, and then delivered to the end user.

If data is being collected that is not currently being stored in the Army ABCD data lake, both parties (project participant and government) must collaborate to register new attributes with the Army ABCD data repository before files can be created. Failure to register new attributes can cause processing issues within the Army ABCD data lake.

The global parameters that Oshkosh added to its data lake can be considered as a form of Government Furnished Information (GFI) that will need to be transmitted to the file developers before files can be created.

The software that was developed to create the ABCD files is Python, an open-source software that can be used for a multitude of purposes. Python requires the use of specific function libraries (typically written in C) to carry out tasks that are directed by the programmer. The software Oshkosh created uses an open-source library called Spacepy. Spacepy also requires the download of a C-based library which can be acquired from NASA's CDF API download site. Python is not required to create CDF or ABCD files; however, it was used in this case because of its flexibility and ability to access and process large amounts of data efficiently.

The Python application first gathers the global attributes from the Oshkosh data lake. The global attributes are required for processing in the Army CBM data lake and are like a header of information describing the CDF file type and other elements about the data collected and the asset it was collected from. At this time, the measurement location is retrieved as well from the Oshkosh data lake where it will be used to set the structure for each of the CDF files.

Next, the application retrieves data for one vehicle for the time frame specified by the application user. Once brought into the application the data is parsed out and assembled on the location determined by the ABCD file measurement location and processed into one file per calendar day per vehicle. Once the file is closed, it moves on to the next day until there is no more data to convert for that vehicle. When the vehicle has all its data converted, the application moves on to the next vehicle identified in the vehicle listing. The application will run until all data from all vehicles is converted to ABCD formatted files.

One of the issues Oshkosh encountered with the file creation is that the Python application would write the ABCD files correctly; however, the files were not compressed nor were they able to be closed properly. This was due to an issue with the CDF API software requiring the person who executes the software to have a CDF_TEMP environment variable set up on their HOME or User Profile. Not having this set up caused the files to not be completed and errors in the software. Oshkosh documented the issue and set up CDF_TEMP environment variables for all ABCD file administrators. If future requirements dictate the use of Python or the CDF API download from NASA, all parties should be made aware of this potential issue. To ensure that ABCD files are transferred and processed correctly to the ABCD data lake, a validation must occur to verify that the structure of the files is correct, the z-variables are all registered within the database, and the metadata and file naming conventions are compatible for processing. This validation requires collaboration between the project participant and Army to identify any deficiencies in the files and provide feedback to the file creators that the file creation software will need updating. Any changes to the GFI, such as measurement locations or other global attributes, must be broadcast to the contractors so they can update their file creation software to stay compliant with new requirements of data file deliveries destined for the Army ABCD data lake.

Oshkosh recommends that future efforts to create ABCD files have a set of requirements and outlined guidance from IRS for the generation of ABCD files. Information needs to be transmitted to the contractors to pass along global attributes as well as any optional attributes the program or platform requires. It should also be noted that any changes to these global attributes on the Army side need to be broadcast to contractors so the files that are created are up to latest standards.

Oshkosh also recommends the identification of commonalities between OEM platforms. Validation of the measurement location needs to be performed when there are discrepancies between data elements that are "common" between multiple platforms for an OEM. This validation will ensure that the data that is truly common will match the data of other platforms. These events may be rare; however, validating the data prior to file creation will add to the integrity of the Army CBM dataset.

For future pilots, Oshkosh recommends that processing of ABCD files be moved to be on-platform. Having the ABCD CDF file software embedded on the system can allow for the timely collection, transformation, and transportation of CBM data to the Army. Oshkosh would recommend that these files be accepted and transferred using the JDMS protocol. Having the data files produced by the vehicle and transmitted directly into the Army data lake will allow the data to be available once it has been streamed or downloaded from the vehicle.

The other option would be to create a software package that could sit on either GDLB, the tablet, the Motor Pool Enterprise Node, or at the Army Logistics Data Analysis Center to create the ABCD files off-board and reduce the requirement for additional hardware or increased hardware capability.

3.7 GDLB

During Phase I, project partners had several conversations about the two paths, OEM and Organic, and Oshkosh identified a potential for a merger of those two strategies. GDLB software relied heavily on two things:

- A Logistics Database that holds all of the data and table relationship information, including a "faults" table, which connected the vehicle fault code to the identified maintenance tracks and information from the technical manuals
- Data transmission from the vehicle

The project partners identified that these two activities overlapped and determined in late 2021 for Oshkosh to incorporate an integration prototype into Phase II. Figures 18 and 19 show the initial outline documented in an in-person meeting between RDI, Ground Vehicle Systems Center, PM TS, and Oshkosh.

After review of the in-person conversation and shared vision for the path forward, Oshkosh established the architecture view for how the integration of GDLB would be accomplished. Figure 19 depicts the resulting architecture.

The first thing Oshkosh did when putting together this prototype was set up a top tier and client tier JDMS node. Although there were copious amounts of documentation, Oshkosh encountered a few issues attempting to set it up:

- Setting up the certs
- Performing an equipment load
- Getting client registered to the top tier
- Setup and getting the two JDMS clients to talk to each other

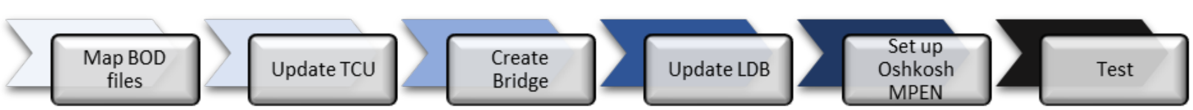

Figure 18. Proposed Process Flow
High level process flow to accomplish the integration effort.

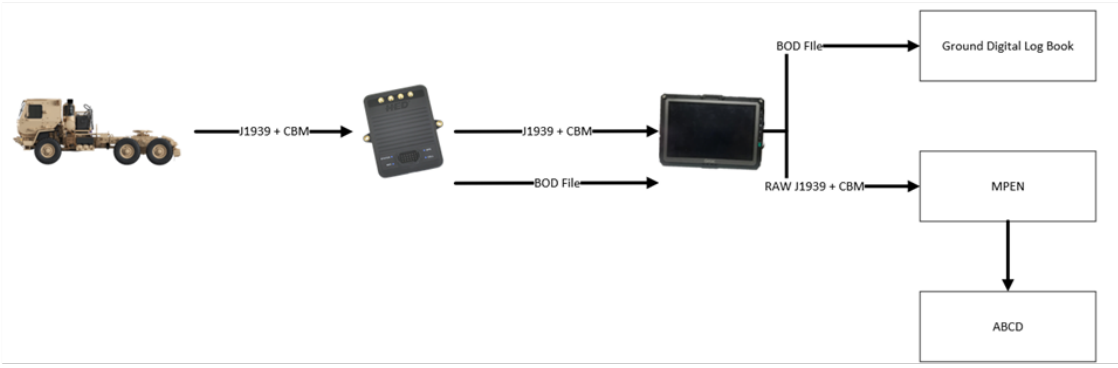

Figure 19. Architecture Incorporating GDLB
High level architecture view for technically accomplishing the integration.

After receiving support from JDMS technical, Oshkosh was able to set up our top tier and client tier nodes. The top tier node was needed to equipment load the client tier tablet with the GDLB software on it. Next, Oshkosh developed an application for the TCU that read DM1 messages via CAN bus. After reading the DM1 message, the application would parse out the Suspect Parameter Number (SPN) and Failure Mode Identifier (FMI). The application would then perform a database lookup using the SPN and FMI. The database lookup would then return the Diagnostic Trouble Code (DTC), DTC Description, and the Business Object Documents (BOD) identifier. Using this information, the application would generate and store a BOD message, that would later be transferred to the GDLB. Due to software incompatibility, Oshkosh was not able to install JDMS on the TCU itself, so a different method of communication to transfer the files to GDLB was needed. To transfer the BOD messages to the GDLB, Oshkosh set up a WebSocket server on the TCU. On the GDLB Oshkosh created a go-between application, Socket Keeper Yoke network manager (SKYnet), which is a WebSocket client. The SKYnet application would connect to the TCU then receive the stored BOD messages. As the BOD messages are received, they are forwarded via JDMS on the GDLB. After the BOD messages are successfully transferred, they are then removed from the TCU. This process essentially demonstrated the use of secure Wi-Fi to transmit data from the TCU on the truck to a tablet – an additional capability for moving data.

1. **Database**
   The current database used for digital logbook is SQL Server 12. Oshkosh received an export of the database from RDI to implement Oshkosh information into it.

2. **Faults**
   All faults loaded in the digital logbook database are located in the dbo.fault table. In order to add any additional faults to this table manually, foreign key restraints must be removed first. The following are the foreign keys that need to be removed:

   - Reffault183 located on the xref_fault_meas_alarm table
   - Reffault244 located on the xref_faultcapability table
   - Reffault29 located on the xref_fault_causal_component table
   - Reffault3 located on the xref_fault_task table
   - Reffault719 located on the xref_fault_model table

   Once these constraints are removed, the dbo.fault table can be updated with any new fault codes. Fault codes in this table are entered multiple times due to them being set by different platform_model_ids. In the current table, there are six platform_model_id codes (1, 2, 3, 4, 6, and 7). These codes are references to dbo.platform_model table, where they break down vehicles by the individual models. The reason the fault codes have multiple entries is because even though platform_model_id 2 and 4 have the same fault codes, they are two completely different model vehicles (Table 7).

The dbo.fault table has a Primary key of fault_id. This is just a numbering sequence to give each record in this table a unique identifier. If the table is manually updated, the sequence for the faults will need to be redone.

3. **Issues**
   Oshkosh tried to add new vehicles to the database (to the dbo.platform table and all tables that reference it), but encountered difficulties when trying to do so. When Oshkosh loaded the database with the new vehicles, they could not be seen in the client tablet. The following are questions Oshkosh cannot answer currently:
   - What is the process of adding new vehicles to the database and what is the data flow process that happens when adding a new vehicle? Is this done on the client side or in an admin platform?
   - What tables are needed to see new vehicles added to the database?
   - Is there another way to add faults to the database besides rewriting the entire dbo.fault table? Is this done on the client table or admin platform?

3.8 Conclusions

Phase II concluded with the primary objectives achieved. Oshkosh demonstrated that the application of the Phase I hardware concept and data collection was transferable to additional platforms. Oshkosh successfully integrated the PLS into the pilot in terms of CBM hardware design, data transfer, and portal visualizations. Oshkosh moved data off the FMTV A1P2 and PLS platforms throughout the entire period of performance.

Oshkosh was able to calibrate the oil degradation algorithms previously applied to FMTV A1P2 to the PLS pilot fleet, proving that algorithms have transferable qualities.

Oshkosh successfully demonstrated the application and integration of GDLB with the CBM+ solution, showing how the two paths can be combined.

Upon completion of the period of performance, all 39 vehicles were returned to pre-pilot configuration.

platform_model_id	manufacturer_part_guid	platform_model_name	platform_family_id
1	776EBB49-4E08-4BF8-AA95-949E02A63C9E	M1075A1 (2320015442251)	1
2	A1A7C170-8654-4443-8F29-BD73391F8276	M1083A1P2 (2320015498610)	2
3	A9FB12AA-5534-49AF-94D4-2ED26AA81A97	M1074A1 (2320015442244)	1
4	30C898C1-4021-48BB-9143-3C61D1CF5484	M1088A1P2 (2320015527759)	2
5	6ECFDA10-46C5-4F96-8B67-E26B8B41837E	MKR18 (2320015427628)	2
6	776EBB49-4E08-4BF8-AA95-949E02A63C9E	XM1075A0ECHU (2320016247308)	1
7	776EBB49-4E08-4BF8-AA95-949E02A63C9E	XM1075A1ECHU (2320016249568)	1

Table 7. dbo.platform_model Table
View of the dbo.platform_model table shows the driver for additional linking and database structure connections.

4. Project Benefits

Phase I identified environmental, material, and readiness benefits that were extrapolated in Phase II. These benefits included saved maintenance costs, reduced vehicle downtime due to maintenance, and environmental benefits reducing the amount of engine oil that had to be consumed and disposed of. The extrapolated values for FMTV A1P2 as seen over the two-year pilot include:

- Environmental benefits – engine oil saved 377 gallons of oil saved over two years
- Material Cost avoidance – $4,350 saved on oil changes for 29 trucks over two years
- Readiness Improvements – 232 hours in vehicle downtime saved over 29 trucks over two years

In addition to these benefits identified in Phase I and carried over into Phase II, the PLS had the potential to recognize these savings as well, however the data was only observed on the PLS for six months which was not long enough to fully validate any savings. Many of the recognized benefits deal with more intangibles: transferrable and common processes and hardware, data sharing pipelines, and the applicability of OEM data enrichment.

5. Recommendations

Looking forward, Oshkosh recommends some continued activities to create an applicable solution that is both cost effective and meets the needs of end users.

GDLB and CBM+ integration: As proven by the proof of concept integration of GDLB, there are many benefits to combining these efforts and allowing partners to collaborate to produce a cohesive solution that addresses the full maintenance burden and resolution. Oshkosh identified duplication of effort and overlap of strategies when addressing this integration which would be mitigated with a cohesive partnership approach.

Edge Capability: During Phase I of the pilot, Oshkosh introduced the concept of Edge Capability, but it was further solidified through Phase II that this would be an ongoing recommendation. Although the data can be moved off-platform effectively through the use of telematics, having the results display and be communicated to the operator and/or maintainer only through a means of a tablet or computer proved to be less than optimal. Over the course of Phase I and Phase II, there was minimal use from the operators and maintainers to consume the analytics results and health indicators from an online portal. Incorporating the capability of onboard computation (Edge Capability) would allow the insights to be able to be consumed closer to the end users. Results could be displayed through an onboard display if the platforms contain that capability (Driver Display Unit), or Oshkosh could deploy a software application that could be integrated onto a tablet or display of choice with the end result being getting this information closer to the end user.

Data flow architecture: One of the foundational pieces to CBM+ and PPMx is data. Throughout Phases I and II, Oshkosh successfully demonstrated the capability to not only autonomously move data off-platform but also distribute to key stakeholders and government entities needing to house and/or analyze the data. Oshkosh moved data off-platform via the use of cellular data transmission, which may or may not be the future preferred method of data movement. While there are certain standards and protocols either in development or deployed in other areas of defense, the TWV community is lagging in adoption and distribution. Oshkosh recommends introducing policy and guidance that requires certain elements of architecture on-platform to facilitate data transmission necessary for CBM+ activities to occur.

Maintenance information integration: Telematics data collection proved valuable to interrogate and perform descriptive analytics. Oshkosh observed a shortfall when targeting predictive analytics as it applies to specific vehicle platforms. Events are witnessed in the data; however, if they are not associated to maintenance activities performed, it is difficult to train models to look for indicators or create algorithms that will proactively predict the occurrence of a future event. In essence, Oshkosh has built an impressive mouse trap, and now are waiting on the mice, where the mousetrap is the algorithm and the maintenance actions are the mice. To fully adopt a strategy where the predictive failures can be targeted, the eco-system of vehicle data, sensor data, maintenance information, technical manuals, and bills of material are all necessary data attributes to connect.