Developing a Scalable Data Analytics Pipeline

Patrick Rupp, Anastacia MacAllister, Jason Garrison, George Hellstern
Lockheed Martin Corporation
Fort Worth, TX
patrick.h.rupp@lmco.com, anastacia.m.macallister@lmco.com, jason.t.garrison@lmco.com, george.hellstern@lmco.com

Colonel Daniel Javorsek
Air Force Operational Test and Evaluation Center
Nellis Air Force Base, Nevada
daniel.javorsek@us.af.mil

Phil Chu
CENTRA
Arlington, VA
philip.chu.ctr@darpa.mil

ABSTRACT

According to the DoD, data is becoming a critical strategic asset for future conflicts. The DoD expects data to drive key decisions within areas such as supply chain and battlefield weapons. However, before data can be analyzed and mined for key insights it first needs to be usable. Collecting and managing data is a continuous challenge for both the DoD and many other organizations. Disparate collection systems and storage makes generating actionable insights from the data cumbersome. In some cases, data management and configuration can make up the bulk of data analysis projects. As organizations seek to become data driven, they not only need to think about analysis strategies, but also data management to ensure data collected is usable.

Work presented describes the development of a scalable data analytics pipeline used to analyze aircraft performance with-in a larger defensive counter air scenario. Currently today, air combat exercises and simulations generate large amounts of data with different structures without any method for combining and analyzing the data to improve chances of mission success. The architecture was designed to work optimally in an environment that ingests data in many different formats. As designed, this novel data pipeline allows for scalability, processing optimization/parallelization, and most importantly provides insulation from data format changes. The method developed pairs an unstructured data lake with a structured data warehouse to ensure a variety of data sources can be used to discover insights for improved warfighter decision making. This architecture allows for raw data formats from many different sources to be ingested, parsed, and stored into a common format such that the analytics techniques are re-usable and scalable. The paper describes the development and scaling process, providing a tangible example of how to manage data from a complex simulation for ingestion into a data analysis process.

ABOUT THE AUTHORS

Anastacia MacAllister is a Machine Learning Researcher at Lockheed Martin's Skunk Works®. Her research focuses on developing machine learning algorithms using sparse or imbalanced data sets, exploratory data analytics, and prototyping novel machine learning algorithms. Dr. MacAllister is currently the data analytics lead for Lockheed's Air Combat Evolution (ACE) work with the Defense Advanced Research Project's Agency (DARPA). The work focuses on investigating how including artificial intelligence in the battlefield impacts complex multi-aircraft scenarios. Dr. MacAllister received her Ph.D. from Iowa State University of Science and Technology in Mechanical Engineering and Human-Computer Interaction.

Patrick Rupp is a Data Engineer at Lockheed Martin's Skunk Works®. Mr. Rupp specializes in developing data pipelines particularly for use in analytics problems ranging in size from small prototypes to enterprise solutions. Mr. Rupp is currently developing the data pipeline for Lockheed's Air Combat Evolution (ACE) work with the Defense Advanced Research Project's Agency (DARPA), and has previously supported Lockheed's work in DARPA's Systems of Systems Integration Technology and Experimentation (SoSITE) program. Mr. Rupp received his Masters in Data Science from Johns Hopkins University.

Jason "Vandal" Garrison is a Project Lead at Lockheed Martin Skunk Works. Mr. Garrison is a former USAF pilot with over 3000 hours in USAF F-15, MQ-1, MQ-9s and USMC F-5 Adversaries as well as a graduate of the USN Strike Fighter Course TOPGUN Adversary program. For the previous 4 years he's worked in Skunk Works Operations Analysis directorate as an F-35 Instructor Pilot and analyst using both virtual and constructive mission effectiveness tools. Most recently Mr. Garrison has focused on leading Reinforcement Learning applications for combat aircraft at the vehicle control and battle management levels.

George Hellstern has 25 years of experience with systems design, including AI solutions for air-to-air combat and sustainment. He is a program manager for autonomy and AI, unmanned air systems C2 and human performance. Previous experience includes operational, programmatic and technical experience from the Air Mobility Command, the Office of the Secretary of Defense and Lockheed Martin Advanced Development Projects also known as Skunk Works®.

Daniel "Animal" Javorsek Colonel Dan Javorsek is the Commander of Detachment 6, Air Force Operational Test and Evaluation Center, Nellis Air Force Base, Nev., and Director, F-35 U.S. Operational Test Team. AFOTEC's Detachment 6 plans, conducts, and reports on realistic, objective, and impartial operational test and evaluation of fighter aircraft. The detachment evaluates the operational effectiveness, suitability, and mission capability of the A-10, F-16, F-15C/E/EX, F-22, and F-35 aircraft, and reports results in support of major acquisition program milestone decisions and combatant command fielding decisions.

Phil Chu is a Systems Engineering and Technical Assistance (SETA) contractor from CENTRA for the Defense Advanced Research Projects Agency (DARPA).

Developing a Scalable Data Analytics Pipeline

Patrick Rupp, Anastacia MacAllister, Jason Garrison, George Hellstern
Lockheed Martin Corporation
Fort Worth, TX
patrick.h.rupp@lmco.com, anastacia.m.macallister@lmco.com, jason.t.garrison@lmco.com, george.hellstern@lmco.com

Colonel Daniel Javorsek
Air Force Operational Test and Evaluation Center
Nellis Air Force Base, Nevada
daniel.javorsek@us.af.mil

Phil Chu
CENTRA
Arlington, VA
philip.chu.ctr@darpa.mil

INTRODUCTION

The DoD's need for detailed battlefield awareness is increasing year over year as systems become more complex to help maintain an edge over adversaries. In addition, Dana Deasy, DoD's former Chief Information Officer, says that the DoD does not lack data, rather its challenged to discover, access, retrieve, analyze, and share all the sources of data whenever and wherever our warfighter needs it (IBM, 2018). This focus on awareness is driven by the need for strategic insight into how to manage future conflicts. In fact, the department of the Navy states that "We are in a global competition to gain better insight and foresight from data" (Department of the Navy, 2017). Unfortunately, much of the data collected resides on many heterogenous systems. These disparate sources make collecting, managing, and utilizing the data difficult because data structures often are unique to generating systems. This problem is because holistic data management and data governance are topics often overlooked when implementing advanced analytics and machine learning (ML) strategies in organizations. However, these topics are crucial to successful ML and analytics implementation. To facilitate this goal, the Navy, Army, and Air Force have their own systems used to explicitly define the naming conventions and standards for various data elements. This focus shows that the DoD recognizes it's need for scalable tools plus techniques that consolidate data for storage and analytics. Unfortunately, having a holistic data strategy implemented in the near term is not feasible. As a result, ways of managing disparate sources of information for analytics and ML development are needed.

One of the largest problems to mitigate lies in existing methods for ingesting data from various sources and even different versions of the same source. That market alone contains many contractors developing their own unique systems with unique sensor tracking data structures. These types of unique system outputs are also encountered in the $2.4 Billion world of modeling and simulation (Harper, 2019). Simulation tools have become a ubiquitous part of the military analysis landscape. They allow for rapid generation and economical access to vast amounts of data on battlefield scenarios, providing warfighters an edge through careful analysis of scenario outcome data. Unfortunately, the problem remains that there are limited data ingestion tools and methodologies which allow for flexibility and scalability between different data structures.

While several strategies have been developed by industry and the DoD to combat this unique data structuring problem, these efforts typically fall short either in flexibility or scalability. The data pipeline detailed in this work aims to answer the problem of extensibility in both ingesting data as well as storing data without sacrificing scalability. The design demonstrates how a unique system's data can be given in its raw form, passed through a guided data processing pipeline, and transformed into a common form for scalable storage into a data warehouse model. While most data warehousing methods focus on the implementation of the data within the data warehouse tool, this work instead focuses on a novel ingestion pipeline design from ingestion to analysis. This design allows ingesting different raw data sources into a data warehouse that is independent of specific data warehouse tools. The level of abstraction provided by the ingestion and processing protects the pipeline from changes in data storage tools as well as from new data formats. Ultimately, these design decisions allow for a change tolerant data pipeline, decreasing cost to maintain and analysis time for warfighters. The cost efficiency is derived from reduced time needed for architecting unique loading schemes for each data source and storage target. This paper focused on the design, implementation, and facilitation of the technology. With the pipeline now implemented, future work will begin to look at quantifying how

the strategies developed can help save time and resources. The paper begins with a walkthrough of classic data storage tools, methodologies, and limitations. From there, the paper details the developed novel ingestion pipeline architecture and its components. Then a case study analysis is presented using the architecture in practice. Analysis focuses on describing how the developed solution performed in terms of scalability, flexibility, and reusability.

BACKGROUND

According to a 2019 article, the US Army planned on spending $8.4 billion for network modernization to help enhance today's battlefield awareness capabilities (Pomerleau & Gruss, 2019). With these and other DoD modernization efforts, the amount of data is growing exponentially. Because of this, Data Management is critically important for supporting battlefield awareness, especially with the tactical emphasis for the DoD to shift to the interoperability between new and legacy systems, data mining, aggregation, standardization, and reusability (Ceruti, 2003). Data pipeline tools and techniques used to facilitate this awareness have continuously evolved from their inception to accommodate the large growth in data size as well as its complexity. However, even with the advancements in technology, problems associated with scalability, data set size, and change tolerance still exist. To understand the current state of data management one must begin by looking at how data management has evolved and how that has shaped today's tools. The following Background sections walk through the evolution of several key facilitating data management technologies and how they impact today's data strategies.

Progression of Data Storage Systems and OLTP

One of the earliest forms of data management systems was developed in 1960 by Charles W. Bachman. The Integrated Database System, which it was called, primary utilized random access memory (Foote, 2017). Since then, many database management systems (DBMS) have been developed and refined for the purposes of fast data processing and information retrieval. One instance of this is the proposal for "A Relational model of Data for Large Shared Data Banks" published by Edgar F. Codd (Codd, 1970). This method demonstrated that information can be stored in an easy to manage tabular method, connected by links such as keys, rather than just the content of the data alone. Due to their popularity and ease of use, by the 1980's, the standard query language (SQL) and relational database management systems (RDBMS) had become common place in industry. However, with the move toward distributed internet based systems in the 1990's the client-server model became the norm. This led to the rise of Online Transactional Processing (OLTP) as databases became a commonplace method for data storage and retrieval. Ultimately, the progression shows that as needs evolved, capabilities shifted towards new and novel methods of data handling. Unfortunately, these existing OLTP methodologies were not capable of solving the foundational analytics and intelligence needs arising in the age of big data.

Growth of Data and Rise of Data Warehouses and OLAP

With the exponential rise of data and the requirement for scalable methods, the need for Online Analytic Processing (OLAP) and enterprise level data storage grew. Enter the data warehouse. The data warehouse is aligned with the typical DBMS, except its usage is quite different under information storage and retrieval applications. The data warehouse focuses on fewer, larger transactions that are often run as batch processing. This focus on large batch processing allows the method to scale, better allowing it to handle big data. Often data warehousing approaches might include migration of large data from external sources into the data warehouse for information collection, intelligence, and reporting. Data warehousing methods have been implemented within Army Installations as a means "to reach across all the activities and functions on the installation to provide effective leadership and management" (Reddy & Schroeder, 1999).

With the usage model for OLAP driving towards data collection and intelligence, data warehouse tools became optimized for data writes, aggregations, summarizations, complex queries, and data manipulation. These design changes enhanced the ability of data consumers to manage and manipulate their data in ways previously too inefficient to attempt. This evolution in capability provided them a competitive edge against their competitors. This edge is recognized by the US Air Force who is currently investing $100 million in the technology (Morgan, 2021) . However, emerging unstructured data formats are breaking away from current data management methods. Even within the DoD, unstructured data is growing, and new methods are needed to navigate these changes beyond the data warehouse. With the DoD's current objectives for heterogenous system communication, data pipeline solutions that can store data to

scale for analysis is the logical next step in value generation for information consolidation. Having all the information from heterogenous systems stored together with a common analytics pipeline provides enhanced intelligence and provides an advantage to our warfighter.

Growth of Unstructured Data Methods & Tools

Along with the data explosion also came flexible data structures often referred to as unstructured data. A particular challenge identified by the research brief on DoD's use of data analytics is the collection and use of unstructured data (Anton, et al., 2019). According to Merrill Lynch, unstructured data makes up more than 85% of all business information (Abdullah & Ahmad, 2013). Unstructured data, data which is not neatly defined by fixed fields in each record, can take on any form such as markup languages, text, and binaries. This flexibility in data structure is optimal for applications with variable structures and changing content. The answer to this problem was the development of systems that store data in an unstructured or raw form. Databases that store unstructured data are often called NoSQL referring to "Not Only SQL".

NoSQL databases are optimized for flexibility, storing data in structured, semi-structured or unstructured forms. Tools like Hadoop have been developed to act as a backbone for storing and manipulating this type of raw data. Such tools were an answer to the challenge of managing and storing complex data shapes since the processing of data into a usable form is often more challenging than storage itself. Unfortunately, these tools alone are not enough to solve the problem of data management. These tools could store data in their raw form, but heterogenous unstructured data must still be processed for scalable analytics purposes. Unstructured data store tools alone are not enough to satisfy the need for scalable heterogenous data pipelines. To create a solution that both stores and processes data, one must look at work combining these different components. Only by effectively combining both data storage and processing can a holistic pipeline be created to provide the warfighter with actionable battlefield insights.

Data Processing & Pipelines

While critical, database and storage tools alone do not provide a holistic solution. Rather, they are just one component within a data pipeline. Another critical part of the pipeline is data processing. Data processing components exist in one or multiple places within the pipeline to ensure data is usable. Two high level data pipeline processes are Extract, Transform, Load (ETL) and Extract, Load, Transform (ELT). ETL is commonly associated with the data warehousing model where data is extracted from its source system, transformed into a new form, and loaded into the target system such that the data in the source system is different than the target system. The second option, ELT, is commonly associated with the data lake model where data is extracted from its source system, loaded into the target system in its raw form, and lastly transformed only upon request. Pipelines are unique in that they utilize a mixture of these methods depending on the tools chosen and the methods required to transform the data from its raw into its finalized form. In some cases, this might mean that data is migrated through multiple systems before it is ready for consumption. Many one-off architectures and tools exist to solve aspects of data pipeline complexities, but many of these tools and frameworks fail to appropriately address the data pipeline flexibility and scalability issue. For example, Data mapping tools often do not support unstructured data mapping, ingestion pipelines are often built unique for each data source with little code re-use, and unstructured data storage tools collect heterogenous data but do not guarantee common analytics processing for all data stored. Our warfighter is in need of a solution which can ingest unstructured data from many sources and perform common data processing for re-usable analytics methodologies which provides advantageous intelligence to our warfighter.

Ultimately, with all the varying components and strategies, data pipelines are as much of an art as they are a science, particularly in the realm of processing data within the DoD for battlefield awareness. There have been programs and tools developed to provide a solution to the data disparity problem. Unfortunately, these methods lack the ability to fully solve the problem of collecting disparate data in a scalable and extensible manner for data ingestion and analytics. Some programs have been developed to handle extensibility in disparate data mapping but lack the ability to scale their data management for analytics. Other programs have developed methods to handle scalable data mapping and management but are severely limited in the types of data structures that can be mapped. Currently, most data mapping tools are restricted to relational data structures which excludes unstructured formats like log files, markup language files, binaries, and many other non-relational data. The method described in this work details a data pipeline that does not have the restrictions of the existing tools while still providing the ability to scale. This design will help take steps

towards developing strategies for ingesting disparate data, mapping to common storage forms, and scaling to meet DoD needs for data management regarding battlefield analysis.

METHODS

In the DoD today, there exists many heterogenous systems with a limited ability to share information. While there are many tools and methods to help with system communication and data mapping, the problem of a holistic extensible data pipeline architecture remains. Demonstrating the need for such a strategy, the department of the Navy identified a common data management framework as critical component to improving readiness to deploy innovations (Department of the Navy, 2017). The goal of this work is to help take steps towards developing a flexible and scalable data management pipeline to produce battlefield analytics. Specifically, this section walks through various aspects of the architecture such as the data ingestion, data warehouse design, analytics views, and the chosen tools for the use-cases detailed in this work.

Data Pipeline Architecture

Data Pipelines are usually a mixture of data handling components developed to solve a particular problem in a unique environment. The two main storage components that exist for analytical data management are data lakes and data warehouses. Data lakes emphasize their ability to store raw data. The positive with data lakes is that one does not need to transform the data to be stored, but the downside is that analytical methods may have to change with the different data that is stored by the system. Data warehousing, on other hand, requires predefined structured data for storage, but the analytics methods are scalable because all the data is in the expected form. The developed approach, as shown in Figure 1, makes use of both by initially storing the raw data in the data lake, ingesting the data, transforming the data to the expected structure, and storing the data within the data warehouse. From there, the data can be pulled into the analytics engine for processing and deriving insights. Combining the data lake and data warehouse concepts allows the pipeline to take advantage of the data lake's ability to handle the increase in unstructured data but provides the data set predictability and structure that analytics methods require.

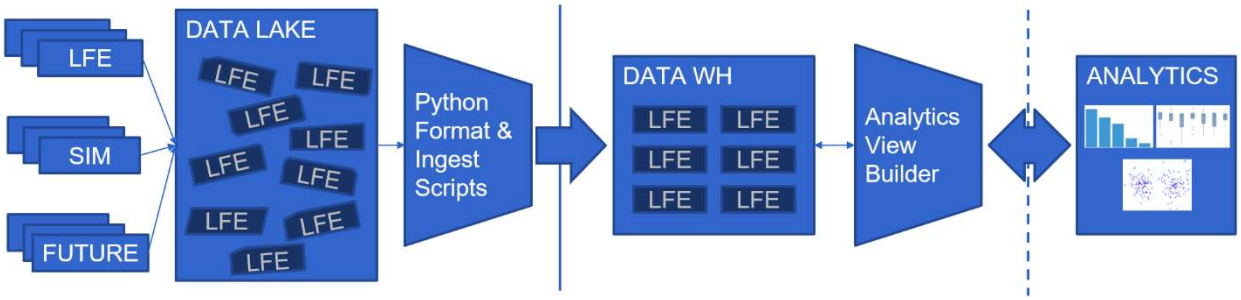

Figure 1: Data Pipeline Architecture

While similar high-level architectures exist, the novelty in the design lies within the sub-components. These sub-components are designed to handle changes in the data sources and the data warehouse tool without disrupting the data pipeline. This cushion is a necessity for managing data pipelines in the DoD because data sources produce heterogenous structures in complex environments that may require specific data warehouse tools. This approach is dually valid because most of the heterogenous systems within the DoD contain common data elements making the data warehousing method feasible. The DoD's data strategy even identifies the need for syntactic interoperability using common data formats (Department of Defense, 2020). Table 1 describes the components and their roles within the ingestion process. The design allows for the user to define how the raw data will be identified by the *Directory Crawler*, parsed by the *File Readers*, transformed by the *Data Processor,* and mapped to the expected form in the *Common Data Loader*. While each data type will require slightly different processing in the front end, once it makes it into the data warehouse it becomes standardized, allowing the analytics to be insulated from any changes. In addition, the method provides a common pipeline architecture decreasing maintainability cost, which is necessary when the number of sources scale up. The following sections will provide further detail on the components design and how this approach could ensure that the warfighter retains the competitive edge.

Table 1: Data Ingestion Component Descriptions

Component	Type	Description
Directory Crawler	Module	Locates paths to all data files within a data directory and creates a File Container
File Container	Container	Contains all data file paths associated with each run
File Readers	Module	Parses the expected data files and creates a Data Container for each run
Data Container	Container	Contains raw data from data files for an individual run
Data Processor	Module	Transforms data into expected form and maps fields to the Common Data Loader
Common Data Loader	Module	Prepares data to be pushed into the chosen data warehouse

Raw Data Ingestion

The first component of the data pipeline deals with data ingestion. Ingestion of data from many heterogenous systems with unique data structures is a necessity for scaling a data pipeline for advanced analytics on battlefield awareness. Typical data warehousing approaches load data from multiple sources into a central repository, but these approaches often rely on feature mapping tools or unique pipelines for every source. Mapping tools are mostly limited to mapping structured data and do not perform well with unstructured data. The unique data pipelines approach, on the other hand, is difficult to maintain since there is no commonality across the various pipelines. Instead, a solution is introduced that embraces unique data pipelines while adhering to a structured common framework. The strategy depicted in Figure 2 demonstrates how each data source will have its own pipe while maintaining commonality. While unique, the components within the ingestion pipeline are consistent. This design makes maintaining the pipelines simple since each pipeline has the same components and data flow. This solution attempts to ease the difficulty managing the pipelines when the number of sources such as text, log or image files scale up. Now that the ingestion process has been detailed, another important consideration will be discussed, which is how much data is ingested.

An important data ingestion design consideration is the batch processing of each run's data. The data is assumed to be exported from the source systems and will be loaded via batch processing rather than live streams. This assumption is valid because the current battlefield post processing initiatives require collecting data from many different isolated and restrictive environments where live connections cannot exist. Loading the data in a batch processing method is also a common practice within OLAP processing which is optimal for data warehousing approaches (DW4U, 2010).

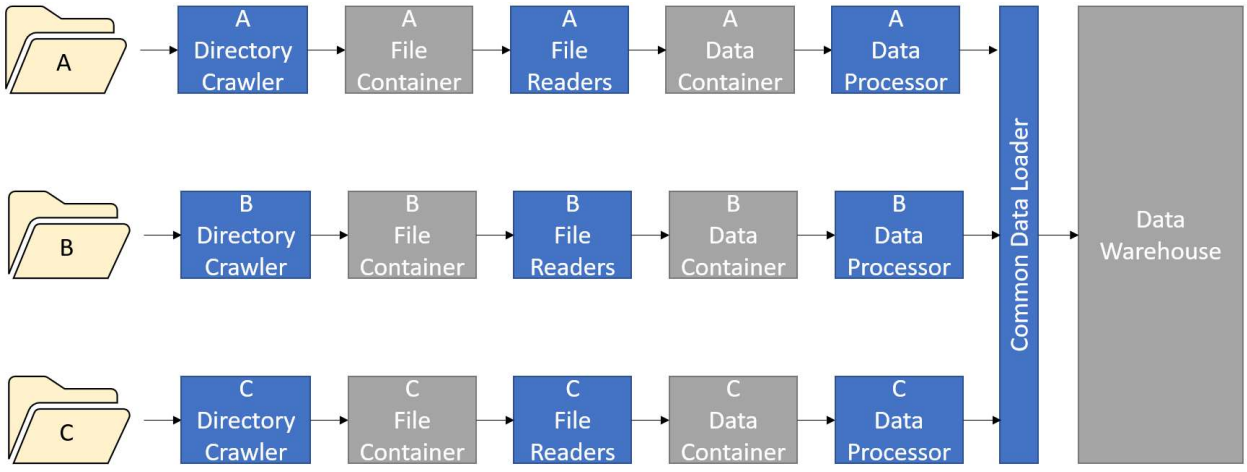

Figure 2: Data Ingestion Components

Additionally, data will be generated in isolated runs where each run contains information for all the platforms and their actions in a given event. This is true whether the data source is a live flight event or a simulation. Likewise, the data will be loaded in the same fashion where each run will be parsed, transformed, and pushed into the data warehouse as an isolated process. The design consideration of loading each run as isolated process has two primary benefits. First, each run's data can be processed in isolation meaning it is a good candidate for parallel data ingestion and processing which ensures faster loading speeds. Second, each run's data should be pushed into the data warehouse together as one transaction. This ensures that if successful, all the run's data is stored into the data warehouse, or if

not successful, none of the data is stored which prevents ill-formed data from muddying the analytics. Using this framework, these benefits span to all the data pipelines ensuring fast and reliable loading of disparate data into the data warehouse and faster insights for battlefield awareness. With the considerations and benefits associated with raw data ingestion identified, the downstream components within the data pipeline will now be explained.

Data Warehouse Design

Having covered extensibility in the data ingestion process, this section will cover extensibility in the data warehouse itself. Data pipelines are often designed around specific data warehouse tools that an organization is currently using or is intending to use. When organizations make the change from one data warehouse tool to another, many of the data pipelines must be redesigned and modified to fit the new tool. The DoD adds an additional layer of complexity by enforcing environmental requirements which allow only certain tools due to program restrictions or classifications. Since operating in the DoD requires working in isolated compute environments, one is not guaranteed that they can utilize the same data warehouse tools migrating from one environment to another. To solve this problem, the architecture provides extensibility in the data warehouse tool itself as shown in Figure 3. The solution does this by adding an additional management module layer between the *Common Data Loader* and the specific data warehouse being used. Once a particular data warehouse tool has been implemented with the specified storage configuration, the user only has the specify which data warehouse to push the data to rather than rebuilding the existing pipelines for a particular tool. This extensibility is not only optimal for migrating environments but also for scalability. As shown in the results section, one can use this design to prototype their data analytics pipeline using small scale tools before switching to larger data warehouse tools without large amounts of rework. This extensibility in the data warehouse solves the problem of migrating environments and scaling up tools to prevent downtime in providing insight to warfighters. Future work using this design framework may explore development and maintainability time savings.

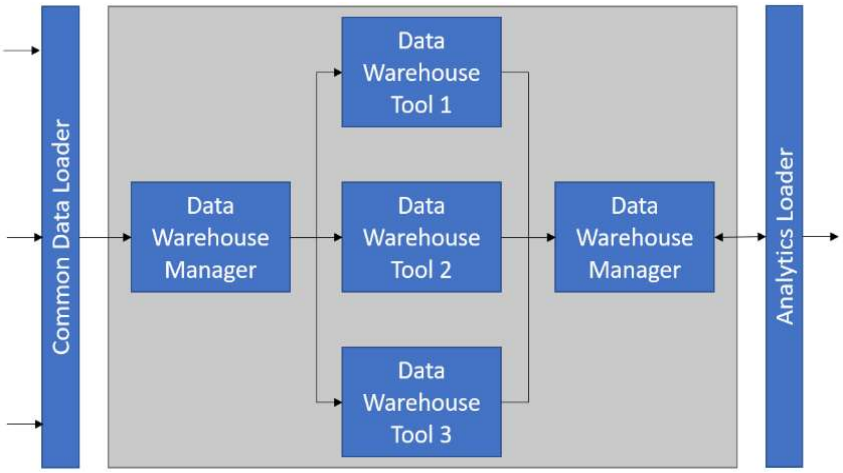

Figure 3: Data Warehouse Manager

Data Warehouse Analytics Views

At this point, extensibility has been identified in both the data ingestion process and data warehouse. However, the solution intends to take the extensibility a step further into the data preparation for analytics. This is important because the analytics process in the DoD today often entails many 'what if' analyses. These analyses are developed with unique processing that does not promote reusability and is often left alone and forgotten. This problem implies that every analysis requires its own unique processing instead of utilizing reusable processing in the data preparation. The answer to this problem is the introduction of a reusable metrics generation process. This reusable process is valid because it takes advantage of the common data elements and common metrics used within the battlespace. Some metrics are calculated at the run level such as weapon utilization (eg. % hit target, % missed target) and force selection metrics (eg. % of blue stealth fighters killed) while other metrics are at lower levels. At the different levels of analysis, there may be many metrics that are constant across scenarios like Defensive Counter Air (DCA), Offensive Counter Air (OCA), Suppression of Enemy Air Defense (SEAD), and more. Reusing existing metrics generation can provide faster analytical insight for battlefield awareness, and the solution solves this problem using a hierarchy of metric views.

Data warehouses are typically designed for structured data which commonly support SQL for data processing. Most of these tools also can create views which are stored queries that can be used similarly to data tables. The difference is that views create the data on request whereas tables pull data already stored in the system. Since a view is a stored query, it can also pull from other views. This capability is important because it allows us to create a hierarchy of views

to generate unique scenario data sets that can pull rely on common metrics generation. For instance, Figure 4 depicts how common weapon metrics can be pulled for various scenario data set metrics like DCA or OCA. This allows users to not have to develop metrics from scratch for each scenario as is normally done in today's 'what if' analyses. Instead, users can pull from the existing common metrics and only develop new scenario specific metrics. This promotes faster turn-around time for new scenario data preparation and analysis which is critical to ensure that warfighters maintain a competitive edge in battlefield awareness over adversaries.

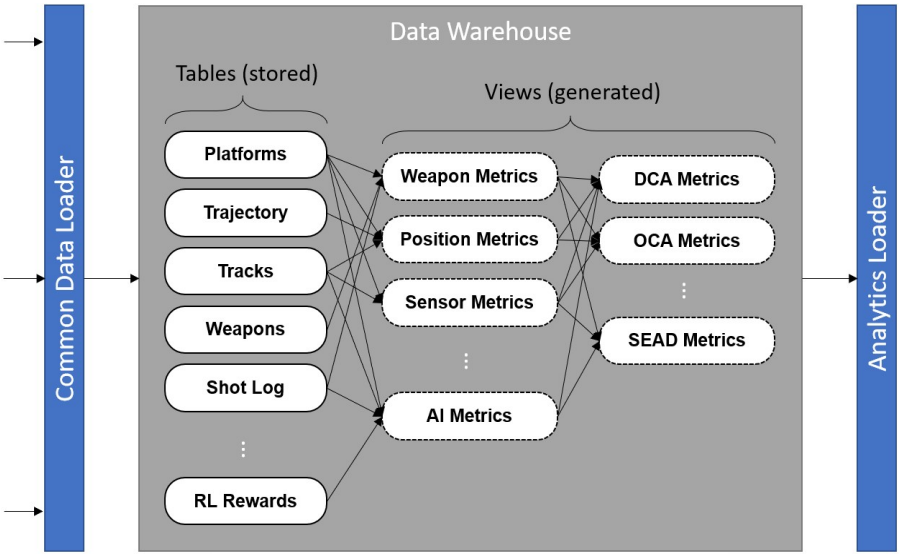

Figure 4: Metric Views Hierarchy

Pipeline Implementation

The architecture shows extensibility in data ingestion, data warehouse tooling, and the metrics generation. As a template, the architecture is not limited to any programming languages or tools. For the purposes of this work, the ingestion process prototype was developed using the Python programming language, and the data warehouse prototype was developed using the SQLite database. As common tools, python has many libraries that support data management such as "pandas" and "sqlite3" library. Additionally, SQLite has most of the functionality that larger database tools would have for data storage and manipulation. Both are optimal for their flexibility and industry support which is necessary to ensure that data can be transformed from its raw form and stored in the expected form within the data warehouse. In the following section, the author will walk through the implementation of these tools across multiple use-cases as the prototyped was developed and demonstrated its ability to solve the problems that are currently facing today's scalable data analytics pipelines within the DoD for battlefield awareness.

RESULTS

The challenge with the DoD's battlefield awareness data management is that there are many heterogenous systems with unique data structures that are generated and stored within restrictive compute environments. These complexities make it difficult to scale data management strategies, but the architecture in this paper attempts to provide a solution that can help solve these problems. This section will walk through the prototype's implementation over five use-cases shown in Table 2. These use-cases start with 20 runs of a human-in-the-loop dogfight scenario with semi-structured data to over 5,100 runs of a complex DCA scenario with over 143GB of structured and unstructured data. This section will detail how the extensible design protected the development from the changes in the source data and from changes in the data warehousing tools.

The initial use-case started as a 2D multi-fighter dogfight scenario with both sides given the objective to destroy all enemy fighters. The scenario had 12 red fighters combating 4 blue fighters where the blue side had a human-in-the-loop pilot making actions to turn or fire a missile. This data set of 20 runs was generated by a special plugin for the AFSIM tool. While this method of data generation is slow, manually intensive, and unable to scale, it does provide an optimal starting point to develop the prototype. This initial scenario generated two types of files. One file type was a semi-structured CSV which contained time-series position data with the associated enemy tracks information as a variable number of columns depending on the number of tracks for each time step. The other file type was a structured CSV containing the time-series action commands that the blue fighters were tasked by the human-in-the-loop. The ingestion process split the semi-structured data into two separate structured entities which then mapped to the table's trajectory and tracks.

Table 2: Use-case Descriptions

Use-case	Description	Data Info	Number of Runs	Data Size
1	Human in the loop 2D aerial combat w/ fighters only	Structured CSVs Semi-structured CSVs Basic Position, Tracks, and Weapon Fire data	20	1.02 GB
2	Human out of loop 3D DCA Cruise Missile Defense scenario w/ enemy bombers only	Structured CSVs Removed Semi-structured data format Include more platform and weapon data	100	2.9 GB
3	Human out of loop 3D DCA Cruise Missile Defense scenario w/ 0, 2, & 3 enemy fighters	Structured CSVs Modified CSVs with new & removed features	300	8.82 GB
4	Human out of loop 3D DCA Cruise Missile Defense scenario w/ varying numbers of enemy fighters, bombers, and missiles	Structured CSVs Inclusion of bullseye position data Modified CSV with new & drastically reduced redundant data	2,400	>11GB
5	Human out of loop 3D DCA Cruise Missile Defense scenario w/ variable weapon loadouts, large number of platforms, ground vehicles, stealth fighters, and ground based air defense systems	Structured CSVs Unstructured YAML file Unstructured text log file Modified CSVs with new & removed features	>5,100	143 GB

At this time, the data warehouse tools were still being evaluated, and the database was not yet chosen. However, this was not a problem since the data was stored simply as structured text data files before SQLite was put in place. While this approach is simplistic, the architecture made it easy to change the data storage method without major code changes in the ingestion process as it did not require code rewrites nor rework of previous developments. Additionally, the most important aspect of this initial use-case was determining what type of data must be stored in the data warehouse for the purposes of storage and analytics. The data storage design may have a large impact on the ingestion process, hence determining the data elements needed for storage is important to understand early in the process. Changes in the data storage can potentially be more expensive later in time, depending on the change. The initial use-case provided a clear picture for what information should be captured and how it should be captured for simulation battlefield awareness data management. The architecture made it easy to prototype the solution without the need for large cumbersome tools or the fear of rework due to changing tools.

Once the initial use-case was prototyped, the focus shifted to growing in both data size and complexity. This use-case focused on simulations with human-out-of-the-loop data generation still with AFSIM. The new use-case was a 3D defensive counter air (DCA) scenario with one side defending designated target(s) and the other is performing an offensive counter air (OCA) attempting to destroy the defended target(s) as shown in Figure 5.

Figure 5: DCA Scenario Example

The second use-case scenario had red bombers on the offensive launching cruise missiles at a blue land target defended by four blue fighters. This new scenario, which generated 100 runs, presented a change in the data structure as well as a change in the mission. The change in the data came from a new way positions and tracks were captured in addition to new data regarding weapons, messages, and platforms. With this new and modified data, the data warehouse design was expanded to include the new data without needed to change the existing trajectory and tracks tables. At this time, the data was completely stored within SQLite unlike the early stages of the first use-case. The new mission did not affect the data design since the underlying information for positions, tracks, and weapon utilization remained the same. This is another demonstration how scenarios and systems can change while still having common data elements. This commonality is a key factor in the design of an architecture optimized for battlefield awareness

The third use-case expanded upon the second with an introduction of red fighters to combat the defending blue fighters. The simulation varied the number of red fighters from zero, two, and three in each run for additional analytical insight. This use-case generated 300 runs where each red fighter force loadout had an even 100 runs. Compared to the previous use-cases, this one increased in data size and variability. Even through this additional complexity the data pipeline did not become disrupted. With each use-case, features are being added or removed from the data structures making them a new data source. Following the framework's design, the new source is hooked into the ingestion process and allows data to flow into the data warehouse without disrupting previous data sources or requiring rework.

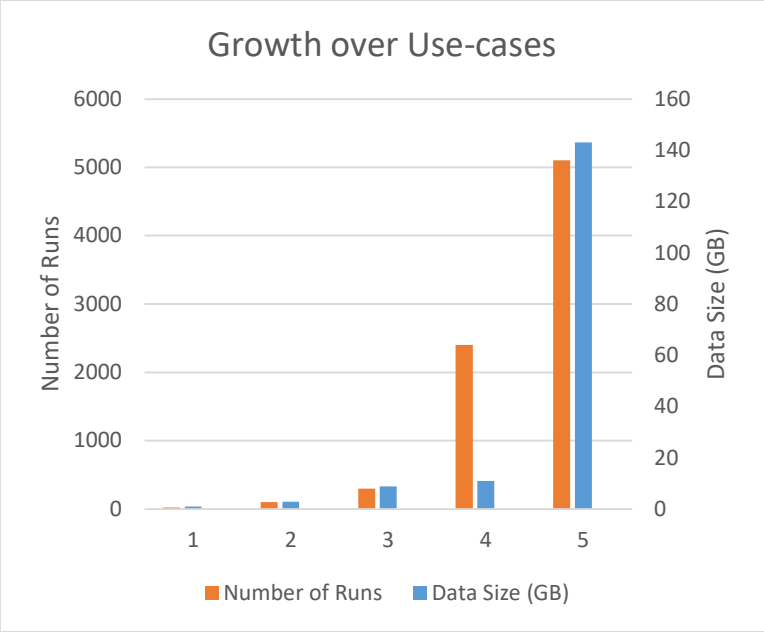

Figure 6: Number of Runs and Data Size over Use-cases

With the data pipeline well established, the concern with the size of data and number of runs required for analytics. Use-case three generated only 300 runs with nearly nine gigabytes of data. With the fear that the data would grow fast, the fourth use-case sought to remove all redundant data and reduce the amount of information output for each run. This use-case generated 2,400 runs of data but used only eleven gigabytes. Large amounts of redundant data were successfully removed from the output. Additionally, the new dataset included variability in enemy fighters, number of land attack cruise missiles (LACM), blue fighter aggressiveness, and bullseye positioning. The more variability is introduced in the scenario, the more runs are required to have a well rounded data set for analysis. This means that the data size grows very quickly with the variability, and that data reduction measures were highly important at this step. Figure 6 shows how quickly the number of runs increases with each new use-case. This need requires a data pipeline that is unaffected by scale. The architecture displayed its extensibility and scalability by having a unique data ingestion for each use-case without disrupting the existing data pipeline. Furthermore, the metrics developed during the previous use-cases allowed for easy reusability and prevented rework in generating new metrics. These benefits even carried over to the scaled up fifth use-case.

The fifth use-case expanded the DCA scenario to include stealth fighters, ground vehicles, ground based air defense systems, and new weapons (eg. Surface-to-Air missiles, Short & Medium range Air-to-Air missiles). The additions to the new platforms and weapons drastically increased the size of the trajectory and tracks data generated. For example, the maximum number of blue fighters grew from 4 in use-case four to 32 in use-case five (16 stealth and 16 non-stealth fighters). In addition to data size of nearly 143 gigabytes, the use-case had two other main objectives. The first objective made the scenario more realistic with the inclusion of ground based air defense systems, larger number of platforms, and more capable weapons. The second objective began to include unstructured data captured from a Reinforcement Learning (RL) agent trained to control the platforms. Even with the growth in data, increased scenario complexity, and inclusion of unstructured data, the architecture's extensible design prevented disruptions in the data pipeline and promoted scaling up from a small prototype to a maturing solution. This extensibility is even shown in the metrics generation where common metrics were generated from reusable SQL views.

The growth of data and the changes that the data went through did not hamper the metrics generation. While each use-case had their own nuanced data structures, the ingestion mechanism for each use-case parsed the data and mapped the fields to the common data format. Storing the data in this common data format allowed the view hierarchy to remain constant across use-cases 2 through 5. This is important to avoid rework when adding new data sources. Solutions which store the raw data within the data warehouse can scale in storage but cannot scale with metrics and analytics. This solution forces the data mapping in the ingestion component to a common data format and allows the metrics generation to remain constant. Reducing rework and unnecessary duplicative development allows developers to build new metrics faster and provide quicker insight to our warfighter.

Throughout the five use-case, the system has been exposed to structured, semi-structured, and unstructured data. However, most of the data lived as structured CSV files which are easy to map than unstructured data. The introduction of the YAML and Log files in use-case 5 began to stress test the solution to ensure that unstructured data can be processed into a form appropriate for data ingestion and storage in the common data format. Future use-case will shift away from the CSV data captures to raw text file event logs which contain the weapon utilization data in higher fidelity. The abstraction from the data source and the data warehouse tool allows the developer a cushion from changing sources and environment restrictions which would normally create rework. The extensible architecture handled changing data sources, fast growing data, and unstructured data with ease allowing the developer to focus on the data management and delivering fast, reliable insights into the scenario performance. This in turn can help our warfighter with advanced battlefield awareness necessary to maintain an advantage over our adversaries.

CONCLUSION AND FUTURE WORK

With the current fast pace of change and data driven strategies in response to our adversaries' push for AI-enabled defense, the DoD must develop detailed battlefield awareness to maintain the competitive edge over adversaries. Unfortunately, with the increase in data size and system complexity, existing data management strategies for advanced analytics and machine learning fail to provide a flexible and scalable solution within the restrictive DoD compute environment. This paper introduces a flexible data pipeline architecture designed to ingest many heterogenous data structures to be transformed, mapped, and stored within the data warehouse for a scalable analytics solution. The abstracted design demonstrated the extensibility in the system when it was applied to the simulation of 2D dogfights and 3D defensive counter air scenarios. In these five use-cases, the number of runs increased from 20 with a human-in-the-loop commander to over 5,100 runs on a defensive counter air scenario producing over 143 gigabytes worth of data. Not only did the data grow exponentially, but the data structure, format, and file types changed throughout this process multiple times. With each change, the design's extensibility allowed for the easy development of a new ingestion pipe to the common data loader. Making use of the common data elements within many of these military simulations, the data was able to be stored to utilize scalable metrics generation. The metrics generation is more than just scalable, it is even designed with reusability in mind. The hierarchy of metrics generation pushes the logic to the right level, allowing new scenarios to make use of common metrics that may span across multiple scenarios. This design allows for faster turnaround for new scenario analytics reports, and insights into whether the new scenarios are effective. Reducing the amount of time spent redeveloping metrics helps increase the speed of decision making and ensuring that warfighters are prepared to run their missions successfully.

The next steps for this work will focus on developing and improving the data access extensibility to the analytics engine. The scope of this paper was to develop the ingestion pipeline to the data warehouse for metrics generation. Future development will allow the analytics engine to access the data without disruption from changes in the data storage tools nor in the raw data structures. Additionally, methods for integrating an AI agent's meta data, training, and performance into the scalable data analytics solution will be researched and prototyped. The stance of the DoD to embrace AI is critical for the future of warfighting. In fact, the DoD recognizes the need for ML and analytics so much that in 2018, DARPA announced a $2 billion investment in accelerating AI integration into U.S. warfare platforms (Whaley, 2019). Lastly, the solution will be further tested in scaling up to larger data storage tools, more data runs, more complex battle scenarios, and even analyzing live flight data instead of just simulation. These additional capabilities and considerations will create a matured solution that can be used to meet the unique challenges faced with data management for battlefield awareness within the DoD and help maintain a competitive edge over adversaries.

REFERENCES

Abdullah, M. F., & Ahmad, K. (2013). The Mapping Process of Unstructured Data to Structured Data. *Research and Innovation in Information Systems (ICRIIS), 2013 International*, (pp. 4722-4726). Bangi. doi:10.1109/ICRIIS.2013.6716700

Anton, P. S., McKernan, M., Munson, K., Kallimani, J. G., Levedahl, A., Blickstein, I., . . . Newberry, S. (2019). *Assessing the Use of Data Analytics in Department of Defense Acquisition.* Santa Monica: RAND Corporation. doi:https://doi.org/10.7249/RB10085

Ceruti, M. G. (2003, October 1). Data Management Challenges and Development for Military Information Systems. *IEEE Transactions on Knowledge and Data Engineering, 15*, 1059-1068. Retrieved 04 18, 2021, from https://ieeexplore.ieee.org/abstract/document/1232263

Codd, E. F. (1970). *A Relational model of Data for Large Shared Data Banks.* Retrieved from https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf

Department of Defense. (2020). *DoD Data Strategy.* Retrieved April 20, 2021, from https://media.defense.gov/2020/Oct/08/2002514180/-1/-1/0/DOD-DATA-STRATEGY.PDF

Department of the Navy. (2017). *Strategy for Data and Analytics Optimization.* Retrieved April 20, 2021, from https://www.doncio.navy.mil/exports.aspx?id=10808

DW4U. (2010). *OLTP vs. OLAP*. Retrieved April 18, 2021, from Datawarehouse4u.info: https://www.datawarehouse4u.info/OLTP-vs-OLAP.html

Foote, K. D. (2017, March 23). *A Brief History of Database Management*. Retrieved from Dataversity: https://www.dataversity.net/brief-history-database-management/

Harper, J. (2019, December 2). *Army Spending Big on Training, Modeling, Simulation*. Retrieved April 18, 2021, from National Defense Magazine: https://www.nationaldefensemagazine.org/articles/2019/12/2/army-spending-big-on-training-modeling-simulation

IBM. (2018). *Data and Defense: Hot to Boost Readiness.* Retrieved April 20, 2021, from https://www.govloop.com/resources/data-defense-boost-readiness/

Morgan, T. P. (2021, February 4). *US Air Force Spends $100 Million To Accelerate Data Warehouses*. Retrieved April 18, 2021, from The Next Platform: https://www.nextplatform.com/2021/02/04/us-air-force-spends-100-million-to-accelerate-data-warehouses/

Pomerleau, M., & Gruss, M. (2019, April 18). *Army budget request adds $1.5B for network modernization*. Retrieved April 18, 2021, from C4ISRNet: https://www.c4isrnet.com/it-networks/2019/04/18/army-budget-request-adds-15b-for-network-modernization/

Reddy, P. V., & Schroeder, C. G. (1999). *Data Warehouse Architecture for Army Installations.* Alexandria: U.S. Army Corps of Engineers.

Whaley, R. (2019, August 09). *The big data battlefield*. Retrieved April 20, 2021, from Military Embedded Systems: https://militaryembedded.com/ai/big-data/the-big-data-battlefield

