Necessity is the Digital Mother of Invention
Developing an IoT-style digital engine in a low-connectivity environment

Lt Cdr L Talbot RN* CEng MSc BSc MIMarEST

** Royal Navy, UK*
* Corresponding Author. Email: liam.talbot103@mod.gov.uk

Synopsis

This paper explores the challenges and solutions in developing an Internet of Things (IoT)-style digital engine replica in a low-connectivity environment, specifically within the Royal Navy (RN). It underscores the inherent limitations of continuous connectivity in naval operations, where communication links can be disrupted or intentionally severed. Whilst alternative solutions to this problem exist, utilising data management platforms such as the Palantir Foundry Software Programme allows an increase in the dataset by capturing data not from one digital twin but from multiple, a strategy termed the ‘digital mother’. By leveraging the collective data from multiple engines, the dataset’s fidelity can be significantly enhanced, nearing the desired level of accuracy with no requirement for additional hardware. Using a lean start up methodology, the paper outlines the fundamental hypothesis and value proposition of the digital mother before demonstrating a methodical approach to building a Minimum Viable Product. Finally, it highlights the unique approach of empowering the end-user to create their own agile digital tools tailored to their specific needs. With training in data engineering, this approach ensures the solutions are not only technically robust but also intuitively designed to meet the demands of front-line RN engineers. The outcome aligns with the ‘Lean, Green, Mean’ approach. In terms of Lean principles, leveraging an organic workforce asset reduces development costs. In addition, the product enhances the quality of decision-making and streamlines maintenance schedules, providing savings in both costs and maintainer resource. From a Green perspective, using predictive analytics to manage inventory and contractor support facilitates a reduction in the engine’s environmental footprint. As for the Mean aspect, employing intelligent maintenance increases availability, maximising resource so the RN can continue to fight and win.

Keywords: Digital replica; Data; Internet of Things; Engine Digital Twin

1. Introduction: The lemon

In 1601 Captain James Lancaster commanded the first East India Company deployment to the Far East. During this trip he performed an experimental study of the effects of citrus fruits on scurvy by providing one of his four ships with a daily dose of lemon juice. By the time the fleet had rounded the Cape of Good Hope, 110 of 278 sailors had died of scurvy on the other three ships. Everyone survived on the ship given lemon juice (Bowen et al., 2012). Lancaster was knighted on his return, and on presenting his report to the Admiralty, his findings of citrus fruits being an effective deterrent to scurvy were only embodied in doctrine in 1795 (Tröhler, 2003).

Whilst demonstrating a staggering 194-year adoption rate of new technologies (Syed, 2015), this anecdote also highlights how the unique operating environment of the Navy forced the requirement of a novel approach to a problem, and the generation of a unique solution. In that instance, the problem was the Royal Navy (RN) taking men into a new frontier where they were dying *en masse* from an unknown illness. That forced the novel approach of experimentation, by a willing amateur, and the generation of a unique solution in providing citrus fruit as an effective antiscorbutic. It also highlights that Captain Lancaster was the end-user, not a health consultant. He intuitively understood the environment that the problem was born from, after all, he was at risk of dying from scurvy himself. He also had the motivation in achieving the desired outcome – not dying of scurvy. Therefore, there was no better person to produce a solution than someone who understood the problem innately and had a vested interest in its success.

With the advance of science, we now know that Vitamin C deficiency is the cause of Scurvy (Szent-Györgyi, 1932), but the humble lemon was the 17th Century RN’s best endeavour to get the required result. A solution that could be iteratively improved through empirical testing. A trained physician may have raised an eyebrow at the simplicity of Captain Lancaster’s approach, but the key context is that this work was not done by trained physicians. This work was done by RN sailors. The customer was the consultant, creating the tools that they needed.

Author’s Biography
Lt Cdr Liam Talbot is a RN Marine Engineer and IMarEST CEng. Educated in Physics & Mathematics, with post-graduate studies in Engineering & Management. Deployed globally, maintaining Ship’s Mechanical & Electrical systems. Shore assignments include Personnel & Training, Navy HQ, Integrated Logistics Support, before developing an interest in data engineering where he recently started to navigate the realms of SQL, JavaScript and Python, making up for any natural talent with a boundless enthusiasm for technology.

 https://doi.org/10.24868/11211

2. Problem – Internet of Things in a high latency environment

423 years after the first East India deployment, and the latest problem for the RN would be unrecognisable to Sir James. It is the Navy taking an engine into a low-connectivity environment when an Internet of Things (IoT)-style datalink, and real-time digital replica, would be beneficial (Rijsdijk et al., 2020). Bandwidth limitations, outdated and fragmented IT architecture, and intermittent connectivity during military operations impede the real-time transmission of data from engine sensors. In addition, integrating this data with supplementary information such as inventory and maintenance schedules is essential to paint a complete picture. Currently, the RN relies on standalone systems, with periodic replication shoreside, to ensure vessel safety and efficiency even in the absence of continuous data exchange with shore-based infrastructure. Although an element of resolution can be found in Edge Computing technologies (Morariu et al., 2021), this inherent limitation precludes the RN from achieving a true IoT-style digital replica.

2.1. Fundamental hypothesis

From this identified problem, the four steps of the Customer Development Model (Blank, 2003) are used to develop a potential product strategy:

1. Customer Discovery - Assuming low-fidelity datasets lead to inefficient and ineffective through-life support, the Equipment Authority (EA) would benefit from an IoT-style digital replica.
2. Customer Validation – Implementing a Minimum Viable Product initiates a Build-Measure-Learn loop (Ries, 2011), enabling the validation of the perceived value of a digital replica.
3. Customer Creation – Expanding to additional users of the equipment, and scaling this model to other equipment grows the customer base.
4. Company Building – Transitioning to Business as Usual, and ‘crossing the chasm’ (Moore, 2014) from early adopters to an established customer base.

The efficacy of using the end-user to develop this product is that the first two steps are innate. If the end-user is building the product, then they already understand their problem and what they need to fix it. This expediency allows a fundamental product hypothesis (Ries, 2011) to be generated quickly:

‘The implementation of an IoT-style digital engine replica enhances decision-making quality, optimises maintenance scheduling and inventory management, increases availability and yields savings in maintainer resources, costs, and environmental footprint.’

3. Solution – The Internet of Lemons

The RN recently undertook a large-scale expansion into comprehensive data management platforms through the implementation of the Palantir Foundry Software Programme. Foundry serves as a versatile software development environment with robust back-end capabilities, allowing for the integration of data streams from diverse sources. Datasets that were once isolated on different systems are now accessible within the same software programme, enabling relational links to be investigated and trend analysis performed. The true strength of Foundry, however, lies in its Low-Code/No-Code development framework (Richardson, 2014), meaning users can create their own workflows and applications to manage their data with minimal or no coding skills. In addition, the keen amateur can delve deeper and by learning the basics of programming languages such as Python, JavaScript, and Subject Query Language (SQL), can extend beyond the in-built functionality and start to code their own custom functions. A software engineer may raise an eyebrow at the simplicity of these applications, but the key context is that this work is not being done by software engineers its being done by RN sailors. Just as Sir James was best placed to pursue his scurvy problem, the 21st Century RN engineer is best placed to pursue their low-connectivity problem.

3.1. Engine and its data

Concurrent to developments in data engineering, the Type-45 Destroyer Propulsion Improvement Plan (PIP) and Type-23 Frigate Power Generation Machinery Control and Surveillance Update (PGMU) develop at pace, with the ships’ existing diesel generators being replaced with the Rolls Royce (RR) MTU4000 Series. The confluence of Foundry alongside the introduction of the MTU4000, the relatively small equipment pool size making this a manageable task, and their increasing numbers and importance to the RN, make the MTU4000 an ideal candidate for digital replication on Foundry.

 https://doi.org/10.24868/11211

Previous endeavours by other navies and industry partners have also encountered the challenge of low connectivity in the pursuit of digital replication of an engine. The Royal Netherlands Navy (RNLN) attributes data-pipeline '*air gaps*', where human action is required to manually move data between networks, to '*limited bandwidth available, the classification and security of data and a fragmented IT architecture.*' (Tiddens et al., 2020). Rather than treat this as a blocker, the RNLN method is to build a semi-automatic data-pipeline, and leave its enhancement for future projects, thus continuing the process of learning-by-doing (Van der Kerkhof, 2020). In addition, Babcock uncovered the same issue and proposed a solution of rigorous testing, from factory and field tests, to derive a bespoke life-model for an individual engine (Edge, 2022).

Rather than investing additional resources in further attempts to retrieve real-time data from a single engine – a challenging endeavour for the reasons outlined - this paper proposes an alternative approach. The solution of data superposition (see figure 1) expands the dataset to include all engines within the same class. By overlaying, or superposing, data from multiple engines into the Foundry data lake, any air-gaps or inconsistencies are addressed. This enhances the overall accuracy and fidelity of the dataset, particularly as the number of engines increases.

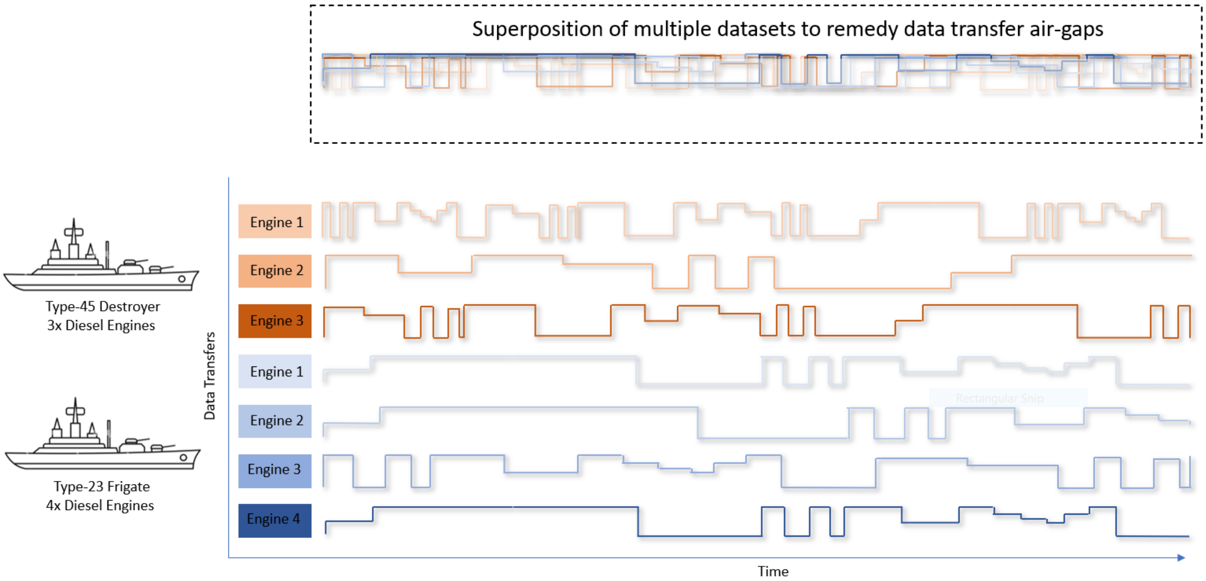

Figure 1: Data superposition highlighting periods of low information density, remedied by increasing the collection of data across multiple engines.

3.2. Value Proposition

Given the scope of this endeavour, initiating the project with a well-defined value proposition (Moore, 2014) serves as a compass, clarifying the purpose of the digital replica, and aligning actions with the fundamental product hypothesis.

'Value proposition – For Equipment Authorities (EAs) who schedule predictive maintenance and stores for defective components, the product is a low latency digital replica that combats the low-connectivity problem by superposing information from multiple engines when data is available. Unlike alternative solutions, the product is developed by the end-user for the end-user and will not require additional hardware to implement.'

 https://doi.org/10.24868/11211

4. Method – What do you call a group of digital twins?

The initial phase of the digital replication process is known as the digital mother – a shared data lake formed from the seven disparate digital twins whose data is collated, examined, and crucially, shared between themselves (see figure 2).

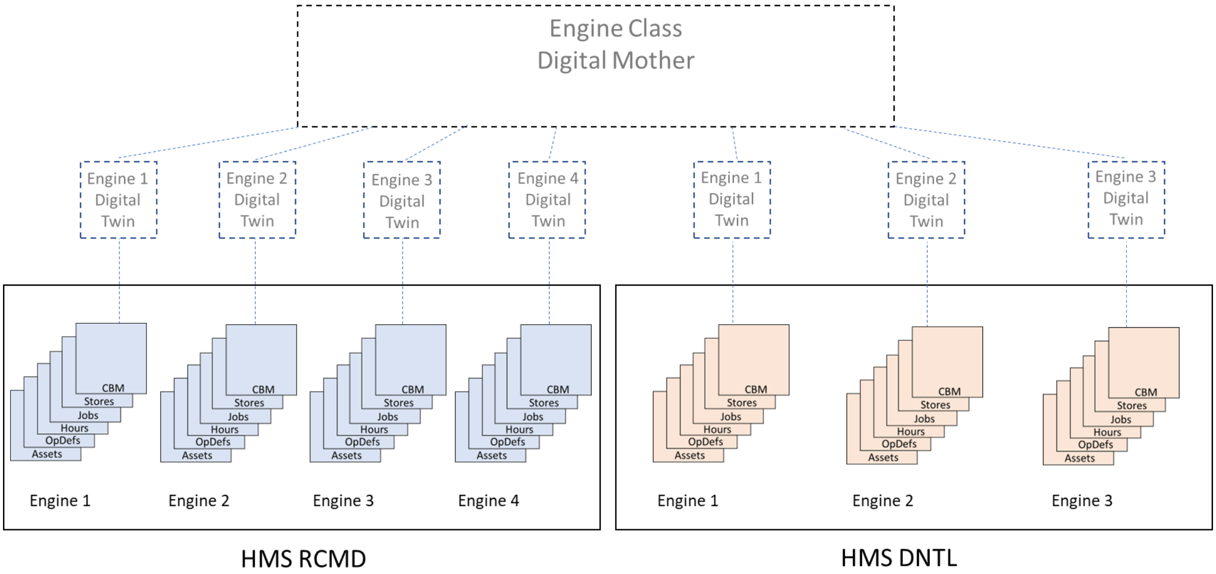

Figure 2: Concept diagram of the first iteration of the Digital Mother which uses Palantir Foundry to collate the superposed data from multiple engines of the same class.

To transform the digital mother from an idea into a product the concept of a Minimum Viable Product is utilised (Ries, 2011), with the high-level system diagram in figure 2 delineated and segmented into major steps, each crafted to validate the fundamental hypothesis detailed in para 2.1:

- Step 1: a repository of relational data from which insights can be generated.
- Step 2: a system of rule-based alerting triggered by set parameters.
- Step 3: utilise machine learning and automation to predict events and resultant actions.

Initiating Step 1 starts the process of learning as quickly as possible. Employing these steps as a handrail, the feedback generated at each stage informs and refines subsequent iterations within a Build-Measure-Learn loop (Ries, 2011). The objective being not to conclude the learning process but to start it, with the systems functionality continuously enhanced – alongside acquiring the skillset to organically develop it. In addition to addressing product design and technical questions, the Minimum Viable Product serves as a constant litmus test for the fundamental product hypothesis, and whether any iterative change will move the concept closer to achieving it.

4.1. Agile development

A key concept is that the digital mother is developed by the user. With training in Foundry, the RN is equipped to build a digital mother with unique flexibility, unrestricted by the constraints of traditional development models characterised by cumbersome bureaucracy and excessive form-filling. Previously reliant on third-parties, RN engineers now possess the autonomy to iteratively refine this digital tool in response to evolving user needs without being beholden to outside partners. Traditionally, a basic change of feature would require a request sent to an outsourced partner, receipt of an estimate, feasibility studies and creation of a change plan along with the back and forth of amends and responses. The organic RN developer can affect a change almost instantly, limited only by an ever-growing skillset to enact them. This simplicity and locality expedite the creation of a digital replica (Kim, 2019). In addition to delivering value both faster and cheaper, this method turbocharges the Build-Measure-Learn loop. Therefore, once the learning journey is commenced, this agility results in rapid iterations where Objects and features are added and withdrawn to suit the required user experience - '*It is much easier to be agile when you are small and light*' (Seagrave, 2023).

 https://doi.org/10.24868/11211

4.2. Digital Mother

The concept of the digital mother revolves around a system health dashboard, where each engine's condition is monitored using several key data inputs, or Objects, as defined in Foundry's terminology. These Objects are created by developing new data pipelines, and their properties and interconnections are visually represented on the dashboard. The system includes write-back functionalities and rule-based alerts, which notify users when certain thresholds are crossed, signalling potential engine distress (see figure 3).

The initial digital model was developed using Foundry's Workshop Module, a drag-and-drop development space that simplifies the creation of complex data workflows without the requirement for extensive coding. The dashboard's homepage provides an overview of the current engine health, highlighting key parameters such as defect health scores and average engine hours conditionally formatted to draw attention to any areas of risk. The dashboard's built-in filter functionality allows users to maintain a comprehensive view of all engines or focus on specific engines of interest (see figure 4).

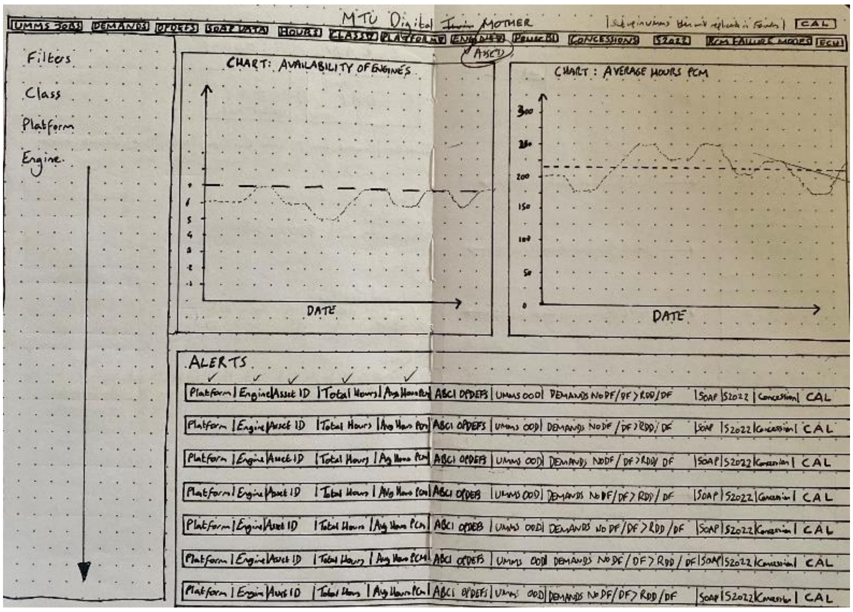

Figure 3: Concept of the Digital Mother Homepage centring on a health dashboard with high-level metrics.

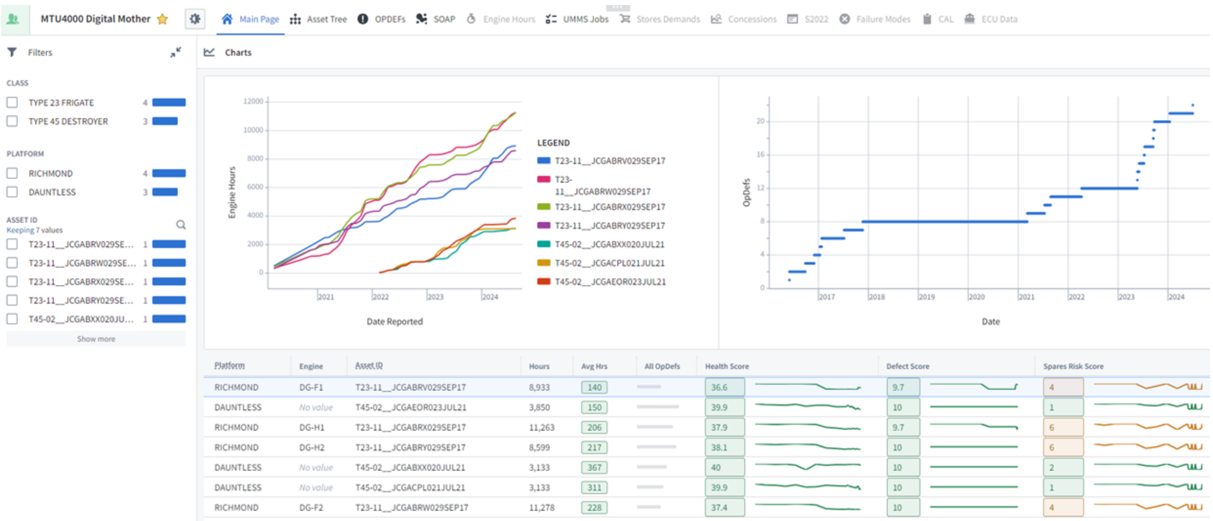

Figure 4: The actual Digital Mother Homepage created with Foundry's Workshop Module, a Low-Code/No-Code drag-and-drop developer space that makes it simple to create Graphical User Interfaces.

The Objects that form the basis for the first iteration are: Assets, Defects, Oil Analysis, Engine Hours, Maintenance Tasks, and Stores. Although these Objects are interconnected within the data lake, they are organised into separate pages, each contributing uniquely to the overall health assessment. This subdivision not only

 https://doi.org/10.24868/11211

enhances the manageability of the project but also improves the user experience by making the interface more intuitive and easier to navigate.

4.3. Assets and Operational Defects

The creation of a holistic view of the engines necessitates a thorough understanding of their subordinate components or Assets. The Asset Object, serving as a digital representation of the engine's physical components, is the backbone of the digital mother. Through this page, users can efficiently navigate the asset tree of engine sub-components, accessing a wealth of information attached to each component. Additionally, users can tag defects against sub-components, enabling further predictive insights, and export selected objects to other Foundry apps for more in-depth analysis.

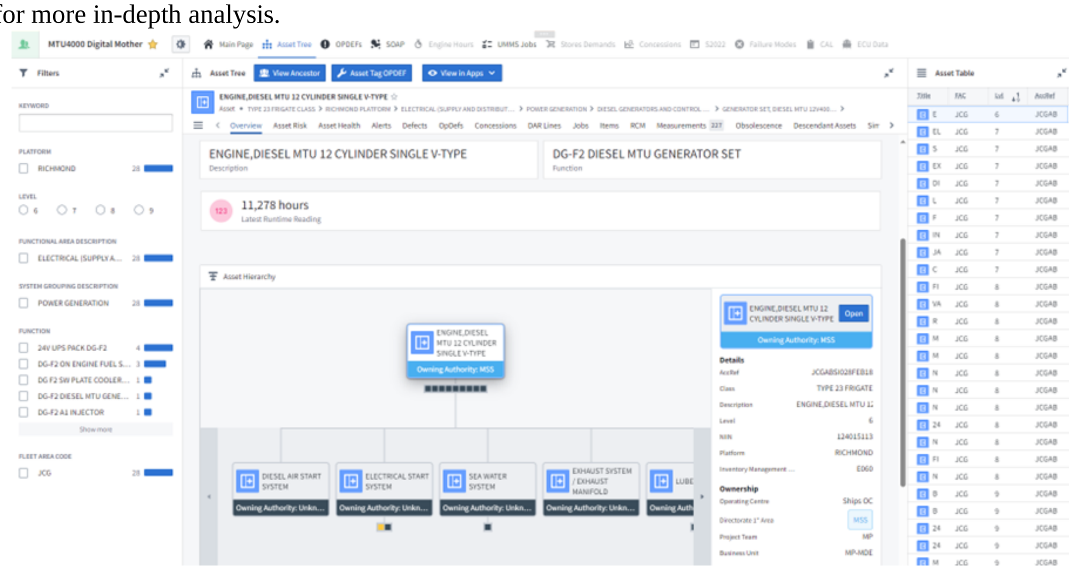

Figure 5: User Interface of the Asset page enabling the management of the engine's asset hierarchy.

A defect that impacts the operational effectiveness of the ship is known as an Operational Defect (OpDef). Establishing a link between the OpDef and Asset Objects facilitates not only the identification of underperforming components but also enables users to analyse causal factors and emerging trends. Furthermore, connecting the OpDef and Maintenance Objects allows for examination of potential correlations between overdue maintenance tasks and defects, while linking it to the Stores Object streamlines supply chain management and facilitates data-driven decisions regarding stock levels based on historical defects. Additionally, this page offers functionalities for data-writebacks, such as appending comments to OpDefs, reassigning to a different Asset, or transferring to another department for action.

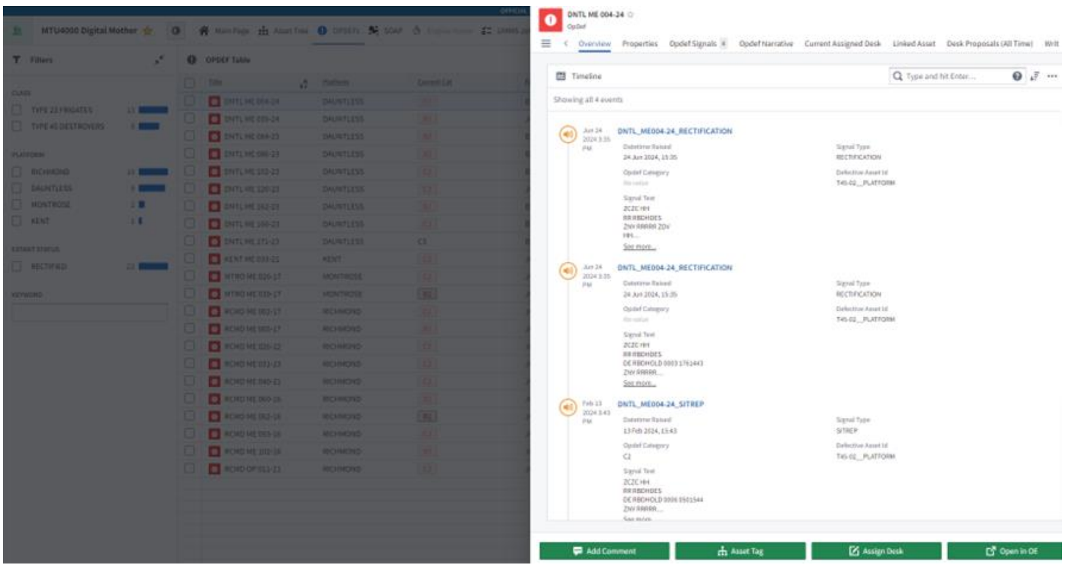

Figure 6: User Interface of engine defects including write-back functions for comments, linking defects to assets, and assigning to specific Equipment Authority desks.

4.4. Spectrometric Oil Analysis

One of the Condition Based Monitoring (CBM) tests performed on the engine is spectrometric oil analysis. Upon receipt of test results, this valuable information is automatically ingested into the RN's data lake through a sophisticated data pipeline engineered using Foundry's Pipeline Builder. This pipeline automates the entire data processing workflow, joining disparate lab results, cleaning them of unwanted or spurious data by validating them against a preset schema, and pivoting the tables into a useful format. The processed data transforms into a Spectrometirc Oil Analysis Programme (SOAP) Object, which regularly updates with new sample test results.

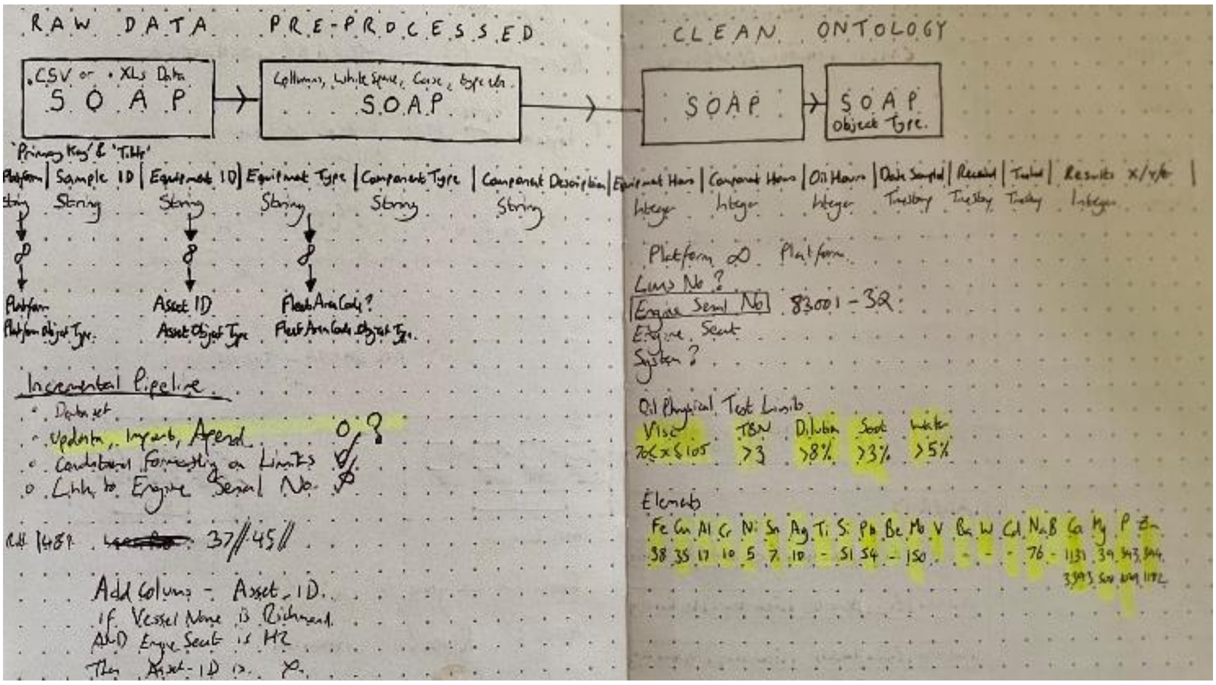

Figure 7: Concept sketch of the data-pipeline used to ingest Spectrometric Oil Analysis Programme (SOAP) data with steps for pre-processing, cleaning, and the assignment of a Foreign Key to enable relational links into other datasets such as Assets and Platforms.

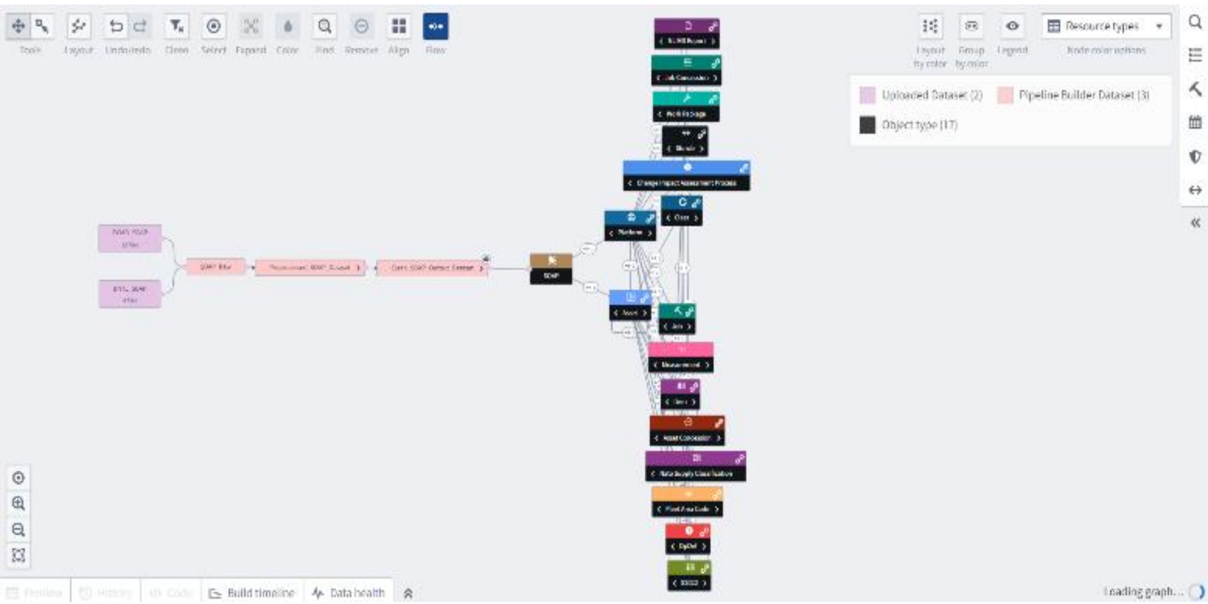

Figure 8: Foundry Pipeline Builder enables the concept sketch to be transformed into an automated data pipeline, where laboratory results of oil analysis are turned into usable Objects in the digital mother. It uses relational links to Asset and Platform to facilitate trend analysis with a variety of other objects.

 https://doi.org/10.24868/11211

To ensure seamless integration and relational links within the data lake, the pipeline uses Subject Query Language (SQL) to assign Foreign Keys, connecting the SOAP Object to specific Asset and Platform Objects. This creates a relational link from the SOAP Object into the broader complex web of interconnected objects, as depicted in figure 8, enabling efficient data exploration and analysis. This connectivity facilitates the use of Foundry's Quiver module (see figure 9) which allows the user to visualise SOAP results in a graph format, establish trendlines, and set up tripwires for automatic alerts in case of contaminant threshold breaches. Additionally, linking SOAP results to other pertinent Objects, such as Operational Defects (OpDefs), supports pattern recognition and predictive analytics. This enhances the capability to anticipate and mitigate potential engine failures.

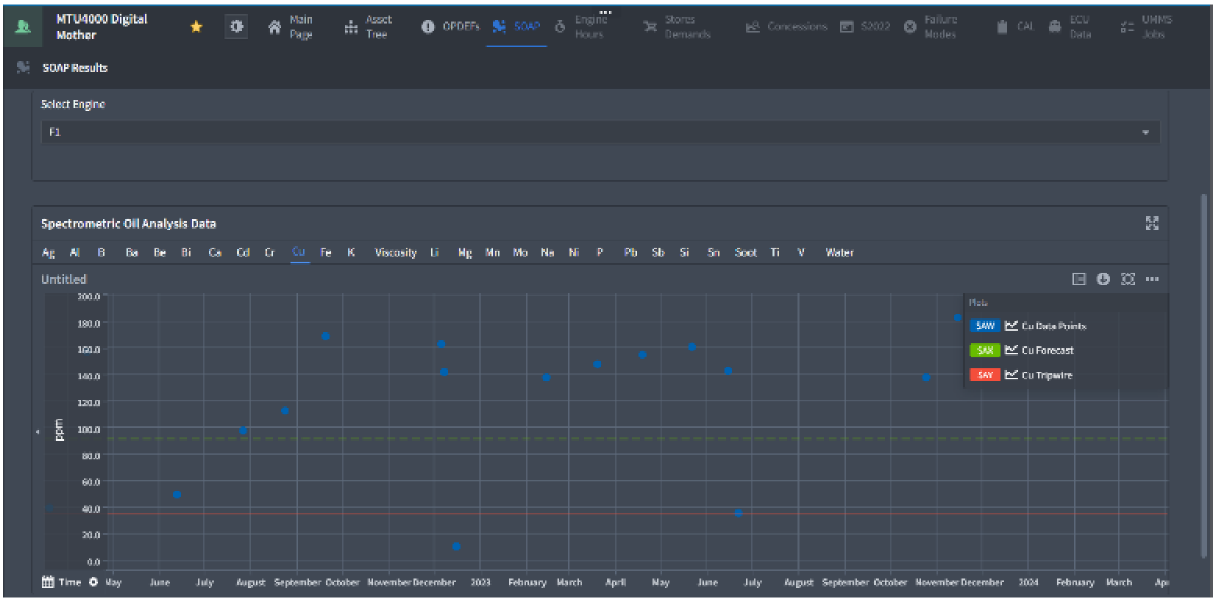

Figure 9: A chart of the oil sample data displayed on the digital mother SOAP page via the Foundry Quiver Module. This provides visual representation of twenty-eight contaminant tests against each engine to enable trend analysis, pattern recognition and automated alerts to the user if thresholds are breached.

The integration of SOAP data with Asset data within the same software platform presents an excellent opportunity to harness advanced machine learning techniques for predictive analytics. This capability represents Step 3 of the Minimum Viable Product outlined in Chapter 4 and is currently under development. The structured nature of this data is well-suited for supervised machine learning, particularly through the development of a regression algorithm that models the relationship between contaminant levels and the 'time-until-failure' of critical components, such as fuel injectors. By partitioning the dataset into training and testing subsets, the model can be trained using supervised learning techniques on the training data and evaluated on the testing data using metrics like Root Mean Square Error. To build and train these models, cloud computing services, such as Microsoft Azure's Automated Machine Learning feature, are being leveraged to identify the optimal regression algorithm and fine-tune its hyperparameters. Once the model is trained, it can be applied to predict future Asset failures based on oil analysis lab results processed through the pipeline depicted in Figure 8.

Currently, the automated alert system is triggered when oil samples exceed predefined thresholds. However, in future iterations, these oil sample results are expected to also generate 'time-until-failure' predictions for assets that are particularly sensitive to oil contaminants, thereby enhancing intelligent supply chain decisions. The advantage of the data superposition method defined in Chapter 3.1 means that the machine learning model used to achieve this will be trained on a comprehensive dataset encompassing numerous engines, thus enhancing the accuracy of its predictions.

 https://doi.org/10.24868/11211

4.5. Engine Hours

The recording of engine hours is a weekly maintenance task within the Unit's Maintenance Management System (UMMS) (MoD, 2020), with the data periodically replicated shoreside and subsequently pipelined into the data lake. This is useful as it allows for the linkage of engine hours (Measurement Object) to an array of other variables. Of particular significance is the ability to discern the operational patterns of these engines. To establish a relational link between the Measurement Object's Asset ID and the corresponding Asset, a simple TypeScript function was coded (see figure 10). This automatically retrieves the latest engine hours, orders them in descending value, then calculates a rolling monthly average based on the preceding three months of UMMS inputs by onboard maintainers.

index.ts ×

```
    2 minutes ago · Uncommitted changes
export class averageEngineHours {
    @Function()
    public async averageEngineHoursColumn(
        myEngines: ObjectSet<Asset>
        ): Promise<FunctionsMap<Asset, Double>>{
        const map = new FunctionsMap<Asset, Double>();
        const myEnginesAsArray = myEngines.all();
        const promises = myEnginesAsArray.map(currentEngine => this.computeMetric(currentEngine));
        const allResolvedPromises = await Promise.all(promises);
        for (let i = 0; i < myEnginesAsArray.length; i++) {
            map.set(myEnginesAsArray[i], allResolvedPromises[i]);
        }
        return map;
    }
    @Function()
    public async computeMetric(
        currentEngine : Asset
        ): Promise<Double> {
        const enginesHours = Objects.search()
                                    .asset([currentEngine])
                                    .searchAroundMeasurements()
                                    .orderBy(Measurement => Measurement.measurementValue.desc())
                                    .take(12)
                                    .filter(Measurement => Measurement.measurementValue);
        const value0 = enginesHours[0]?.measurementValue;
        const value12 = enginesHours[11]?.measurementValue;
        const calculation = (value0 !== undefined && value12 !== undefined)? (value0 - value12) / 3 : 0;
        return calculation;
```

Figure 10: Foundry's Code Workbook used to create a TypeScript function to automatically retrieve engine hours input by the maintainer at sea, then calculate a rolling monthly average based on the last 12-weeks of data.

Given the absence of historical data for these engines, a statistical framework is used to derive insights from the limited available readings. Assuming a Gaussian distribution, the empirical rule is employed as a convenient alert system. One standard deviation from the mean is acceptable, two standard deviations warrant amber conditional formatting, whilst three standard deviations from the mean prompt an alert to the user, indicating that the average running hours of the engine exceed statistical norms. The emerging trend reveals a heightened level of usage compared to their predecessors, the Paxman Valenta and Wartsila12V200 for Type-23 Frigates and Type-45 Destroyers respectively. So much so that discussions are already taking place on potential changes to major service periods and out of service dates which, due to the increased operation seen, may be earlier than anticipated. Interestingly, Babcock independently came to this same conclusion by monitoring Load Factors (Edge, 2022).

 https://doi.org/10.24868/11211

4.6. Maintenance Tasks

The maintenance task, or Job Object, is another pivotal component of the digital picture. Principally, it furnishes invaluable metrics on engine health, affording users the ability to graphically track any maintenance backlog. Integration with the OpDef Object further enables pattern recognition, potentially revealing correlations between defects and neglected maintenance tasks, in addition to prompting the introduction of new tasks as needed, with suggested frequencies derived from historical defect occurrences. As the dataset expands, historical records evolve to drive maintenance decisions based on empirical data rather than time-based or conjectural estimations.

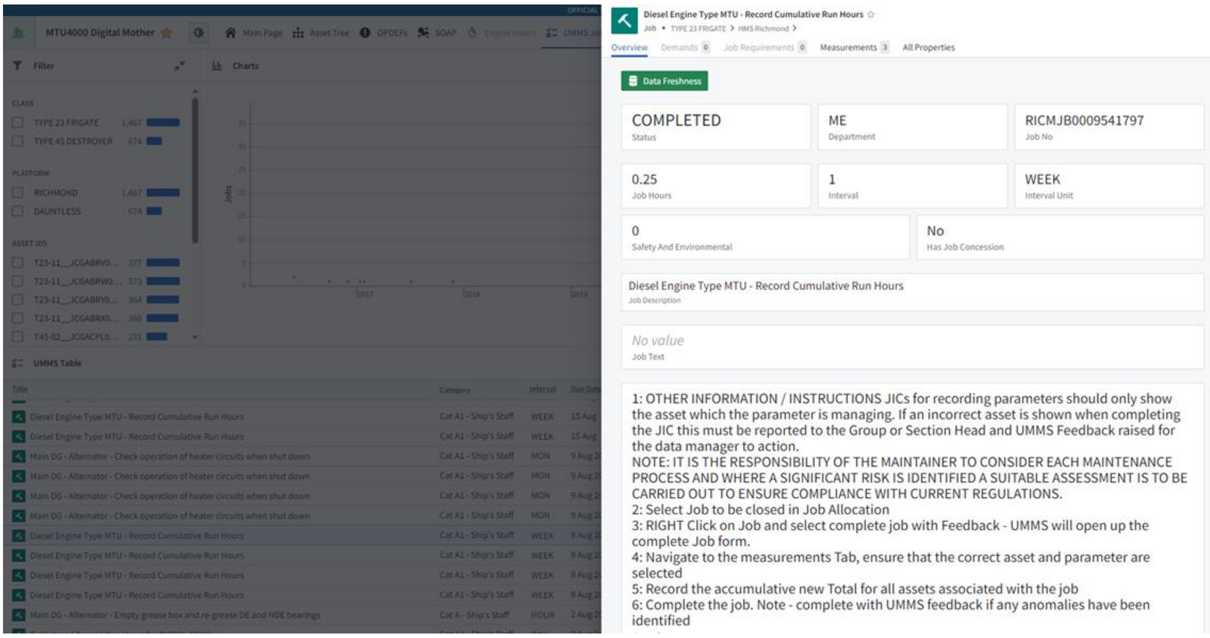

Figure 11: User Interface of maintenance task management page including the Job Information Card and completion status alongside relational links into the supply chain if stores are required to complete the task.

The Job Object contains numerous properties, including the Job Information Card and crucial Object links to requisite stores, facilitating a relational pivot into the logistical supply chain data. Consequently, the transition to a data-driven maintenance system inherently translates into efficiencies within the supply chain. Furnished with these datasets, and the relational links between them, provides an opportunity to implement an intelligent maintenance schedule. To a degree, the RN already operates a predictive maintenance schedule based on routine Condition Based Monitoring (CBM) and a Reliability Centred Maintenance (RCM) system informed by manufacturers' input and historical Lessons from Experience. However, the integration of Complex Data technologies enables the construction of this framework from a substantially larger dataset, with scalable pipelines enhancing information latency. The outcome is a higher quality of decision-making and a more efficient maintenance schedule, leading to savings in maintainer time, costs, and environmental impact.

4.7. Stores Items

As the Item Object is already present in the RN Foundry Database, utilising the Asset ID as a primary-key allows a search-around from the engine's assets into stores and demands. This framework, coupled with the information discussed above, affords the opportunity for more accurate predictions of required stores. Such enhancements not only facilitate the deployment of vessels with optimally provisioned storerooms but also enables proactive planning for forthcoming major services. The result of an intelligent maintenance schedule, where the right tasks are conducted at the right time, is a reduction in unnecessary maintenance tasks which were prompted by a low-fidelity dataset. This efficiency will emanate throughout the support enterprise, allowing for timely ordering of stores, arrangement of contractor support and an informed ship's programme based on accurate predictions of engine maintenance requirements.

5. Future development

Whilst developing the first prototype of an engine’s digital replica in Palantir Foundry, numerous areas for development were noted.

5.1. Improve the effectiveness and efficiency of the inputs

- Regarding SOAP data, there is potential to automate the ingestion by utilising a webhook system or having the tester send their lab results directly to Foundry. Additionally, this scalable data-pipeline could be expanded to other Condition Based Monitoring data such as cylinder pressures and vibration analysis. This data can be easily sourced from maintainers and integrated into UMMS as part of routine tasks, facilitating automatic feeding into Foundry.

- Engine Control Unit (ECU) data is currently decoded and analysed by RR MTU IT Teams. However, due to the issue of low-connectivity, and lack of Foundry access, this is dependent on downloading and manually sending data. This ‘*air-gap*’ is similar to the one experienced by the RNLN project (Tiddens et al., 2020) and further collaboration is required to seamlessly ingest the ECU data for integration of load profiles and error code history.

5.2. Automate the outputs

- Leveraging machine learning techniques would allow Condition Based Monitoring (CBM) data to be used to produce real-time failure forecasts. The format of the SOAP data ingested by Foundry’s Pipeline Builder already lends itself to supervised machine learning via regression algorithms. Once this is initialised, the existing alert system could be further developed to provide ‘time-until-failure’ predictions for Assets sensitive to contaminants.

- By extending machine learning to the supply chain, an additional degree of automation could be achieved. Potentially, the RN could then have a situation where the engine’s digital replica orders its own stores and arranges its own contractor support, to conduct maintenance it predicts it will need on a future given date.

5.3. Royal Navy Engineer Digital Consultancy

Establishing a RN engineer digital consultancy would be a worthwhile investment. This would involve creating a dedicated team of RN engineer data consultants who can swiftly tackle organisational challenges by leveraging Complex Data to inform decision-making. These consultants, drawn from SME-users within the RN, would possess a unique blend of frontline experience and technical proficiency developed through training in SQL, JavaScript, and Python, combined with secondments to organisations like Palantir. Following the lean-startup methodology, the consultancy would prioritise rapid deployment of low-complexity apps to address identified problems. This approach allows for quick iterations based on real user feedback, ensuring that solutions are continually refined and improved. By replicating successful models at pace and scale, the consultancy can overcome the sluggish adoption rates previously seen. A key aspect of the consultancy's approach is its hands-on engagement with end-users. Consultants would parachute into teams, solicit their input, and quickly develop solutions tailored to their needs. Moreover, they would not only deploy these solutions but also provide training to empower teams to utilise and even develop their own digital solutions in the future.

Establishing a RN Digital Engineer consultancy would enable the organisation to harness the power of data effectively, drive innovation, and expedite decision-making. By combining frontline expertise with technical proficiency and a proactive, user-centred approach, the consultancy would represent a significant step forward in modernising and optimising RN digital engineering.

6. Results and Conclusion

The inherent limitations of low-connectivity persist in impeding the realisation of a true IoT-style digital engine replica. However, by aggregating data from all engines within a class, the dataset's fidelity can be significantly enhanced, nearing the desired level of accuracy. Leveraging Foundry as a robust platform for constructing digital models presents a viable solution, particularly given its accessibility to end-users. With minimal training in coding and data engineering, users can advance towards more intelligent and automated systems. In that context, the initial prototype of the engine's digital mother has been broadly successful with the immediate results given below:

- The Asset page is used for scrutinising sub-components and has uncovered disparities between the ships operating these engines. Subsequent data clean-up measures are underway to pay down this technical debt (Cunningham, 1992).
- The OpDef page is used to scrutinise historical defects, ensuring their accurate classification under the correct sub-components to facilitate future predictive analyses.
- The SOAP page is being used to identify potential correlations between CuNi contamination and recent fuel injector failures, aiming to discern any underlying patterns.
- Engine hours serve as a basis for forecasting long-term milestones such as major services and anticipated out-of-service dates.
- Investigation of the Job Object has brought to light numerous discrepancies in the allocation of maintenance tasks across different engines, necessitating further analysis and corrective measures.
- The Item Object and its relational links to defect data, engine hours, and operational usage information, enhances the advisory capability to recommend optimal inventory levels for ship's stores.
- Endeavours to ingest Engine Control Unit data have unearthed various challenges in transnational data transmission to RR MTU, and in providing Foundry access to other industry partners. While interim solutions are viable, collaborative efforts hold promise for enriching the existing Foundry dataset with engine load profiles and error periodicity.

Finally, incorporating RN engineers into the development of these tools is paramount. Drawing on their wealth of lived experience, they possess an intimate understanding of operational challenges and can intuitively envision the most effective solutions. By prioritising investments in expanding their skillset to include coding and data engineering, the continued enhancement of these tools is assured.

As the RN enters its third year of embracing Complex Data, there is reassurance in observing a slight improvement in its 194-year adoption rate. Whilst this problem would be incomprehensible to Sir James Lancaster, the solution method utilised would be very familiar.

Acknowledgements

The author would like to thank Cdr Seagrave MBE RN for advancing data engineering in the RN. Palantir for delivering training and answering technical questions. Navy Digital for laying the foundations of many of the Objects used in this project. RR MTU IT for their ongoing engagement in processing Engine Control Unit data, and Lt Cdr Prior RN for his judicious peer-review.

The views expressed in this paper are that of the author and do not necessarily represent the views of the RN.

 https://doi.org/10.24868/11211

References

Blank S.G. (2003). The Four Steps to the Epiphany: Successful Strategies for Products that Win. K&S Ranch Publishing Inc.

Bowen H.V., McAleer J., Blythe R.J. (2012). Monsoon Traders: The maritime world of the East India Company. Scala Publishers Ltd for National Maritime Museum, citing Markham C.R. (ed). The Voyages of Sir James Lancaster, Kt, to the East Indies. The Hakluyt Society.

Cunningham H. (1992). The WyCash Portfolio Management System. In Proceedings of the OOPSLA-92 Conference on Object-Oriented Programming Systems, Languages, and Applications. ACM.

Edge W., Fox M. (2022). Delivering the UK MoD's support DEAM. Proceedings of the International Naval Engineering Conference 2022.

Kim, G. (2019). The Unicorn Project. IT Revolution Press.

Ministry of Defence. (2020). Book of Reference (BR) 1313. Hampshire: Navy Publications and Graphics Organisation.

Moore G.A. (2014). Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers. Harper Business Essentials.

Morariu A-R., Ashraf A., Björkqvist J. (2021). A Systematic Mapping Study on Edge Computing Approaches for Maritime Applications. Proceedings of the 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA).

Richardson, C., Rymer, J. (2014). New Development Platforms Emerge For Customer-Facing Applications: Firms Choose Low-Code Alternatives For Fast, Continuous, And Test-And-Learn Delivery. Forester Research, Inc.

Ries E. (2011). The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business.

Rijsdijk C., da Silveira N.N.A., Tinga T. (2020). Using ship sensor data to achieve smart maintenance? Proceedings of the International Naval Engineering Conference 2020.

Seagrave S. (2023). Data Bites #38: Getting things done with data in Government [Video]. Institute for Government. https://www.instituteforgovernment.org.uk/ecent/data-bites-38-getting-things-done-data-government. Accessed April 2024. Timestamp (45:08).

Syed M. (2015). Black Box Thinking: Why Some People Never Learn from Their Mistakes - But Some Do. John Murray Publishers.

Szent-Györgyi, A. (1932). Hexuronic Acid as the Antiscorbutic Factor. Journal of Biological Chemistry, 97(1), 39-44.

Tiddens W., Pollmann, B., Curvers D., Teunisse M., Shaneh Saz S., Zegers J. (2022). Smart Maintenance for Modern Naval Ships. Proceedings of the International Naval Engineering Conference 2022.

Tröhler U. (2003). James Lind and scurvy: 1747 to 1795. JLL Bulletin: Commentaries on the history of treatment evaluation (https://www.jameslindlibrary.org/articles/james-lind-and-scurvy-1747-to-1795/)

Van de Kerkhof R. (2020). It's about Time: Managing Implementation Dynamics of Condition-Based Maintenance. CentER, Tilburg University.

 https://doi.org/10.24868/11211