UNIVERSIDADE Đ
COIMBRA

Manuel Costa Carvalho

Visualizing Remaining Useful Life prognostics in Aircraft Maintenance

Web based tool for visualization of actionable information in the ReMAP project

Dissertation for Master in Design and Multimedia supervised by Professor Penousal Machado, co-supervised by Professor Catarina Silva, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

October 2021

VISUALIZING REMAINING USEFUL LIFE PROGNOSTICS IN AIRCRAFT MAINTENANCE

MANUEL COSTA CARVALHO

1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

Master in Design and Multimedia
Department of Informatics Engeneering
Faculty of Sciences and Technology
University of Coimbra

October 2021

Manuel Costa Carvalho: *Visualizing Remaining Useful Life Prognostics in Aircraft Maintenance,* Web based tool for the visualization of actionable information in the ReMAP project, October 2021

This work is partially funded by national funds through the FCT – Foundation for Science and Technology, I.P., within the scope of the project CISUC–UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020. Also, this Paper is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N 769288.

SUPERVISORS:
Penousal Machado
Catarina Silva

ABSTRACT

This work arises in the context of the Real-time Condition-based Maintenance for Adaptive Aircraft Maintenance Planing (ReMAP) project, under the Horizon 2020 (H2020) program.

The main goal of this work is the development of a web tool for visualizing and extracting actionable information relating to aircraft components health state, in order to aid with maintenance decisions. We aim to leverage the study of Information Visualization for the abstract analysis of Remaining Useful Life (RUL) prognosis and prognostic results from ReMAP members, in order to develop a visual idiom for its adequate presentation and interpretation. By analysing and experimenting with prognostic model results as well as generated synthetic data, we built a web based Visualization tool that is able to present and analyse RUL prognostics, for all aircraft components of a fleet. This makes the information readily available to support the maintenance decision teams. Finally the Visualization tool can also support prognostic model developers to analyse and compare model results.

The visualization tool is available at: https://student.dei.uc.pt/~mcc/ReMAP/v4/

RESUMO

Esta dissertação surge no contexto do projecto ReMAP do programa H2020.

O principal objectivo do trabalho é o desenvolvimento de uma ferramenta web para a extração e visualização de informação útil, relativa ao estado de saude de componentes de aeronaves. Esta informaçao é comunicada com o proposito de auxiliar os processos de decisão de manutenção. Neste sentido pretende-se aplicar o estudo da area de Visualizaçao de Informação para analisar os resultados obtidos por parceiros do projecto bem como o processo de realizar prognosticos RUL, para desenvolver uma linguagem visual adequada para comunicaçao dos prognosticos obtidos e a sua correta interpretaçao. Através da análise e de experimenetação com resultados de modelos de prognóstico e com dados ficticios gerados no contexto do projecto, construímos uma ferramenta web que permite a visualização de prognósticos RUL de todos os componetes de aeronaves de uma frota.

A ferramenta de Visualização está disponivel em: https://student.dei.uc.pt/~mcc/ReMAP/v4/

ACKNOWLEDGMENTS

First and foremost I dedicate this work to my brother Óscar and my parents, Óscar and Fátima, for making my education their top priority and supporting me unconditionally.

I would like to thank all the ReMAP team, specially Floris, Iordanis, Marie and Raul for sharing their work, knowledge and time.

I would like to thank the WP5 UC team for their help and guidance trough out the project.

I would like to thank my supervisors for their time and for helping until the very end.

I would like to thank my friends and family for their patience and support.

For last I want to thank Mariane for being there for me through all the hardships.

Thank you!

CONTENTS

LIST OF FIGURES

ACRONYMS

BCE	Before Current Era
CAD	Computer Aided Design
CBM	Condition Based Maintenance
CE	Current Era
EoL	End of Life
EEoL	Expected End of Life
H2020	Horizon 2020
HI	Health Indicator
ICS	Integrated Cooling System
CACTCS	Cabin Air Compressor Temperature Control System
InfoVis	Information Visualization
IFHM	Integrated Fleet Health Management system
PHM	Progostics and Health Management
PDF	Probability Density Function
ReMAP	Real-time Condition-based Maintenance for Adaptive Aircraft Maintenance Planing
RUL	Remaining Useful Life
VHI	Virtual Health Indicator
WP	Work Package

1 INTRODUCTION

The present dissertation arises within the ReMAP project [1]. ReMAP is a project funded within the H2020 program by the European Commission. ReMAP aims to develop an open-source solution for Aircraft maintenance, the Integrated Fleet Health Management system (IFHM). The IFHM aims to offer all the processes or steps required for Condition Based Maintenance (CBM).

Within the ReMAP framework, one of the steps is predicting the RUL of the aircraft components, so that the maintenance tasks are scheduled according to the components' condition. In order to quickly and efficiently provide the prognostic information to the maintenance teams, a Visualization tool that allows the users to extract useful and actionable information relating to the aircraft and components prognostics and availability is required.

In addition to the opportunity to design and develop an application that is meant to be incorporated in a challenging environment, and address a critical task, the desire to explore and showcase the benefits of using Information Visualization (InfoVis) techniques, to better communicate and understand complex information, specially when dealing with large quantities of data and with time-based data, is also one of the motivations for this thesis.

Although there is a part in the development of the project which might be considered as prognostic estimation, this work is mainly focused on the visualization of the prognostics, the idiom and the framework in which component and aircraft prognostics are organized.

The main objectives for this work are the development of a web based tool for the visualization of the prognostics, the development of an idiom for visualizing multiple prognostics for an aircraft or fleet, and delivering other actionable information related to the prognostics. This includes allowing the user to better understand and compare the incoming prognostics with past predictions and with the average behaviour, as well as making visible the degradation of the systems. Furthermore, the prognostics should be presented in a in way that allows the user to evaluate the generated schedules (which are based on the prognostics) and identify eventual corrections.

The key contributions of this dissertation are:

- The design and development of an idiom for the visualization of RUL prognostics ; this idiom builds upon the visual language already used by prognostics developers for evaluating and communicating prognostic results, which promotes its immediate understandability and adoption;

- The design and development of a web-based InfoVis framework for the visualization of the prognosis of aircraft componets; this tool allows for "plug and play" action, allowing users to simply provide a ".csv" document for the dataset of the visualization, requiring no other resources or software other than the browser;
- The "generalization" of the framework to allow the visualization of an entire fleet, allowing the user to quickly have an overview of the fleet state and infer on aircraft availability;
- The possibility of visualizing the average of the recent prognostics behaviour and to compare it with incoming prognostics;

Chapter 2 starts with an historical summary of early InfoVis contributions, up to what has become known as the "Computer Era". In this chapter we also go over the "What-Why-How" framework proposed by Munzner, as well as the the related data and task classifications. Chapter 2 will report on the technologies used for the development of the project, as well as related visualization works and techniques.

Chapter 3 will report on the work plan and calendar, methodology, the project stakeholders as well as other project members.

Chapter 4 starts by introducing Prognostics and Health Management, then describing how the "What-Why-How" analysis was used, both for synthetic data generation, and for the tools' development. It explains how we modeled the synthetic data and it describes the structure of the tool and how it works.

In Chapter 5, we report both on experimental and early results obtained with synthetic data, as well as subsequent results obtained using RUL prognostics generated by real models developed under the ReMAP project.

Chapter 6 concludes the dissertation summarizing the work done, the main contributions and indicating opportunities for improvement.

2 STATE OF THE ART

There are several proposed definitions for the field of InfoVis [2]. Some definitions are very broad, while others can be a bit too strict, effectively excluding some classic works such as the ones of Playfair, simply because they mention or require interactivity or the use of computers.

Some other definitions differentiate InfoVis from Scientific Visualization and even Data Visualization, while others might consider the latter a branch of the first.

Regardless of how strict, most definitions, if not all, share a common core principle. That principle is to produce an artifact that allows us to make use of the several capabilities of our vision, to facilitate and improve our understanding or cognition of information.

Using visual metaphors and representations to convey information provides two main advantages [3, p6]

- Our visions have a very wide bandwidth or capacity to receive information/stimuli which makes the process of contextualizing information, establishing relationships, identifying patterns and shapes much easier;

- By externalizing the information, we relieve the viewers of having to imagine everything on their own. This is specially important when dealing with new, complex, or large amounts of information because we can't rely so heavily on our memory.

Therefore InfoVis can be considered as a field that studies the use of visual artifacts and metaphors to convey (more) information, more efficiently. Usually in a way that would be much more difficult or impossible, without relying on visualization.

In the remainder of this chapter we summarise information gathered for the development of the project.

As the project's goal is the production of a tool for visualizing time-oriented data, the first step was an historical exploration into InfoVis, specifically on how time oriented-data has been displayed.

Tackling an InfoVis project starts with a varying amount of questions, the three main being:

WHAT? What is it that we are trying to communicate;

WHY? Why is this information being presented, why and who is it relevant to;

HOW? How are we going to present the information?

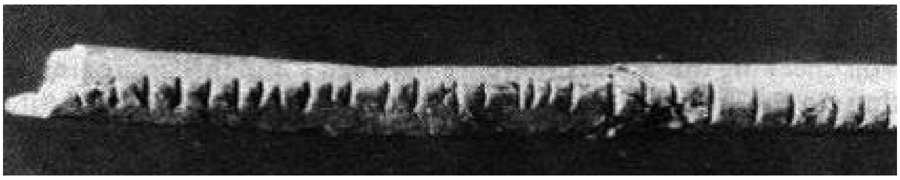


Figure 1: Picture of the Lebombo Bone. This bone presents several vertical lines carved along its length. It is believed that it was used as some sort of annotation tool. This bone has been dated to be around 42,000 years old. Extracted from: [7]

This research was supported mainly on two books: Visualization Analysis & Design, Munzner [3] 2014 and Visualization of Time-Oriented Data Aigner et al. [4]. Both books are presented as arising from the lack and need of a generalized and unifying view on the research field of InfoVis (and sub-field of time oriented data).

The "What?-Why?-How?" approach is presented and explored in both books and will be applied in the context of the current project. Both books cover and present a wide array of techniques and how and when to use them.

Section 2.1 presents a brief historical overview of the InfoVis field with focus on closely related time-visualizations.

In Section 2.2 we describe the most relevant parts for our project regarding data and tasks classification and analysis, as described by Munzner [3] in the " What?-Why?-How?" framework.

Section 2.3 will report on the aspects concerning the technologies used within the development of the project.

For last, Section 2.4 reports on related web-based visualization tools and techniques which contributed to the projects development.

2.1 HISTORICAL PERSPECTIVE ON INFORMATION VISUALIZATION

There are several variants of the saying "A picture is worth a thousand words", in several languages and attributed to several historic figures such as Napoleon, Da vinci and as far as Confucious. This expression captures effectively the essence of InfoVis, or visualizing information as a more efficient tool for acquiring and understanding information [5].

Given the field definition's plasticity, it is hard to pin-point the concrete time and place where we started creating or using visual aids for cognition.

We can trace back its roots to the prehistory, with human's first visual expressions. This can be in cave paintings, with its earliest figurative representations dating around 43,000 years or as markings in bones, presumably as an aid to counting or keeping time, 20,000 to 50,000 years old, Figure 1[6, 7].

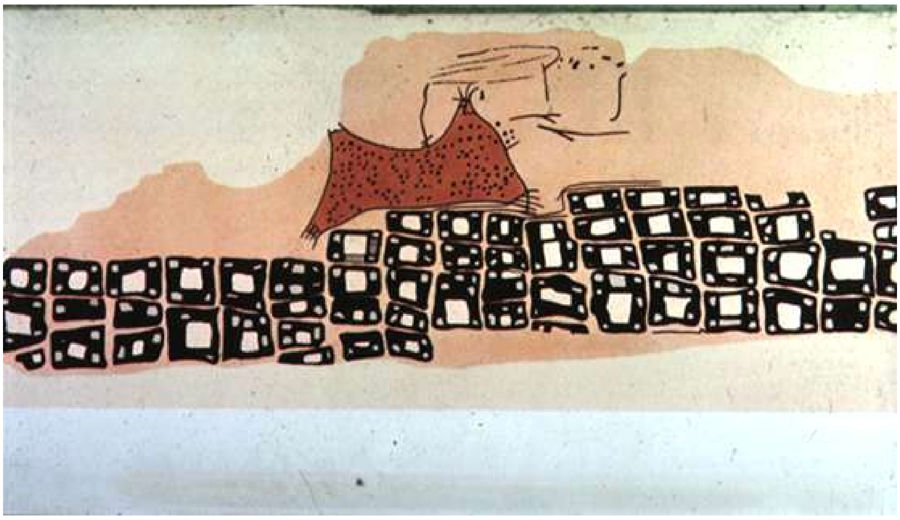


Figure 2: Cave painting depicting the layout of a town, matching the city of Catalhoyuk. This painting was discovered in a cave in Ankara, Turkey. It is believed to be 8,000 years old. Extracted from: https://www.infovis.info/visuals/Gallery_of_Data_Visualization/Historical_Milestones/1600-/oldest_map_konya_town_6200BC.jpg

In the same sense, it is believed that the earliest attempts at cartography, or map-making, date from 25,000 to 7,000 years ago, including landscape representations, settlements and constellations.

Figure 2 shows a cave painting that predates the Babylonian clay tablet maps, usually considered the oldest maps [8].

The early and proto-writing systems were developed in at least three different places and in different times, namely the Sumerian, the Egyptian and the Chinese, from 7,000 to 4,000 years ago [9].

Using a visual aid for speech unburdens our memory load, and also removes the time related constraints of spoken language, either by allowing for multiple readings at different times and paces or by preserving the message exactly as it were intended or firstly relayed.

"Because the messenger's mouth was heavy and he couldn't repeat [the message], the Lord of Kulaba patted some clay and put words on it, like a tablet. Until then, there had been no putting words on clay" - Sumerian epic poem Enmerkar and the Lord of Aratta. Circa 1800 Before Current Era (BCE)(Uruk; present day Iraq)

Although modern InfoVis definitions portray Visualization as an alternative or as a diametrical opposition to text, it is hard to deny the benefits of visualizing information as text when compared to spoken language.

In what concerns to record keeping, historically, text proved to be a much more reliable medium than spoken words and folktales

In the 10th century, about 700 years before the invention of the Cartesian Coordinates, we find the first record of multiple variable data being plotted relative to time. This artefact, show in Figure 3,

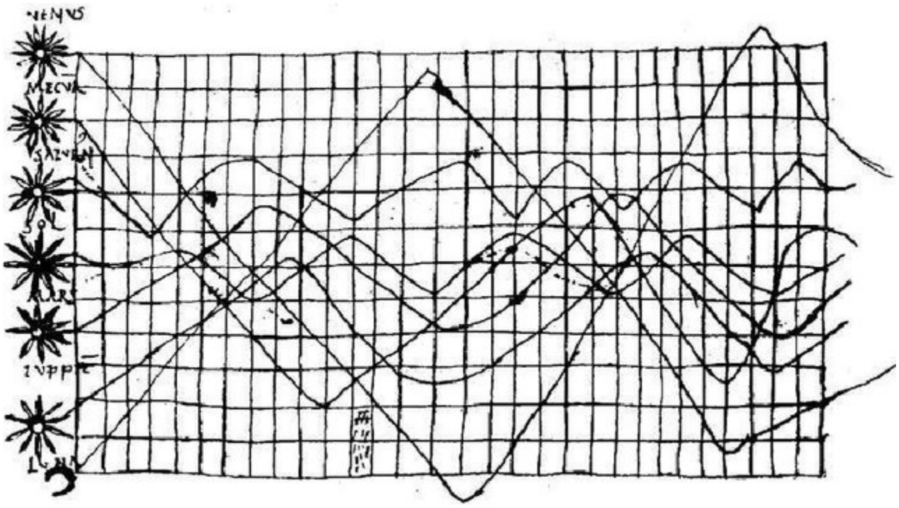


Figure 3: Planetary positions over time, 10th century, unknown author. On the vertical axis several celestial bodies are named. It is believed that the horizontal axis represents time and that the drawn lines correspond to the bodies' paths in the night skies or their height above the horizon. Extracted from: https://commons.wikimedia.org/wiki/File:Mouvement_des_plan%C3%A8tes_au_cours_du_temps.png

represents the position or the angle of rotation of several celestial bodies during the year [4].

Its contents are debatable due to the lack of rigor and contextual information, as well as due to the unknown authorship.

In the 17th century, building a bridge between algebra and euclidean geometry, René Descartes developed the Cartesian Coordinate System which directly contributed to the origin of analytic geometry, calculus and cartography. To this day, it is still one of the most common tools used both in Mathematics and in InfoVis.

In 1765, Joseph Priestly, a multifaceted scientist and educator, developed the first timeline, called "Chart of Biography" where the lives of several historical Statesmen and Men of Learning are plotted as lines, a sample is show in Figure 4. The lines' lengths represent their lifespan and their spatial position represents their temporal location, that is the lines start in the year that the person was born and end in the year they died. With this idiom we can compare the lives lengths easily, as well as assess who lived at the same time or how far apart two lives were. Another very important detail is that Priestly even managed to deal with the uncertainty of some of the dates, by using dots and dashed lines [4].

This Visualization is attributed as the inspiration which led to William Playfair inventing the bar chart.

Around 150 years after Descartes' time, a little younger and contemporary of Priestly, we have William Playfair. Playfair is considered today as the father of statistical graphics, and he is responsible

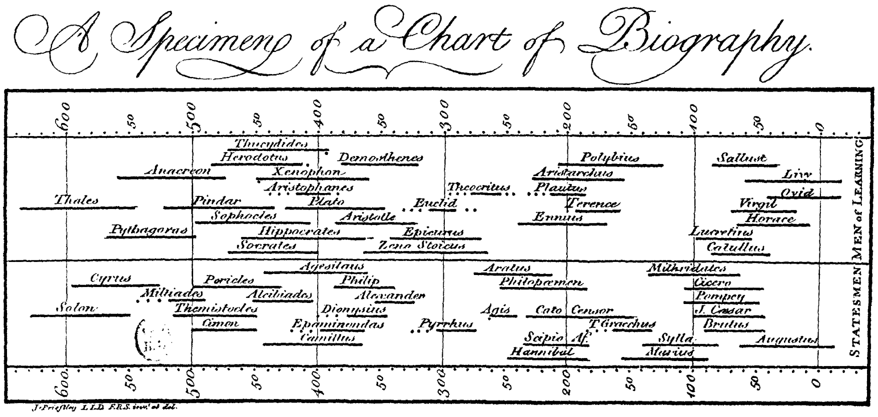


Figure 4: "A specimen of a Chart of Biography" Josehph Priestly, 1765. The figure shows a sample version of the original "Chart of Biography" which was much larger. In the top and bottom of the graph we have a time scale disposed horizontally, counting the years from around 600 BCE to the first years of Current Era (CE). The lines plotted represent the lives of important historical figures. Their position and length are relative to the timescale. Extracted from: https://commons.wikimedia.org/w/index.php?curid=2285294

for creating and improving the most common graphics we still use. Examples are the bar, silhouette, pie and line charts.

Specifically in the time-dependent information domain, Aigner et al. [4] outline Plaifairs' works published in "The Commercial and Political Atlas" and "A Letter on Our Agricultural Distresses, Their Causes and Remedies". Both publications include an array of time-based visualizations, the first is relative to the economic state and balance of England in the 17th century. The second, being the first that combines several types of graphics in the same idiom / visualization, presents the relationship between salaries, wheat price and the reigning monarch, from the 16th century to the 19th.

In 1869 Charles Minard creates the "Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie" or the figurative chart of successive losses in manpower by the French army, during the Russian Campaign, from 1812 to 1813. In this chart, considered by many, one of the best statistical graphs ever created.

With a map as the visualization basis, Minard represents Napoleon's army advancement as trace of a band; the width of the band corresponds to the number of men in the army, the color of the band represents the direction, if the army is attacking or retreating. The location of the band corresponds to the geographical location of the army. Additionally there is also a representation of temperature, which is mapped relative to the army's retreat.

In addition to allowing us to have good overall comprehension of the course and outcome of the Campaign, we can clearly see how

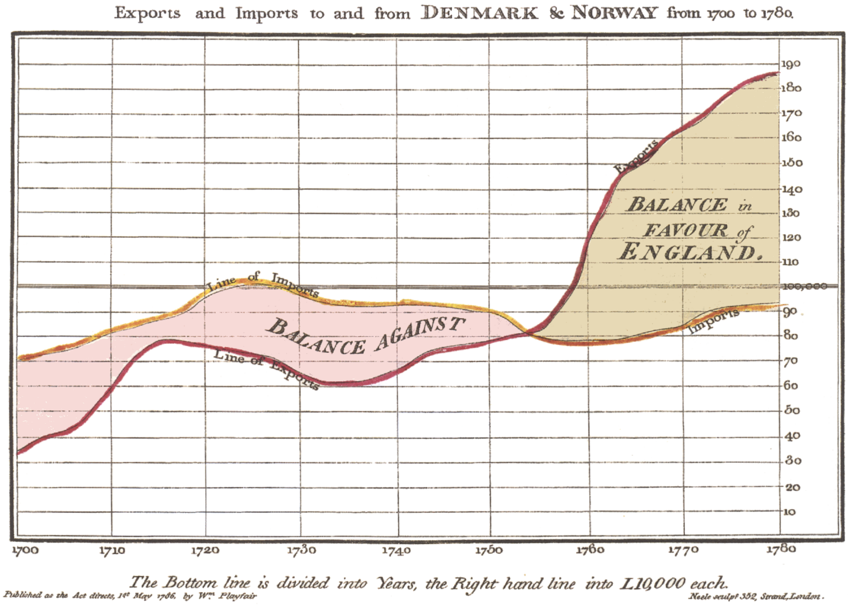


Figure 5: "Exports and Imports to and from DENMARK & NORWAY from 1700 to 1780" by William Playfair, 1786. The graph combines two line plots of time series to create an area chart. The yellow line shows the imports from Denmark and Norway to England. The red line shows the exports. The area between both lines represents the balance of the trades. It has two colors, red for when imports are higher than exports, called "Balance Against" and yellow, when exports are higher than imports and the balance is in favour of England. Extracted from `https://commons.wikimedia.org/w/index.php?curid=12674977`

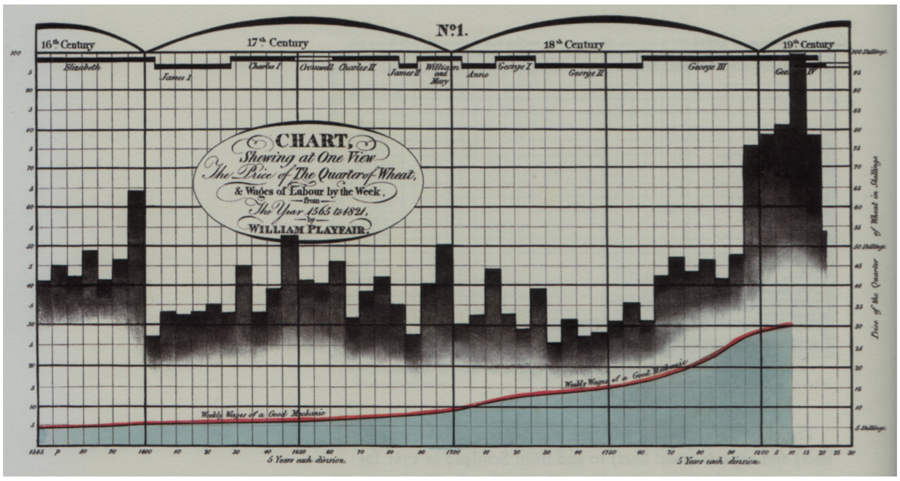


Figure 6: "Chart Showing at One View the Price of the Quarter of Wheat, and Wages of Labour by the Week, from 1565 to 1821" William Playfair, 1822. In this chart, Playfair combines different idioms to represent different types of information in the same view. The horizontal axis is time, show in the bottom and the vertical axis is price, shown on the left. The blue area and red line shows the "weekly wage of a good mechanic". This can compared with the average price of a quarter of wheat, shown in the black vertical bars. This comparison is enriched with information about the reining monarch, represented as timelines, and marks delimiting each century, both at the top of the graph. By William Playfair 1805 (1759-1823) - Edward Tufte, The Visual Display of Quantitative Information, Graphics Press USA, 2001, 2e éd. (1re éd. 1983), 190 p., Public Domain, https://commons.wikimedia.org/w/index.php?curid=29717191

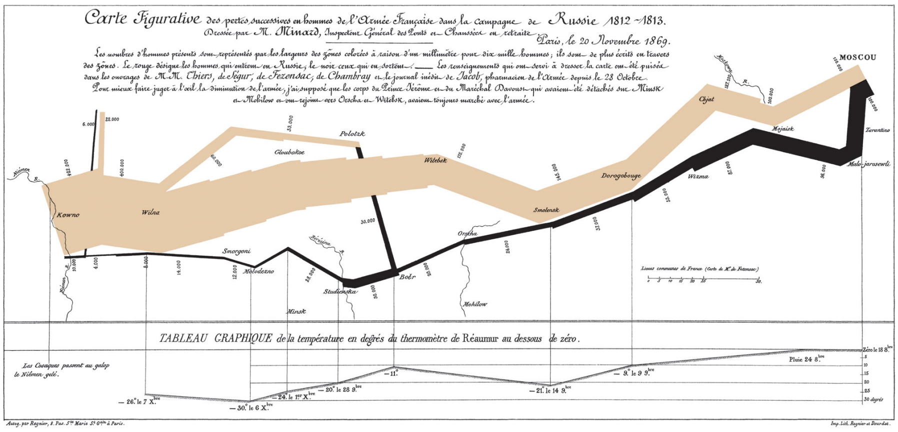


Figure 7: "Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie" by Charles Minard, 1869. Considered one of the best statistical graphs made, it is very information dense but is surprisingly easy to read and to decipher. By Charles Minard (1781-1870) - see upload log, Public Domain, https://commons.wikimedia.org/w/index.php?curid=297925

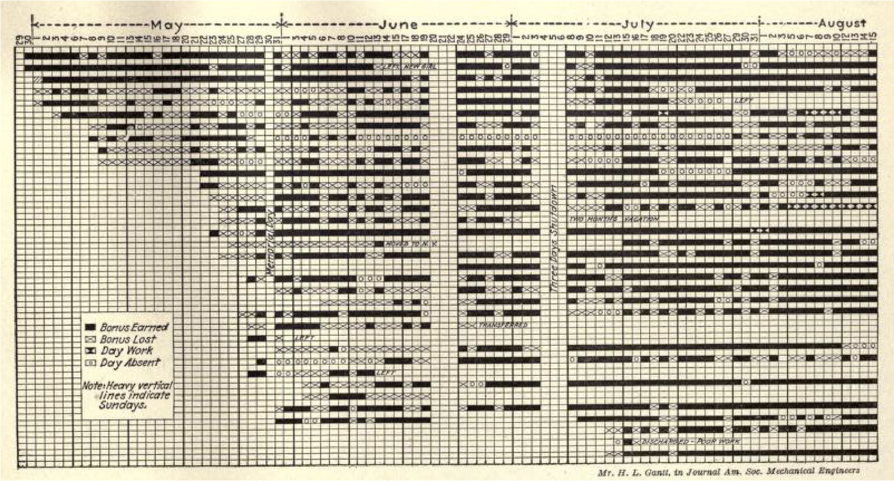


Figure 8: In this chart, each line represents one worker. On the top we find an horizontal axis, measuring days, from May to August. The legend notes that black squares mark days when workers earned a bonus; Crossed white squares mark the days when the worker lost his bonus; The square with 2 black opposing triangles represents a normal work day and squares with a circle represent absent days. Additionally, the heavy horizontal lines mark Sundays. Extracted from: https://archive.org/details/graphicmethodsfo00brinrich/page/52/mode/2up

poor was the French Army performance, the giant number of losses, as well as the date and place where they happened. We can also relate losses while retreating with the temperature [4].

With the advent of the Second Industrial Revolution resources, staff and task management became ever more important became a important. Although it is considered that he was not the first to experiment with this type of idiom, Henry Gantt (1861-1919) became attributed with invention of the Gantt charts. This type of chart is design to deal specifically with planning and representing activities, their location in time and duration. Alongside activities, other elements can be displayed like milestones and sporadic events [4].

Figure 8 shows an example by Gantt himself.

In a Gantt chart, tasks are organized vertically as a textual list, in one of the sides, usually the left side and usually taking only up to a small percentage of width of the visualization. Then, on the other side, each task is paired with a timeline, the lines are vertically aligned with the task names, their length represents the duration, and their starting and ending points represent their location in time. Additionally tasks can be grouped and hierarchically organized via indentation and/or color of background [4].

2.2 DATA AND TASKS

There are two main questions that arise when facing an InfoVis problem: "What?" and "Why?"

The first question relates to the data we are working with. What it is, what it means, how it can be represented. In the context of the ReMAP project the information to visualize was the RUL of aircraft systems and/or components. In other words, how much time can those components fly before having to be replaced. We know that this RUL estimation is going to be performed at regular intervals, therefore we know that data is going to be some sort of time series.

Munzner [3] presents an assortment of classifications for data, user and idiom, including data type, dataset type, attribute type, attribute semantics, marks, visual channels, idioms, user tasks and needs etc. This classification is built upon the three questions framework "What?-Why?-How?" and equally divides the answers in data-task-idiom. This frameworks is presented in relation to the Nested Model, which divides the InfoVis problem into 4 nested levels. From the outer level in, the Nested Model consists of: Domain or Problem characterization; Data and Operations Abstraction; Encoding and Interaction techniques; Algorithm Design. Here, the what and why questions would correspond to the 2nd level and the how question to the 3rd level.

The "What?-Why?-How?" framework is used and referred by other authors with slight differences to the one detailed by Munzner.

2.2.1 *What?*

The "What?" question refers to the data: What is it? How is it organized? What does it mean? To answer this, data is classified according to its type, data set type and semantics.

Some data types are bound to some data set types. In the same way, certain data set types have their intrinsic semantics, or relationships between the content, that might not exist in other types. I.e., tree datasets can only have nodes and links; Time-dependent data such as in time series have an intrinsic order.

DATA TYPES For data type we have: **Items**, **Attributes**, **Links**, **Positions** and **Grids**. An **item** is an individual discrete entity, usually used to identify, group or address other information. An item can be a row in a table or a node in a network.

An **attribute** is some specific property that can be measured, observed or logged like the height of a person, the price of a stock, colors, temperature, names etc.

Links are relationships between items. For instance between nodes in a network.

Positions are specific spatial location data. These can be pairs or trios of coordinates in 2D or 3D space. For instance latitude and longitude on the surface of a globe. Or the xyz coordinates on a medical scan.

Grids can specify a sampling strategy or a geometrical or topological relationship between cells.

DATASET TYPES Regarding the dataset classification, four main dataset types are presented: **Tables**; **Networks** and Trees; **Fields**; **Geometry**;

Tables are datasets that are organized with rows and columns. Each cell is identified by a combination of row-column and contains a value. Following the classification presented, this value is an observable or loggable attribute, which is identified by the column, and belongs to the item identified by the row. Munzners book also specifies multidimensional tables from flat tables. The main difference being that, while on a flat table, the differentiation of an item can be done with one attribute only and therefore one item will be one row, in a multidimensional table, several attributes are required to identify one item, and therefore one item might need to be represented by two or more rows. For instance if there is a time component and we are taking several measurements of an item, we can stack the tables that correspond to each moment, vertically, and result in a three dimensional table. Otherwise we each item will be composed of an additional row for each instant logged.

Networks are datasets composed of items (nodes) and their relationships (links) and are usually focused on the relationship between the nodes the latter. The nodes may or may not be linked and both the nodes and the links might have attributes associated with them. It is necessary to distinguish this as the abstract concept of a network as opposed to a visual representation of a network in which the spatial localization of the nodes is most relevant. One example for this type of dataset can be a dataset detailing a Neural Network and the latter would be its representation. Another specific example in network like data sets are trees. These are a specific type of network which is hierarchical. In this sense the nodes of the same hierarchy don't connect amongst each other, only to their parents and children, that is the previous and following levels in the hierarchy.

Fields are analogous to tables but are used in the context of continuous data. Fields are intrinsically related and constrained by the sampling and the interpolation techniques used to log and present the data and are most commonly used in physical phenomena sensing and modeling in the engineering, mathematics and medical fields.

Geometry datasets pertain to the shape of items and their spatial position. These can contain information about simple primitives like points and lines or they can detail complex shapes like 3d models.

In contrast with the other types, geometry datasets might have no attributes at all.

Other types of dataset are described like set, lists clusters and paths. A set being an unordered group of items and a list being an ordered set. Clusters focus on the similarities between items and paths are sets of links, in a network or tree.

This list focuses only on the basic types of data structures. Although the complete list would be very extensive, these are presented as the foundation as most of the remainder are some sort of amalgamation of these.

On the other hand, this list does not intend to pose as a strict divider, instead, this list and all of these classifications are proposed as tools to help us answer the question of "What?" is the data we are working with.

There is also the matter of availability, regarding the dataset, as it might be a static or a dynamic dataset. A dataset is considered static when it does not change during the visualization. That is when the dataset is fully available from the start. A dynamic dataset implies that we will get new information over time, as the visualization is being used. This new information might be in a form of extra information or it can be a change in existing values or measurements instead.

ATTRIBUTE TYPES As before, there are a lot of attribute types we can use to describe our data.

Munzner draws a major distinction between **Categorical** and **Ordered** attributes.

Categorical attributes do not have implicit ordering. An extrinsic order can be applied at will, but the data in itself does not have one. The most common example is the name or type of things. As mentioned, names can be sorted alphabetically, for instance, but as names they don't have an intrinsic or natural order. **Categorical** attributes serve mainly to distinguish if two things are the same or not.

As opposed to categorical attributes, **Ordered** attributes have an implicit ordering. This type is divided into **Ordinal** attributes and **Quantitative** attributes.

Ordinal attributes have a well established order but it cannot be quantified or we cannot apply mathematical transformations to it. Its order can also be on a relative or on a self-referenced scale. The example of shirt size is presented in the book, in which although it is clear that the size Large is bigger than the size Small, we cannot, for instance, add small to large (small+large =?) to obtain a meaningful piece of information. Another example used is rankings, we can work with numerical rankings but there would be no meaningful information resulting from mathematical operations.

On the other hand, with **quantitative** attributes we can indeed apply meaningful mathematical operations, like arithmetic comparisons, as these attributes are measurements of magnitudes. These include measurements of time, height, temperature, weight, etc.

Ordered attributes can be further subdivided into **Sequential**, **Diverging** and **Cyclic**.

While **sequential** attributes fall between a specific homogeneous range, from a minimum to a maximum, **diverging** attributes are usually expressed in relation to another measurement, that is a new "zero".

For instance, if we're measuring a stock price, as this price will necessarily be higher than 0 and lower than a very high number, we can say that it is a Sequential scale, or that this attribute is **ordered** and **sequential**. On the other hand, if this dataset is, for instance, comparing the stock price over a year with its average price, then it will be diverging as our values will be roughly centered on 0 and growing outwards in both directions, as this zero is in fact our average price, and as an average there will be values above and below. See Section 2.4.3.

Ordered attributes may also be cyclic instead. This means that when the values go over (or under) a certain threshold, or value, they go back to the start. These types of attributes are extremely common when working with time-oriented data, as the way we measure time is intrinsically cyclic. After 60 seconds it resets to 0 and we get one minute, after 60 minutes, the count goes back to 0 and we get 1 hour, and so on and so forth. This cyclical character is present in all magnitudes/scales for measuring time and, when trying to compare two behaviors over time it is usual to compare them within one of these ranges or cycles. A time-oriented attribute is not cyclic by default, or just by being time related, it needs to be a conscious decision. I might have the dataset for the price of a stock, during a whole year, but unless I choose to group the data by hour, day, week or month it will not be automatically determined that it is cyclic.

There is also the matter of hierarchy and attributes might be or have **hierarchical** characteristics. For instance, in the case of the ReMAP project, the prognostics are done at the component or sub-system level, but these are part of a greater group, a system, which in turn belongs to an aircraft and the aircraft belongs to a fleet.

The hierarchical information might be an attribute by itself, or for instance, it might be encoded in an identifying attribute (name). Another common example is in time, and time related data. Especially when there is a cyclical grouping as there will be a need to match the relative time with the real one, that is to identify which month is a certain week relative to. If not only to identify and differentiate the most recent information from the historical or past data.

2.2.2 *Why?*

Following the 3 question set, the second stands for "Why?" - Why is the visualization needed?

In order to answer this, Munzner proposes the identification and classification of the user tasks and needs, in an abstract and general way so as to not get lost in a project's individual characteristics and to be able to compare the strategy and techniques used with those of other domains and projects. This is also necessary as it is common to have the same types of actions or similar processes being referred to in different ways, in the same way that is common for a term or concept to have different meanings in different research fields.

It can also be helpful to first define who is or are the tool users, so that we can identify not only different tasks and needs but also if all the design choices remain with the tool designer or not.

2.2.2.1 *Users*

If we grade Visualization tools from specific to general use, most commonly, the classification will (co)relate with the amount of freedom the user has in terms of design choices. InfoVis tools for very specific usages, are usually designed with a narrow context in mind and, although they might present several design choices and allow the user to see the data in different ways, these are usually built-in and the user can only switch between views or choose from a defined set of tasks or actions. Therefore being limited as well in the type of data they can work with. On the other end of the spectrum, generic InfoVis tools, or for general use, are designed to work on broad and different contexts, by different people, with somewhat different goals. These might allow for a user to visualize all datasets of one type, or even allow the user to build the visualization itself, like Google Sheets and Observable. One tool having more user freedom does not imply that it is automatically better than the others. It is necessary to adapt the tool's breadth to the user's tasks and needs because too much freedom might translate into a lengthy and inefficient process, it can introduce errors or clutter the visualization space. On the other hand, if we narrow the tool too much we might not be able to respond to the user needs at all, or not allow for an analysis with enough depth or with a useful focus.

These design choices include ones like focusing one element or group, displaying and inferring data, like statistics and other transformations, adding or displaying different and/or simultaneous views, pan and zoom etc. These functionalities can exist in either end of the spectrum, the difference is if they are included as the normal or default behaviour or if they are included as opt-in features, or as something that the user can build, manipulate or display by himself.

2.2.2.2 *Tasks*

Three different levels of actions are described and proposed in order to understand the user goals. These are the analysis, search and query level.

The analysis level is divided into consuming and producing Information. The most common use for a visualization is for the purpose of consuming information.

ANALYSIS One of the most common goals of using a InfoVis tool is to **consume** information. The tool can be used by the user alone, in which it can be a matter of discovery or pure enjoyment, or the visualization might be used by someone to corroborate or illustrate something he already knows, to someone else, in this case its purpose is to present information.

With the goal to discover, it is perfectly common to simply look at the overall picture, that is the whole data, and try to formulate a hypothesis. On the other hand, the visualization purpose might be exactly to verify previously made hypotheses or conjectures instead. This action or goal is considered to be the classical motivation for the creation and sophistication of interactive idioms and visualizations, as the designer has no way of knowing in advance what the user will want or need to see, and, therefore, needs to accommodate for the user to explore the data, hopefully, in all the ways the user needs.

The enjoyment is considered to be aligned with the discovery goal being that one of their main differences is that while discovery is driven by need the other is most commonly driven by matter of simple curiosity. They might be or lead to different visualizations or idioms, but any kind of visualization can be ("mis")used with the purpose of enjoyment.

As a free-time InfoVis enjoyer myself, that goal usually surges unexpectedly. That is, I don't set out to look for visualizations in order to quell my enjoyment needs, but instead I'll stumble upon a news article or a InfoVis tool itself and I will find myself wanting to know more about the information being presented and I will end up exploring the tool and the data on my own, usually for an extended period. Although InfoVis has been used in communication media for some time, more and more we are seeing investigative journalism leveraging the interactive capabilities of the browser and web pages to communicate their investigation results with interactive visualizations. Especially in the case of big data we see tools that, for one are used by the journalist or author to present information to us, the reader, but at the same time can also be used by us and allow us to delve deep into the whole data and absorb it at our own pace and leisure, thus also covering our user goals of discovery and enjoyment.

The purpose of presenting information implies that there is one entity or user trying to relay information to another. This can be done by presenting the data and the related summarized information or it can be achieved by guiding (showing) the audience through the several steps that lead to this information or knowledge. InfoVis tools and techniques for this purpose are extremely common in the pedagogical context. There is a very wide array of subjects that are taught in schools with the aid of graphics and other techniques, from learning the correct techniques for the long jump to z-tables for statistics and innumerable others, especially the more complex the subject gets.

Presenting is also common in the context of collaboration and work organization, with tools to help with decision making, planning, forecasting and instructions. As mentioned earlier, the area of journalism also shows a recurrent use of InfoVis tools to present information, either it being due to its better efficiency in visual space usage and on relaying information, but as well and even more so due to the shift in consumption from physical based formats to digital ones. As journalism and information consumption shifted to the digital and especially to the social media and web era, it was now possible and easier to integrate and use InfoVis tools with interaction capabilities and produce those kinds of settings where a InfoVis serves both or all of this 3 purposes, that is presenting the results from the investigation and allowing the audience to use it at will, either for discovery and/or enjoyment

Another purpose, which might be simultaneous with the consume purpose, is to **produce** information. In this context, the InfoVis are used with the intent of outputting some kind of (new) information, usually being part of a set of processes or tasks in which the visualization of the information is not the final or main goal. The produce goal can be subdivided into three: annotate, record, and to derive information.

Annotating refers to the context in which related information is added on to the visualization, usually by the user. This can take form as highlighting certain data-cells, or groups, adding textual information like adding comments or context for the next step or user in the process. Bookmarking data items, queries and views can also be considered annotating.

Using a tool with the purpose of recording data means the user will want to externalize and save -something- of the data, it can be the visual representation, the visualizations parameters, as well as everything we mentioned in the annotation purpose. They might also want to export the information selected or derived, in any kind of information format, to the extent of a text summary. This time frame is the main difference between annotation and recording, in the sense that the annotations are temporary additions that last while the visualization is being used while the recording purpose implies that data is externalized or outputted and kept. Annotations might be saved and made persistent in which case the purpose might change to record.

The purpose of using the InfoVis tool might also be to derive information. The derivation process is one of the most present aspects in the visualization context. It may or may not be the user end goal (to derive new information) but it certainly is a very common task the InfoVis designer might do, for any kind of usage purpose. Even if there is no derivation purpose, deriving new or transforming the original data might still be required for the tool to be effective. Deriving data means we are creating or inferring new data based on the data we already have. This can translate to new information that is only obtainable by mixing or matching different existing data (or attributes, most commonly) or it can just be a simple reformulation of the existing data, either by compressing the data and removing unnecessary details, or by changing the data types or groupings in order to have more or better options to choose, in order to arrive in our final solution or visual idiom. Naturally, the derived information can be the visualization in itself, or it can be annotated and/or recorded as well, so it is important to not consider user goals and tasks as exclusive, or better, we should take in all of the user needs and goals and decide on how much each purpose should influence the InfoVis idiom. This means that, for instance, if our problem requires a solution that presents the monthly sales of some company as they are trying to improve that metric. We might decide on providing a tool that presents the sales data for some months, additionally derive some other information like the average sale rate, or the average per month, and then it can also compare them. We could stop at this point, because the company just asked for a tool to view their monthly sales, in order to improve them. If we take an overall look at the tool usage and user goals we can see that the tool is going to be used repeatedly, and if they want to compare specific month rates, they would have to run the tool several times and even screenshot the visualizations. If they want to add a cause or a justification to some specific change in rate, they would also have to screenshot the visualization and comment on their own. Having this in mind it would make sense to further discuss the tool usage with the company or the users and assess which features make sense to have under the visualization and which should remain separated, these are also defined by other conext-related constraints like the time and cost to finish the project, computational power and even the literacy or technical background of the users.

SEARCH The second archetype or main branch of goals and purposes is to search for information. The search goal can be subdivided into lookup, locate and browse and explore. All the goals referred under the analyze goal require that the user takes a previous step of searching for data elements. The search subdivision is done according to what the user knows about the data he is looking for.

When the user knows what item he is looking for and where to find it, the type of search is called lookup. This is the case when we go on Google maps to see our house from satellite view, we know what it is and where it is.

When the user knows what the item is but he does not know where to find it then he is trying to locate it. This is the case when we are searching for a specific entry on a sorted list, like the results of a race. I want to know in what place did player X finish, I know what it is but I don't know where it is so I have to locate his name in the results list.

On the other hand, when the user does not exactly know what data he is looking for but knows where to find it, then it is called browsing. Using the race example, if I want to know who is the winner of the race, I know I can find it on the first entry of that sorted list, whoever it was that won. The same applies when I want to buy a shirt and I browse a clothing shop online. I don't know which exact shirt I want, but I know I can find it under the shirts category and maybe narrow it down by color, size and availability.

If the user can't even be certain about the location of the target then the search is of the type explore. This type is common when looking for outliers, for instance, as we know what we're looking for but we don't know how far and in which direction will those outliers fall, in relation to the average grouping. It's the same when we go to a gift shop and we want to buy a souvenir for someone else: we don't know exactly what it is at first, and we don't know exactly in what part of the store we will find it but we know what the person likes. Therefore we will explore the store until we find some item that fits our budget and that person's preferences and we will probably just know what it is and where it is, when we find it. In a more general way, if we have a set of measurements and operations, and some result appears to be very unexpected or just plain wrong, we might try to explore the whole data and the operations, step by step, in order to find what it is that got us that result - we don't know what we're looking for, just that something is not as it is supposed to be, and we are not sure if it is a problem with the raw data or with the following operations. Exploration usually starts with an overview of everything, as a quick way to discard areas we can see are working as intended and to narrow our search scope.

QUERY The third branch of user goals is to query data or information. It is not completely strange to the search branch of purposes as queries and querying features can provide the user for most of his goals that fall under the search branch. That being said, in order to query something, the user has to have a target in mind so it is common for the query goal to succeed a search goal or task where the targets are identified. Once the targets are defined, we can query them at 3 different levels: identify, compare and summarize.

Identify refers to a single target. Following a search where we lookup or locate some known targets, identifying them will inform us on their characteristics, that is it will return or present the target's remaining attributes. Re-using the race example, after learning who was the winner, by browsing the top of the list (looking at the first entry), I can then cross his identification number or name with a hypothetical participants list or table and I will be able to query or identify the runner and learn the rest of his information, like age, residency etc. On the other hand, if there is just one table with all this information, I can simply query and identify the runner whose attribute of run time is the lowest or whose finishing position is one or first.

The second level of queries, compare, has a scope of multiple targets. One of the most common cases of querying multiple targets is in order to compare them. For instance, given a dataset of atmospheric temperature for several years, a very typical scenario would be to subdivide that data by years and months. Then, very common as well, would be to query the same month for all the years in order to compare them. Or even to query and compare each year in its whole.

The third category of queries is to summarize. Another word for it is overview, both the verb and the noun. The first usually refers to provide a comprehensive view of all the data, and the second, as a noun, usually refers to a summary (display) of the data instead. Naturally the scope of these queries is every target available in the data set. This concept and goal of providing an overview of the data is common core in visualization, both as a feature in InfoVis tools but also as one of the steps in the process of building a tool in itself.

TARGETS Queries are defined in relation to the number and awareness of the target. If tasks are verbs, then targets refer to nouns. Depending on the context, we might not need to refer to them as they are implicit. On the other hand, as in search and query contexts, the targets are explicit and the user needs to refer to them. Hence the abstract categorization of the targets should also be considered, especially if there is not a concrete answer to some questions or if the data is not available or completely known, while the tool is being built.

In the broad context of the "whole data" there are 3 high-level kinds of target: trends, outliers and features. Trends refer to the main aspect or structure of the data, and usually try to understand what is happening or what is going to happen. Trend examples include, increases, decreases, peaks, troughs and plateaus. When elements of the data don't fit an existing general trend, these are considered outliers.

Features are context dependent, they usually represent a specific structure of interest, within the overall data. For instance, in the RUL estimation context, there is a step referred to as feature selection. In this step there is a selection of the sensor's data, possibly including derived and metadata, discarding irrelevant data, or data from sensors that don't contribute that much to the diagnostic / prognostic. As shown in MathWorks's guide, in its simplest form, it can be done just by visualizing the data. In more complex cases when this is not enough, other techniques and calculations are used instead, but its visualization remains relevant. On another example, we have the daily stock prices, for some companies, for a whole year. One feature might be the companies prices behaviour on the weekends, or on some specific months, as the companies might have some inherent seasonal characteristics.

Our target might not pertain to the whole data, but instead to a single item or attribute. Attributes are specific properties that can be visually encoded. The lowest level target is to find the value of a certain attribute. Other common examples include finding the extremes, or the distribution of the values of an attribute.

We might also be interested in multiple attributes instead, common examples include finding dependency between attributes, correlation and similarities. Dependency is when a change in values in a certain attribute produces a change in another, that is the second attribute is dependent on the first. This is the case when working on derived or inferred data, for instance. Correlation is when one attribute presents a similar trend or behavior to that of another but in a way that does not configure dependency, as there is no certain causation. Similarity between two attributes is a metric which takes into account all their values and calculates how far they are from each other. Finding trends, outliers, similarities and relations are very common tasks or goals while using visualization tools. As mentioned earlier, if not looked at abstractly, it might be difficult to find common ground between approaches and different fields, as tools tend to be domain-specific and use domain-specific language and representations.

2.2.3 *How?*

After classifying the data we are working with and what the user needs to do and learn from it, we can look at the different types of design choices to build our idiom. The design choices can be broken into 4 major types: Encode, Manipulate, Facet and Reduce. Encoding data encompasses the five main choices for spatially arranging data and another category for non spatial visual channels. The first group is referred to as arranging data, the last category is referred to as mapping data. Arranging data includes the cases when the spatial position of the items is based on its attributes. We can express the values, separate, order and align regions or just use the spatial data when it is the case.

Mapping data means we are transforming categorical and ordered attributes into other visually perceptible attributes, instead of the spatial position. These mapped attributes include color, size, angle, shape, motion and many others that can be visually interpreted.

The second type of choices, manipulate, pertain to any that directly affect the view, that is the context and overall aspect in which the data is being presented. It includes changing the type of visualizations being presented, it includes selecting elements and it also includes ones that affect the viewpoint of the user like navigation and zoom.

The third type of choices, referred to as to facet data includes the choices in which data is presented in multiple and/or coordinated views, choices related to separating and partitioning data into different views and, also, vice-versa, how to aggregate multiple instances in to one view. In other words, when we present different data in simultaneous views, when we partition the (same) data into different views and when we overlap or mix and aggregate different views into just one

The last type of choices includes the ones in which we reduce the amount of data being shown or at least the space some of it occupies. This can be done by filtering away irrelevant or less important data, by grouping or aggregating data elements, representing a derived attribute, accounting for that group, or by embedding the focus and context information together, in the same view.

2.3 TECHNOLOGIES

For this project, several technologies were used, in different stages of the development of the project, as there were several different tasks and objectives to fulfil.

The resulting Visualization tool from this project is developed in HTML, CSS and Javascript, using the D3.js library.

Before building the framework and coding the visualization tool, other technologies were used, both for learning the data characteristics and aspect but also for quickly deploying and trying out ideas.

The first technology we should mention is Google Sheets [10]. Google Sheets is an online tool which allows users to create, edit and view datasets in the table format. Although its main use is not the creation of InfoVis artifacts, it proves itself to be a very useful tool, as it has an array of inbuilt idioms that the user can deploy on demand, requiring only simple settings and adjustments.

After having an overview of the data, knowing how it looks like and how it is modeled to look like, the second technology used, was the Rhinoceros 3D [11] software, which is a 3D computer graphics and Computer Aided Design (CAD) software. Rhino, as is most commonly referred to, has a plugin, called Grasshooper 3D [12], which is a graphical algorithm editor and allows for parametric and generative design. Because of that, it also has some useful tools for dealing with data and datasets. As such Grasshopper is also able to produce InfoVis artifacts. Although it is not very fast at dealing with huge amounts of data in real time it has one big caveat when compared with other technologies: Having near the same amount of possible features and operations, it does not require the user to write any code. The user simply needs to connect data blocks in to and out of operation/task blocks, and it is done visually with the mouse only. This means that the tasks we can do in Google Sheets, for instance, can be done just as fast, and in the end, we can use and manipulate the resulting artifacts as one usually does in a CAD software. This in-depth geometric study and manipulation of the resulting curves was a big contributor for experimenting with the time-series comparison method, which we will detail in Chapter 4.

The most important tool for the development of the tool was, as mentioned earlier, the D3.js [13] javascript library. This library allows for the manipulation of documents based on the data they represent. The main focus of the library is to help dealing with the data related problems and it offers a neat way to deal with data-driven transformations with requiring the user to type very long lines or to rely on temporary states. Another great feature is that we can pass styles and other attributes as a function of the data instead of as a constant, as is the case in the JQuery library. Additionally, D3 does not introduce new drawing functions, like processing (p5) or protovis, instead D3 relies on inbuilt html, svg and css capabilities, meaning that if these are updated, D3.js will still work without requiring a new an update of its own. For last, D3.js is an open-source library and is paired with several tutorials and guides [14] covering a wide array of topics and techniques.

Another technology which need to be mentioned is the Observable [15] platform. This platform allows the users to code in Javascript and immediately see the results, directly on the browser and without needing to build the whole web page or packaging the software. Additionally, most D3.js tutorials and techniques offered by D3.js authors are only available on the observable platform.

2.4 RELATED WORK

Other works and techniques, from different areas, specifically Epidemiology and Finance (stock exchange), were very useful for the development of the project.

For one, one of the major aspect in epidemiology is the modeling of the number of infections, on a population, specially how will it evolve over time and also when will the infection numbers slow down and eventually stop. This is usually communicated to the public with line charts and timelines.

On the other hand , in the stock exchange context its is also quite present this need to compare different time series and predict their behaviour, most commonly the stock price over time. Along with line plots, other types included are point plots, bar charts and box plots. Is also possible to find several of these combined in the same visualization

2.4.1 *COVID'19 dashboard*

The most significant event occurring during the development of this project was, without a doubt, the SARS-COV-2 / COVID19 pandemic [16, 17]. A few days after it grew into a pandemic, we were being blasted with huge amounts of information, specially regarding the infection numbers. Every day we would see line plots and graphs showing and comparing the daily and weekly infection numbers, between different countries and, later on, between the same months of different years. Of all, the most readily available were the ones presented by Google [18], embedded in any COVID related search. These visualizations use data from several international sources [19].

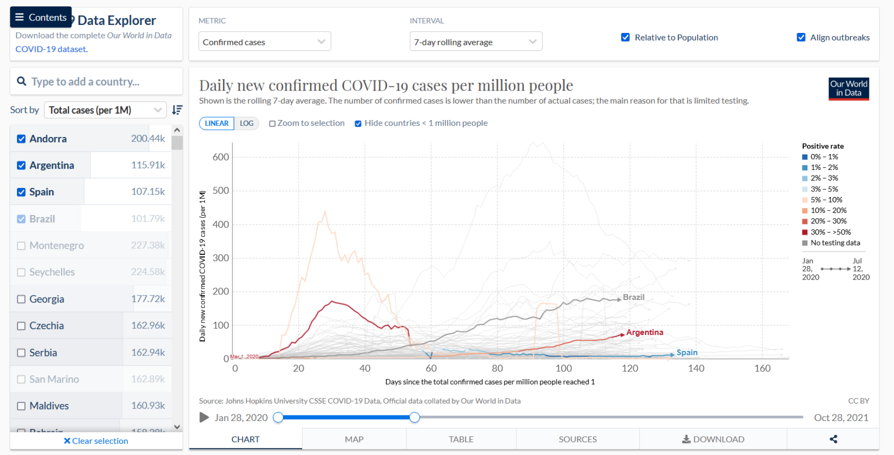
One of the most interesting type of graphs I saw, was a graph comparing the total infection numbers over time, between different countries. The aim was to try and predict if and how bad could a second wave of infections hit. And also how long until the numbers would reach a safe zone, based on the behaviour of countries which were hit sooner.

While some graphs were showing all the countries lines on a common time scale, this one would align the infection curves by the date each country registered their first infection, effectively making all the lines start in a common point.

To do this we just need to use the number of days elapsed since the first case recorded instead of the real dates.

As it got clear that it was not going to slow down on its own, this kind of graph progressively lost its relevance and I can no longer find the article where I saw the graph described above. Now, nearing the second year of the pandemic, data is much more available and there are endless dashboards online. Fortunately I could find an example, as the one described, show in Figure 9, by Ritchie et al. [20].

On their COVID-19 statistics page, they present around 8 different metrics with matching visualizations. If we chose the metric "Cases" or "Deaths", there is an option on the top left corner that lets us "Align outbreaks", making all the selected countries time series aligned by their starting point.



Data Explorer

Download the complete *Our World in Data* COVID-19 dataset.

METRIC: Confirmed cases

INTERVAL: 7-day rolling average

☑ Relative to Population

☑ Align outbreaks

Sort by: Total cases (per 1M)

☑ Andorra	200.44k
☑ Argentina	115.91k
☑ Spain	107.15k
☑ Brazil	101.79k
□ Montenegro	227.38k
□ Seychelles	224.58k
□ Georgia	177.72k
□ Czechia	162.96k
□ Serbia	162.94k
□ San Marino	162.89k
□ Maldives	160.93k

Daily new confirmed COVID-19 cases per million people

Shown is the rolling 7-day average. The number of confirmed cases is lower than the number of actual cases; the main reason for that is limited testing.

□ Zoom to selection

☑ Hide countries < 1 million people

Source: Johns Hopkins University CSSE COVID-19 Data, Official data collated by Our World in Data

CC BY

Jan 28, 2020 – Oct 28, 2021

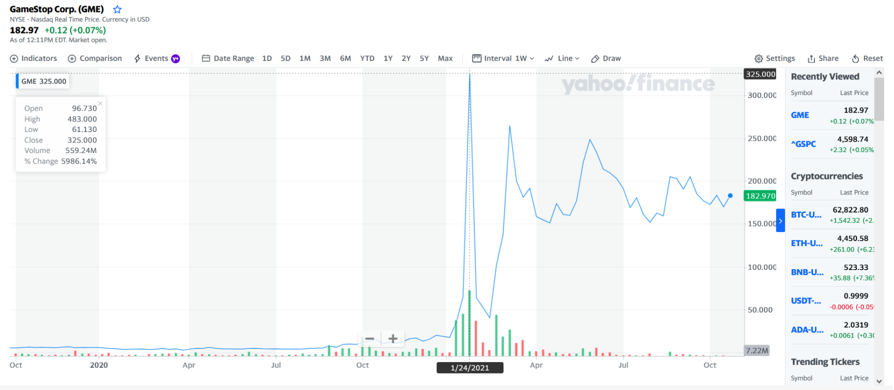
Figure 9: Graph showing the Daily new confirmed COVID-19 cases per million people from the "Our World in Data" COVID-19 Dashboard. In this instance, we checked the option "Align Outbreaks" and all time series are align by their starting point. Additionally the time scale is measured in days after the start, instead using the date. Besides aligning the outbreaks, the user can change the metric being viewed, change the sampling and switch between absolute and values relative to the countries populations. Screenshot from [20].

2.4.2 *Stocks dashboard*

In the stock exchange context, one of the most important focus is to study the behaviour of the price of the stocks overtime. Usually this is presented either in a table or, increasingly common, with a visualization. The visualizations, usually takes form as point, line or bar charts. It is also common to find combinations of all these and also other types, example in figure 10. Another important interest in stocks is to be able to predict the future of a stock price and also to compare it with other stocks. This is usually done with the aid of indexed charts, more details Section 2.4.3.

Another event which prompted the surge of multiple visualizations on a daily basis was the "GameStop Short Squeze" [22]. Without going into too much details, it was a matter that saw the stock price of a brick-and-mortar games store climb from 30$ to 325$ in less than a month. This rise was attributed to a dispute, sometimes described as a "David-vs-Goliath" situation. Goliath representing the Hedge Funds that were betting on the price drop and, supposedly, David being the every-day internet users that got together on the Reddit platform, to stand up against the giants, protecting the GameStop brand and possibly profiting in the process.

As the rise in the stock price was believed to go up indefinitely, until one of the parties gave up, at its peak, users of the subreddit were posting new graphs every day, comparing it with other past events and trying to predict the outcome of the situation. During this


GameStop Corp. (GME)
NYSE - Nasdaq Real Time Price. Currency in USD
182.97 +0.12 (+0.07%)
As of 12:11PM EDT. Market open.

GME 325.000

Open	96.730
High	483.000
Low	61.130
Close	325.000
Volume	559.24M
% Change	5986.14%

Recently Viewed

Symbol	Last Price
GME	182.97 +0.12 (+0.07%
^GSPC	4,598.74 +2.32 (+0.05%

Cryptocurrencies

Symbol	Last Price
BTC-U...	62,822.80 +1,542.32 (+2.
ETH-U...	4,450.58 +261.00 (+6.2
BNB-U...	523.33 +35.88 (+7.36
USDT-...	0.9999 -0.0006 (-0.05
ADA-U...	2.0319 +0.0061 (+0.3

Trending Tickers

Symbol	Last Price

Figure 10: Stock chart from Yahoo! Finance dashboard. Here we see a visualization of the stock price of a company, over time. This chart combines lines and bars. It allows the user to modify the visualization with the tools displayed on top bar: Changing the time scale, the time range and the appearance of the visualized items. It also allows the user to draw and annotate over the visualization. In the top left corner we have a small area with textual information about the time were our mouse is hovering, show by the dark rectangle at the bottom. Screenshot from [21];

period, the subreddit grew from 600,000 to 11,000,000 users and a good portion of the media shared was related to or involved InfoVis.

Both this event and the COVID-19 pandemic, in differente scales, showed how much people are ever more relying on InfoVis tools to communicate large quantities of (complex) data, and also how wide spread it is being used. More importantly, the data presented for both events is time-based data and predicting future behaviours was a main goal.

2.4.3 *Indexed charts*

Indexed charts [23, 24] are line charts that show relative changes in a group of time series relative to a reference point. To do this, we transform the Y values into a more comparable metric. This can be done by displaying all the values in relation to a specific value, be it a maximum, a minimum, an average etc. In both cases aboce, the reference value is the price that each time series has at the selected time. This makes it easier to compare relative change, not only within itself but also with other lines. This transformation can also be called normalizing the data, although normalization usually involves a static value.

One of the most common uses of these charts are in Finance, commonly used to track changes in stock prices.

As show in figure 11 we have several lines plotted. Each line has its own color and is named after one of the stocks companies. The horizontal axis marks time. Instead of displaying the absolute values for the price, as in a normal stock chart, the prices are displayed relative to the price of each stock, at the time selected. The vertical axis shows the “gain/loss” ratio. With interactivity, we can move the time back and forth and change the reference price.

In our project, we use this idea of indexing the data according to a value in time but we do not normalize the RUL values relative to that point in time. Instead, we normalize time, like described in Section 2.4.1. With this, for a selected RUL value of a reference time series, we shift the other time series back or forth so that all are aligned and present the selected RUL at the same time as the referenced time series. More details in Section 4.2.1.

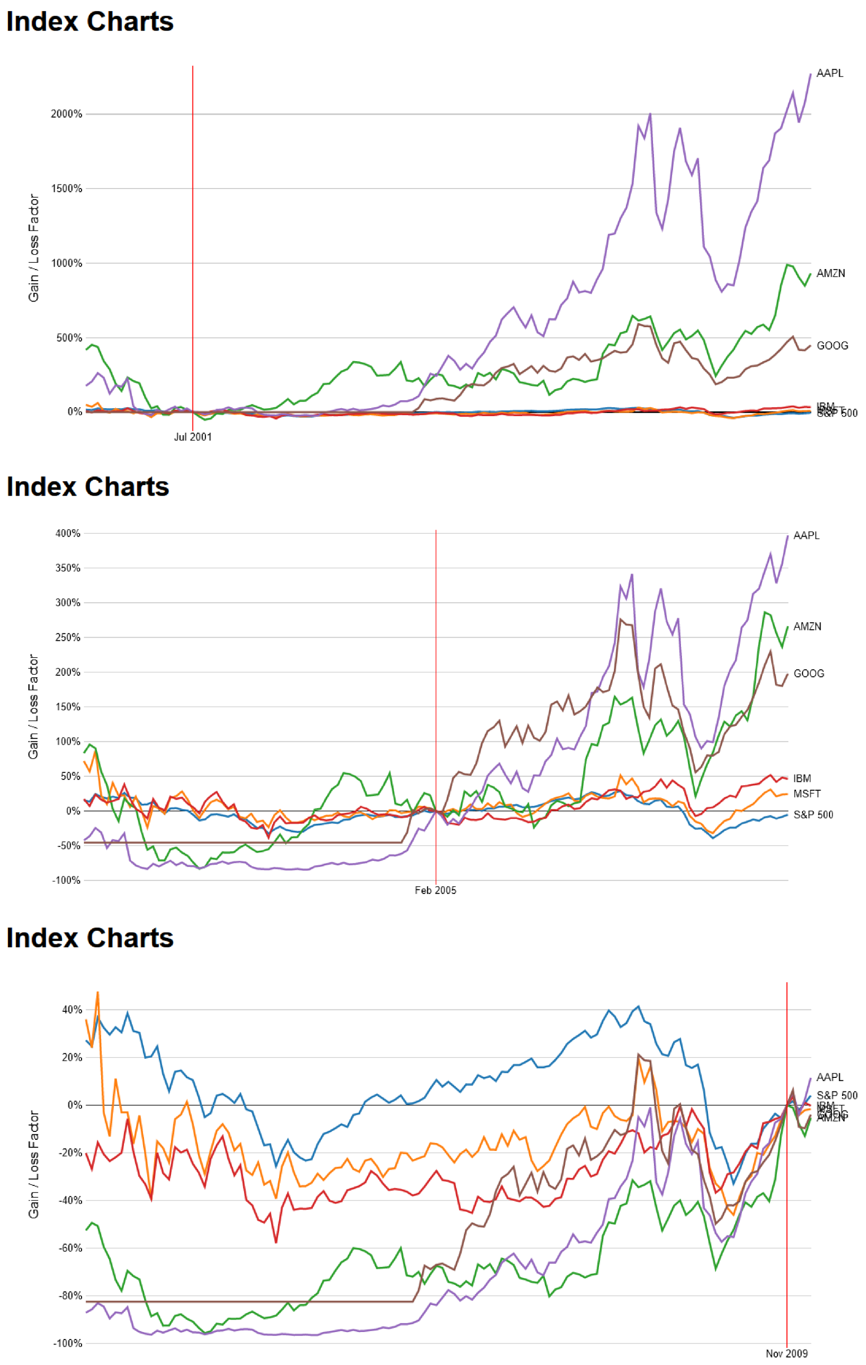


Figure 11: Indexed chart in three different moments in time. As we can see, all prices "align" at the point we select (red line). This way it is easier to compare growth rates between the different stock prices. Screenshot from *Protovis - Index Charts* [23]

3 WORK PLAN

In this chapter we will describe the work plan and organization and we talk about other parts and members of the ReMAP project that relate with our task. In Section 3.1 we describe the stakeholders interested in our project as well as the ReMAP teams more closely connected with our tool. In Section 3.2 we describe the objectives of the investigation. In Section 3.3 we describe the methodology used for the development of the tool and we describe how our project relates with the techniques described in Chapter 2. Lastly, in Section 3.4 we report on the proposed work plan and how it actually happened.

3.1 STAKEHOLDERS

The ReMAP project [1, 25] is a major project, comprised of several teams, with members from different countries, universities, companies and members from the airline industry. The project is divided in 9 different Work Package (WP)s, each with its own work focus. The tool developed in this project is proposed within the WP5: System Level Diagnostics, Prognostics and Health Management. This WP is focused on the development of methodologies for system and component level diagnostics, prognostics and health management for data and physic-based approaches. Along with WP4: Structural Health Management - Diagnostics & Remaining Useful Life prognostics, the resulting data is used as input for WP6: Maintenance Decision Support Tool, so that maintenance and scheduling tasks are preformed according to the aircraft components condition. This tool is proposed as (part of) the link between WP6 and WP6, Task 5.4: Actionable information extraction and visualization for optimal maintenance.

Besides the project members and the European Commission, other stakeholders include Airline companies as well as other logistic related companies or organizations which need to manage fleets (trucks, ships, cars, etc). Anyone developing prognostic models, as well as models for condition based maintenance scheduling.

3.2 OBJECTIVES

Our main objective is the development of a Web-based tool for the Visualization of RUL Prognostics. This tool needs to display time-based prognostic data for all aircrafts' components in a fleet, simultaneously. Its most important purpose is to support the WP6 Maintenace Decision Support tool, by enhancing and augmenting the RUL prognostic comprehension.

One of the objectives for this investigation was to follow Munzner [3] guidelines for an abstract analysis of the problem as well as applying the resulting classification to the ReMAP context. Another goal was to try and stay close to the visual representation already used within the PHM context, as described by NASA, mainly in order to not add a new layer of complexity or unfamiliarity. Additionally we aim to alter the technique used in the indexed charts in order to get a result more appropriate for the problem in hands. As is common in charts depicting the price history of companies stock, we wish to replicate the same organization for the outmost level of detail, therefore we will use vertical navigation for browsing the items, (aircraft and systems) and will show their details and respective visualizations horizontally (RUL and the other attributes).

3.3 METHODOLOGY

As described, this project embodies the link between two different WPs, WP5 and WP6, possibly three as WP4 end product can also be a RUL prognostic. Every week there were WP5 meetings, one week just the local team and in the following, the whole team. These meetings were for updating the team on the members work progress, to trade feedback and ideas as well as to establish the following objectives for development.

In an early stage, given the unforeseeability for when there would be real data available, we decided that there would be a focus on producing usable mock-ups quickly. For two reasons: first we wanted to have a "base" visualization ready to use as soon as we would get the data, so that it would only require minor changes, or none, to be able to see the new data. Secondly there was a need to learn the D3.js library so the early usable mock-ups also served as a learning aid, again, so that we could be ready to explore the real data as soon as possible.

As there were a lot of open questions about the data, its availability and its format, the early stages were as push in different directions simultaneously and was mainly focused in experimenting, exploring, iteratively refining the the ideas that work.

As such, the method of work resembles the growth process of a Slime mold, show in Figure 12: First there is an early growth in every direction, looking for food. After a certain period this growth stops and the mold starts shrinking its size, maintaining only the pathways necessary to reach any food it found. As these paths are refined / established, the process repeats [26, 27].

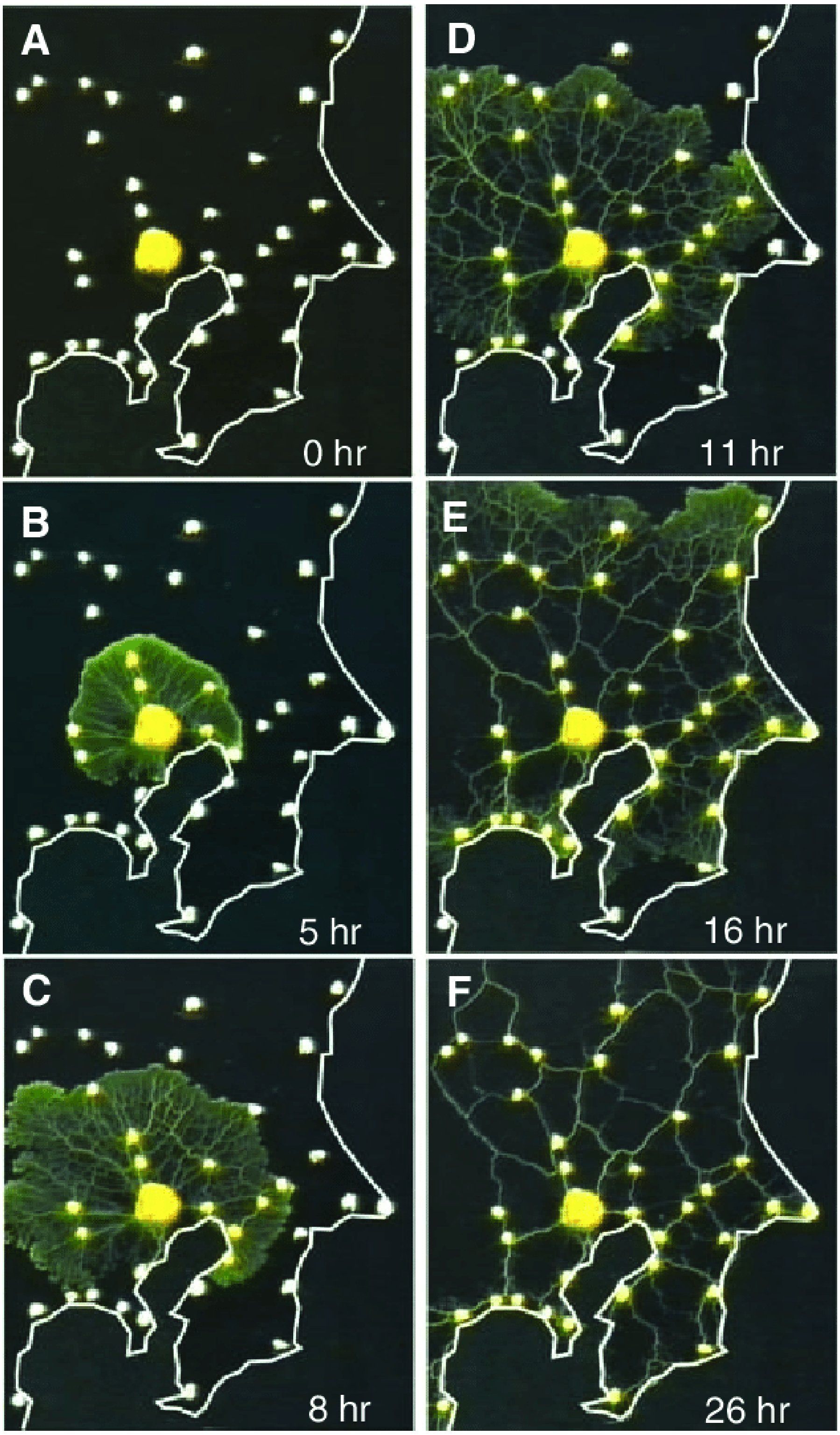


Figure 12: Summarized view of Physarum polycephalum growth pattern. Available from: `https://www.researchgate.net/figure/Network-formation-in-Physarum-polycephalum-A-At-t-0-a-small-plasmodium-of-Physarum_fig1_41111573` [accessed 31 Oct, 2021]

In the project context this works as process with 3 steps, exploration, refinement and feedback. The feedback step represents the weekly meetings, and this step is repeated after any of the other two.

Therefore we grow in "every" direction - build one or several mock-ups with certain characteristics or implement certain capabilities. Then there is a feedback moment where we decide which capabilities are good or useful and which are not. Then the mock-up is adjusted accordingly, we refine and hold on to the interesting points, followed by another feedback stage. When we establish a satisfactory point of development or refinement, we can repeat the process from the beginning and explore new paths.

Concretly, we start with an analysis into RUL estimation, so as to understand the problem context, the flow of the data and the visual language that is already commonly used in the PHM context. After a brief round of graphic mock-ups, focused on the overall visual structure and organization of the data we quickly started building usable mocku-ups, implemented in html and javascript, and working on a set of synthetic data to feed the visualization.

The first task was to find a suitable framework for displaying the RUL prognostics of several components of the same aircraft. At this point, we were able to receive a small set of real prognostic data and the focus shifted to finding a suitable representation for RUL prognostic and its behaviour over time. Aditionally we also started exploring prognostic quality and how to visually compare the different trajectories, building upon the technique described in the indexed charts. After having very interesting results with comparing trajectories in the real data, the following task was to model and generate a new dataset that would look and behave like the data received. After this we focused on building a new mock-up which could generate new datasets, on demand, so that we it was easier to see how it worked under different conditions / different prognostics. With prognostic models reaching new stages of development, it was specefied that the prefered RUL format would be as a PDF. Instead of generating a new dataset with RULs as PDF we use the resulting distribution of the trajectory comparison to derive a PDF, ending up with a result visually similar as the one shown in Figure 21 [28]. For last, the framework was changed so that it could display prognostics for a whole fleet and not just one aircraft.

3.4 WORK PLAN - PROPOSED VS. REAL

The proposed plan assumed that we could have access to real prognostic data sooner and in greater quantities. In reality we could only access Prognostic data from two models running on a synthetic datasets. Further exploration into visualizing components relationship, for in-

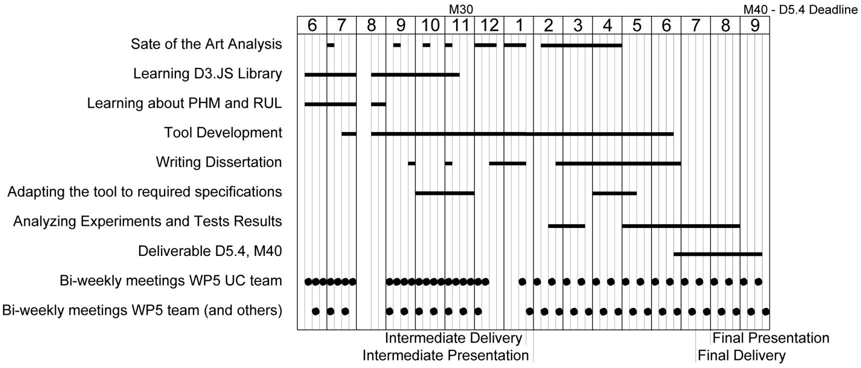


Figure 13: Proposed scheduling of tasks and goals.

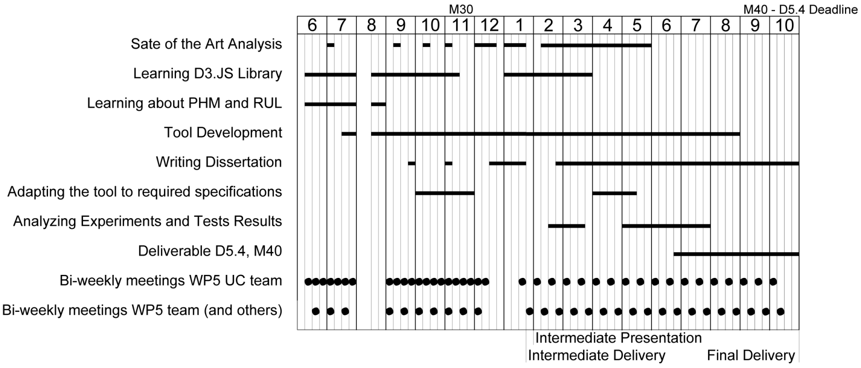


Figure 14: Calendar with actual tasks realized.

stance in terms of dependency were not possible. We also did not explore Minimum Equipment List dependencies.

The mentality of rushing usable mock-ups meant that the tool was ready to experiment on the real data just in time to complete the task and deliverable 5.4.

This explorative and practical approach helped with learning the d3.js library and allowed use to try several different approaches.

On the other hand, the theoretic research needed a more informed and organized plan. Without it, there were some delays and difficulties in producing the final reports.

4 VISUALIZING RUL PROGNOSTICS IN AIRCRAFT MAINTENANCE

In this chapter we will talk about how we are going to visualize RUL prognostics. In section 4.1 we briefly describe the different steps in Prognostics and Health Management in order to contextualize the problem of RUL estimation. Then we identify and classify the data, data structure and tasks As for the most of the duration of this project there was not much RUL data available, it was necessary to model and generate data that would mimic real RUL prognostics, in section 4.1 we explain how we modeled and generated the synthetic data.

Finally, in section 4.2 we describe the framework of the tool. We explain its different levels of detail as well as the search and filtering capabilities.

4.1 PROGNOSTICS AND HEALTH MANAGEMENT

RUL prediction or RUL prognosis is just one process within a bigger system, referred to as Progostics and Health Management (PHM).

Broadly this system can be separated into 5 (or 6) steps [28]:

1. Acquiring sensor data from an aircraft systems' components, relating to their condition like temperature, vibration etc.

2. Feature Extraction. As we're dealing with very large amounts of data, since some sensors are taking measurements several times per second, it is necessary to understand which data is going to be helpful or not at establishing the components health condition.

3. Diagnostics. After selecting the right data sources, the data is used in diagnostic algorithms that give us an estimate for the components health state.

4. Prognostics or RUL estimation. The data and its estimated health state is then used in Prognostic algorithms which will infer on how long the component can still be used, that is its Remaining Useful Life.

5. Decision/Management. After having an estimate on how long a component can still be used, this information is then used by yet another algorithm which deal with the maintenance planning or scheduling.

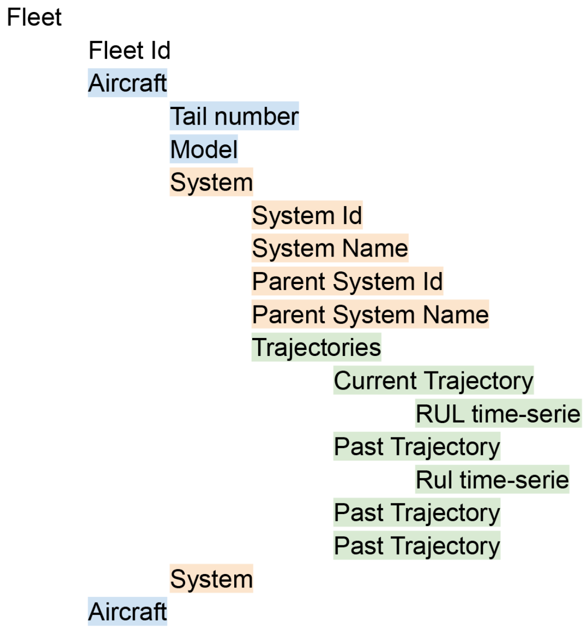


Figure 15: Cascading schematic of the dataset semantic hierarchy

In the context of the ReMAP project, the RUL estimation is thought of as happening at hourly intervals. Therefore we can assume the data type is some kind of time series. Different prognostic approaches can present different RUL formats, as there were different approaches under development, three RUL formats were considered: Single Value, Interval, and for last, PDF.

Semantically, the overall data structure of the modeled problem is the following: A fleet has a list of Aircraft; Aircraft have a list of Systems; Systems have a list of trajectories and trajectories are time series composed of timestamp-prognostic pairs.

This data structure is encoded in a dataset of the type of multidimensional table. Semantically, our key is composed of two attributes, the timestamp and the system identification. If the systems id is the same across different aircraft then it it is necessary to use the Tail number as a key attribute as well. All other attributes are dependent on this pair/trio. Therefore, the data is classified as a multidimensional table with time-varying data. The list of attributes generated/used for the tool simulation is the following:

- Timestamp – Key
  - Ordinal sequential attribute
  - Temporal attribute
  - Integer or Date
- System Id – Key

  - Ordinal sequential attribute
  - Integer

- System name
  - Categorical attribute
  - String

- Parent System Id
  - Only if exists
  - Ordinal sequential attribute

- Parent System Name
  - Categorical attribute
  - String

- RUL (or Virtual Health Indicator (VHI))
  - Quantitative attribute
  - Float or [Float,Float]

- Tail number
  - Quantitative attribute
  - Integer

Another attribute may be included, or it can be generated, if necessary, namely the Trajectory ID or Index. That is some attribute that lets us identify which trajectory does the prognostic pertain to. If all the prognostics are stacked in a time-stamped list, then trajectories are derived and numbered based on their age. This attribute is also important to distinguish the current trajectory, the one we're trying to predict, from its past relatives. The RUL attribute can be received in any of the aforementioned formats. If a VHI is used instead, then it is preferred to be as a single value. When the RUL is given as PDF, if it follows a normal distribution, only the mean and standard deviation are necessary. The Visualisation tool also derives a new RUL attribute, referred to as historical RUL, and it can take the same formats as the input RUL.

As broadly described above, RUL estimation is followed by a scheduling step. In this step other algorithms use the RUL data to generate schedules for the aircraft's flights and maintenance operations.

Aircraft are very expensive and complex systems. Although statistically, are one of the safest traveling options, its also one of the most regulated. In order to be so safe it is common for there to be several degrees of redundancy in order to prevent dangerous scenarios. This is also the reason why preventive maintenance is the most common strategy used. As such, it is not surprising that the scheduling can't be done autonomously and instead requires a technician to generate and analyse the schedules to make sure that everything is as should be. As mentioned earlier, one of the goals of the RUL visualization tool is to allow the users to get to know deeply the RUL prognostics so that they can know if and how much they can rely on the generated schedules and also to identify possible corrections. As such this tool is expected to be used as an interface for the prognostic data and its expected to be used as one would use a glossary or dictionary, so that the users can quickly refer to the information they need while doing their tasks that rely on RUL data. The visualization tool is expected to be of utility not only for the people working in the scheduling phase but equally for the people devolping the RUL prognostic algorithms.

- Identify which components need intervention and when
- Identify priority actions
- Identify the amount of specific spare parts need in a given time period
- Identify the probability of a system failing within a time period
- Visually compare current trajectory with past trajectories

Other mid and low-level tasks and needs include:

- Query prognostics for specific Aircraft or Component(s)
- Query prognostics which are believed to happen in a specific time period
- Mixed queries
- Analyse prognosis accuracy
- Go back to past prognostics

Under the what-why-how classification, we can identify four of the high level user goals: to Discover, Present, Derive and Record information. Regarding the type of search, all types are identified: Lookup, Locate, Browse and Explore. As for the types of query, all are present: Identify, Compare and Summarize. Finally, the type of targets identified were: Trends, Outliers, Features, Distributions, Extremes, Dependency, Correlation and Similarity.

As there are tasks identified in all the abstract branches, it strongly indicates that only one idiom might not be enough to provide for all the user needs. This is further supported by early experimentation, shown in Figure 16, Figure 17 and Figure 18, as some idioms couldn't show more than one aircraft, and others had several readability problems.

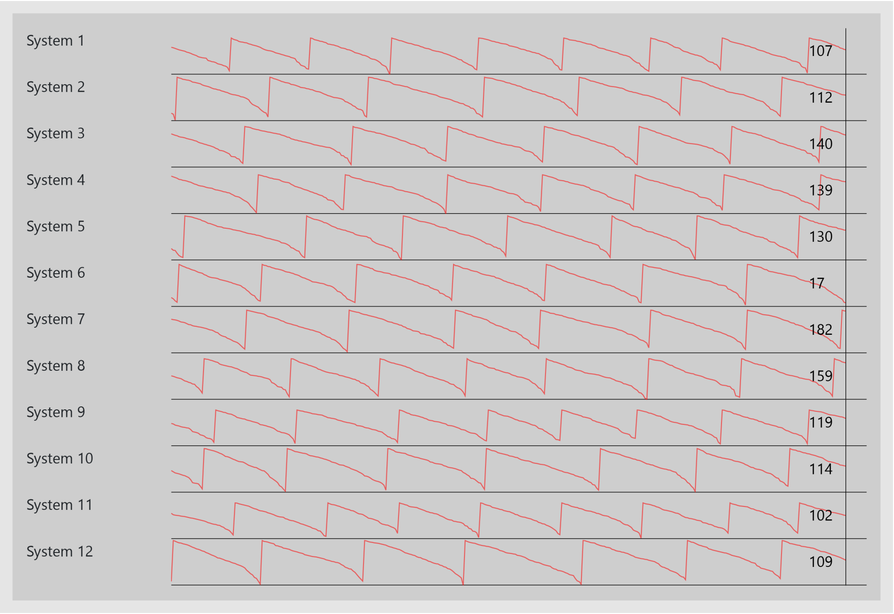


Figure 16: Early mock-up showing only past information. Trajectories are all grouped in one curve.

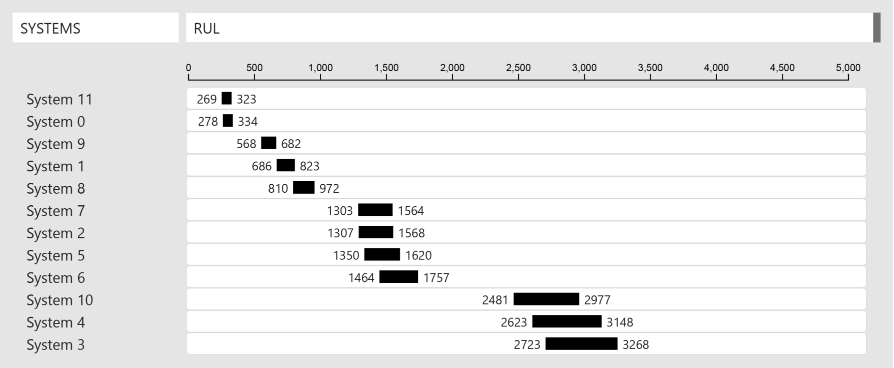


Figure 17: Early mock-up showing only Current information displayed in the future time.

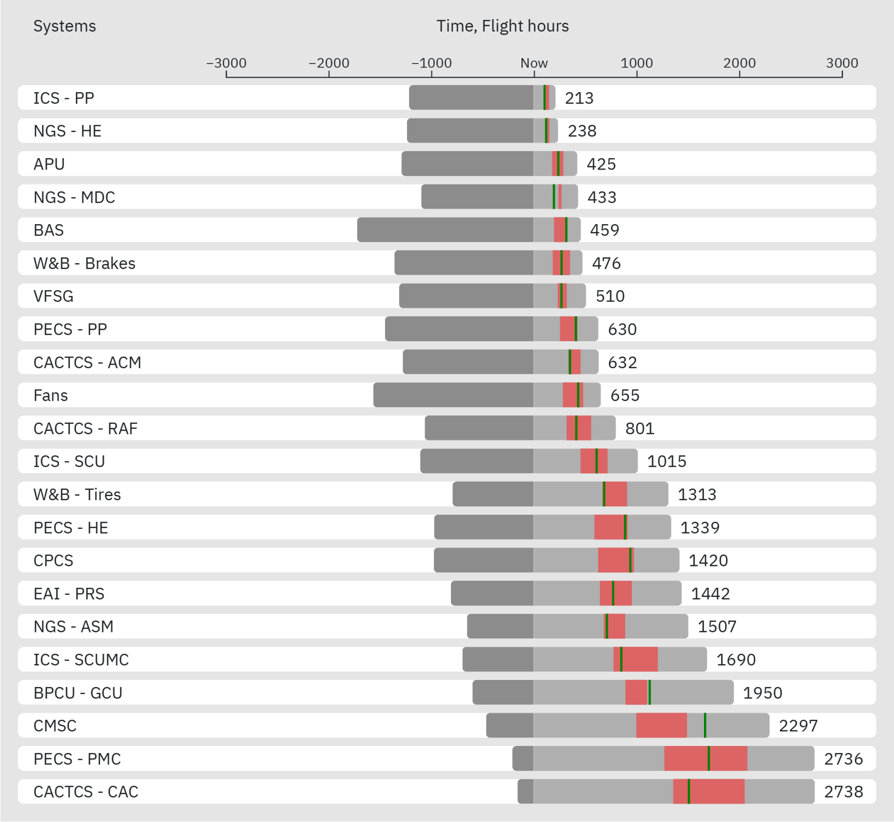


Figure 18: Early mock-up showing past, present and future information, but only for one aircraft.

4.1.1 *Synthetic data Generation*

In order to implement and test different idioms in a quick manner, it was necessary to produce synthetic data that simulated the degradation experienced by the components or the RUL estimated by the models. Based on an abstract analysis on the basis of RUL estimation and preliminary RUL prognostic results on a Cabin Air Compressor Temperature Control System (CACTCS) system, the data generation was also modelled after a mix of two analogies, the autonomy of a car and of a phone. A phone's autonomy is broadly presented as a percentage of charge left; some phones also couple this with an estimate of time of use remaining. Some cars also present a separate autonomy measure that gives a rough amount of kilometers we can still drive, based on the amount of fuel left and average fuel consumption. The trajectories received were all monotonic functions. That is, they started at a high RUL value and as time goes on, the RUL only decreases. Unlike the real state of the degradation, the RUL ground-truth can only be inferred after the End of Life (EoL), this means that for all we know, all instance RULs might be wrong or just slightly imprecise, as they are dependent on future use to be accurate. Although we do not use our cars' fuel or phones battery until they reach 0, this notion of a prediction that is most likely bound to be corrected is present and is used in the data modulation. With this, trajectories are generated by first selecting a random value close to an average component's life. More specifically, the first prognostic of a trajectory is generated with 2400 to 3200 flight hours of Remaining Useful Life. At the same time, it's created the overall degradation rate, which is the value that is going to be subtracted from the RUL every hour. It starts at 1. Then, a random flight duration is selected, ranging from 2 to 18 hours. Along with the flight duration, we decide randomly if the degradation rate will increase or decrease, every hour, for that flight. Then, the degradation rate will either increase or decrease by an amount that is randomly generated every hour. Higher values mean that it can change from a low degradation rate to a high one very quickly. Different limits for the hourly change in degradation rate are shown in Figure 30.

The hourly degradation is limited to a range of 0.9 to 8 hours lost per hour flown. Once the flight duration is over, a new flight duration is selected, and the cycle repeats until RUL reaches 0 or another established threshold. After a minimum number of trajectories is generated per system, the most recent one is trimmed randomly and considered the current trajectory we're trying to predict. Then, the system and aircraft identifications are randomly selected from a set list and the hierarchy is established. The number of systems and aircraft for which trajectories are generated can be configured.

As is modelled, our RUL is almost always higher than it should be, at any given time. Therefore it is not considered as an accurate RUL estimation but as an Health Indicator (HI) instead. Regardless of this inbuilt error, the derived historic RUL still manages to be (relatively) accurate and to react to drastic and unexpected changes in the input RUL behaviour.

More details are provided in the next section. The generated data has been extremely valuable for the development of the Visualization tool, and the resulting datasets, joining the generated synthetic input data with the derived historical RUL, are being used to experiment with scheduling models within WP6.

4.2 FRAMEWORK

As mentioned above, early experiments with simpler idioms didn't quite manage to provide for all the user tasks identified, therefore the visualization consists of two idioms embedded in the same visualization, with three different hierarchical or detail levels which present both the context and detail on demand.

The user experiences the tool starting from the highest level and can progressively uncover the lower levels, which contain more details and more visual information. the lower levels start hidden/-collapsed. At the starting point, the user sees a list of aircraft and, for each aircraft, their corresponding RUL and a graphic, showing a combined PDF, on a time scale, which tells the user all instances the aircraft is expected to require maintenance. Suppose the user is not satisfied with this level of detail or he needs information regarding specific systems. In that case, the user can click on any aircraft and expand the space it takes on the screen, revealing a new embedded visualization. This mid-level presents the user with all the aircraft's systems prognostics. The user is presented with the list of systems, and like in the upper level, each system is coupled with its corresponding RUL and a graphic with the PDF. In this level, the aircraft combined PDF is split into its parts, and the user can interpret each system's PDF individually. If this detail level is still not enough, the user can click on any system again and expand the area it takes on screen, revealing yet another embedded visualization. This lowest level is the richest in detail and in data/information. In this level, the user sees textual information about the system and the current prognostic. He also has a graphic that shows the current trajectory and compares it with past trajectories of the same component. Figure 19 represents all the visualization levels.

Here the user can analyse more deeply the behaviour of the system over time to find trends and patterns which might either reinforce his confidence in the scheduling generated down the line or, on the other hand, realise that, for instance, a certain trajectory's behaviour

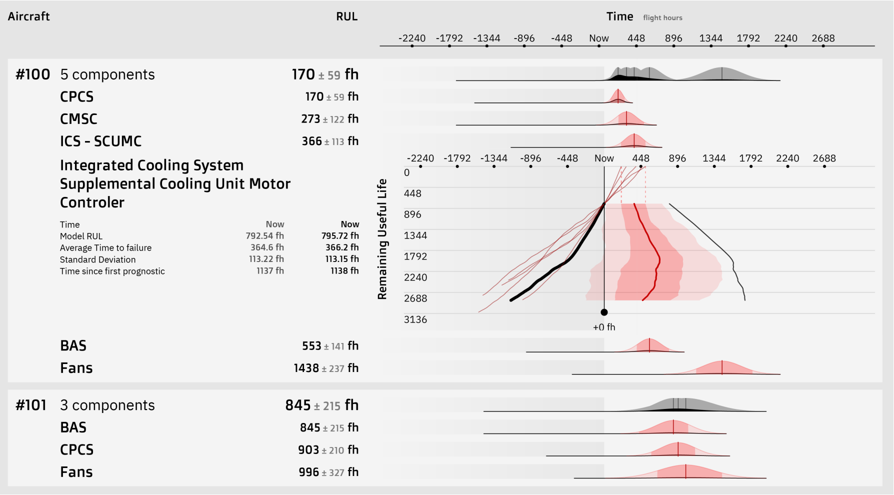


Figure 19: All three visualization levels working at the same time. They all share the same horizontal axis. The fleet level has the main horizontal axis at the top, visible in all detail levels.

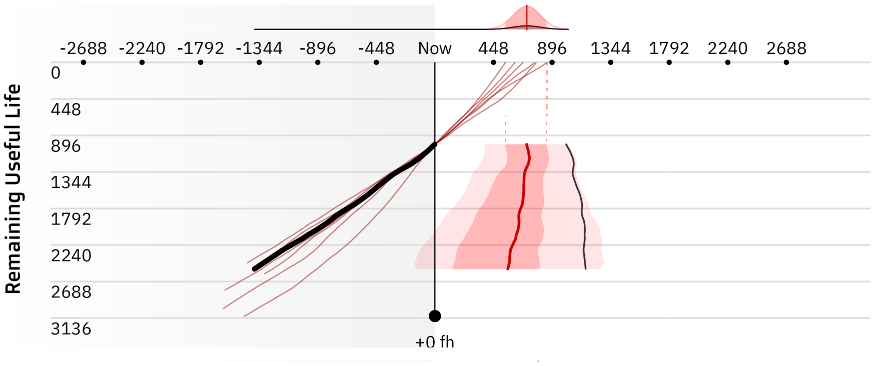


Figure 20: System level view. Current trajectory as black curve, past trajectories as thin red curves, and PDF as red areas.

has never been seen before or is extremely rare. Therefore, he should be more cautious with the auto-generated schedules. As each level also validates or justifies the level above, they will be detailed from the bottom up.

4.2.1 *System level*

At the System level's lowest hierarchical level, represented in Figure 20, we have a composite visualization with two concurring idioms sharing the same horizontal axis. On the top, we represent a PDF, the historic RUL. On the bottom, we have the current and past trajectories' time series aligned in a manner that allows us to derive the historical RUL PDF.

In the lower plane, both axes are measured in flight hours. The vertical axis is mapping the RUL, and the horizontal axis maps the time

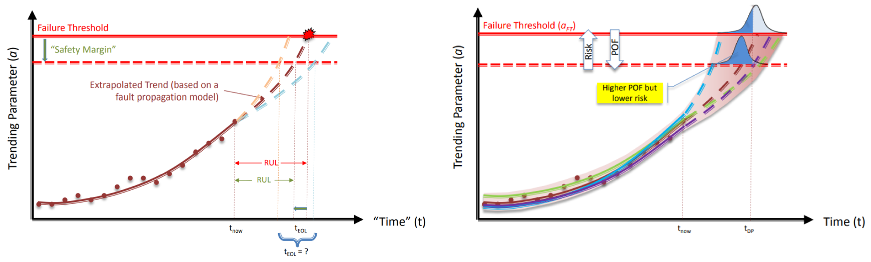


Figure 21: Examples of the idiom presented in NASA Prognostic Tutorial. Left: Comparing similar trajectories with current trajectory; Right: Representation of the resulting PDF in the same idiom

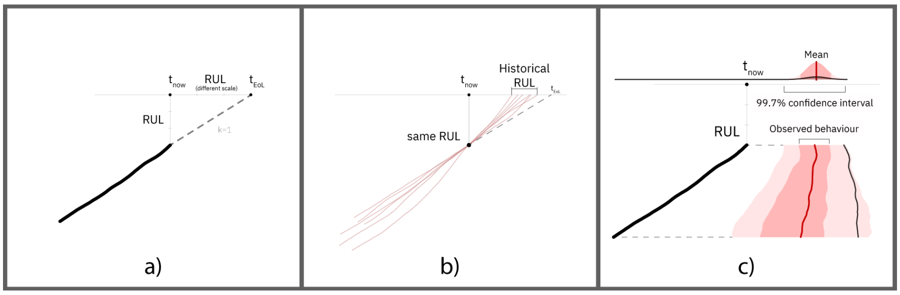


Figure 22: (a) Current Trajectory and EEoL; (b) Past trajectories aligned at the same prognostic, resulting in the Historical RUL; (c) The PDF temporal trace is aligned with the time axis and the Current Trajectory.

relative to the present moment - "Now". This idiom tries to deviate very little from the one presented in Nasa's Tutorial Prognostics [2] shown in Figure 21 and allows for calculating the RUL(based on/) of a "trending parameter".

In our case, the trajectories are not selected based on similarity or any metric, and their starting point does not align with them. Instead, our simulation assumes that, for instance, the 7 most recent trajectories of the same component are being compared with a current one. They are aligned by the point in which their prognostic was the same as the current trajectories most recent one. For an RUL of 500 flight hours, we look back and check, on average, how many hours those past trajectories flew after having the same prognostic. Like in NASA's, the derived RUL is measured in the horizontal axis, and a PDF is generated based on the observed trajectories behaviour, giving a sense of where, in the time axis, the trajectory is expected to end. Since our trajectory is already composed of RUL estimations and is measured in flight hours, we can compare the observed behaviour with the expected behaviour and compare it with the historical RUL just derived.

In our idiom, Figure 22, the current trajectory is represented as a black curve. The past trajectories are represented as thin red curves. Instead of mapping the current trajectory from a starting point into

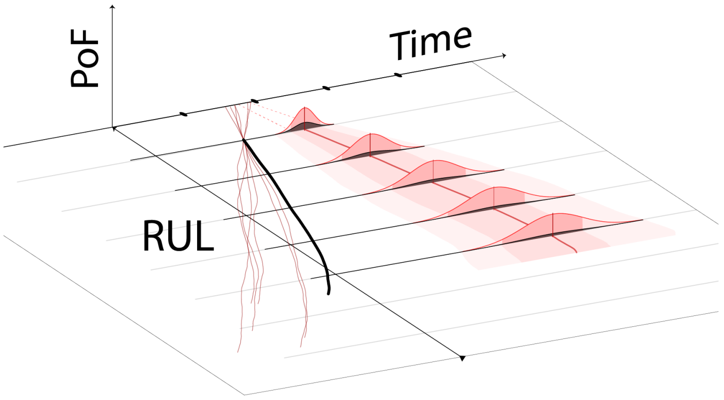
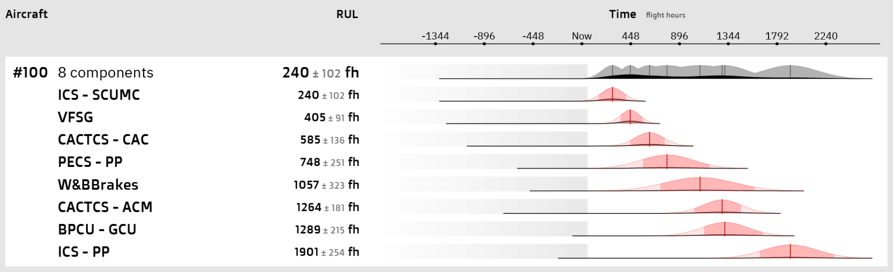


Figure 23: 3D view of PDF behaviour over time, traced on the horizontal plane.

the future, we map it from its end, in the present moment, into the past as in a real time stamped axis, and only the past trajectories are mapped abstractly on the time scale. The black dashed line shows the point where we would expect our component to fail based on the input RUL. In other words, if at the "Now" moment the trajectory presents 100 flight hours of RUL, the dashed line will end 100 hours after the "Now "point. As mentioned before, the ending points of the past trajectories are used to generate a new RUL as a PDF. This means that for every input RUL we will have an output PDF. In order to show the full output behaviour, on top of showing the most recent PDF, we represent all of the past PDFs as well, but each as a colour coded line instead of as a bell-shaped curve. These lines then produce the coloured areas that are displayed exactly under the PDF. The analogy here is that the upper graphic is represented as a front view, while the trajectories graphic is considered as the floor plane, and we're viewing it from above, as seen in Figure 23. So, we stacked the PDFs exactly where we see them on the front view and aligned them with the relative RUL input on the ground view.

The darker red area represents the observed interval and corresponds to the shortest and highest time flown by the past trajectories. The light red area represents the 99.7% confidence interval according to a normal distribution derived from the past trajectories' time flown. The thick red curve represents the mean of the distribution and is looked at as the EEoL, point in future time we most expect the trajectory end. Generally, this last artefact will take on a shape resembling a pyramid. It starts wider, and as our confidence increases, or as the standard deviation decreases, the shape's width thins out, and it points more and more closely to the actual end of the current tra-



Aircraft		RUL
#100	8 components	240 ± 102 fh
	ICS - SCUMC	240 ± 102 fh
	VFSG	405 ± 91 fh
	CACTCS - CAC	585 ± 136 fh
	PECS - PP	748 ± 251 fh
	W&BBrakes	1057 ± 323 fh
	CACTCS - ACM	1264 ± 181 fh
	BPCU - GCU	1289 ± 215 fh
	ICS - PP	1901 ± 254 fh

Figure 24: Aircraft level view displays the combined PDF for the aircraft and all individual systems.

jectory. The different shapes and meanings will be discussed in the results section.

4.2.2 *Aircraft level*

In the mid hierarchical level, the Aircraft level, shown in Figure 24, the user gets an overview of all of an aircraft components prognosis.

The lowest mean RUL orders the systems and the lowest is attributed to the aircraft for aircraft ordering purposes. This lowest RUL attribution needs to be reworked for MEL restrictions to be included. In this level, the user sees only the PDF representations present in the lowest level and not the whole behaviour.

.

The PDF representation consists of 3 waves or functions, overlapped, and one line, as shown in Figure 25. These functions are plotted on an omitted xy plane, where the horizontal axis represents the time and is aligned (is the same) with other the horizontal axis represented just below and above in the other two levels. The vertical axis shows the Probability of Failure for a given time in the future. There are two red functions and one black/grey. The red areas and lines have the same meaning as described in the lower-level areas as they represent the same thing. The darker red function/area represents the observed behaviours, that is, the interval within which all of the shown past trajectories ended. The lighter function represents the normal distribution derived from the observed past trajectories. The black function is the exact same as the light red one, with the one difference being the scale of the vertical axis. In the light red case, the functions are scaled to fit the entire height of the plane. This artefact is presented to help the user gauge the shape of the distribution. In the black function case, the scale is the same across all visualizations shown, and these serve the purpose of allowing the user to gauge the actual probability values. The actual probability values are secondary in importance to the distribution's shape and the minimum and maximum values of a specific confidence interval. All of the systems PDF is then roughly

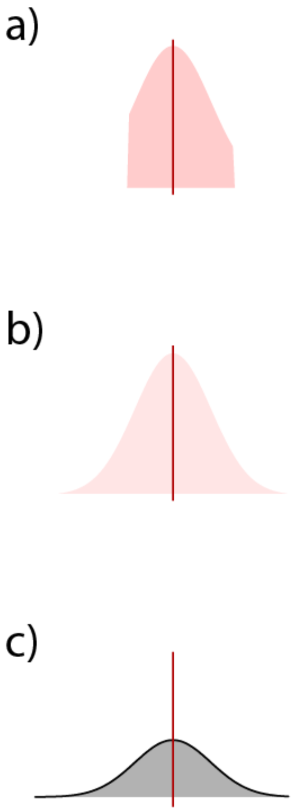
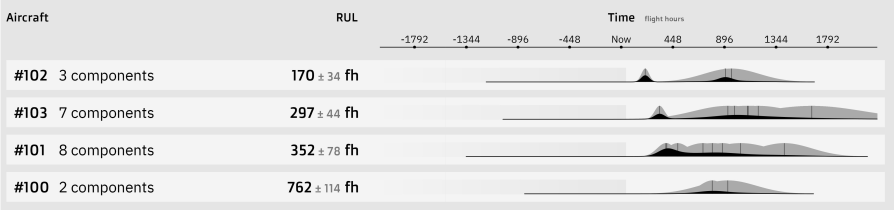


Figure 25: Decomposition of the RUL PDF (a) Observed Interval; (b) Normal distribution, 99.7% confidence interval, scaled to fit; (c) Normal distribution, 99.7% confidence interval, real scale. In all, the red line represents the mean.



Aircraft		RUL	Time flight hours
#102	3 components	170 ± 34 fh	
#103	7 components	297 ± 44 fh	
#101	8 components	352 ± 78 fh	
#100	2 components	762 ± 114 fh	

Figure 26: Fleet level view, displaying only the combined PDFs for the aircraft, their systems' lowest RUL, the number of systems and the aircraft tail number.

combined to create a PDF representing the different places in the time axis that we expect the aircraft to require maintenance.

4.2.3 *Fleet Level*

Finally, the highest hierarchical level, referred to as the Fleet level, is the first the user encounters and provides mainly summarised and abridged information from the previous levels. Here, as shown in Figure 26 the user will start by seeing all of the aircraft queried. Like in the aircraft level with the systems, the aircraft will be coupled with their relative RUL and the corresponding combined PDF representation mentioned in the previous section.

.

There are two combined PDFs represented, one lighter and the other darker. Like in the Aircraft level, the lighter PDF is scaled to fit the plane's height and give a sense of the shape of the function(s). This is done by representing the probability as a percentage of the highest possible probability (at the mean) and then stacking the curves; If there are overlapping probabilities, only the highest value is considered. All the systems mean RULs are represented as thin vertical lines to reinforce their location. The darker PDF tries, again, to portray some sense of the real probability, although in this case, it is done in a less rigorous manner. Several combinations were experimented with, involving the average of the curves, but the one who provided the most distinguishable visual representation was simply adding all the curves together. Although this doesn't translate into a real and logical concept, for instance, if systems overlap, the resulting probability might be higher than 100%. It is useful as a way to give context to the scaled to fit PDF curve and works as a visual reinforcement that there is something that needs to be looked into.

4.2.4 *Search Function*

The search function allows the user to individually search for specific aircraft tail numbers, models, components and time periods. In other words, the user can search for a tail number, and he will see all com-

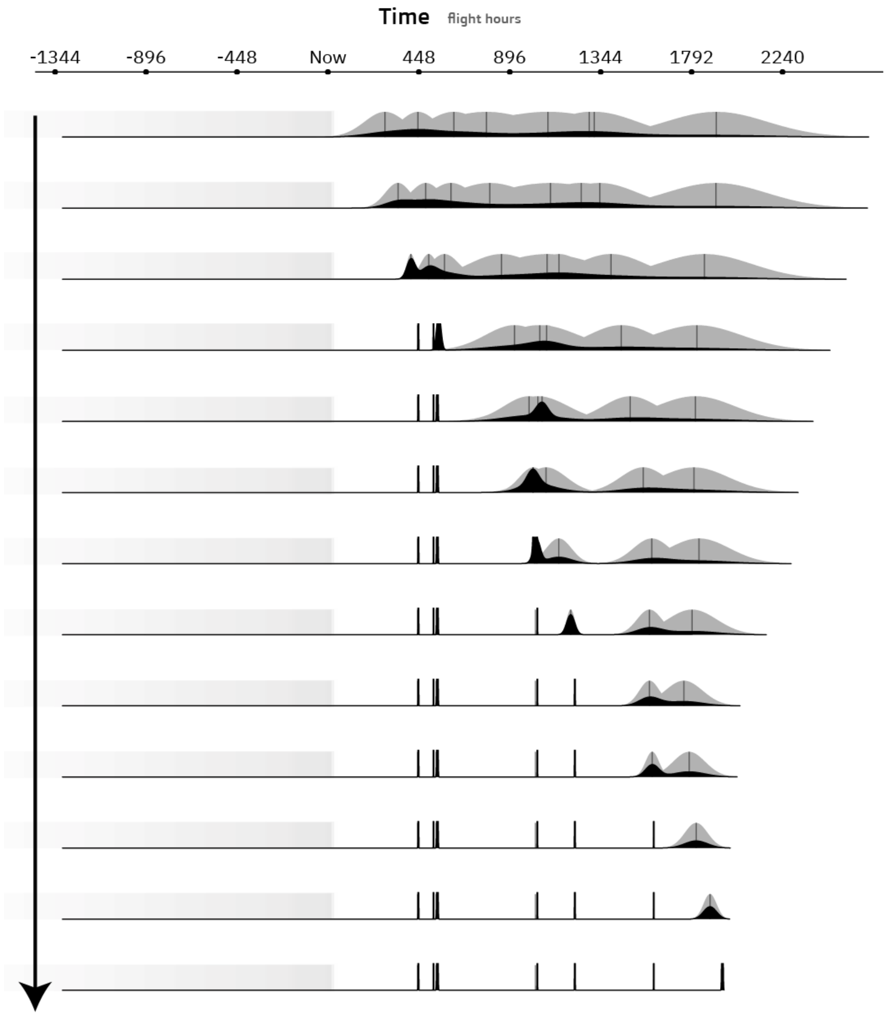


Figure 27: Time evolution of the combined PDF of one aircraft, starts at the first row and moves forward in time, going down the rows.

ponents which have prognostics for that specific aircraft. The user can search for a components name or initials, for instance, Integrated Cooling System (ICS) and all of the aircraft which have current prognostics for ICS will be displayed (and the systems as well). Additionally, the user can query a specific time period from 2 to 3 months in the future, and the tool will display all aircraft whose systems are expected to fail in that period (and not the systems that are expected to fail outside of it). The user can also mix and match the queries, i.e., show all prognostics for ICS and CACTCS. We can quickly know how many aircraft and components are being displayed/queried in the summary area, top left side.

Our visualization builds upon a Gantt chart structure which allows us to organize and navigate aircraft and components easily. More over, the visualization of the RUL derivation

5 EXPERIMENTATION

5.1 PRELIMINARY RESULTS

Early experimentation relied on two different types of datasets. There were the synthetic datases which were repeatedly generated and refined throughout the project and one small set with 10 trajectories, resulting from a real prognostic model, developed by another teammate, running on real sensor data from an aircraft component.

The real dataset was composed of 10 RUL trajectories in the format of single value. That is every time-RUL pair would tell us how many hours that component could still fly, at that moment.

All the trajectories in the dataset would start with a high RUL value and it would continuously decrease, and only decrease, until it reached 0 hours.

As this dataset was from a model in early development it was a established that the prognostics would not be very accurate. For example the first prognostic in any trajectory would be 1.5 to 2 times higher than the time the component actually flew. Regardless, this dataset was very useful because we could meaningfully experiment with trajectory comparison and think about ways to deal with less accurate prognostics.

For the trajectory comparison, we take the same principle explained in Section 2.4.3 and in Section 2.4.1, when we align the Outbreaks.

We move the trajectories back and forth so that they are aligned in a point where they all had the same prognostic. We can do this because we want to know how long are the trajectories, if they all starteded in that common point.

Figure 28 shows an instance of the method applied, relating the behaviour of one trajectory with all others.

For the experiment, we select one reference trajectory. We trim it at a random point in time and we save the respective RUL as our reference value. Then, we move the other trajectories horizontally so that they all have the referenced RUL aligned with the reference trajectory cutting point. Finally, we analyse the interval created by the ending points of the other trajectories and we check if the real ending point of the reference trajectory is inside or near it. The best scenario we can expect is for the reference trajectory to end exactly in the middle of the interval.

If the reference trajectory is always centered within the others, for its whole length, it means that the trajectory is quite similar to the

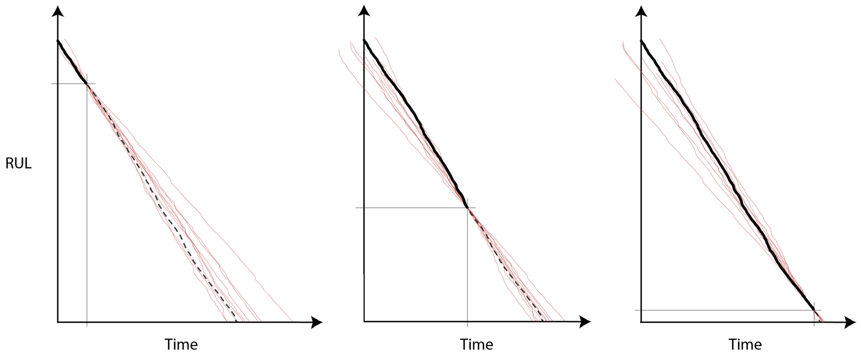


Figure 28: Comparison method applied to data from a real aircraft component. The black line represents the reference trajectory. The dashed line represents the real behaviour of the reference trajectory. In the context of the simulation it is the "unknown" future of the reference trajectory. The red lines are the other 9 trajectories. In the context of the simulation they are looked at as historical behaviour; for instance, the component's most recent past trajectories. There are 3 different moments represented ordered from left to right. We simulate that the dashed part of the line has not happened yet and that we are trying to predict it. We can see that in the three moments, the end of the reference trajectory is inside the interval; moreover, as we get closer to the end, the interval shrinks, meaning that we can be more confident.

group as whole, that is, the trajectory is similar to the average of the other trajectories.

If we apply the comparison for all the time instances and save the resulting intervals in a time series, we can then compare it with the reference trajectory and with the ground truth.

The ground truth is the real behaviour that the models are trying to predict. It can only be established after the trajectory ends and this poses an increased difficulty.

The ground truth can be modelled in two common ways: linearly or with an elbow point. A linear ground truth is modeled as a straight line, starting with the total number of hours the component operated, decreasing at a rate of one hour every hour until it reaches 0.

To deal with lower accuracy while far way from the end of a trajectory, the ground truth can be modeled with an elbow point. If we cant make accurate predictions until we are closer than 500 hours away from the end of a trajectory, we can make that threshold the elbow point.

With this, the ground truth starts as an horizontal line repeating the 500 hours value. When it gets closer than 500 hours from the end, the ground truth curves (like an elbow) and starts decreasing linearly, just like the linear ground truth.

Figure 29 shows a summary of the method applied to all trajectories at all instance RUL / time values.

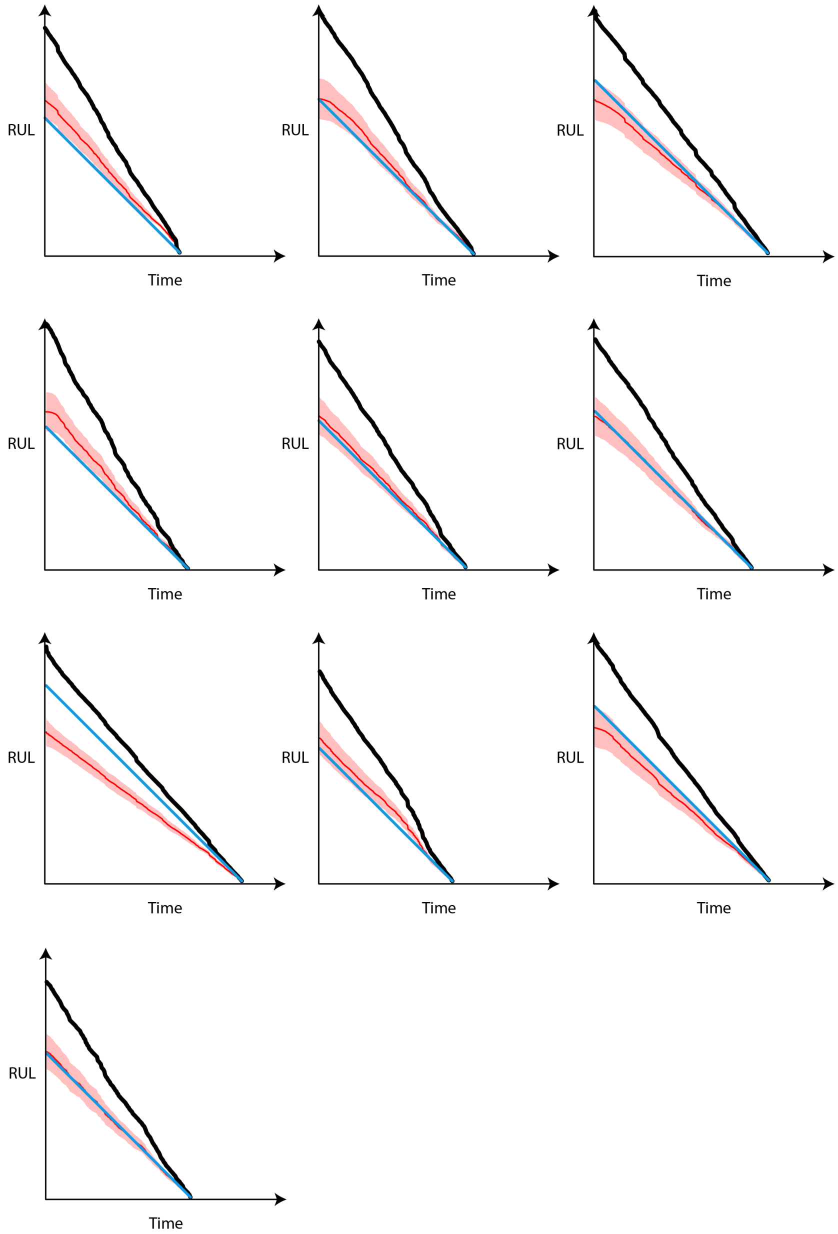


Figure 29: Summary of the comparison applied to every trajectory. The black line represents the reference trajectory. The red area is created by a band representing the interval of the other trajectories endings at the corresponding time. The red line shows the average value of the endings.

Additionally, we can see that the area of the intervals is quite close to the ground truth.

The synthetic datasets were generated to experiment and test different ideas. The most relevant experiments were related to the tool framework, and to test the comparison developed with the real data.

The trajectories were modeled to behave as the real data and were also monotone functions with the RUL as a single value.

Early experiments indicate that navigating through the components list horizontally works better and is easier to understand therefore future experiments all build upon the Gantt chart structure for the tools framework.

Trajectory comparison also showed very promising as we could tell with some degree of certainty where all the real trajectories would end at any given point.

5.2 EXPERIMENTAL SETUP

As RUL prognostic models matured throughout the project, new RUL prognostic data was made available, now with the PDF format. For the later part of the project, 3 different datasets were used. Again, the synthetic data had a crucial role as there was not enough data to meaningful simulate different components, much less different aircraft, simultaneously.

In order to include different aircraft, the syntethic data now attributes tail numbers to each trajectory. We keep generating trajectories in single value format but with the trajectory comparison we can derive the historic PDF, therefore we can consider the input RUL as a VHI.

Later on in the project we had access to datasets resulting from 2 different prognostic models running on the Turbofan Engine Degradation Simulation Data Set [29].

For the first model, we got results from 3 sets, FD01, FD02, FD04. In the first set, the model was trained using a linear ground truth. On the other hand, for the other two sets, FD02 and FD04, the model, was trained with a ground truth modeled with elbow point, at the 120 flight hours mark. This means that we don't consider RULs higher than a certain threshold - the elbow point - and instead of the ground truth growing back (from the end to the start) linearly, it stops increasing past that threshold.

All three sets were composed of 100 trajectories as we chose to use the results of the model running on its training sets. For two reasons: first we wanted to use complete trajectories and second, because we wanted the prognostics to be accurate, even if it was more accurate than expected in normal usage.

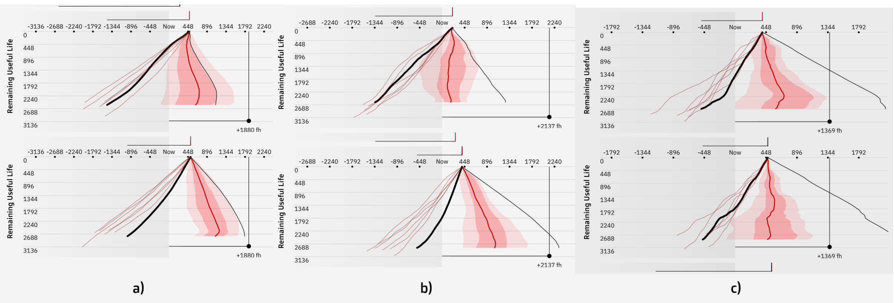


Figure 30: Two unrelated components, top and bottom, illustrating different values for the maximum change in degradation rate. a) 0.01 hours per hour; b) 0.03 hours per hour; c) 0.1 hours per hour;

The last dataset is also from a model running on the FD04, with ground truth modeled with an elbow point and contains 10 trajectories.

As both models are running on synthetic data, the experiments were focused only on the highest detail level, the system level, since we can't relate or infer on correlation or dependency between different time periods or different systems.

5.3 EXPERIMENTAL RESULTS

This section summarizes the results obtained by experimenting with both the generated data and the real prognostic data using the C-MAPSS

5.3.1 *Synthetic data*

By varying the degradation rates defined in the synthetic data generation phase, we can produce trajectories with different aspects and different degradation rates.

As we can observe in Figure 30, the trajectories start presenting more variation as we increase the amount that the degradation rate can change every hour. Generally, the pyramidal shape still maintains, but as we increase the rate of change, it is more common for the mean curve to present a more zig-zag shape.

As this change is increased even further, the trajectories start presenting step-like behaviour, as seen in Figure 31. Still, the overall accuracy of the historical RUL is not guaranteed to increase, usually the opposite.

Figure 32 represents a normal behaviour of the tool and the trajectories. All the past trajectories start by presenting some length and max RUL variance. As we get into the future, we see the Standard Deviation (the width of the red area), generally, decreasing, meaning that we're more confident about where we are pointing. At the same

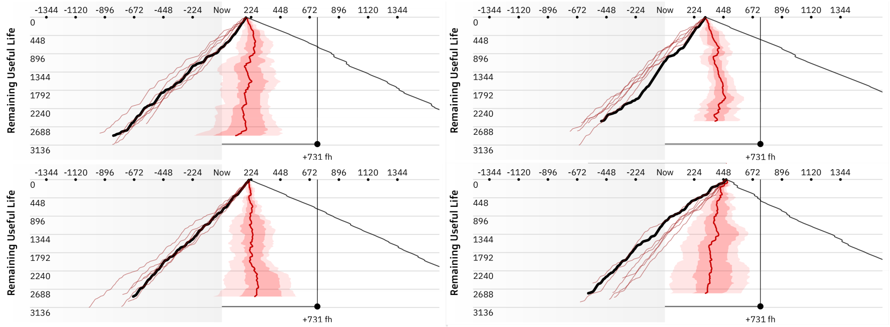


Figure 31: Visualization of four different systems, generated with maximum change per hour of 0.5 hours. Although the mean line shows many zig-zags, it does not turn far away from the EoL.

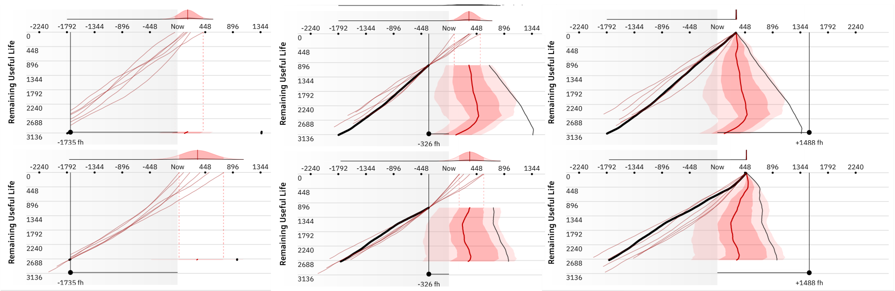


Figure 32: Normal behaviour for the tool and trajectories, for two different components, in 3 moments in time, starting from left to right.

time, we see that the red line is almost vertical and has little overall variation in the X axis. This means that we've been predicting the EoL to happen at the same place/date for most of the time.

Figure 33 compares input and output RUL with the ground truth, showing that the ground truth line was always inside the derived intervals and very close to the mean historical RUL. We can see how the input RUL relates to the ground truth by representing the input trajectory as "where it will end", that is in the same mapping as the PDF trace. Its the same we see on Figure 29 but we map the RUL axis parallel to time.

When comparing the input RUL and the historic PDF with the ground truth, if the ground truth is on the left it means our expectations were

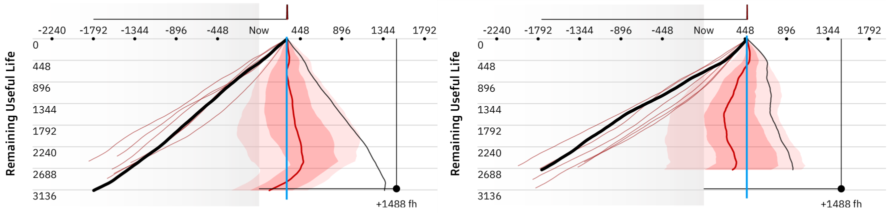


Figure 33: Output RUL (red areas and line) comparison with current trajectory EEoL (black line on the right) and with the ground truth (blue vertical line).

too high and the component lasted lest than we expected. On the other other hand, if the ground truth is on the right it means that our expectations we're too low or that he component lasted more than expected.

5.3.2 *Turbofan Engine Degradation Simulation Data Set*

For testing the visualization with real data, 2 different models were used. Both models run on the Turbofan Engine Degradation Simulation Data Set [29] and produce an RUL prognostics in PDF format.

The first model was trained separately with linear ground truth and with elbow point which we will call group A.

Figure 34 and Figure 35 show the method applied to group A - FD01 linear ground truth. At a first moment, some trajectories seem to not be aligned because they never had such an high RUL. As trajectories are very different, the historic PDF points to the right, showing that every moment until now we have been postponing the EEoL. At an intermediate time, the reference RUL is low enough for all trajectories to have experienced it. Therefore as trajectories present less variation, the historic PDF trace starts pointing up. In the final moment we can see that the historical PDF trace kept going up vertically with minor horizontal deviation. This tells us that, since a little before the second moment, the EEoL has not changed and it is quite close to the EoL.

Figure 36 shows trajectories from group A - FD02 with elbow point.

The second model, group B, only used ground truth with elbow point.

Figure 37 shows the visualization of group B.

5.4 ANALYSIS OF THE RESULTS

In most of the experiments, the EoL of the current trajectories would be either within the 99.7% confidence of the derived PDF or within the observed intervals. Moreover, it was also common for the EoL to be within the red line's (mean) variation range, meaning that that exact point in time was predicted several times, as time went on.

The only case when this would not be the correct behaviour is when the current trajectory presented a very different behaviour while the past trajectories were very similar to each other. This is the case in Figure 29, on the only trajectory where the ground truth is between the input RUL and the PDF trace (column 1, row 3). This is can be also seen in Figure 30 in the lower system of a) and b).

The synthetic data is in itself biased, because if we are generating values within a certain distribution, it can be expected that the same distribution will show up when we compare the trajectories. That is to say that if, the generated trajectories can decrease 0.9 to 8 hours,

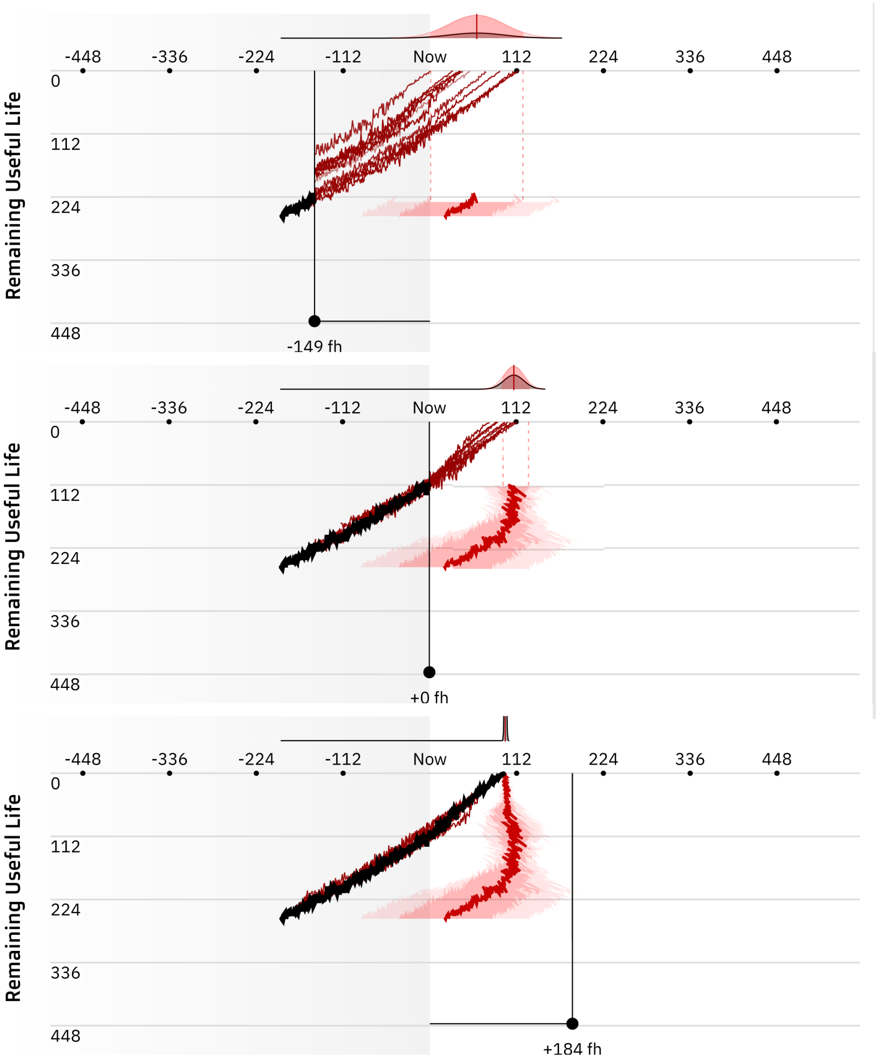


Figure 34: Test with 10 trajectories picked randomly from group A - FD01. The ground truth is modelled linearly.

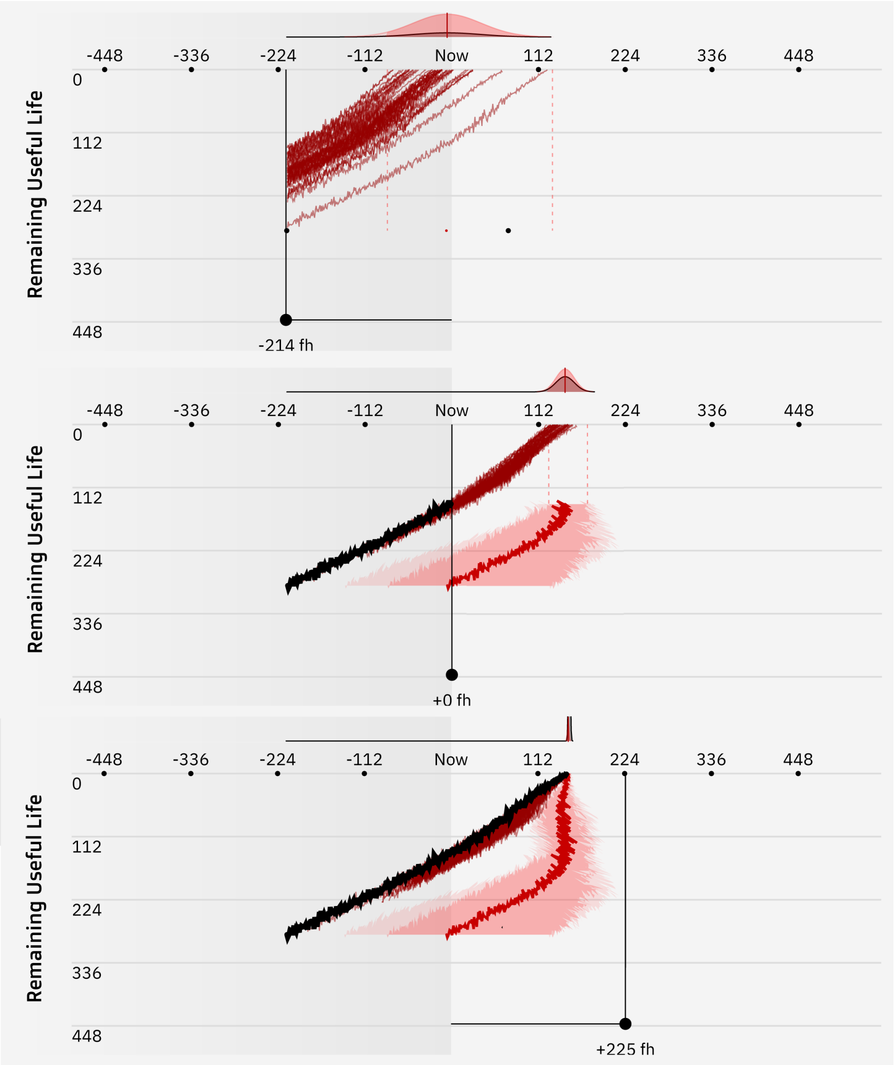


Figure 35: Test with 50 trajectories picked randomly from group A - FD01. The ground truth is modelled linearly.

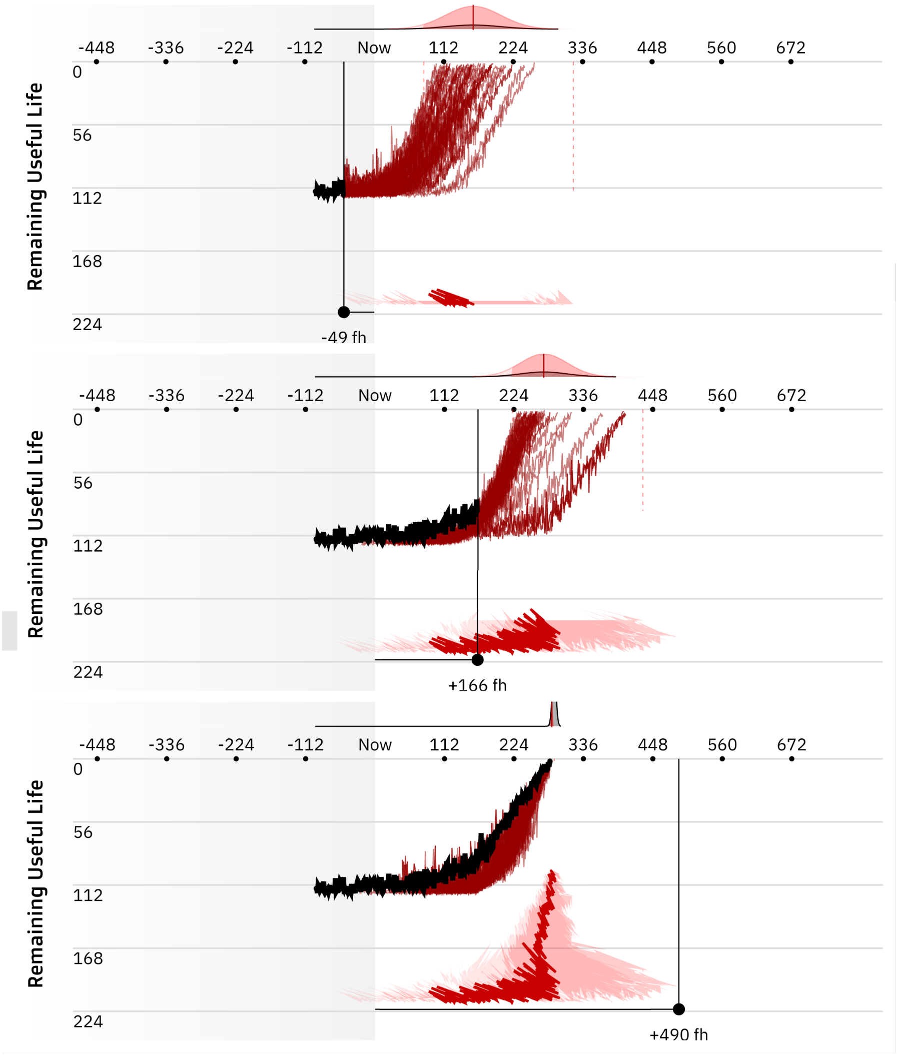


Figure 36: Test with 50 trajectories picked randomly, from group A, FD02. The ground truth is modelled with elbow point.

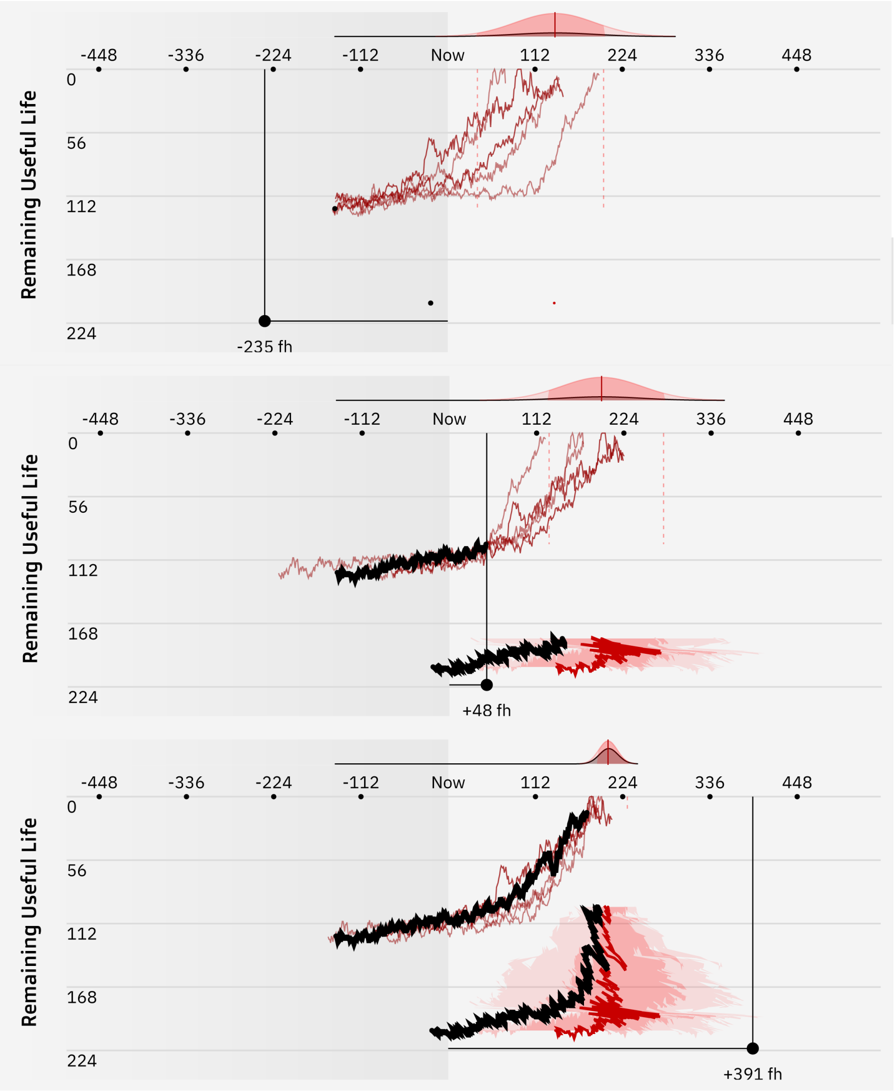


Figure 37: Test with 10 trajectories, from group B. Ground truth modeled with elbow point

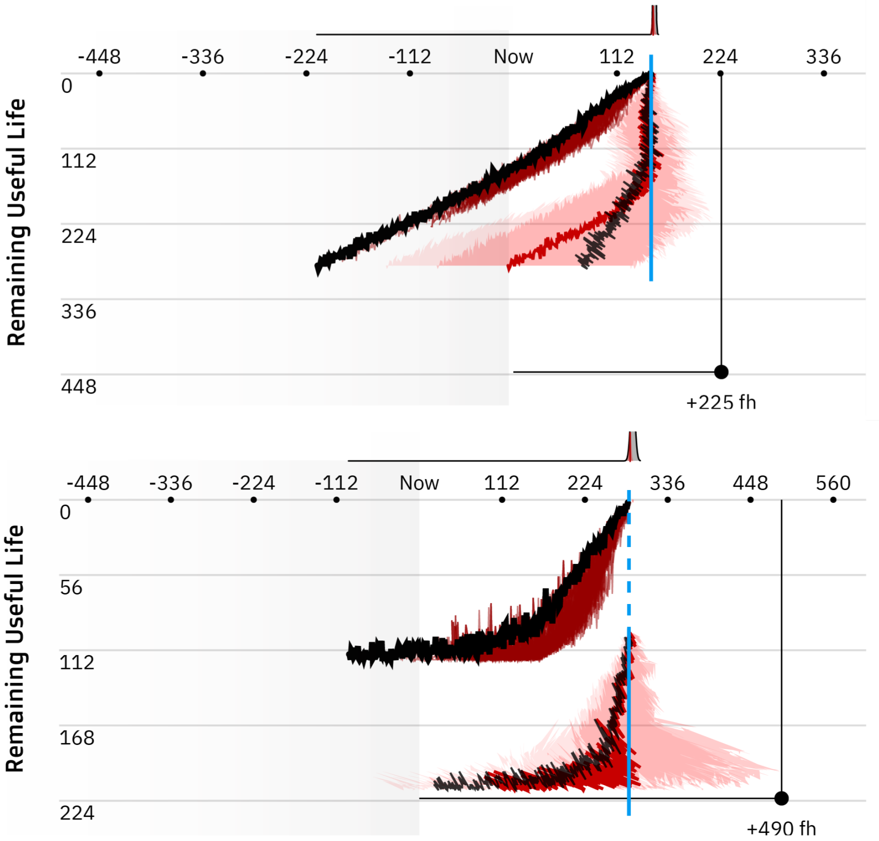


Figure 38: Comparison of model and derived RUL with ground truth. a) Applied to the dataset trained with Elbow Point, the comparison is moved down for visibility; b) Applied to the data set trained without Elbow Point.

every hour, then if we generate an infinite number of trajectories and apply the comparison, the aligned trajectories should approximate this interval. This would be true for both sides of the cut off point and the group of the whole trajectories should aproximate a diagonal hourglass shape.

The synthetic data was very useful for the structuring of the tool, it let us experiment with aggregating the different RULs of the same aircraft.

The most surprising result is that the method worked very well on the first set of real data, specially since the input is not considered accurate, Figure 28.

The two most important aspects identified while working with real prognostic data were the fact that the RUL prognostic data is much noisier than the synthetic data and, unlike modelled in the generated data, the real RUL is not a monotone function. Both these two aspects cause some problems with the visualization, but they do not break the idiom. The first makes the current trajectory occlude the past trajectories and somewhat its own behaviour.

The second makes it so that the PDF historical trace is both occluded and incorrectly positioned. It also causes those "diagonal" traces instead of horizontal ones, like in the generated data.

In both cases, with and without elbow point, the input RUL is quite accurate (as we are working with the models training set). If we compare both the ground truth and both the input and output RULs, on the horizontal plane, we can see that all are quite close, as in Figure 38.

It should be noted that, as the input RUL approaches 100-120 flight hours, the historical RUL stop pointing sideways (right) and start pointing straight up, overlapping with the ground truth. This might indicate that it is unnecessary to train the RUL prognostic with a preset elbow point or that it is possible find the "true" elbow point with the visualization.

As RUL prognostics capabilities and effectiveness increase, it is expected that the last level of the visualization will lose some of its relevance, eventually phasing out the statistical RUL computation as it will lead to redundancy. Either way, it can still be useful as a "second guess". Better yet, maybe it can be used to visualize the models input data, the HIs and VHI.

6 CONCLUSIONS

This work started with an analysis into InfoVis, presenting some classical ways of visualizing time-based information. This was complemented with an analysis on"What?-Why?-How?" framework. This abstract analysis framework was then used to classify our problem, in the context of PHM and CBM There was a search into different research fields for visualization techniques that dealt with time-based data and time-based data prediction. We found two different contexts which used or relied on the same principles for the technique of aligning the trajectories. In Section 2.4.3 is the same principle of using a relative scale but on the contrary axis. In regards to Section 2.4.1, in the "Align Outbreaks" example, we apply the same principle, on the same axis, the time-scale, but we apply it for every time instance a not just the starting point.

The resulting tool and technique developed owe much of their strengths to the medium and technologies used. As such there is a chapter quickly describing and referencing them. The comparison technique developed showed favorable results and shows great potential for generalizing its use to virtually any kind of simple time series comparison.

Additionally, it might be useful as an extra comparable metric to any compatible prediction problem. For example, comparing an informed prediction with the statistical analysis. In the context of RUL estimation, we can treat the input RUL as a VHI and effectively run a form of statistical prognosis (the trajectory comparison) on top of a data-driven prognostic result. We can also just display and output the input RUL, using the comparison result just as a visualized second guess. It is not clear yet which one is preferable.

Using a Gantt chart structure, relying on indentation for hierarchy, as well as grouping and ordering the systems inside each aircraft, look like robust solutions to represent and easily navigate the data.

We assume that the historical RUL PDF trace does not look straightforward, but that in a dynamic context, if the user can see the tool operating / moving, it becomes apparent how the area is being derived and the reason of its location.

Further investigation can be done in regards to the PDF groupings for the outer and middle levels of detail, 4.2.3 and 4.2.2, as is not a straight forward matter.

Much more real RUL prognostic data is required in order to do meaningful visualization result analysis. It would be of great help if there was a major public dataset available containing RUL prognostic results.

Future work can be directed towards implementing the last level of detail more closely with RUL prognostic developers so that we visualize the model and not a generic technique.

It is also necessary to wait that other key components of the CBM framework mature, so that the parts that this tool connects, WP5 and WP6, are ready to do a meaningful simulation.

As WP6 Maintenance Decision Support Tool also displays RUL prognostic, it is not clear yet what should and should not overlap in both visualizations; as more prognostic models and date are made available we will get a further understanding into what the prognostic developers need to communicate to the maintenance teams.

Future work might also include integrating the RUL Visualization tool more closely with the Maintenance Decision Support Tool.

BIBLIOGRAPHY

[1] *H2020 ReMAP*. H2020 ReMAP, July 2021. URL: https://h2020-remap.eu/ (visited on 10/30/2021).

[2] *Information Visualization - InfoVis:Wiki*. Infovis-wiki.net, 2013. URL: https://infovis-wiki.net/wiki/Information_visualization (visited on 01/08/2020).

[3] Tamara Munzner. *Visualization analysis & design*. Crc Press, Taylor & Francis Group, 2015.

[4] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Et Al. *Visualization of time-oriented data*. Springer, Cop, 2011.

[5] Wikipedia Contributors. *A picture is worth a thousand words*. Wikipedia, Nov. 2019. URL: https://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words.

[6] *The Lebombo Bone: The Oldest Mathematical Artifact in the World*. African Heritage, 2019. URL: https://afrolegends.com/2019/05/17/the-lebombo-bone-the-oldest-mathematical-artifact-in-the-world/.

[7] *Lebombo bone*. Wikipedia, 2021. URL: https://en.wikipedia.org/wiki/Lebombo_bone (visited on 10/30/2021).

[8] Axel K. Schmitt, Martin Danišík, Erkan Aydar, Erdal Şen, İnan Ulusoy, and Oscar M. Lovera. "Identifying the Volcanic Eruption Depicted in a Neolithic Painting at Çatalhöyük, Central Anatolia, Turkey." In: *PLoS ONE* 9 (Jan. 2014). Ed. by Victoria C. Smith, e84711. DOI: 10.1371/journal.pone.0084711. (Visited on 11/19/2019).

[9] *Proto-writing*. Wikipedia, Aug. 2021. URL: https://en.wikipedia.org/wiki/Proto-writing.

[10] Google. *Google Sheets: Free Online Spreadsheets for Personal Use*. Google.com, 2014. URL: https://www.google.com/sheets/about/.

[11] *Rhino 6 for Windows and Mac*. Rhino3d.com, 2019. URL: https://www.rhino3d.com/.

[12] *Grasshopper*. Grasshopper3d.com, 2009. URL: https://www.grasshopper3d.com/.

[13] Mike Bostock. *D3.js - Data-Driven Documents*. D3js.org, 2013. URL: https://d3js.org/.

[14] *Gallery*. observablehq.com, Feb. 2020. URL: https://observablehq.com/@d3/gallery.

[15] *Observable - Explore, analyze, and explain data. As a team.* observablehq.com. URL: https://observablehq.com/.

[16] World Health Organization. *Coronavirus disease (COVID-19) pandemic*. World Health Organization, 2021. URL: https://www.who.int/emergencies/diseases/novel-coronavirus-2019.

[17] *COVID-19 pandemic*. Wikipedia, May 2020. URL: https://en.wikipedia.org/wiki/COVID-19_pandemic.

[18] *Coronavirus (COVID-19)*. Coronavirus (COVID-19) - Google News. URL: https://news.google.com/covid19/map.

[19] *Coronavirus (COVID-19) statistics data - Google Search Help*. support.google.com. URL: https://support.google.com/websearch/answer/9814707?p=cvd19_statistics&hl=en (visited on 10/31/2021).

[20] Hannah Ritchie, Edouard Mathieu, Lucas Rodés-Guirao, Cameron Appel, Charlie Giattino, Esteban Ortiz-Ospina, Joe Hasell, Bobbie Macdonald, Diana Beltekian, and Max Roser. "Coronavirus Pandemic (COVID-19)." In: *Our World in Data* (2020). URL: https://ourworldindata.org/coronavirus.

[21] *GME Interactive Stock Chart | GameStop Corp. Stock - Yahoo Finance*. finance.yahoo.com. URL: https://finance.yahoo.com/chart/GME.

[22] *GameStop short squeeze*. Wikipedia, Mar. 2021. URL: https://en.wikipedia.org/wiki/GameStop_short_squeeze.

[23] *Protovis - Index Charts*. mbostock.github.io. URL: https://mbostock.github.io/protovis/ex/index-chart.html (visited on 10/30/2021).

[24] *Line Chart, Index Chart*. observablehq.com. URL: https://observablehq.com/@d3/index-chart (visited on 10/30/2021).

[25] *CORDIS | European Commission*. Europa.eu, 2021. URL: https://cordis.europa.eu/project/id/769288 (visited on 10/30/2021).

[26] A. Tero, S. Takagi, T. Saigusa, K. Ito, D. P. Bebber, M. D. Fricker, K. Yumiki, R. Kobayashi, and T. Nakagaki. "Rules for Biologically Inspired Adaptive Network Design." In: *Science* 327 (Jan. 2010), pp. 439–442. DOI: 10.1126/science.1177894. (Visited on 02/09/2021).

[27] Laura Sanders. *Slime Mold Grows Network Just Like Tokyo Rail System*. Wired, Jan. 2010. URL: https://www.wired.com/2010/01/slime-mold-grows-network-just-like-tokyo-rail-system/.

[28] Kai Goebel, Abhinav Saxena, Matt Daigle, Jose Celaya, Indranil Roychoudhury, and Scott Clements. "Introduction to prognostics." In: *European PHM conference*. 2012.

[29] Abhinav Saxena and Kai Goebel. "Turbofan engine degradation simulation data set." In: *NASA Ames Prognostics Data Repository* (2008). URL: https://ti.arc.nasa.gov/c/13/.