NPL REPORT MS 53

DATA MODELLING GOOD PRACTICE GUIDE

REECE SAINT, PAUL DUNCAN

FEBRUARY 2024

DATA MODELLING GOOD PRACTICE GUIDE

REECE SAINT, PAUL DUNCAN,
and with contributions from
Joao Gregorio, Jean-Laurent Hippolyte, Frédéric Brochu

Informatics group, Data Science department

ISSN: 1754-2960

https://doi.org/10.47120/npl.MS53

This work was funded by the UK Government's Department for Science, Innovation & Technology through the UK's National Measurement System programmes.

National Physical Laboratory
Hampton Road, Teddington, Middlesex, TW11 0LW

Approved on behalf of NPLML by
Marina Romanchikova, Science Area Leader Informatics/Data Science department.

CONTENTS

ABBREVIATIONS & TERMINOLOGY

Data Creator	People (users) who create new data and information of any type.
Data User	People (users) who primarily manipulate and/or analysis data, either explicitly through platforms such as Excel or languages such as Python.
DCC	Digital Calibration Certificate. Relating to the digital equivalent of calibration certificates, structured to be machine readable. See **SmartCom.**
D-NMI	Digital National Metrological Institute. See **[1]**.
DS	Data Science (used here in reference to the Data Science department, NPL).
D-SI	The Digital-SI. Relating to the digital representation of measurements and their associated unit. See **SmartCom.**
FAIR Principles	Fair, Accessible, Interoperable, Reusable. General good-practice principles for data quality. See https://www.go-fair.org/fair-principles/, accessed on 2022-02-22.
GUI	Graphical User Interface.
GUM	Guide to Uncertainty of Measurement, See JCGM publication, https://www.bipm.org/en/committees/jc/jcgm/publications, accessed on 2022-02-22.
JCGM	Joint Committee for Guides in Metrology. Creators of the VIM and GUM documents. See https://www.bipm.org/en/committees/jc/jcgm [Accessed 2022-06-10]
Machine-Readable	Data is considered machine readable if it is presented in a common interoperable data format, with any associated data standards.
Machine-Understandable	No standard is available for the term, however is general refers to the ability for a machine to infer meaning from data. E.g., a program may be able to infer that an article is about geopolitics and can categorise it as such.
NLP	Natural Language Processing. An umbrella term for techniques for textual data processing.
NPL	National Physical Laboratory
SmartCom	A project developed within the European Metrology Research Program (EMPIR), including NPL, which concluded in 2018. Established concepts for Digital Calibration Certificates and metrological work based on the digital representation of the SI (D-SI). See https://www.ptb.de/empir2018/smartcom/project/, accessed 2022-02-22.
UUA	Ultrasound and Underwater Acoustics, research group department at NPL.
VIM	Vocabulary of International Metrology. See JCGM publication, https://www.bipm.org/en/committees/jc/jcgm/publications, accessed on 2022-02-22.

1 INTRODUCTION

1.1 DOCUMENT PURPOSE

This document is intended as an introductory guide for those within NPL, and external partners, who are looking to design a data model which best represents the type of work they conduct and aligns to the organisation's goal of becoming a digital national measurement institute (D-NMI) (1).

The aim of this guide is 1) to start a discussion as to what data modelling means for NPL, 2) to provide a template to build a data model for measurement and business data, and 3) to discuss case examples of data modelling within the organisation.

1.2 DOCUMENT STRUCTURE

Section 1 details the benefits of data modelling to data-creators, data-users, and NPL overall. Furthermore, The D-NMI context is given, as well as several related projects and resources which will be referred to in later sections.

Section 2 discusses general concepts around data and metadata. Scientific metadata standards are discussed, and resources provided for deeper investigations.

Section 3 introduces the general concepts and components of a data model.

Section 4 provides a suggested step-by-step guide to data modelling, including a list of considerations to be aware of when formalising methods and processes. A simple and detailed example are given, and a few pre-existing **case examples** studied. Users already familiar with formal data modelling practices who are looking for a quick-start guide are advised to start reading from this section.

Sections 5 and 6 provide a more in-depth Glossary of terms relating to Data Modelling and Metadata, for those wanting deeper insight into the topics.

1.3 MOTIVATION AND BENEFITS

Digitalisation is a key strategic goal for NPL, as it is for metrology worldwide. The organisation must demonstrate its capability in making the change to a digital way of working while not impacting its operational capability adversely. The achievement of this goal will require many staff within NPL to engage in a new way of thinking, with scientists and engineers alike having to incorporate the concepts of data engineering (and/or 'data science')[1] into their core activities.
However, there are significant knowledge gaps in understanding the scope and lifecycle of NPL data from its creation, curation, storage, and final archiving. Collectively these components form the foundation of data modelling as they reflect the intended short- and long-term use case of the data.
In parallel, users must also ensure their data adhere to the Findable, Accessible, Interoperable, and Reusable (**FAIR**) principles (2) and the concepts of the **digital-SI,** while including **uncertainty** information where applicable.

---

1 The distinction between data science and data engineering is fuzzy, however, typically data engineers typically transform and manage data through data pipelines whereas data scientists analyse, test, and present data to drive decision making.

To create useful data models, data creators and users must be engaged in the earliest stage of the data timeline. This change will be a challenging transition, with the primary hurdle being motivating the change from current ways of working, while balancing the financial burden of allocating time to indirect development of good practices. However, good data modelling practices will ensure:

- A reduction in errors in software & database development.
- Increased consistency in documentation.
- Increased performance in data retrieval and/or storage.
- Improved communication of data within NPL, and with external customers and partners.
- Better adherence to data requirements from customers, academic journals, government bodies and agencies, as well as the principles of the scientific method.
- A smoother interface with NPL’s new data infrastructure, e.g. Azure, Databricks and Data Factory.
- Expertise and intellectual property are not lost in traditionally unstructured textual sources, e.g., paper and digital laboratory and workshop notebooks.

More generally, good practices in data modelling will provide an easier roadmap to NPL to becoming a successful D-NMI.

2 INTRODUCTION TO DATA

2.1.1 What are data?

Data, for the purpose of this report, are anything which can be measured, collected, reported, and analysed[2]. These data may already exist in a digital format, such as in Excel sheets, comma separated variable (csv) files, or in structured folder repositories. However, they could just as likely still exist only in paper records, or unstructured digital notebooks.

Understanding how and what data are generated throughout a specific process will become a critical first step for everyone within NPL who wishes to digitalise their work. This can be a time-consuming process. It requires data creators to consider both what is already explicitly recorded in their processes, but also to capture the implicit data which may not yet be recorded but might provide insight in future. For example, many processes in a laboratory or workshop have process workflows, user logs, equipment inventories and geolocation information which is only discoverable in physical notebooks known only to those who work daily with them.

This is a significant challenge to NPL, as any data that is not recorded increases the risk of scientific and commercial value being lost. However, the intention of digitalisation is to not to arbitrarily generate data for the sake of it but to balance data collection with the potential future use-cases for that data. This balance will never be perfect, and it is unavoidable that over-time, it will be discovered that some now-critical element was missing from the original data plan. Conversely, over-scoping data will be inhibiting, and potentially destructive, to any organisation which hopes to manage how much data it must store and manage.
To guide the selection of relevant data, consider:

- The minimum viable data required to deliver on a measurement service, publication, customer requirements, and scientific standards.
- The minimum viable data required to adhere to NPL's quality control, uncertainty requirements, health & safety regulations, HR requirements and legal responsibilities.

2.1.2 What are metadata?

Metadata are the auxiliary information which describe and enrich the (measured or simulated) data. Some examples are the timestamp relating to a measurement or a string describing the format of a data entry. The lack of standards of metadata for metrology presents a challenge to groups (within NPL in particular) as the implicit scientific metadata generated by individuals can be extensive and unstandardised. The definition is further complicated as scientific groups tend to pursue cutting edge developments, meaning that methods, requirements, and domain-standards can be prone to rapid changes. While some disciplines have begun to introduce metadata standards for their specific field of work (3), there is the risk of too many standards becoming counterproductive and unsustainable.

This section will introduce a summary of metadata types to guide data-creators. A more detailed discussion of metadata types is included in **Section 6 Appendix B – Metadata Details.**

2 See the OECD Glossary on Statistical Terms "Data" here (3) https://stats.oecd.org/glossary/detail.asp?ID=532 [Accessed 2022-06-10]. The Wikipedia article on the disambiguation of Data vs Computation Data is also insightful, https://en.wikipedia.org/wiki/Data_(computing) accessed on 2022-02-22.

Types of Scientific Metadata

The distinctions below present the authors' views and are intended as a guide rather than a set of formal definitions. Many standards, see section 2.2, split metadata types within data structures to better enable human navigation and improve machine retrieval times. No distinction is made as to which of the metadata types are 'more' machine-readable or machine-actional, as this is context and use-case dependent.

METADATA TYPE	EXAMPLES
DESCRIPTIVE	Document language; authorship; topic labels; abstract.
STRUCTURAL	Hierarchy architecture type; parent/child relationships; tabular data layouts
ADMINISTRATIVE (INCL. SECURITY)	Access rights; security restrictions; document creation information
FORMAT AND STORAGE	Data sizes; compression methods; precisions; standards
VERSION CONTROL	DateTime logs; change comments; version number
METHODOLOGY	Lab notes; peer review paper method sections; applicable scientific standards (e.g. D-SI)
UNCERTAINTIES	Measurement uncertainties as per the **GUM,** see **Appendix A – Glossary of Terms**
D-SI	Digital representation of SI units**,** see **Appendix A – Glossary of Terms**

2.2 METADATA RESOURCES & STANDARDS

- The Metadata Standards Catalogue https://rdamsc.bath.ac.uk/. Provides a rich list of metadata standards organised by schema and by scientific domain.
- The Digital Curation Centre (DCC) https://www.dcc.ac.uk/. Maintained by the University of Edinburgh. Provides a number of publications on data curation but see https://www.dcc.ac.uk/guidance/standards/metadata, accessed on 2022-02-22.
- The Text Encoding Initiative. Aimed at standards for the exchange of textual information. https://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.html, accessed on 2022-02-22.
- Dublin Core, https://en.wikipedia.org/wiki/Dublin_Core, accessed 2022-02-22. A series of definitions for describing digital resources and metadata, now formally part of ISO 15836.

3 AN INTRODUCTION TO DATA MODELS

3.1 WHAT IS A DATA MODEL?

A data model is any system which aims to describe and connect data and metadata through understanding the relationship between data elements. A column of data containing cells with rows of measurements for a device is a simple relationship structure in which all entries relate to that device. More complex models could describe the interaction of data elements when there are machine-person interfaces, such as a person committing a project update via a graphical user interface (GUI) or vast networks of geographically distanced storage servers containing data for whole organisations.

Any data model created, however, is only valuable if it serves the needs of the data users. Therefore, the key guiding principle when designing a data model is that modelling should be driven by the specification of the use-case. In the following, we present several possible examples of the model design concepts.

3.1.1 When to make a DATA model?

While the definition given of a data model in **Section 3.1** is intentionally vague, the principle of aligning to requirements is critical to data model design. If not followed, a designer may attempt to create a highly granular and nebulous data model only to discover that the result is unused or even unusable due to its complexity. It is prudent to consider if there are cases when a data model is not required.

Below we list the conditions that elicit the use of data models:

C1. If digital exchange of data between devices or users is required, a model should exist which represents and describes the data interface and the data packages exchanged. Ideally, the result would adhere to FAIR principles and ensure that no data is unintentionally lost or edited during the exchange.
C2. If data is to be stored or archived, a descriptive model is required to ensure accurate, fast, and/or long-term retrieval.
C3. Any process which requires strict auditing and/or traceability will need a model which can integrate compliance requirements. This includes scientific journals and funding agency data publication requirements such as EMPIR.
C4. Any measurement and accompanying uncertainty and (SI) unit, which depends on other measurements, or will have dependents itself, requires a model which captures this information.

Purposefully the examples do not imply a *digital* data model. Most pen and paper researchers do have an implicit or explicit method to organise their work and data. For example, tabulated data will be annotated with dates, units, uncertainties; and typically, the annotations and notes alongside data instances will describe the methodology, experimenters name, laboratory/workshop and so on. All this information is part of a data model, even though it is not digital. The next section, **Section 3.1.2**, will discuss the specific benefits to digital data modelling, followed by the basic principles of data models.

3.1.2 Benefits of digital data modelling

In the context of NPL and its goal of becoming a D-NMI, the benefits of digitalisation are best described by the internal NPL report “Digital NMI: NPL’s Internal Landscape” (1), which highlights the following key benefits:

- Automation of data and metadata capture and analysis,
- Digital delivery, including digital calibration certificates,
- Shorter calibration chains and realisation of units at point of use.

Whilst the last of these benefits is specific to NPL, the other two aspects are more widely beneficial. These benefits go further than just having a 'digital-enabled science lab'. It is not enough to just store data in digital formats such as digital notebooks, Excel documents or within database architectures such as SQL[3]. A well-constructed data model should enable data which is findable, accessible, interoperable, and reusable (i.e. consistent with the FAIR principles) to the widest audience. This is only achievable if the digital models are well structured, have rich metadata, are understandable to its human users and, if the aim is to leverage data for machine-learning processes, machine understandable [See **Abbreviations & Terminology**].

While reconstructing processes within a laboratory or workshop from analogue (pen and paper) methods to a fully digital equivalent will take a significant time investment, it is an inevitable hurdle that the organisation must overcome if it wishes to continue to output high-quality work and measurement services to rest of the UK.

3.2 DATA MODEL VS PROCESS DIAGRAM

The following sections will discuss data models in abstract terms, and so it is pertinent to describe here the distinction between a data model and a process diagram (also known as a procedural flow, experimental method, or process timeline).

A **process diagram** should describe all the required steps in the physical world to complete a task. This includes anything that a human or machine does, including the decision-making elements, and interfaces with other humans or machines. The diagram should be followed in real-time and be logical enough to capture all the relevant use-cases.

Throughout the process model, data elements are created, edited, and potentially stored. A data model is a representation which reflects the relationships between these data elements. It is typically a diagram in which elements are connected with lines which imply some sense of the type of relationship the elements have.

3.3 TYPES OF DATA MODELS

There are many ways of describing data and constructing data relationships. The primary driving force behind selecting which type of data model to construct is the specific use-case of the task at hand, and the restrictions of the network/system the data will be held within. It would be beyond the scope of this report to discuss the application of each type, and so the reader is directed to the selected external sources as follows below. Of particular interest due to their common usage are **relational models** (most applicable to the scientific and metrological work at NPL)**, network models** (e.g. for computer network architectures)**,** and **entity-relationship models**.

- Lucid chart's "types of database models", accessed on 2023-04-24, https://www.lucidchart.com/pages/database-diagram/database-models/#section_0
- Wikipedia's page on Data Models, accessed on 2023-04-24, https://en.wikipedia.org/wiki/Data_model
- Wikipedia's database model page, accessed on 2023-04-24, https://en.wikipedia.org/wiki/Database_model

3 The Structured Query Language (SQL) is a standardised language for handling, primarily, relational databases. See Section **3.3, Types of data models.**

3.4 PRINCIPAL COMPONENTS OF A DATA MODEL

To provide a guiding tool for data model designers, models are often presented in terms of three “perspectives”, collectively known as the **degrees of abstraction** (4)**.** This section presents these perspectives in order of decreasing abstraction, i.e., increasing technical contents, as follows: the **conceptual model**, the **logical model**, and the **physical model.** It should be noted that these are not rigidly defined concepts, and so specific contents can differ between domains and applications. Further, the detail of each perspective will be determined by the technical ability of the designer and, critically, the use-case.

3.4.1 Conceptual Models

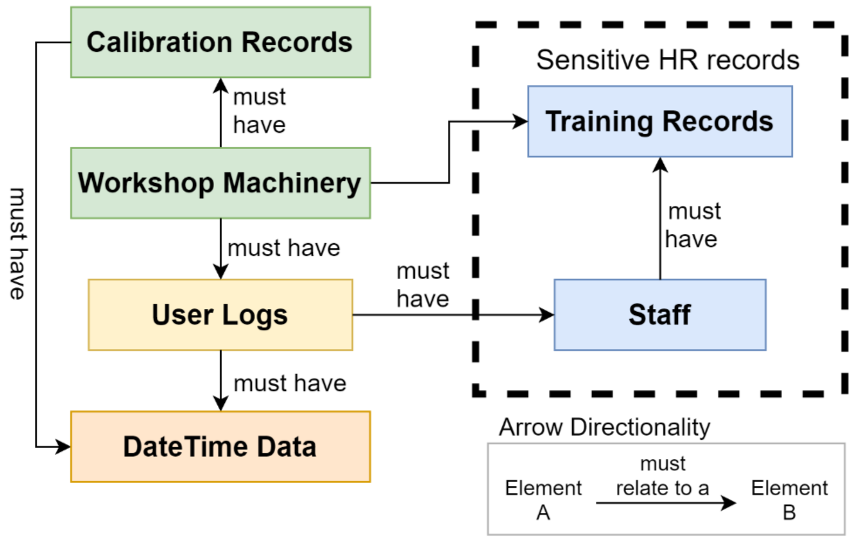

Figure 1 - Conceptual Model Example. A simple conceptual model diagram, showing how staffing records may be linked to user logs of a piece of workshop equipment. Staffing data is ringfenced as sensitive data. Many of the elements have dependencies, shown with the “must have” text next to the connecting arrows[4].

A **conceptual model** is a high-level picture of the data components within a system connected through their relationships. An example is shown in **Figure 1** for a situation where a piece of calibrated machinery is used by a user who must have some relevant training records. The model designer has chosen to separate elements into primary concepts, such as Staff data, in which entries (i.e., individual staff members) have an associated Training Record. These records in turn must relate to the specific piece of Workshop Machinery that member of staff is trained on. Similarly, DateTime data interfaces with the User Logs so we

4 In more technical data modelling work, connecting arrows have both a directionality and include additional symbolic terms in reference to the types of connections and relationships. I have not included this formal notation here, but point the curious reader to the Unified Modelling Language (UML) here https://www.uml.org/ (11) and https://en.wikipedia.org/wiki/Unified_Modeling_Language (10) both accessed on 2022-03-25.

know who did what and when, and the Calibration Records, so we know how long ago the machinery was calibrated.

While the conceptual model is the most abstract view of the data model, **it is critical that it is defined based on the intended use case of the data**, as we have done above.
As the conceptual perspective builds in complexity, it may include basic rules for data administration and data security as well as storage and backup interfaces. External requirements such as customer system and process restrictions may need to be integrated. These may include those determined by GDPR compliance, I.P. and legal considerations, and health and safety information.

3.4.2 Logical Models

The **logical model** is the bridge between the high-level conceptual model and the more technical physical model (**Section 3.4.3**). It involves expanding on the details of the conceptual model to provide descriptive contents for data elements, if applicable, and clarifying the relationships between them.
Continuing our example from before, the logical model for our system may look like **Figure 2**. Here, the individual data elements are expanded and populated with descriptive terms which reflect some of the specific entries that might be contained within each. Some of these elements are highlighted in blue, e.g., “Equipment ID”, as they will form the relationship keys[5] which connect one element to another.

5 More technically, an element may have a primary key which uniquely identifies itself. It may then contain many foreign keys which refer to the primary keys of other elements.

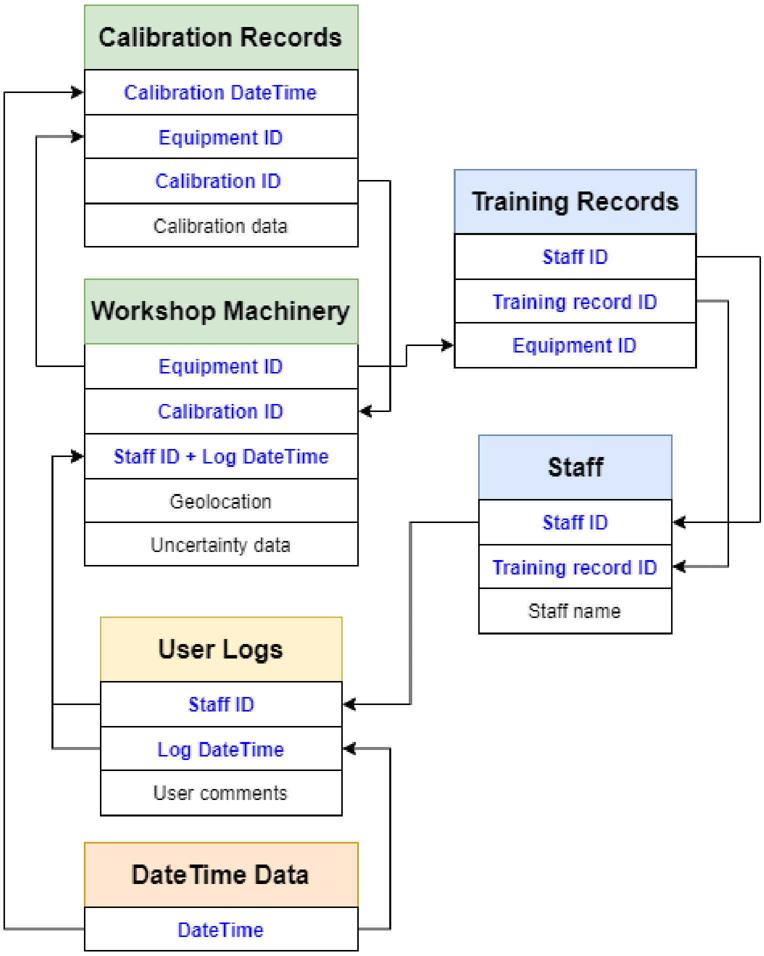

Figure 2 - Logical Model Example. Extension of the Conceptual Model diagram from Figure 1. Outlining the contents of the data elements allows for specific relationships to be mapped between the ID fields. Elements in blue indicate entries which are to be used as unique identifiers (UIDs).

The logical model is gradually becoming more technically complex, to the limit of the requirements of the use case. For example, each data entry may be further described to outline the data type, data format, or associated applicable measurement uncertainty (as per the **VIM & GUM,** see **Abbreviations & Terminology**), to name a few. This extra information is generally considered metadata in this context.

Relational information may require an associated rule book which establishes restrictions on data, their structure, and their relationships, as well as mandatory and optional flags. For instance, our example may require that any member of staff entering a user-log for a piece of machinery to have a Training Record ID and Equipment ID associated to their staff data profile. Perhaps this is enforced through an additional data field which flags restrictions, or as part of some metadata file listing requirements.

At the logical stage, it may become apparent that the use-case(s) considered are more complex than initially scoped or use-cases are added/removed. This may require sub-models to be developed bespoke to individual cases. However, it is important to focus on the essential components for the project(s) to limit the amount of **scope creep** in models, see Appendix A – Glossary of Terms.

3.4.3 Physical Models

The **physical model** enables the practical implementation of the model design. This requires every data element to be enriched to include specific technically descriptive features, such as its data type, storage requirements, precision, and format. A full physical model may also describe how the data interfaces with one or more databases, and how users and machines can access and edit elements.

Figure 3 demonstrates how the logical model can be extended to the physical model. This now includes details on some of the data types, as well as describing how some of the tabular elements will be stored in specific database architectures, such as comma separated variable files or in Extensible Markup Language (XML) documents[6].

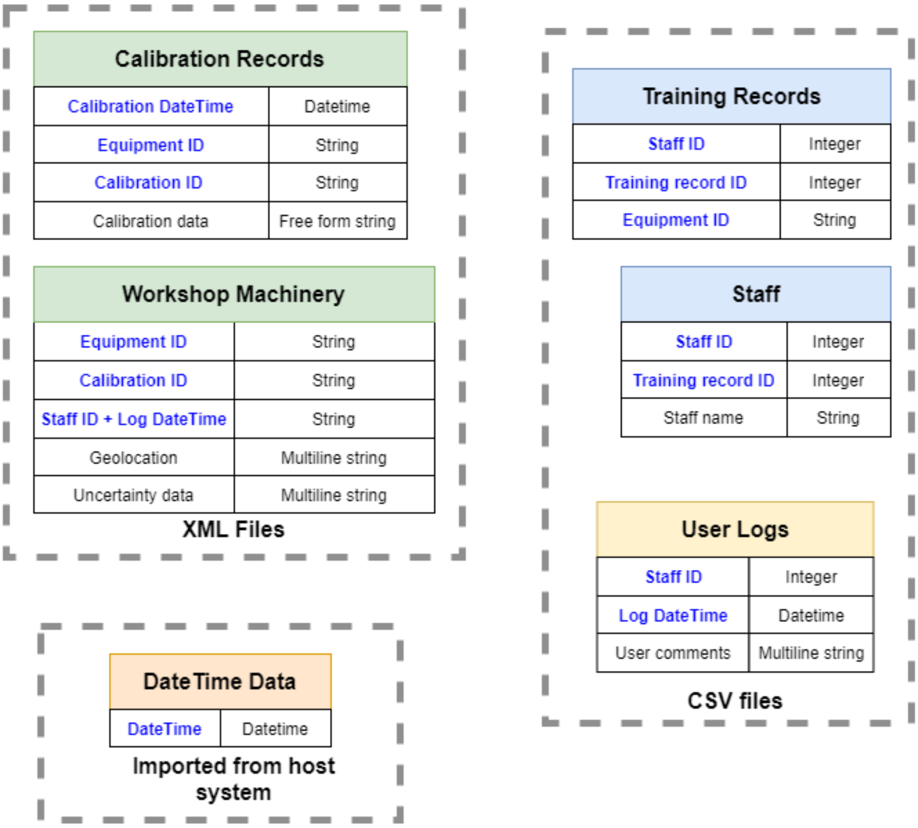

Figure 3 - Physical Diagram Example. Expanding on Figure 2, this diagram now includes information regarding the likely data types for each entry. Elements are collected to show the data structures they will be stored in. This is not a complete model but provides some core information for data engineers to begin constructing the necessary environment for a real-world implementation.

This view is the most technically challenging element of data modelling, and **Figure 3** would not be sufficient on its own to build an equivalent database and interface. The physical model requires input from data architects and engineers, as well as business/scientific users to ensure the implemented solution does not drift away from the model requirements. For example, a physical model designer will have to decide if the model should be optimised for fast retrieval or for storage efficiency.

6 The XML structure is a highly flexible text-based format that can be used to represent many common digital data objects using semi-human understandable labels or flags, like HTML. See https://www.w3schools.com/xml/xml_whatis.asp (12).

The physical model presents a specific challenge for NPL, and its goal of becoming a true D-NMI. To best align to the principles of FAIR, the D-SI, uncertainties, and any application of DCCs, the specific technical requirements will have to be understood from the conception at the more abstract conceptual level. The data owner must be data literate enough to understand how their model can adhere to these requirements. This application is unlike many commercial data modelling tasks where “business users” are often ringfenced away from specifical technical discussions. Similarly, the data engineers involved will need to understand the requirements of the data owners to ensure the physical model is a net gain to the organisation, and not an inhibiting factor.

3.5 DATA MODEL DESIGN TOOLS

The diagrams presented in this report were generated using the open-source software **DrawIO**, which comes with several benefits. Digital diagrams are easily editable, and DrawIO specifically can be integrated into a version control pipeline within Gitlab via a Visual Studio plugin. It is an intuitive design package, and in principle can also be used collaboratively within Visual Studio.

Microsoft Visio is a commercial equivalent, providing both cloud services and collaboration through OneDrive and Office 365 service, at the expensive of an individual or server-based licence fee.

In principle, markup language tools such as **Mermaid** can be used **within Gitlab** to fully design diagrams while also managing version control. However, this will need the user/designer to be comfortable using or learning the Mermaid markup language.
SysML (System Modelling Language, (5)) is a more technically oriented tool which is specifically aimed at the development of architectures for System Engineering application (6)**.**

These three options represent a fraction of the data modelling tools available. A Google search for “data modelling tools with version control” presents many options, from database design packages to simple diagram-making tools like DrawIO. This is an on-going challenge and discussion within NPL, as having a consistent and understandable data modelling standard will be critical as more data owners aim to digitise their processes.
In the short term, some simple projects may suffice with pen and paper in initial design work. Any software should come with ways to easily document changes and versions, such that future editors can easily and safely edit models.

4 DATA MODELLING GUIDE

The success of digitisation across the organisation is dependent on providing data users and creators the skills to understand and begin to develop their own data models to augment their primary work. This section aims to enable this process by presenting a workflow to develop a generic data model, followed by a worked example to explain the concepts in practice.

The intention of this workflow, and this guide overall, is to bridge the knowledge gap between data creators (scientists, engineers, HR, sales, strategy, etc.) and data engineers/architects. By creating a common understanding of data modelling practices, good practice in data management can be written into procedures and workflows at an early stage and data use and reuse can be improved throughout an organisation. .

4.1 DATA MODELLING WORKFLOW

Figure 4 shows a suggestion for the steps required to construct a generic data model. The steps make use of the concepts introduced in the previous sections but can otherwise be understood through the worked example that follows.

Figure 4 – Data Modelling Workflow. Suggested steps required to build and understand a data model, demonstrated through the worked example in Section 4.2. Documentation is necessary at all stages, and so is not explicitly connected to the other notes. Data Engineer Involvement is an optional node connected to steps three and seven. The figure layout is based off a DrawIO template which was customized for this document.

4.2 DATA MODEL EXAMPLE: ACME CORP

An organisation, Acme Corp, wishes to establish continuous air quality measurements at one of their manufacturing properties[7]. For Acme, this means capturing data from disparate sources into a central monitoring service, which then activates automatic air purification units when needed. The solution developed will have a direct influence on the safety of those

7 Acme Corp being a fictional brand from the old Road Runner cartoons. While I have tried to ensure this example is somewhat scientifically sound, there may be shortcomings. The example will be presented as if real conversations were had with the Acme organisation.

involved in the manufacturing process, as well as having an environmental and financial impact. They have approached NPL for their air quality measurement services, as well as requesting help in developing a suitable data model to process and act upon the data collected.

4.2.1 Acme Corp Air Quality – Start Simple

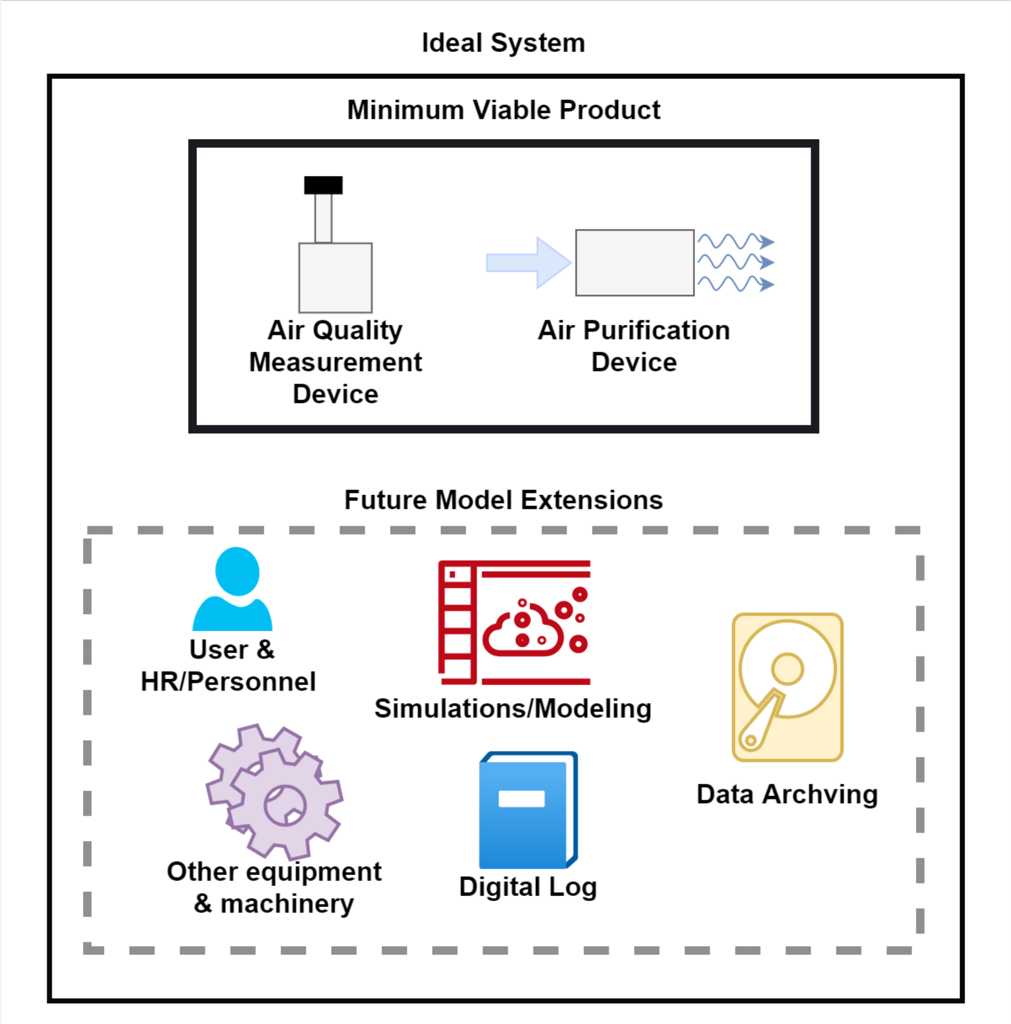

Figure 5 - Start Simple. Diagram capturing some of the potential 'ideal' features that may be integrated into the Acme data model. The overall ideal system is split into the minimum required systems/features (solid line), and what can be ring-fenced for future development (dashed line).

In **Figure 5,** the principle of ***start simple*** is demonstrated. While the digital representation of measurements, monitoring systems, machinery, personnel, along with the associated data, will be complex, it is best to start by focusing on the critical system in its simplest representation and then build in complexity gradually.

For Acme, we consider the interaction of the air quality (A.Q.) measurement device with the air purification (A.P.) device the critical part of the model, and so a model representing it will be the minimum viable product. This is ring-fenced in **Figure 5**. Outside of this boundary includes Future Model Extensions that Acme would like in an idealised system, and these can be developed in future.

By selecting a simple starting point, the data model is purposely limited in scope during early development. This allows a minimum viable model to be constructed, one that will satisfy

Acme's needs. A smaller scope also means this can be achieved quickly by avoiding rabbit-holes of development time in components that are not critical.

4.2.2 Acme Corp Data Model – Outline Use Case

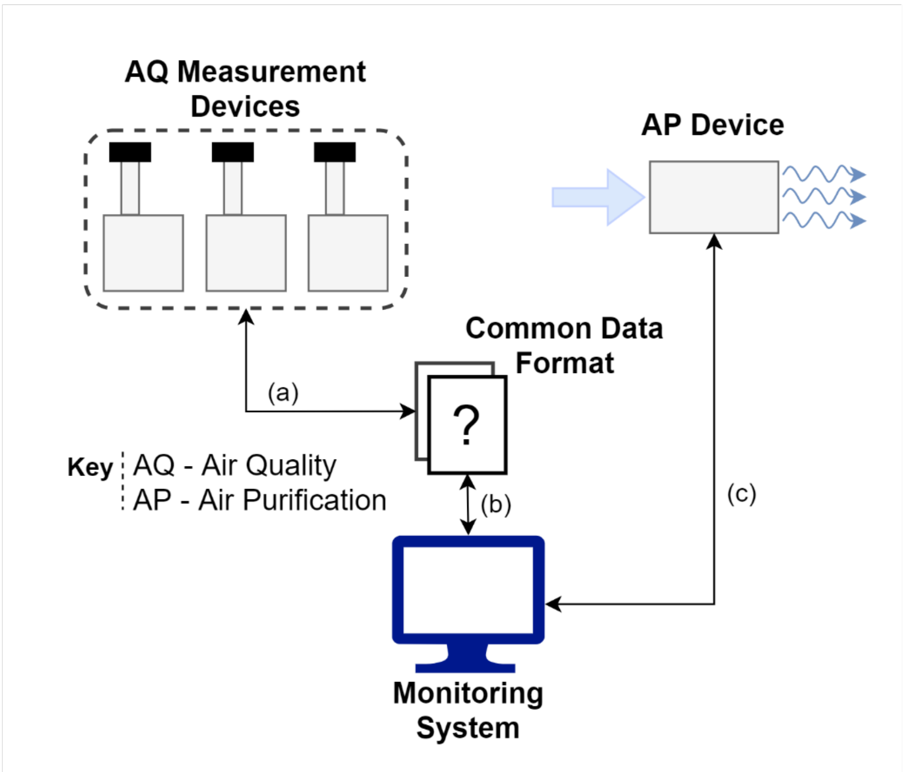

Figure 6 - Outline Use Case. Further detail is added to capture how the systems might interact to meet the requirements of the use case. A monitoring system is included, and a common data format is required for the collective air quality measurements.

Building from ***start simple,*** we then consider the specific **use case** of our model, captured in **Figure 6**. We understand that our reduced Acme example needs to interact with an Air Quality (AQ) measurement device to produce some air quality metric and respond with an action for the Air Purifying (AP) device, but what is specifically required of the data model to achieve this?
After speaking to the measurement team and Acme, the minimum requirements are established for our process. These are shown in **Figure** 6:

- The **measurements** from several AQ measurement devices will be collected.
- **(a)** The AQ measurements must be **aggregated** into a **common data format** so they can be compared.
- **(b)** These AQ measurements will feed into a **central monitoring system**, which is being developed by a different team.
- **(c)** The single AP device will **communicate** with the monitoring system, allowing for automatic **response** and **user-control** of the device.

The team developing the monitoring system will need to be consulted to determine the necessary input and outputs to communicate with the AQ & AP devices.

4.2.3 Acme Corp Data Model – Define Minimum Input/Output (I/O)

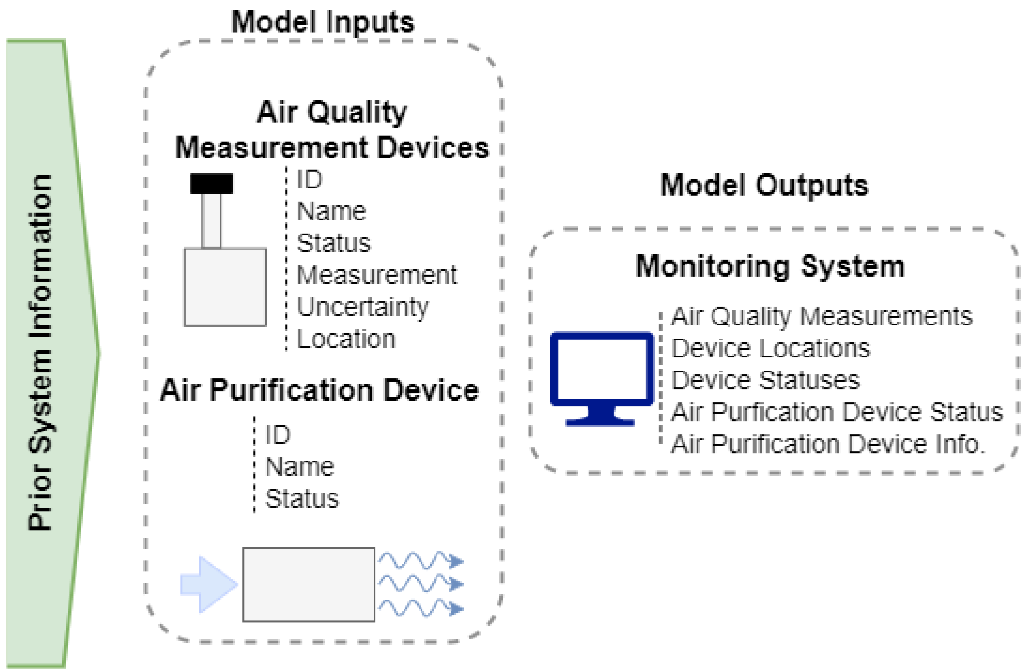

Figure 7 - Minimum I/O. The customer-defined minimum model inputs and outputs are grouped based on their most relevant system. "Prior System Information" captures any data without an explicit source, and so is assumed to already exist.

Figure 7 demonstrates how the Acme **use case** can be used to shape the minimum required inputs and outputs. The ***minimum inputs and outputs*** represent the absolute minimum amount of information and data necessary to achieve the customers' requirements and deliverables, as well as to meet NPL's quality, scientific and metrological requirements.
For Acme, the model ***minimum inputs*** are generated by the two devices in the system, the AQ measurement device and the AP device. However, there is an amount of implicit information which is taken as prior and pre-existing knowledge such as details about each piece of equipment, measurement formats and calibration details.

The effective ***minimum output*** is a monitoring system which collates the measurement data and provides the automatic and user-control commands for the purification device. In a discussion with the Acme engineer, it is determined that the monitoring system will accept and return simple comma separated variable (CSV) files[8]. This format is quite common, so should not pose an issue in the simple model. The two input devices similarly accept and receive CSV files, according to the NPL measurement service staff. This means the aggregation of the measurement data will be easier to implement in the monitoring system.

4.2.4 Acme Corp Data Model – Process Model

8 CSVs files are a common and well-supported data format, conceptually similar to Excel rows and columns (13).

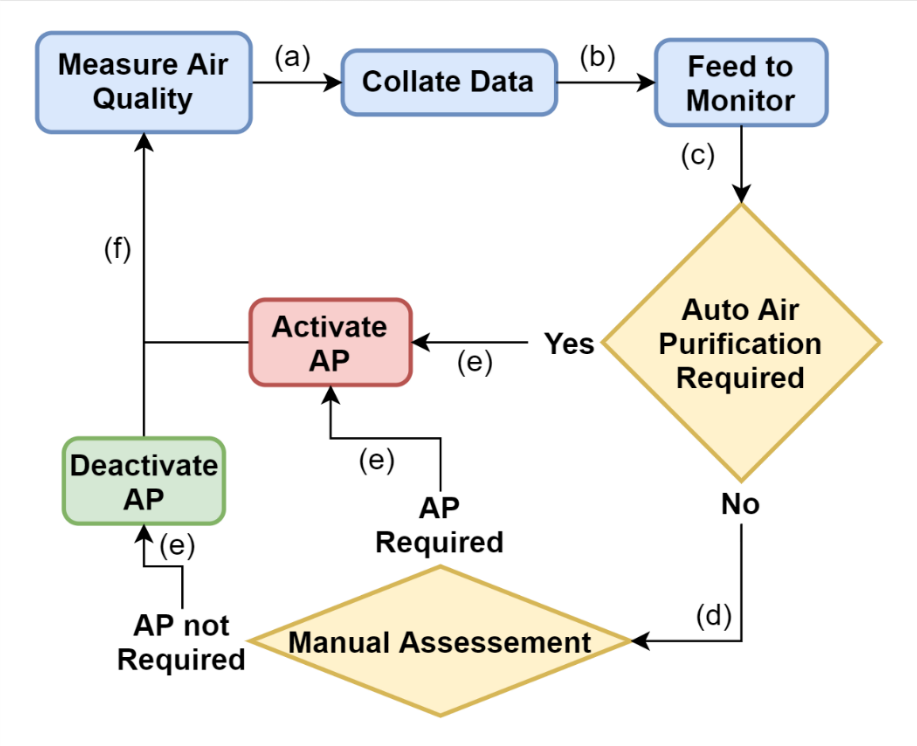

Figure 8 - Process Model. A simple process flow that captures the main logical steps to achieve the air quality and purification monitoring system. Each connecting line (a)-(f) represents an interface which enables the step between decision nodes, described in the text.

Figure 8 establishes the ***process model.*** This diagram reflects the timeline of steps and decision points required to meet the requirements of the use case. Developing the process model needs the input of the customer, who in this case is represented by the process experts, to ensure the steps are logically consistent and do not omit critical stages.

This example has a simple set of steps, however there are implicit interfaces/interactions captured by the connecting arrows, labelled from (a)-(f). Some of these reflect specific processes that enable the progression between steps, while others are just workflow connections. These steps also ensure that the AP device is only ever deactivated after Manual Assessment which in this simple case is deemed the safest option, but in practice may be overly zealous. Through discussion with Acme, the implicit steps are understood to be as follows:

a) AQ measurement device delivers data through physical cable to monitor system PC.
b) Collated data is delivered to monitor system database.
c) Data monitoring – Not a distinct process/step.
d) User access of monitoring system to view AQ measurement data.
e) Monitor system delivers signal to AP devices to activate/deactivate by physical cable.
f) Return to start – Not a distinct process/step.

4.2.5 Acme Corp Data Model – Conceptual Model

[See **Section 3.4.1** for a fuller discussion of Conceptual Models]

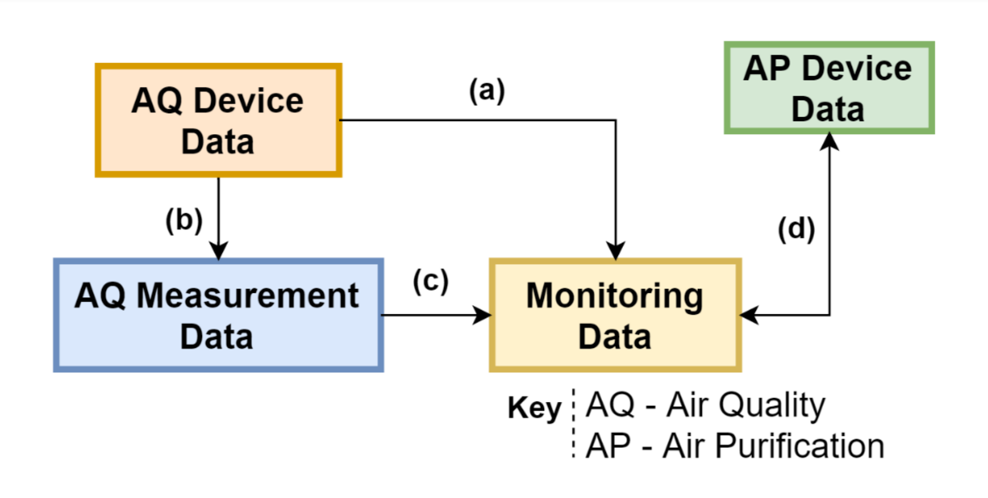

Figure 9 - Conceptual Model. A high-level diagram demonstrating data elements and their connections, labelled from (a)-(d).

Figure 9 demonstrates the ***conceptual model***, the data-compliment to the **process model** described in **Figure 8.** The previous steps have allowed us to understand the minimum data requirements for the use case. The process diagram captures steps which may generate, transform, archive, or monitor data. The conceptual model demonstrates how this data can be grouped into data elements, which are then defined with respect to other elements through relationships.

These relationships allow designers to explore which data elements are interdependent and, as complexity is added later, allow for explicit logical rules to connect and merge data elements in an efficient way.

For the Acme customers, the diagram in **Figure 9** is composed of the following[9]:

- **AQ Device Data:** Fixed information about the measurement device. This must connect to **Monitoring Data** via some relation (a) to link devices to the monitoring system.
- **AQ Measurement Data:** Measurement data related, via some connection (b), to the **AQ Device Data** element. This must also link to **Monitoring Data** via some connection (c).
- **AP Device Data:** An element capturing data on the air purifying device. This must have a two-way connection, (d) to **Monitoring Data** so the device can be (de)activated.
- **Monitoring Data:** A central data element which contains relations to the other data elements, allowing for the monitoring system as required by the use case.

9 This diagram is essentially a Star Diagram, one of many design architectures for data models, where many elements are related to one central data element (14).

4.2.6 Acme Corp Data Model – Logical Model

[See **Section 3.4.2** for a fuller discussion of Logical Models]

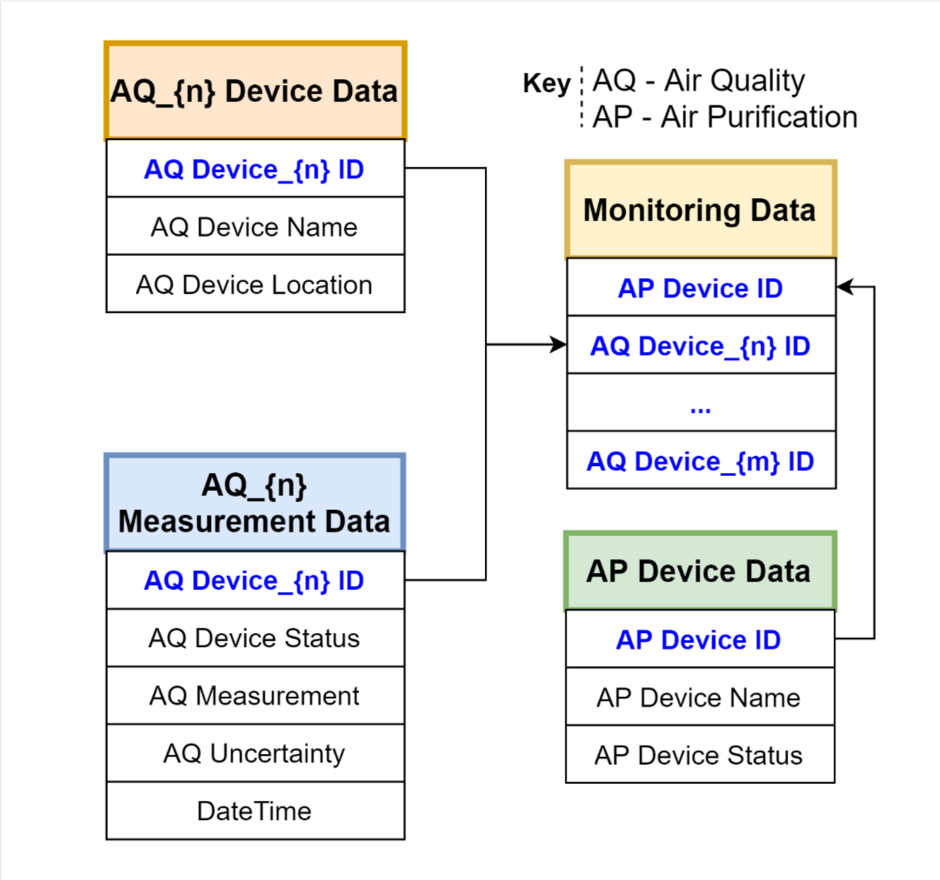

Figure 10 - Logical Model. The conceptual model is extended by considering the data fields that exist within each element, using the minimum input/output for the use case. Each element contains a unique identifier in blue, which is used to build the relationship to the monitoring system.

The ***logical model*** in **Figure 10** shows how the conceptual model is extended by naming specific data entries the data-elements contain and begins to link data elements through relational keys. The **minimum I/O** step should be used to populate the data elements.
The Acme customer model is composed of elements which are uniquely identified through their **primary keys**, highlighted in blue for the AP Device Data, AQ Measurement Data and AP Device elements. However, the Monitoring Data element is described only in reference to **foreign keys,** which are keys belonging to other data elements[10].

This process is known as **normalisation**, which is the act of removing redundancies (repeated data fields) by using reference keys between data elements. It allows for the element such as the AQ Measurement Data element to be the single source of truth for any data associated to that specific device. When, for example, the AQ Device Status changes, it will only have to be updated in one location.

This model has also allowed for additional measurement devices by using the notation "_{n}" where {n} would contain some integer between 1 and m. E.g. Device_{1}, Device_{2}, and so on. This is a personal choice of the designer, and not a specific standard.

---

10 I am using the primary/foreign key terminology from the Entity Relationship Model process, which the curious reader can learn more about here (15).

4.2.7 Acme Corp Data Model – Physical Model

[See **Section 3.4.3** for a fuller discussion of Physical Models]

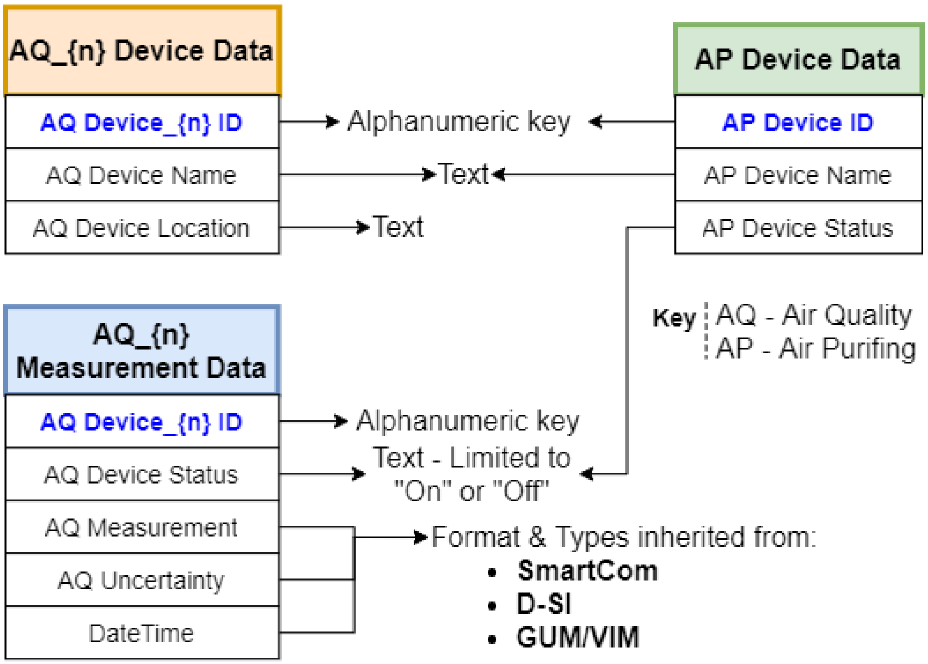

Figure 11 - Physical Model. First steps in capturing some of the data specifications that will be required to create a fully realised physical model.

Figure 11 demonstrates some of the foundation information required to start developing the fully realised ***physical model.*** The physical model should extend the logical model to the conclusion of a fully implemented data model which interfaces with an associated database and may have some additional user or machine interface for editing purposes. As with the previous steps of this guide, the physical model design will be iterative, and the solutions developed will likely need to change over time.

For the customer, Acme, the physical model required us to discuss the nature of every data entry within the data elements described in the **logical model. Figure 11** begins to capture where certain standards will affect the data format (e.g. **SmartCom** [see Appendix] and the **D-SI**), and what values entries will take. It could be further extended depending on Acme's expertise by considering the physical storage size required for each element, the data type, and by creating more technical names for each field.

However, in practice this step needs a **data engineer** to be engaged into the design process. This person will translate the structure of the data model, and the elements it contains, and create an interface which connects the model to one or several databases. In our simple Acme use-case, we are fortunate that all the data is in a simple CSV format and will be delivered in packages through physical cables. More complex examples may make use of application process interfaces (APIs) and cloud-based services to exchange data.

4.2.8 Acme Corp Data Model – Documentation

Documentation is key to ensuring a model is understandable to new users and allows for future developers to understand the design choices made. There is no one correct method to document a model, its design, and its log of changes. Within NPL there is an organisational push towards Gitlab-based solutions where appropriate. However, even a well-structured and regularly updated Word file or .txt file may work for some cases.

For NPL specific software quality requirements, see the following:

- NPL Intranet Page on Software Quality
- NPL Procedure QPNPL/M/013
- Personal insight into the M/013 doc by Trevor Esward of NPL

4.3 CASE STUDIES

In this section, three existing case studies are presented from work carried out by the Data Science department in NPL with both external and NPL internal customers. To product the identity and intellectual property of the customers, some of these case studies have been scrubbed of identifying information.

4.4 CASE STUDY A – EXTERNAL

Background: An external customer approached NPL to help design a data model to homogenise data originating from multiple sources, expedite their project design and ensure the safe outcome of their work. Their system includes experimental sensors and computational simulations which would need to be merged so that an activate machine-learning algorithm could determine corrective measures. A homogenous data package should then be fed this ML process, as well as an additional third-party validation system.

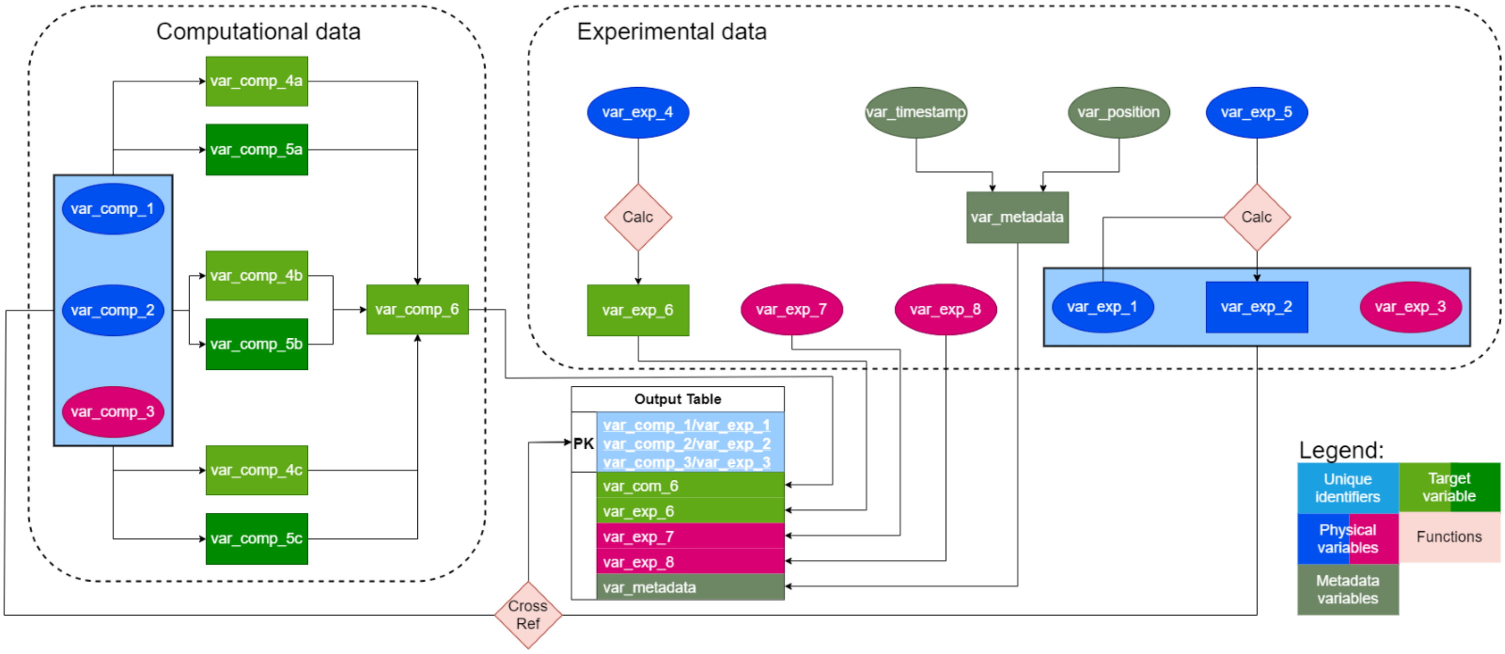

Figure 12 - Case Study A - Data Model. Data model schematic as designed by NPL DS designer. This model combines data from a set of experimental sources with a computation simulation, creating a merged table to be used in two third party validation and control services.

1. Start Simple: The customers made it clear that the external validation process and ML tool were already established. The data model design was to focus on the experimental and computational processes, specifically focusing on data capturing or representing the physics of their process.

2. Outline Use Case: The data model must be able to capture data elements from measurement sensors and computational simulation outputs and merge these into a comparable format. The result must align with the requirements on the ML & external validation processes.

3. Minimum I/O: The customer was not able to provide raw data for their processes to help determine the minimum I/O, however their expert knowledge of their own systems allowed for the field names to be mapped out. It was determined that the ML & validation model would both require CSV files, effectively fixing the model output, and that the inclusion of Geolocation and DateTime metadata was critical to the experimental data.

4. Process Model: Through regular discussions with the customer, a high-level understanding of their process was established. However, because this project was so new, it could not be fleshed out to include practical details on the system design. Enough was collected to generate the minimum data model requirements.

5. Conceptual & 6. Logical Model: Figure 12 shows the conceptual/logical model that was created for this project. This demonstrates how data elements from the experimental and computation inputs are merged through specific relational keys, generating the Output Table shown. Variables are colour coordinated to reflect their commonality, i.e., unique IDs, variables, and so on.

7. Physical Model: The early stage of this project meant that a physical model was not within scope. This would be handled by the customers own engineering team, based on the modelling work carried out by NPL.

4.5 CASE STUDY B – INTERNAL – BATTERY PROJECT

Background: Members of the electrochemistry group (referred to as the Battery Team from now) at NPL approached the DS team to help in the development of a Graphical User Interface (GUI) to enable data fusion between two sets of experimental data gathered from batteries. DS was requested to specify a relational database to capture multi-experiment battery data.

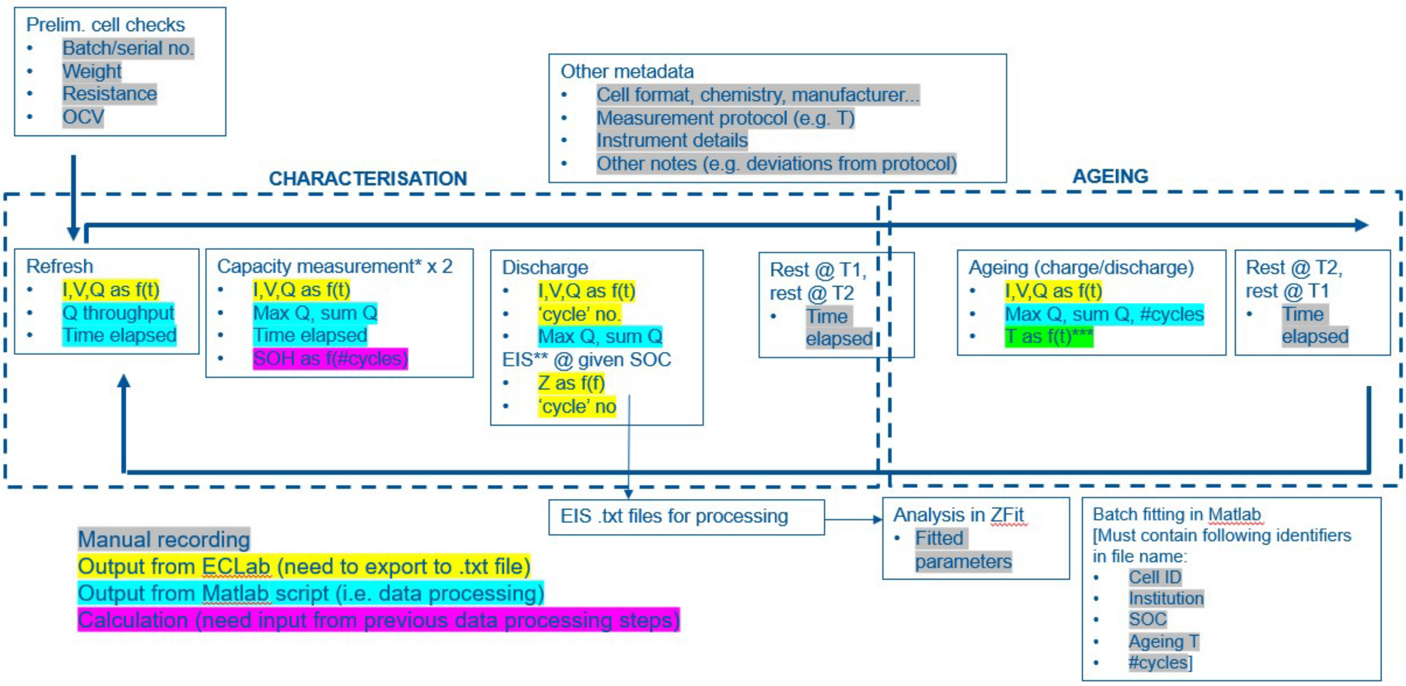

Figure 13 - Case Study B - Customer Diagram. Elements of the experimental setup are laid out by the customer based on each sub-elements use-case and primary task, including physical units.

1. Start Simple: The Battery Team had a strong understanding of their data and processes and provided the diagram shown in **Figure 13.** The diagram captures the data elements required for the two primary battery testing processes: “Characterisation” and “Ageing”, as well as outlining metadata, and some system processes. With this foundation this project started by further scoping the physical variables and equipment used, and then trying to match this information to the pre-existing processes depicted in **Figure 13,** e.g., “Characterisation”.

2. Outline Use Case: The data model developed would be required to capture two distinct experimental processes. These processes generate a mixture of time-series and non-time series-based data structures. Ideally the model design should be scalable, i.e., be able to capture many experimental runs.

3. Minimum I/O: The minimum inputs were initially driven by the Battery Team’s diagram, and then were extended through discussions to include all the data elements required by the physics, the experimental equipment data, and the desired sampling times. These collectively would feed into the minimum output, which in this case would be the digital storage of their measurement data in a connected relational database. No standards were applicable to this project at the time of development.

4. Process Model: The measurement procedures, being known and fixed, became the process model. The specific practical implementation of the process steps was out of scope for the DS teams’ deliverables.

5. Conceptual: Building from the diagram in **Figure 13,** and the establishment of the Minimum I/O, the conceptual diagram was developed through iterative discussions with the Battery Team. They had a clear idea of how they would like to manage and store their data to enable automation, and how best to achieve that in relation to their experiments.

6. Logical Model: Figure 15 shows the established data schema developed for the battery project, essentially capturing the logical model stage of data model development. The diagram primarily consists of data tables containing instrument data and measurements, related to experimental runs and physical values via relational keys. This is a living document, as seen by the number of as-of-yet unclear elements, e.g., the variables appended with a question mark in the LCT_cycle_index table shown in **Figure 14**.

7. Physical Model: The physical model is currently under development as part of a project extension of this work, where a demonstrator will be created to showcase the model.

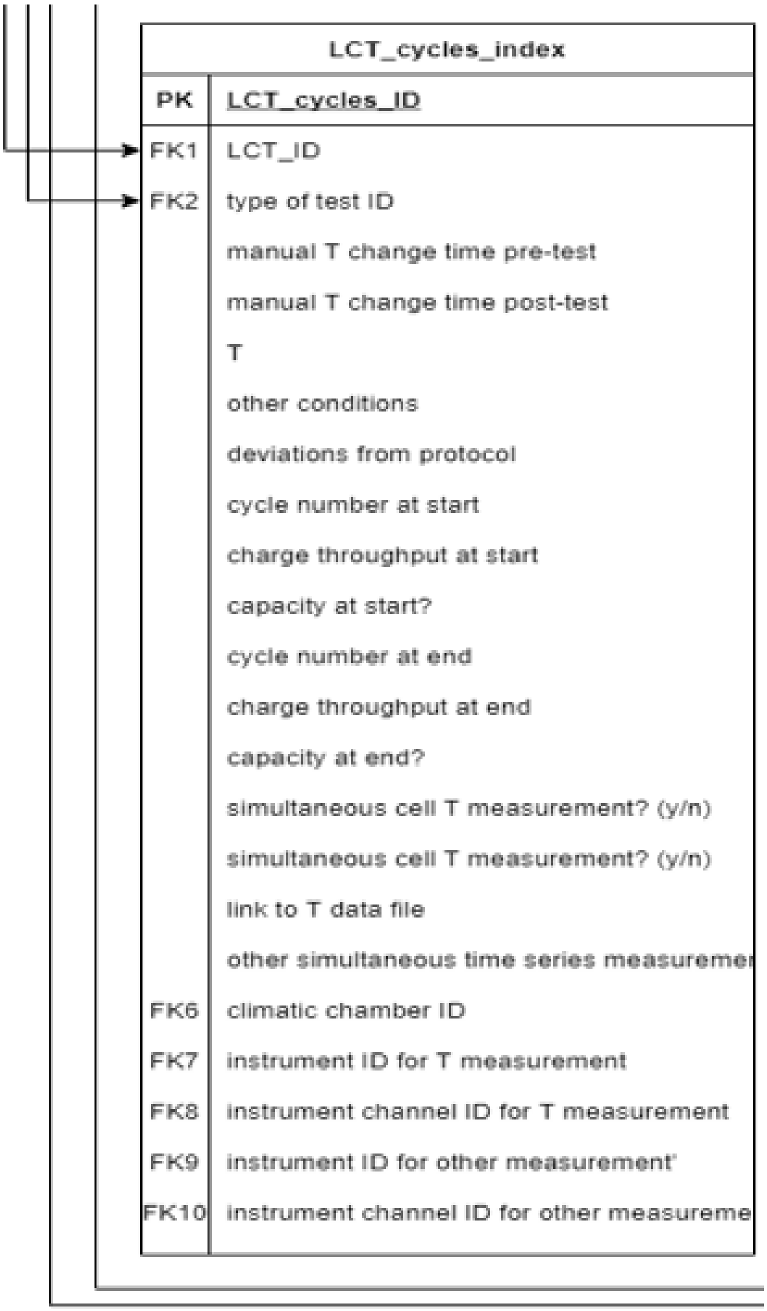

Figure 14 - Case Study B - Data Schema. A cropped view on the data schema created for the Battery Metrology project. The full image is given in Figure 15.

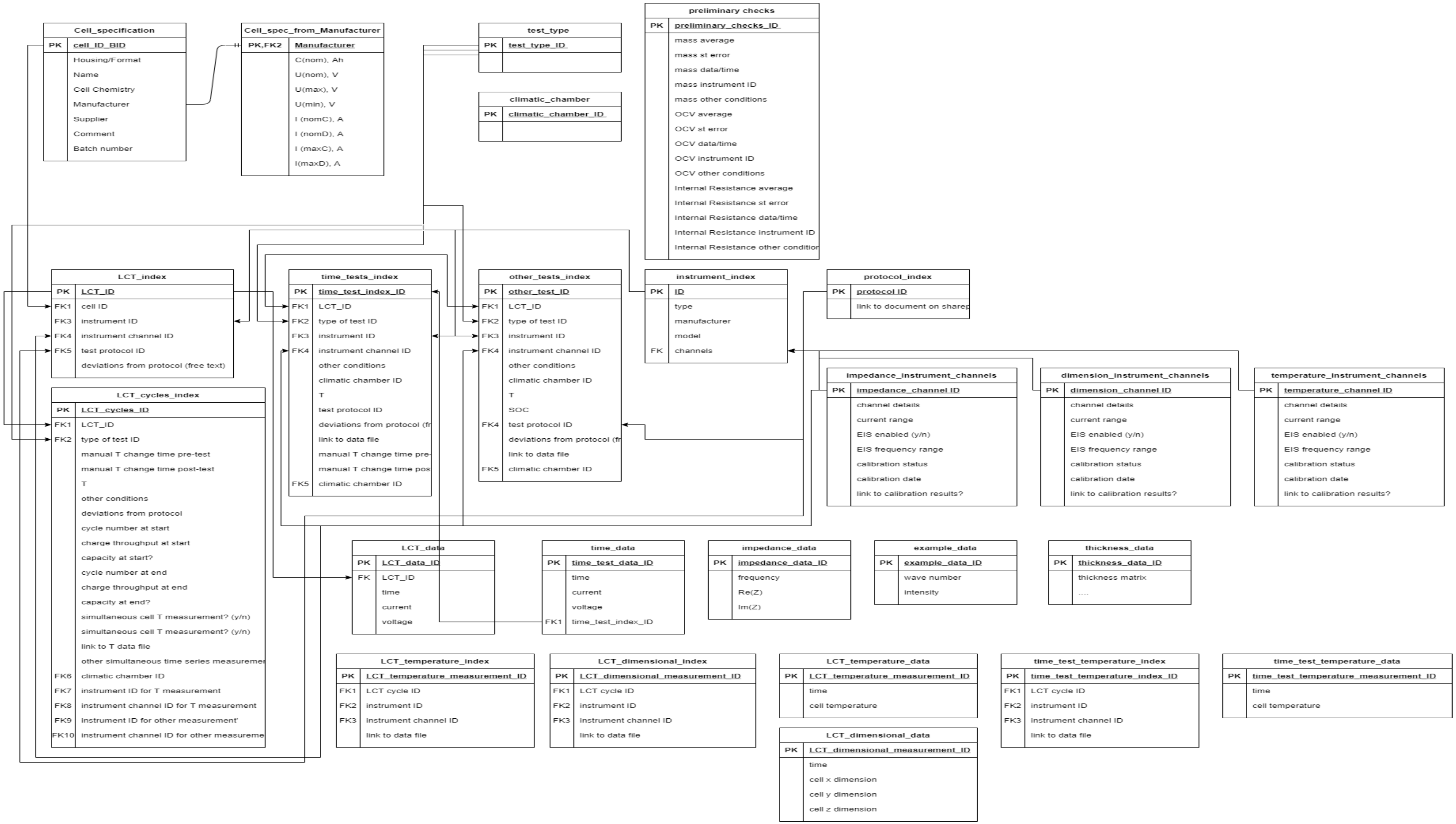

Figure 15 - Case Study B – Data Schema. The full data schema created for the Battery Metrology Project. The foundation of which is based on the original customer diagram in Figure 13. Image is shown to demonstrate the complexity associated to one project data model.

4.6 CASE STUDY C – INTERNAL – HYDROPHONE PROJECT

Premise: The Ultrasound and Underwater Acoustics (UUA) group approached the Data Science team to assist in the modernisation of their hydrophone calibration procedure. This process aimed to digitalise their well-established and fixed process which, at the start of the project, required users to manually access a cumbersome document database, as well as several LabView programs. A centralised digital process would simplify calibrations; enable historical data analysis (a key requirement); and allow for the creation of Digital Calibration Certificates (DCC).

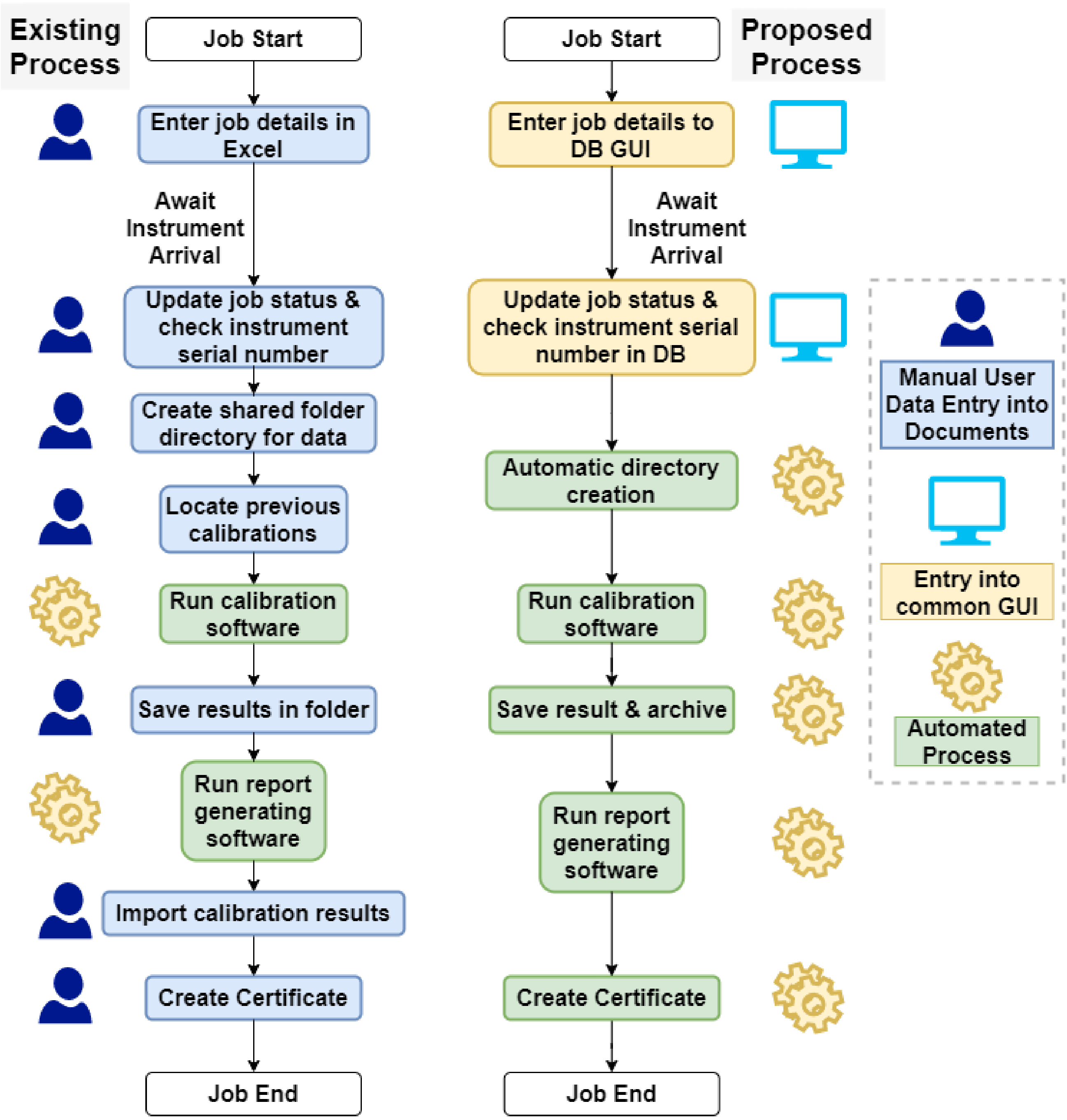

Figure 16 - Case Study C - Hydrophone Diagram. The pre-existing hydrophone calibration procedure (left), in which users manually access and edit several documents and LabView programs. The idealised process making use of a central database (DB)

and GUI dashboard which would interact with LabView directly. The LabView interfaces are not shown here.

1. Start Simple: The hydrophone team have an immutable process to which the data model must conform. This provided a natural starting point for the project, and with the customer a workflow was created to capture the current process, shown on the left-hand side of **Figure 16.** From this, key data and metadata elements could be identified and extracted to form the foundation of the model.

2. Outline Use Case: The data model generated would interface with a new storage database and a web dashboard for document collecting and parsing. The right-hand side of **Figure 16** shows the outline of the use case developed with the hydrophone group.

3. Minimum I/O: The minimum inputs for the model align to the calibration procedure, and so include: calibration data, calibration conditions, and calibration references and metadata on measurands. The overall minimum deliverable is to establish a database to capture this information, with the GUI as a complementary deliverable. The database for DCC creation was deemed a project extension if time allowed.

Regarding the application of standards, the customer outlined that they would like to implement the Digital-SI (D-SI) SmartCom, see **Appendix B – Metadata Details**, standard into their physical value representation. They also made the model designer aware that a number of their data values would need to be checked for GDPR compliance due to the contents, in this case requiring that the outputs aligned to the customers' existing GDPR compliance guidelines.

4. Process Model: The primary process model is the procedure outlined in the right-hand side of **Figure 16.**

5. Conceptual: The customer was able to provide an understanding of the mapping between key data and metadata elements for the calibration processes, allowing for high-level relationships to be established.

6. Logical Model: Extending the conceptual model, relational tables were constructed to populate the data and metadata aspects. Again, this was done with close guidance from the customer. Specifically, they provided sample data on which to base the model.

7. Physical Model: The DS project lead was also the responsible person for the development of the database, i.e., the data engineer. Their team successfully created the relational database and integration service, with a message brokering system to manage interactions. The relevant schemas were integrated into Gitlab for version control, following NPL software quality guidelines, paired with Python and Java scripts to manage systems and a PostGreSQL database component.

5 APPENDIX A – GLOSSARY OF TERMS

5.1 DATA MODELLING TERMS & TOPICS

Data quality

Defining data quality or ‘goodness’ is highly subjective. Broadly, the quality of the data can only be understood in reference to how it will be used and how it conforms to applicable standards.

Data granularity

Data which is comprised of many small and low-scope elements can be considered highly granular. For example, A name entry “{name: Joe W. Bloggs}” can be made more granular by division of the name components into “{{firstName: Joe}, {middleInitial: W} and {lastName: Bloggs}}”. (7)
Increasing granularity will lead to greater storage requirements, but can improve data querying and retrieval if the data model is well designed.

Data mutability

Data entries which can be change are considered mutable. In a design process, data mutability needs to be tracked to understand when and how data elements are altered. Similarly, immutable entries should be flagged to ensure they never change.

Data sensitivity

Data are considered sensitive when they must adhere to all or some of the following requirements: health and safety, security, GDPR, intellectual property, patent, legal, and financial.

Data bifurcation/duplication

Data elements may need to duplicated, or new parent-child hierarchies established, throughout a process flow and/or data model. The creation of data branches based on a parent-child hierarchy relating to the data source can be considered data bifurcation.
For example, a single batch of samples represented in the data model with one ID value is sub-divided, requiring a new identification for each sample while retaining the parent batch information.

Optional and mandatory data

Some complex use-cases may have scenarios with optional entries and workflows. Enforcing mandatory data elements and data formats gives better confidence in the data model output, but this can be very rigid. Process checks may be needed to limit errors when both optional and mandatory elements are present in the model.

Future proofing

All data models will eventually require changes. Through thorough documentation of the design choices made, models can be better future proofed to enable later users to confidently make changes.

Scope creep

This is a common issue within data modelling and general project management tasks. It is largely unavoidable but can be managed through understanding the requirements and use-cases early, and then relegating non-critical scope changes to later model iterations.

Version control (of model)

Documentation which captures the historic record of changes is increasingly important within NPL. Adhering to NPL's software quality control procedures will help manage this, as well as integrating Gitlab pipelines where applicable.

Version/change control (of data)

Some applications will require changes in data to be documented as part of the model. This may include a datetime-stamp, user details, and change reasons all stored in some historic database relating to the primary model. This secondary data could be considered metadata to the primary data. This can be a non-trivial problem which occurs regularly in complex procedures, especially those which handle physical systems which can break down or need resetting. Such systems will need the engagement of data engineers and architects early to ensure it is well implemented and, critically, useful to the data model use-case.

Normalisation

A more complex concept, which relates to the task of reducing redundancy in the data model. In simple terms, this means reducing the amount of repeated data across the model. A more normalised model is more efficient to store, at the expense of how complex queries on the data may need to be (8).

6 APPENDIX B – METADATA DETAILS

- **Descriptive**

Information primarily for discovery and identification of the data, including human readable elements. In the case of a document or paper, it may include the language, author/creator, title, or abstract. Elements such as topic labels, URIs (universal resource identifier), parent/child hierarchies, and primary data types, (e.g. classes, type primitives such a string, float, Boolean etc.) could be included here for both human and machine readability.

- **Structural**

Data contextual information such as hierarchy and relational information may be separated into their own category. For example, a hierarchy may describe experimental data belonging to measurement objective X, as part of Project Y, within the group Z.
It may also include information on the structure of an item, such as document page ordering, or tabulated data format.

- **Administrative (incl. security)**

Information pertaining to access rights, security restrictions, creation datetimes and creator information, including legal, copyright, licence, and IP information.

- **Technical**

Information regarding data storage sizes, compression methods, and may also include data types if not included in the descriptive information. May include data precision required for storage optimisation. For text and mathematical notation, Unicode standards and formatting information could be included here.

- **Version control**

Data structure and information pertaining to the version control implemented for the primary data.

- **Methodology**

Information regarding specific steps and processes carried out to generate the data. The may include external references to digital lab notes, equipment used, materials, users, and academic journals. One of the most challenging aspects of metadata to capture digitally in the context of machine-actionable data. If not contained in their own elements, this section may report on the data uncertainties and digital-SI (D-SI) standards used, as follows:

  - **Uncertainties (SmartCom, GUM)**

Information regarding the uncertainty values, types and standards used. The **SmartCom** project outputs, paired with the **Guide to Uncertainty Measurement (GUM)** documents, provide a suggested uncertainty metadata structure.

  - **SI units (D-SI)**

The CIPM Task Group and Experts Group are working with BIPM to develop infrastructure, standards and tools that will enable traceability chains back to a unique point of reference for the SI to be created and reported digitally.

- **Digital Calibration Certificates (DCC)**

Both a type of metadata, as well as a metadata standard. Describes architecture for calibration certificates, such as those created by NPL's measurement services. Integrates the D-SI and SmartCom standards into its structure. A DCC may be included as part of a larger data package for a whole group, or as a stand-alone instance. It includes both human- and machine-readable elements (9).

7 BIBLIOGRAPHY

1. **Wright, Louise.** *Digital NML: NPL's Internal Landscape - Not for Distribution .* Data Science, NPL. s.l. : NPL, 2021.
2. *The FAIR Guiding Principles for scientific data management and stewardship.* **Wilkinson, M., Dumontier, M., Aalbersberg, I. et al.** 160018, s.l. : Sci Data, 2016, Vol. 3.
3. **OECD.** Glossary of Statistical Terms - Data Definition. *OECD Stats.* [Online] [Cited: 10 June 2022.] https://stats.oecd.org/glossary/detail.asp?ID=532 .
4. **GeeksforGeeks.** Data Abstraction and Data Independence. [Online] [Cited: 10 June 2022.] : https://www.geeksforgeeks.org/data-abstraction-and-data-independence/.
5. **Lines, K and Krishnamurthy, H.** *Model-based Systems Engineering at NPL: An initial investigation.* s.l. : NPL, 2023. MS 42.
6. **SysML.** SYsML - What is SysML? [Online] [Cited: 10 June 2022.] https://sysml.org/.
7. **Technopedia.** What is Granular Data? [Online] [Cited: 10 June 2022.] https://www.techopedia.com/definition/31722/granular-data.
8. **w3.org.** Database Normalization. [Online] [Cited: 10 June 2022.] https://www.w3schools.in/DBMS/database-normalization/.
9. **Davenhall, C.** Scientific Metadata. [Online] DCC & National E Science Centre. [Cited: 10 June 2022.] https://www.dcc.ac.uk/resources/curation-reference-manual/chapters-production/scientific-metadata.
10. **Wikipedia.** Unified Modelling Language. *Wikipedia.* [Online] [Cited: 10 June 2022.] https://en.wikipedia.org/wiki/Unified_Modeling_Language.
11. **UML.** Welcome to UML. [Online] [Cited: 10 June 2022.] https://www.uml.org/.
12. **W3.** XML Introduction. *W3schools.* [Online] [Cited: 10 June 2022.] https://www.w3schools.com/xml/xml_whatis.asp.
13. **Python.org.** CSV File Reading and Writing. [Online] [Cited: June 10 2022.] https://docs.python.org/3/library/csv.html.
14. **Microsoft.** Understand STAR Schema and the Importance for Power BI. [Online] [Cited: 10 June 2022.] https://docs.microsoft.com/en-us/power-bi/guidance/star-schema.
15. **w3.org.** Entity Relationship [ER] Model. [Online] [Cited: 10 June 2022.] https://www.w3schools.in/dbms/er-model .