Intro

Defining a master-and reference data model by solely configuring the meta, classes, properties and relations. That is what our approach on MRD is all about.

This article describes the challenges and approach regarding master reference data at TenneT. We were blind with respect to our generic data model. We have developed a **prototype** and **working method** that enables us to **quickly and effectively gain insights into our generic data model**.

At TenneT, we operate within a complex and widespread working and data environment that experiences a high level of dynamism in the use of Master, Reference, and Transactional data. Like most companies, we have entered the world of big data and realized that we just started our journey to become data driven.

The prototype is made available with this article and can be downloaded. If you would like to first delve into the background of our approach, please read this article. Alternatively, you can dive directly into the possibilities of our approach; download the prototype (**Tool**) and read the attached step-by-step instructions (**TOOL MANUAL**).

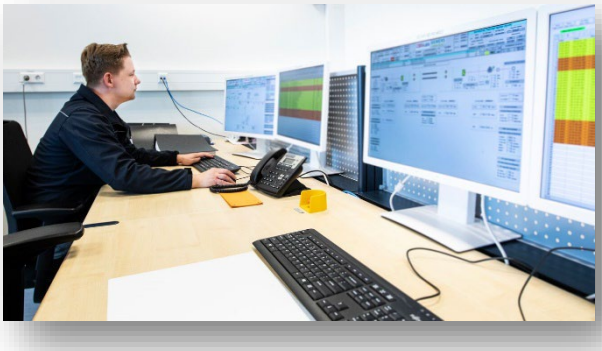
TenneT TSO

TenneT TSO is a transmission system operator responsible for overseeing the operation of high-voltage electricity grids (110 kV and higher) in the Netherlands and part of Germany. Our main task is to ensure a secure electricity supply for the nearly 43 million people residing in the areas we serve. To accomplish this, we provide electricity transmission services, system operations services, and facilitate the energy market. Additionally, we are involved in the design, construction, maintenance, and operation of our grid.

Our challenges

To execute our operations effectively, we handle vast amounts of data, encompassing grid operations such as forecasting, monitoring, steering, verification and settlement. This data exchange extends beyond our domestic market to include neighbouring countries and ENTSO-e partners, necessitating the reliance on master reference data.

TenneT operates in diverse environments to support electricity market facilitation, transmission of electricity, and maintaining the balance of the electricity system. Collaboration between our business units, amongst them Asset Management (building and maintaining the grid), Customers and Markets (Customer info/management) and System Operations (SOP) is essential. SOP, my department, focuses on transmission services, system services, and market facilitation, relying on master and reference data sourced from other units.



TenneT comprises German and Dutch divisions, adapting its operations to each country's unique market model. Although the processes for preparation, operations, and settlement yield identical outcomes, they follow distinct approaches within each country.

In the Netherlands, TenneT manages high-voltage and extra-high-voltage grids responsible for energy transmission and exchange internationally and with medium- and low-voltage grids operated by regional grid operators, ultimately supplying consumers including companies and residential homes.

International energy exchange via interconnectors involves careful coordination to avoid exceeding physical capacity limits. Transmission System Operators (TSOs) collaborate to maintain a reliable supply and efficient energy management across borders.

Similarly, domestic energy delivery in the Netherlands requires continuous coordination to balance supply and demand, involving collaboration between TenneT and regional grid operators such as Liander, Stedin, Enexis, Rendo, Coteq, and WestlandInfra Netbeheer.

Besides physical grids, the energy sector operates within designated areas, each serving specific functions to streamline cross-border energy transactions and enhance collaboration between stakeholders.

Within these administrative areas are physical grids with connections to other girds, and production / consumption installations. Market participants, including suppliers, metering operators, grid operators, and TSOs, need to be linked to all relevant assets.

The operational process is carried out by the System Operations unit, in close collaboration with the Asset Management and Customers and Markets units. These units in turn collaborate with other units within TenneT.

In the System Operations unit, we have distinct processes for trade matching and market facilitation, transmission monitoring, market exchange, electricity exchange with Norway, Denmark, and England (High Voltage Direct Coupling via sea cable), matching via land connections with Germany and Belgium, monitoring and settlement of grid operator activities, and more.

All the processes within the different business units are performed using various applications. Some applications are designed to execute multiple processes, while others are specifically tailored to handle a single process. This setup is functional, but not optimal.

It is common that master data managed in one information silo (application) is partially or completely present in one or more other silos. For example, a market participant may have multiple instances with their own name, Chamber of Commerce number, and role in different systems. The same situation applies to assets, where identical physical assets are stored in numerous different environments, including databases of the Asset Management unit, System Operations unit, Grid Planning unit, and others, each serving their specific purposes. However, many of these records contain characteristics that are largely or even entirely present elsewhere in the organization's databases.

The maintenance intensity is, as you will understand, very high.

Therefore, our goal is to transition towards a centrally managed Master/Reference Data (MRD) environment. This centralized system will hold the single source of truth that is valid for all systems. Any changes made to a record within the MRD will be implemented at the central location, and the updated information will be distributed to all the relevant systems that require this data. By adopting this approach, we aim to ensure consistency and accuracy throughout the organization, eliminating the need for multiple local information silos and reducing data duplication.

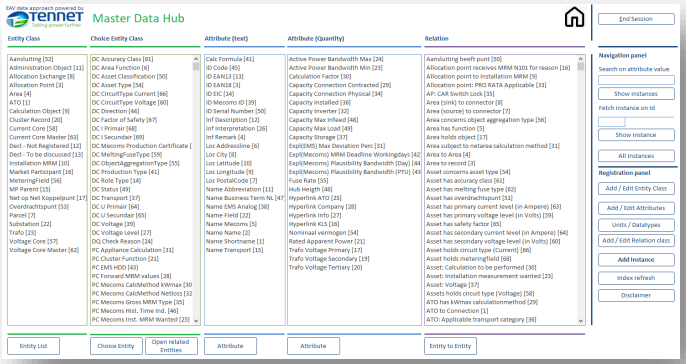
The Data Management Body of Knowledge (DAMA/DMBOK) emphasizes the importance of Master/Reference Data (MRD), and it is evident that a well-established MRD organization plays a crucial role in enabling effective and efficient business processes. This principle applies to any company or organization, regardless of its industry or sector. By having a centralized MRD system, businesses can ensure data consistency, accuracy, and reliability across various systems and processes. This not only enhances operational efficiency but also enables better decision-making, data analysis, and overall data governance. A robust MRD framework is essential for maintaining data integrity and driving business success in today's data-driven landscape.

Certainly, large organizations, such as TenneT, commonly aspire to establish a centralized Master/Reference Data (MRD) system. This desire is particularly heightened in the case of TenneT due to operating in a highly dynamic environment. Each year brings forth its unique set of challenges and considerations.

In the context of 2023, several key issues are at stake: The energy transition and Congestion issues the bidding function area split (NL to NL North and NL South), the possible sale of TenneT Germany , blockchain technology, expanding time-to-market activities, data driven developments, and so on.

Defining a central MRD environment

Establishing a central MRD requires the development of a comprehensive and unified data model. This process can be complex due to variations in local practices and differing approaches to data management. In decentralized environments, each system or department may maintain its own set of data with inconsistencies, such as registering different names or failing to update information like name changes consistently across all systems.



Moreover, local administrations may employ their own unique key values, leading to conflicting or duplicated data. Additionally, data might be stored at different levels of aggregation in different systems, further exacerbating the challenge of achieving data consistency and coherence across the many interconnected systems.

Creating a central MRD involves addressing these complexities by harmonizing data structures, standardizing naming conventions, and implementing data governance practices. It requires collaboration among various stakeholders, including system administrators, data managers, and subject matter experts, to ensure a common understanding and agreement on data definitions, relationships, and rules.

Through this concerted effort, organizations like TenneT can overcome the obstacles posed by disparate local systems and establish a unified data model that serves as the single source of truth. This central model facilitates accurate and reliable data exchange, enhances data integrity, and enables effective decision-making and analysis across the entire organization.

Every "local" application has the potential to possess a robust database for storing the MRD. A well-designed MRD, based on a relational model and adhering to the standardization principles, ensures its effectiveness. However, it is unfortunate that within TenneT, there are numerous applications where the database suffers from flaws.

Reverse engineering our data model instead of normalizing

A comprehensive re-evaluation of TenneT's core purpose could lead to the development of a completely new model, which can then be normalized based on the analysis. However, this process would take several years to complete. Furthermore, even after successfully creating a robust and consistent model, there will still be existing applications that rely on MRD that may not be available in the new enterprise data model, or may require modifications. In essence, while working towards implementing a central enterprise master reference data model, it is crucial to ensure the continued functioning of these local applications by establishing connections between them and the new central environment.

Instead of pursuing normalization, we opted for a “reverse engineering” approach. We began by examining individual applications (referred to as local information silos) and analysing the data stored within them.

Our initial focus was on the grid operator application, which contains a central "main" table housing all records (referred to as instances) where we capture attribute values related to their characteristics.

In our new model, we aim to register instances in separate classes instead of having them all in one main table. In a traditional relational model, this would involve creating separate tables for each class. However, in our prototype, which is based on a variation of the Entity-Attribute-Value (EAV) model, this is not necessary. Instead, we can simply register the class itself without the need for separate tables.

In addition to registering instances and classes, we can also register attributes in our model. Attributes can be compared to columns in a relational database. They capture specific information about an instance, such as the name of an asset or an identifier like the EAN code. Additionally, there are other types of attributes, "quantity attributes" that represent values like installed or contracted capacity. These attributes provide additional details and characteristics associated with the instances in our model.

With this reverse engineering approach, we first fully processed the actual data from the network management environment into the prototype. This resulted in a significant number of entity classes (tables) with attributes (columns) and attribute values. The relationships between the classes were also established.

After fully processing the data from the network management environment (thus setting up the model), we proceeded to the next environment. The complete dataset from that environment was once again processed, and the new data introduced not only expansions but also brought changes to the initial data model. Designing a relational model for the entire TenneT organization, including units such as SO, AM, and CMT, along with all the operational environments (which consist of numerous local mrd-tables), would inevitably lead to a database model of significant complexity. The complexity arises from the interconnectedness of various components. This complexity exists in both the relational model and the EAV model. However, in the EAV model, you don't need to create actual tables for each class or attribute; Instead, you simply enter the class or attribute as needed. This flexibility allows for a more dynamic and adaptable approach to capturing and organizing data within the model.

How EAV differs from a relational database

EAV (Entity-Attribute-Value) is a data modelling approach that differs from a traditional relational database model in several ways. In a relational database, data is structured into tables, where each table represents an entity class (subject), and each column represents an attribute of that entity. The values are then stored in the rows of the table.

In contrast, EAV takes a more object-oriented approach to data modelling. Instead of using fixed tables with predefined columns, EAV allows for more flexibility in storing and retrieving data. In an EAV model, data is stored as a collection of triplets, consisting of an entity, an attribute, and its corresponding value.

This flexibility allows for the dynamic addition of attributes without altering the structure of the database schema. It can be useful in scenarios where the number or nature of attributes may vary significantly across different entities or when dealing with highly dynamic or sparse data.

In an EAV model, all records or instances are stored in a single table, regardless of what those instances represent. Each record (instance) belongs to a specific class, such as Customer or Product. The classes themselves are stored in a separate table.

In comparison to a relational database, where each table represents an entity class, the columns in a relational table correspond to attributes in the EAV model. Attributes can be seen as the "questions" that you can ask about an entity. Examples of attributes include Name, Code, and Address. The answers to these questions, specific to each record or entity, are stored as attribute values.

To summarize, in EAV, the data is organized into a single table for all records, with separate tables for classes and attributes. Attributes correspond to the columns in a relational database, while the attribute values represent the specific answers or values associated with each record.

In an EAV model, the relationships between instances are also stored in an object-oriented manner. The relationship "class" (entity-to-entity type) describes the type of relationship, such as "Customer Bought Product." The actual relationship between two specific instances or records is then stored in the Entity-to-Entity table.

This table captures the associations or connections between instances of different classes. It specifies which specific instances are related to each other based on the defined relationship class. By storing the relationships centrally, the EAV model allows for flexible and dynamic connections between entities, accommodating various types of relationships and their instances.

The essence of EAV : In the EAV model the actual data concerning records themselves are written in three tables. The “subjects” (entityclass), the attributes (characteristic/property of the subject) and the values. Besides, relationships between the records are registered centrally. One table explaining the type of relationship, and one table registering the relationships between the instances. This model offers flexibility and is reminiscent of object-oriented approaches.

In EAV model, the schema or structure is dynamic or flexible, because any instance can have any number of attributes, and any attribute can have any type of value. This makes it easier to add or remove attributes without changing the overall structure of the database.

The downside: EAV model can be slower and is more complex to query compared to a relational model.

The EAV principle is well-documented on various platforms, such as Wikipedia and YouTube, where you can find detailed explanations. Additionally, you can explore its practical implementation and our simplified approach by following the step-by-step instructions provided in the enclosed manual.

Why we chose our EAV approach

Setting up the initial model in the EAV approach is easier and more cost-effective compared to a traditional relational approach. However, the main advantage of working with an EAV model is the ability to make continuous changes during the reverse engineering process. At TenneT, the inventory process began with the grid operator environment, and once that step is completed and incorporated into the prototype, the next application was processed. This process builds on the existing dataset, enriching it with new data while maintaining the same model structure.

Incorporating new data into the existing model involves adding attributes to relevant existing entities and introducing new entities that belong to either existing or new classes. It is also possible to identify the need for splitting existing classes into multiple classes or promoting a class to a main class with subclasses. EAV allows for the flexibility to adapt the model as needed during this data processing phase.

Handling data conflicts is also easier in the EAV model. During the modeling phase, where the database is solely focused on compiling the data model, you can introduce, terminate (delete), modify, or split a class and its related information. If the MRD environment already provides production environments with MRD connected to applications or provides applications with MRD in other ways, you can make changes to attributes, classes, and relationships by terminating the old situation and starting from a given date with the updated dataset for the new situation.

This level of flexibility and adaptability is highly valuable in TenneT's dynamic environment.

Just follow the data: defining the MRD model made easy…

The EAV Prototype allows us to export the MRD from a local information silo, normalise/process it and then incorporate it into the prototype in such a way that it enriches the current MR dataset with the specific info from the relevant information silo. However radical; Classes can be introduced or terminated. Attributes can be attached to one or more classes. Relationships between classes can be introduced, removed, replaced or changed. And once the info from the current information silo has been completely processed and all data quality requirements have been met, we move on to the next application/information silo. Just until everything that is scope of MRD is processed.

So once we're done (if ever …., given, again, our dynamic environment), we will have collected and linked all the data (because we are really processing the data) from all those sources. The data is obviously not usable (you work with exports that immediately become obsolete from day 1) but you have full insight into all your different types of objects (classes) including how those objects relate one to the other and what you need to register per type of object.

Perfect input to set up your canonical data model.

To experience the simplicity of our approach, I recommend working with the enclosed MsAccess file, which contains the tool used for "reverse engineering modelling" within our business unit. The accompanying manual (TOOL MANUAL) provides a step-by-step explanation of the tool's operation. Additionally, you will find an explanation of the EAV storage methodology, further detailing the approach.

Credits; Thank you to…

My TenneT colleagues incl. Bert Huijskes (Bunsiness Analyst), Bob Mantel (Projectleader), Hans van Keulen (DataGovernance Officer), Nico Vlug/Willem Nijntjes/Alfred Kayser (BA), Caspar Derksen (Domain Architect), Henrie Mathijssen (Manager), our consultants from Twycis/Florijn/EZAccess and ChatGPT:-). With valuable insights we made a very interesting and easy to adapt tool helping us to gain a lot of insight..