

November 2025

BEV Data Vocabulary

EVMO Project Technical Report



Prepared for the Electric Vehicles in Mining Operations (EVMO) Project



5th November 2025

Authors

Edward Fagan, *Zero Nexus*

David Lyon, *Zero Nexus*

Funded by the Natural Resources Canada (NRCan) Zero-Emissions Vehicles Awareness Initiative ([ZEVAI](#)) program.

Funded in part by a grant from the Mining Innovation Commercialization Accelerator ([MICA](#)).

This report is licensed under [CC BY-NC-SA 4.0](#). This licence enables re-users to distribute, remix, adapt and build upon the material in any medium or format for non-commercial purposes only. Attribution must be given to the creator of the material, and any newly generated material must be licensed under identical terms.



Table of Contents

Executive Summary	3
What is a Vocabulary?.....	4
The Need for a Vocabulary.....	5
The SKOS Format	5
Example Use Case	6
Using the Vocabulary in Your Work.....	9

Executive Summary

Battery electric vehicles (BEVs) introduce a range of new systems into a mining operation, bringing with them some data interoperability challenges. Organizing, cleaning and preparing BEV data takes considerable time and effort and presents a barrier to consistent analysis.

Tackling these data interoperability issues will help enable us to make better use of data, spend less time on clean-up, make data transfer easier and ultimately lead to more effective analyses. Reducing the barriers to data use also supports easier integration of digital technologies within wider digitalization initiatives.

The first step on this journey is standardizing the format of BEV data and creating common definitions for the data types used in data analytics projects. The vocabulary described in this project aims to make data more accessible and useful for engineers, analysts and researchers working with mining BEVs. This open vocabulary has been published in its own [GitHub repository](#) along with examples showcasing its use.

What is a Vocabulary?

A vocabulary is a curated list of terms to make data collection and analysis more standardized and consistent. It is a simple form of knowledge organization system.

Our BEV data vocabulary aims to define the different types of data that describe BEVs and BEV support equipment (like chargers). The types of data included in the vocabulary range from data describing sub-systems in the vehicles like “battery ID”, “average voltage” or “rated power” to data on aspects of mine production like “availability”, “operating delay” or “mean time to repair”.

The data types in our vocabulary are grouped by category to make them easier to find and interpret. *Figure 1* shows the BEV-related categories covered by the vocabulary and the data types associated with the “Battery Data” category.

The vocabulary creates a standard set of definitions for the characteristics of BEVs and the data they generate. We are aiming to make this vocabulary a useful tool for engineers, operators and analysts working with BEVs for a couple of key use cases:

1. **Basic terminology management:** creating a documented list of the relevant terms for BEVs and their support equipment to help with identifying and interpreting data.
2. **Data transfer and analysis:** providing descriptions of BEV data types to help make data transfer and data analysis tasks more consistent across sites, projects and over time.

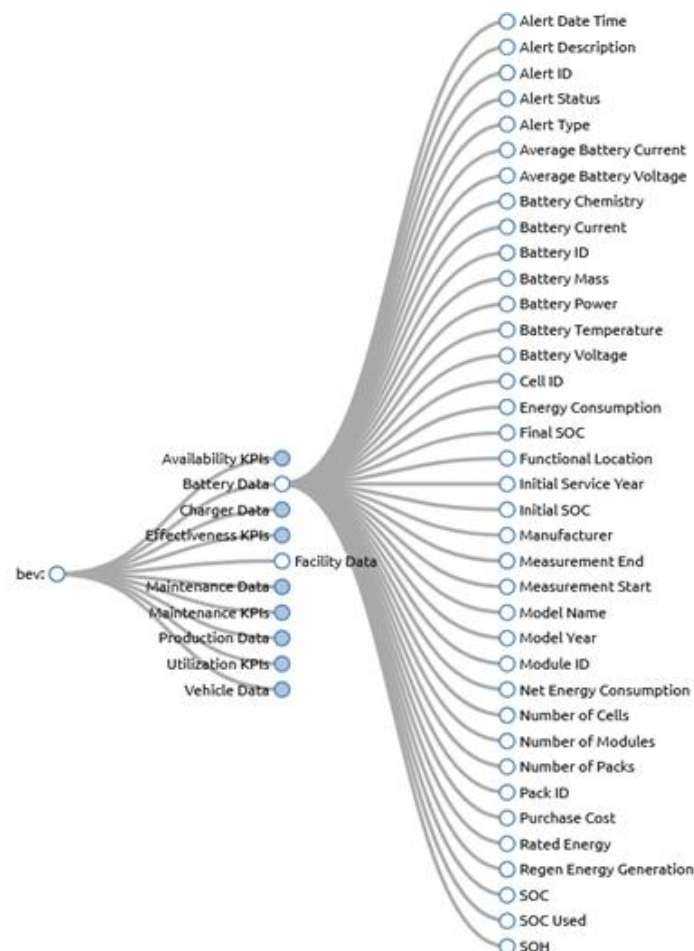


Figure 1: Categories of BEV data, with the Battery Data category shown in full.

The Need for a Vocabulary

BEV deployments throughout the mining industry are still relatively rare. Even in Canada where most of the world's BEVs have been trialled or are [operating](#), there are only a handful of cases where BEVs make up the core production fleet.

As companies introduce more BEVs into their operations, we need to ensure that vehicle/fleet analytics properly accounts for the differences between BEVs and conventional diesel equipment. There can also be challenges analysing and comparing data from mixed fleets (those containing vehicles from multiple different suppliers) due to inconsistent terminology and data definitions.

Ultimately, this vocabulary is the first step on the path to building tools and methodologies for greater BEV data interoperability.

The various types of data relating to, and generated by, BEVs that are covered by the vocabulary include:

- **Equipment characteristics:** the set of documented technical characteristics describing a piece of equipment, like what is found on a data sheet.
- **Sensor measurements:** the types of data captured by sensors on machines, such as battery voltage/current, charger power, vehicle speed/acceleration and payload.
- **Measurement metadata:** data describing how measurement data was captured, such as sensor calibration information or the date/time/location of a charging measurement.
- **Process records:** logs of production and maintenance activities captured in digital form by mine planning/scheduling software or computerized maintenance management systems.

The format for storing the data definitions is addressed next, but it should be noted that a formal information management approach was selected so that the data type definitions will be machine-readable. This enables automation of data processing tasks and the potential for future connection to other projects outside of mining, like the [Battery Data Format for Time Series Data](#).

Future work will expand on the vocabulary to define data schema for common tasks, such as BEV charging data analysis.

The SKOS Format

The format we have chosen for the vocabulary is called the Simple Knowledge Organization System (SKOS), which provides us with a standardized way of representing the data in the vocabulary. SKOS is a generalized system for classifying and describing knowledge or data.

SKOS defines the approach to structuring data, but the actual file format we use is called [Turtle](#) (using the .ttl file extension). This is just a text file with a defined structure. We can explore a simple example of the SKOS format in table form (*Figure 2*).

URI	skos:prefLabel@en	skos:definition@en	skos:broader@en
bev:vehicle_data	Vehicle Data	All data relating to the description and use of a battery electric vehicle (BEV)	
bev:battery_data	Battery Data	All data relating to the description and use of a BEV battery	
bev:vehicle_id	Vehicle ID	The ID number for the vehicle	bev:vehicle_data
bev:model_name	Model Name	Vehicle model name/model number	bev:vehicle_data, bev:battery_data, bev:charger_data
bev:manufacturer	Manufacturer	Equipment manufacturer	bev:vehicle_data, bev:battery_data, bev:charger_data
bev:energy_system_architecture	Energy System Architecture	Type of vehicle powertrain	bev:vehicle_data
bev:battery_id	Battery ID	The ID number for the battery	bev:battery_data
bev:battery_chemistry	Battery Chemistry	Battery cell chemistry	bev:battery_data
bev:rated_energy	Rated Energy	Rated energy capacity of the battery	bev:battery_data
bev:battery_mass	Battery Mass	The mass of the battery	bev:battery_data
bev:soc	SOC	Battery State of Charge (SOC) reading	bev:battery_data

Figure 2: Simple SKOS example in table format.

The first entry in the table defines one of the main categories of data, the Vehicle Data. This entity acts as the broadest definition for all the data types related to vehicles in our vocabulary. The first column is the URI or “Unique Resource Identifier” for each entity. Basically, a unique machine-readable name for referencing this entity elsewhere in our vocabulary.

The second column lets us set the preferred label for this data type, and the “@en” at the end of the property name indicates that the entry is in English. We can add another column for any other language; if we wanted to define a preferred label in French, we just need to add a column with the heading *skos:prefLabel@fr* to the spreadsheet.

Using this structure, we can define the data types and link them to these higher-level entities. In the example, each data type gets its own row, where we can define its properties and links to other entities. Looking at the “Vehicle ID” data type, we can link it to the broader concept of “Vehicle Data” by enter the URI for the “Vehicle Data” entity in the “broader” SKOS property on this row.

There are many more properties we can use to define each data type, but for the purposes of this initial example we’re done! If you want to learn more about the different ways to describe data using SKOS check out these resources: [Wikipedia](#) and [W3C Recommendation](#).

Example Use Case

Defining the data types is all well and good, but how can we put this to use? To help demonstrate the benefits and power of the vocabulary, we have put together an example based around the analysis of BEV maintenance records. The goal is to show how a standard set of definitions for the data types can help with data sharing and quality control.

In our example, our mine planner, Bob, generates a spreadsheet of the maintenance records from his company’s mine planning software for all their BEVs (Figure 3). The data covers the last 6 months of BEV operations. Bob needs to share the data with the analysts at the corporate office as part of a company-wide review of all BEV operations. (Note: the maintenance cost data used in this example is randomly generated and does not represent actual BEV fleet costs).

Sarah is the analyst at head office in charge of aggregating the data from three different mine sites. She immediately runs into the issue that all three sites have set up their maintenance systems differently, each using some custom variable names. Sarah needs to ensure she’s comparing the same data across her whole analysis. For example, she needs to know if the cost data from each site is for part costs, labour costs or total costs.

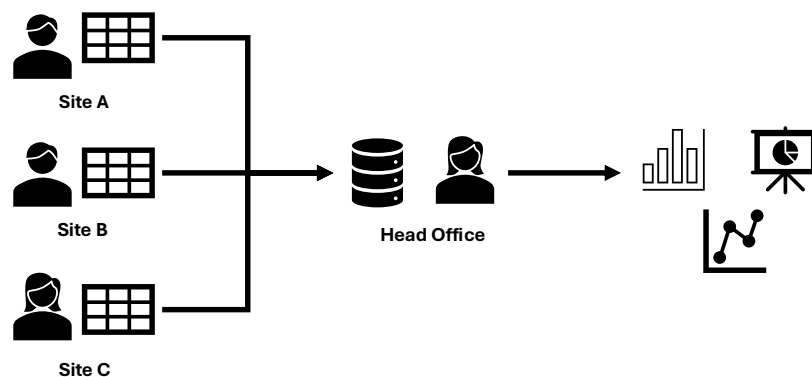


Figure 3: Aggregating and comparing data across sites.

Sarah has written a script in Python to read in the table headers (i.e. the variable names) from each Excel workbook she receives from the sites and check them against the vocabulary (Figure 4). She aims to use the standardized vocabulary to confirm if the variables have set definitions and flag any that need confirmation.

Many variables match the vocabulary, but several are marked as unknown. Variables like “Equipment ID” in one site’s data set can clearly be mapped to the “Vehicle ID” in the vocabulary. But others, like “Ordr Strt” are less clear. Sarah takes the list of unknown variables back to Bob and the other sites’ planners to confirm their definitions. To save time in the future, she records the links between any of these unknown variables and the equivalent variables that exist in the vocabulary.

For variables that are not already defined, she submits an issue to the [GitHub repository](#) containing the BEV Data Vocabulary with the details of the data types and requests they be added to continue to grow the open vocabulary.

Sarah can then combine the three site’s data and perform her analysis of the maintenance costs, splitting the work order data between electrical and mechanical work centres (see Figure 5). If you would like to see how this works in detail, we have put together an example Jupyter notebook to [demonstrate this use case](#).

Check if the data headings match the Vocabulary variables

Find the variable names in the datasets that do not appear in the Vocabulary

```
# Find the variable names in each dataset
site_A_vars = list(site_A_df.keys())
site_B_vars = list(site_B_df.keys())
site_C_vars = list(site_C_df.keys())

# Find the variable names that are missing from the open Vocabulary
site_A_missing_vars = [_ for _ in site_A_vars if (_ not in vocab_vars)]
site_B_missing_vars = [_ for _ in site_B_vars if (_ not in vocab_vars)]
site_C_missing_vars = [_ for _ in site_C_vars if (_ not in vocab_vars)]

print(f"Site A variables not included in the BEV Data Vocabulary ({len(site_A_missing_vars)}/{len(site_A_vars)}):\n")
for _ in site_A_missing_vars:
    print(f"\t{(_)}")

print(f"\n\nSite B variables not included in the BEV Data Vocabulary ({len(site_B_missing_vars)}/{len(site_B_vars)}):\n")
for _ in site_B_missing_vars:
    print(f"\t{(_)}")

print(f"\n\nSite C variables not included in the BEV Data Vocabulary ({len(site_C_missing_vars)}/{len(site_C_vars)}):\n")
for _ in site_C_missing_vars:
    print(f"\t{(_)}")
```

Site A variables not included in the BEV Data Vocabulary (8/10):

- Order
- Category
- Equipment
- Description
- Actual Costs
- Planned Costs
- Created On
- Owner

Site B variables not included in the BEV Data Vocabulary (4/10):

- Category
- Equipment ID
- Planned Costs
- Start Date

Figure 4: Example workbook where Sarah checks the site data variables against the vocabulary.

BEV Fleet Maintenance Costs

Monthly maintenance costs reported by **Electrical** and **Mechanical** Work Centers, aggregated across Sites A, B & C

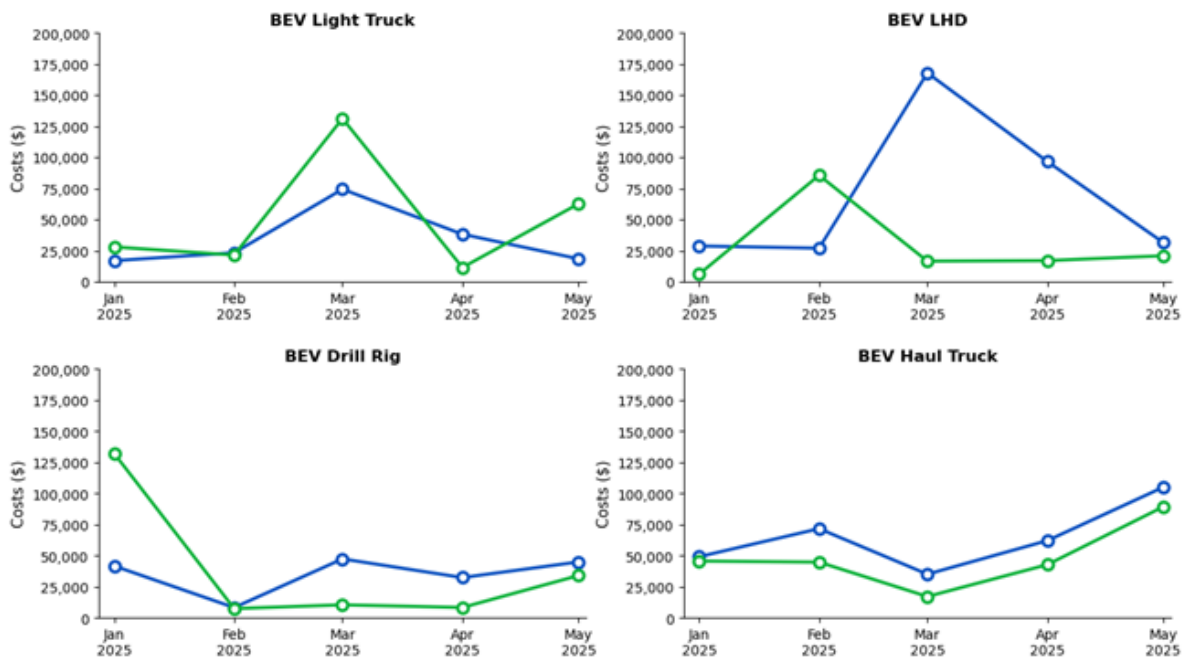


Figure 5: Results of the maintenance costs analysis.

Using the Vocabulary in Your Work

Hopefully you can see the value of using the vocabulary in your own work. One approach would be to adopt the standardized definitions into the data management practices of your team, department or organization. This could mean applying the variable names in your SAP or telemetry systems, so when you share data between teams there is no ambiguity in the definition of the variables involved. People working with your data can reference the open vocabulary and know exactly what data they are dealing with.

Or you could make an internal copy of the vocabulary and mapping allowing you to extend and control the definitions, pulling in updates from the public repository as you see fit. That way the open vocabulary contains the standard definition for each variable, and internal mappings between the open variable definitions and site-specific variable names can be kept private.

If you are comfortable with coding, you could set up some simple data verification checks using the open vocabulary for dealing with data from multiple sources. A data quality control script could confirm all the column headers in an Excel workbook match the defined data types in the vocabulary and flag any data that requires investigation.