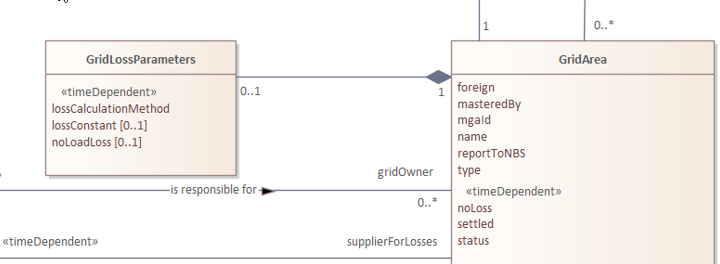
Date: 2025-03-30

# Loss Curve

1. **Elhub model**

To align CIM with Norwegian [elhub.no - Elhub](https://elhub.no/) (Smart data collector) we need to provide values and method to calculate the losses in a grid area:



**2. Modeling of Power Losses**

The Elhub model implicitly assumes a linear loss behavior, which is increasingly insufficient. As we move closer to the operational limits of the grid, losses exhibit nonlinear characteristics—especially as power increases, losses rise disproportionately (typically quadratic or even exponential in nature).

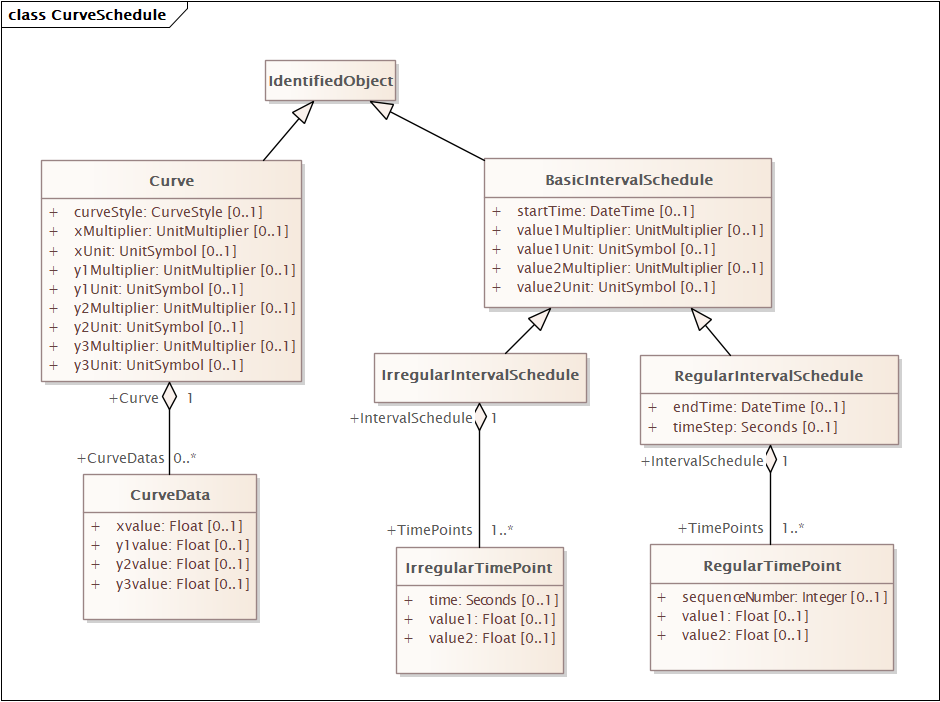
A more flexible model is needed to accurately represent this behavior. The proposed PowerLossCurve supports this by allowing more advanced interpolation methods or even explicit quadratic function modeling via a QuadraticFunction class. This ensures that the system can better reflect real-world performance and support future grid analytics and optimization.

The losses are also related to season so validity period that are reputable has been added.

**3. Generic Curve vs Specific Curve**

The current CIM model (as shown below) relies on a generic Curve structure, with fields such as xvalue, y1value, y2value, etc., whose actual meaning is defined only through documentation. While this structure offers flexibility, it also introduces significant ambiguity. Developers and integrators are left to interpret what each axis represents, based solely on conventions or external texts.

In contrast, the proposed model introduces a domain-specific PowerLossCurve with semantically clear properties like x\_P: ActivePower and y\_LossP: ActivePower. This makes the model self-explanatory and less error-prone, improving usability, interoperability, and maintainability.



**4. Validation and Semantic Richness**

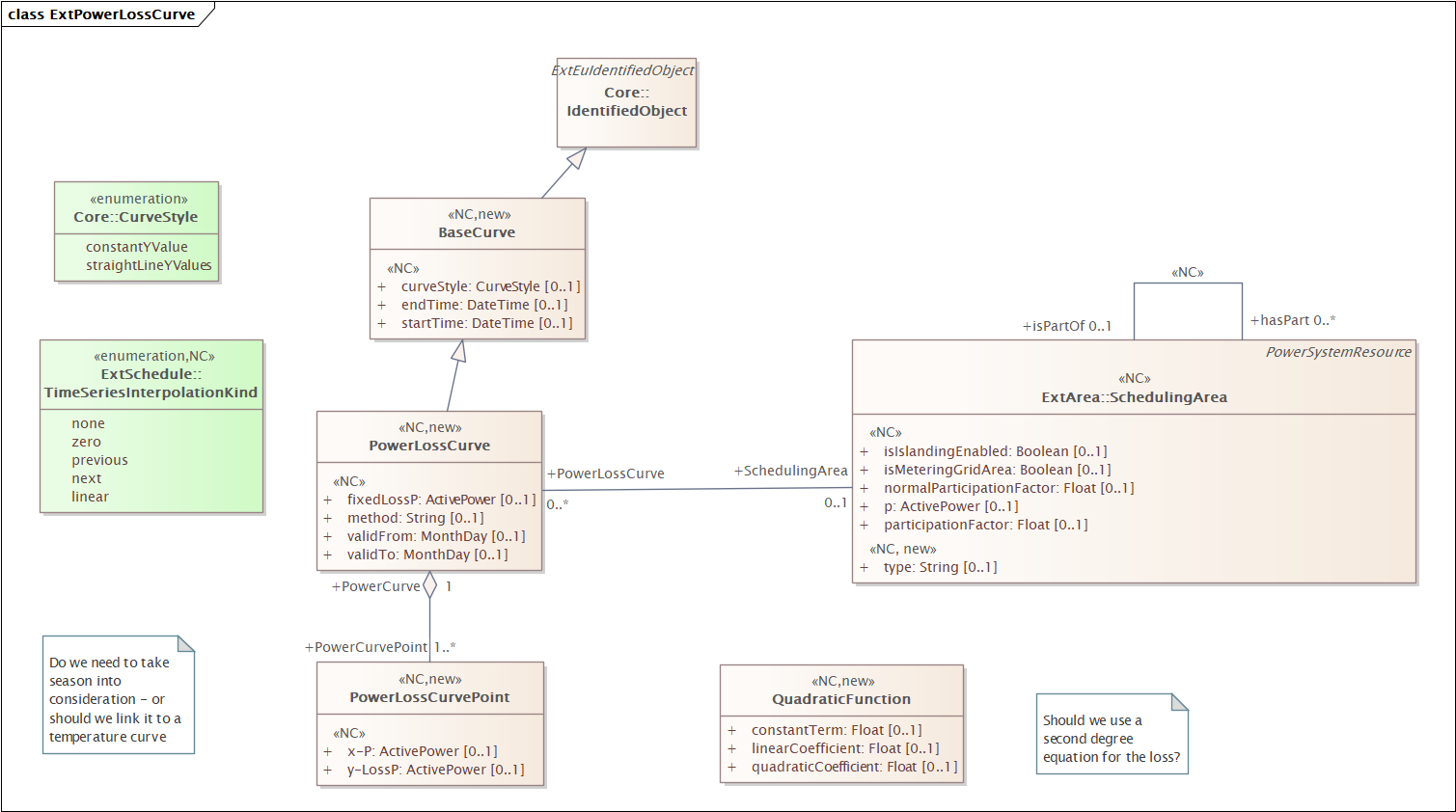
Another key drawback of the generic Curve model is that validation must happen at the instance level. Since the structure lacks explicit semantics, schema-based validation (e.g., using XSD or SHACL) cannot catch modeling errors. Instead, tools must validate each curve instance by interpreting external documentation or applying custom business logic.

By contrast, a semantically rich model like the proposed PowerLossCurve enables:

* **Schema-based validation**: Ensuring consistency and correctness can be automated through schemas that understand the domain semantics.
* **Better tooling and interoperability**: Systems can interpret and validate curves based purely on model structure without human intervention.
* **Improved machine learning and LLM support**: Large Language Models (LLMs) and machine learning algorithms benefit significantly from well-defined semantics. A clearly structured, self-describing model allows for more accurate training, better feature extraction, and explainable AI, as the intent and meaning of the data are embedded in the model itself—not hidden in documentation.

**5. PowerLossCurve WIP proposal**

A temporal proposal are as follow:



Quadratic function is: Y = ax2 + bx + c

Need to discuss:

* Shall we "deprecate" (still keep it) the current curve model and make all new curve specific?
* Shall we use CurvePoint of QuadricFunction – or give the option for both?
* Is the current curveStype sufficient?