AVEVAWORLD

OCTOBER 2024

Vattenfall Hydro Nordic

Using the SDK to create and monitor standardized data structures

Magnus Holmbom & Mattias Wångblad Inga, Vattenfall Hydro

AVEVA

VATTENFALL

ENERGY – HYDRO POWER | SWEDEN

Vattenfall standardizes AF and PI System Point structures by using SDK

Challenge

- Old AVEVA PI System structure built up over long time
- Geographically spread out with strong local focus
- Transition from focus on technicians to use by analysts

Solution

- Creation of SDK scripts to create a well templatised AF structure, PI Point naming and ensure a fleet wide comparable data quality

Results

- **Standardized AF Structure with 150 turbines and 40.000 PI Points**
- **Increased Capacity: PI System team's increased ability to support new business cases**
- **Improved Efficiency: Reduced time spent on developing and maintaining PI AF templates**
- **Improved Efficiency: Reduced implementation time of new sensors**
- **Increased Capacity: Enabled fleet wide analysis and comparisons**
- **Increased Capacity: Enabled use of CBM in the organization**

This is Vattenfall

One of Europe’s largest producers and retailers of electricity and heat

Introduction

100%
Owned by the Swedish State

8 million
Electricity customers

1.0 million
Electricity network customers

2.1 million
Heat customers

2.3 million
Gas customers

20,995
Employees

Location of our operations and major plants

Wind

Biomass

Hydro

Gas

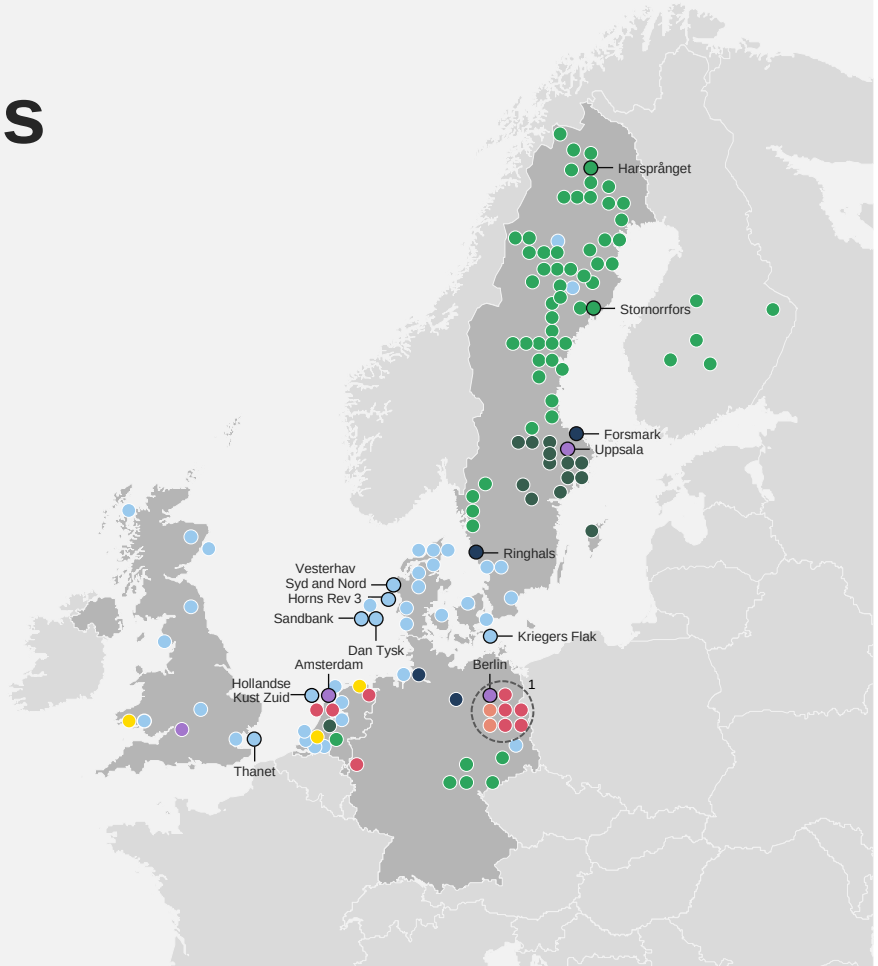
Nuclear

Coal

Solar

District heating

◯ Largest facilities marked with a circle



[1] Heat Berlin is to be divested to the State of Berlin during 2024

Our challenges with AVEVA PI System

- → Spread out over large area with isolated technicians
- → Varied fleet of powerplants built from 1910 to 1980s
- → Used PI System for over 10 years starting from small scale
- → Historically PI System has been used as monitoring tool for the technicians
- → Main focus on having a local connection between measurement value and equipment

Increasing demand for analysis

- → Production statistics/optimisation
- → Predictive maintenance
- → Weather and hydrology forecasting
- → Environmental reporting
- → Investment planning
- → Root cause analysis

AF SDK usage

- **SDK is .NET but possible to wrap in Python**
  Inspired by the package PIConnect
- **Versatile and thorough in usage**
- **Great documentation**
- **More automated use of scripts instead of manual updates**

```python
def createDrifttid(base_child: AFElement, target_parent: AFElement):
    try:
        childTemplate = AFTemplate(base_child.element.Template)
        base_att = base_child.element.Attributes.get_Item('Tekniskt id')
        base_id = base_att.GetValue().Value[-3:]
        if not target_parent.hasPumpWithId(base_id):
            name = 'tmp'
            newEl = target_parent.element.Elements.Add(name, childTemplate.element)
            att = newEl.Attributes.get_Item('Tekniskt id')
            att.ConfigString = att.ConfigString + base_id
            pi.AF.AFNameSubstitution.ResolveName(newEl)
            print(newEl.Name)
    except Exception as e:
        print(e)
        newEl.UndoCheckOut(True)
```

[3] ✓ 0.0s

```python
def createChildElements(base_parent: AFElement, target_parent: AFElement):
    for child in base_parent.getChildren():
        childTemplate = AFTemplate(child.element.Template)
        if not target_parent.hasChildWithTemplate(childTemplate):
            name = 'tmp'
            try:
                newEl = target_parent.element.Elements.Add(name, childTemplate.element)
                PI.AF.AFNameSubstitution.ResolveName(newEl)
                print(newEl.Name)
            except Exception as e:
                print(e)
                newEl.UndoCheckOut(True)
        if childTemplate.isBasedOn('_Drifttid'):
            createDrifttid(child, target_parent)
```

[4] ✓ 0.0s

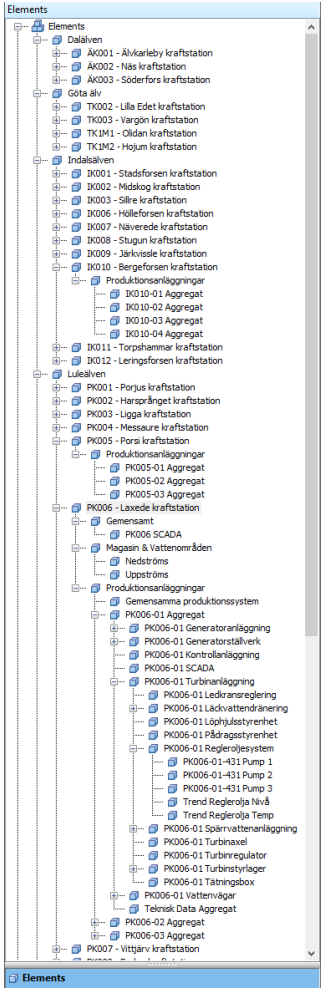
```python
def createElements(base_parent: AFElement, target_parent: AFElement):
    createChildElements(base_parent, target_parent)
    for target_child in target_parent.getChildren():
        template = AFTemplate(target_child.element.Template)
        base_child = base_parent.getChildWithTemplate(template)
        if base_child:
            createElements(base_child, target_child)
        else:
            print("No: " + target_child.element.Name + " in base")
```

[5] ✓ 0.0s

SDK

AF Structure generation

- ➔ Tedious to set up 150 turbines manually
- ➔ We have created SDK script to generate the entire template structure
- ➔ All is templatised, easy setup of PI Vision displays
- ➔ Uses a referenced template structure
- ➔ Roll out update on all turbines in case of change
- ➔ Gap analysis on missing PI Points
- ➔ Reduces risk of errors during AF tree generation



SDK usage

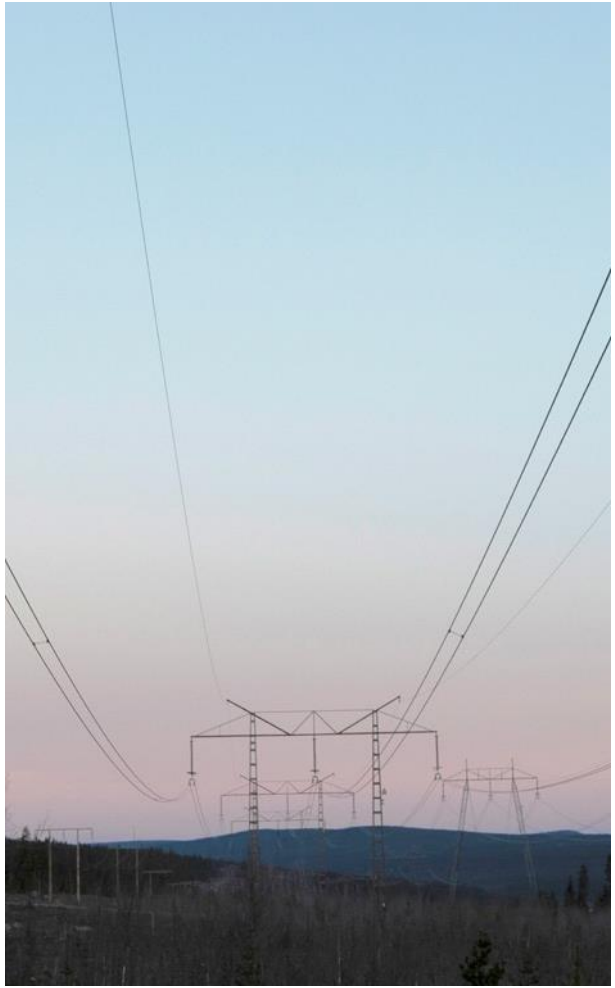
PI Point renaming

- ➔ Loops through existing PI Points and recommends new name that fits AF templates
- ➔ Mixture of language and AI models
- ➔ Generating descriptions to couple the point to the actual equipment
- ➔ Some manual work

SDK usage

Data analysis through SDK

- ➔ Quick combination/analysis of data
- ➔ Transfer data to cloud using SDK
- ➔ Healing faulty data
- ➔ Combining after sensor switch



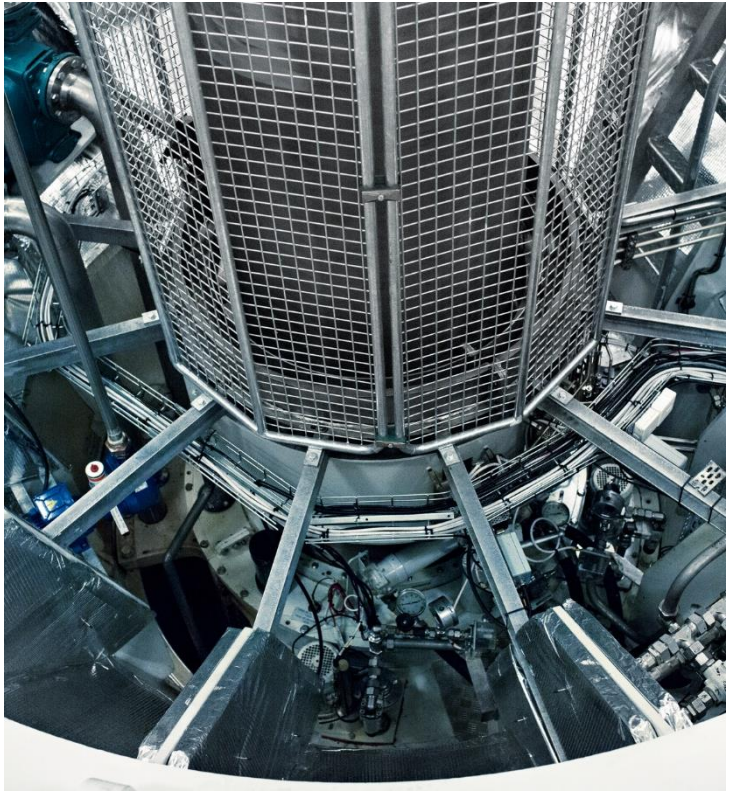
Next step

Data quality monitoring

- ➔ **Next step**
- ➔ **Compare all standardised signals and check for deviations**
- ➔ **Data rate monitoring**
- ➔ **Fluctuation deviations**

Outcome

- ➔ **Enabled Condition Based Monitoring**
- ➔ **Simplified the setup of any new analysis**
- ➔ **Show increased maintenance costs in the fleet due to new production patterns**
- ➔ **Gap analysis to identify stations which lacks important features**



Lessons learned

- **The Importance of Naming Standards**: Ensure consistency and clarity in all data management.
- **Data/AF Structure**: Essential for effective analysis deployment.
- **Analysis Needs Drive Structure**: Tailor the structure to meet specific analysis requirements.
- **Data Quality and Standardization**: The foundation for reliable and comparable results.
- **Automation and Efficiency with Python**: Maximize productivity through scripting.
- **Using PI SDK for Data Transfer**: Secure and efficient data transfer.
- **Preliminary Data Studies (Business Cases)**: Analyse and visualize data to simplify understanding before setting up the entire workflow

VATTENFALL

ENERGY – HYDRO POWER | SWEDEN

Vattenfall standardizes AF and PI System Point structures by using SDK

Challenge

- Old AVEVA PI System structure built up over long time
- Geographically spread out with strong local focus
- Transition from focus on technicians to use by analysts

Solution

- Creation of SDK scripts to create a well templatised AF structure, PI Point naming and ensure a fleet wide comparable data quality

Results

- **Standardized AF Structure with 150 turbines and 40.000 PI Points**
- **Increased Capacity: PI System team's increased ability to support new business cases**
- **Improved Efficiency: Reduced time spent on developing and maintaining PI AF templates**
- **Improved Efficiency: Reduced implementation time of new sensors**
- **Increased Capacity: Enabled fleet wide analysis and comparisons**
- **Increased Capacity: Enabled use of CBM in the organization**

This presentation may include predictions, estimates, intentions, beliefs and other statements that are or may be construed as being forward-looking. While these forward-looking statements represent our current judgment on what the future holds, they are subject to risks and uncertainties that could result in actual outcomes differing materially from those projected in these statements. No statement contained herein constitutes a commitment by AVEVA to perform any particular action or to deliver any particular product or product features. Readers are cautioned not to place undue reliance on these forward-looking statements, which reflect our opinions only as of the date of this presentation.

The Company shall not be obliged to disclose any revision to these forward-looking statements to reflect events or circumstances occurring after the date on which they are made or to reflect the occurrence of future events.

linkedin.com/company/aveva

@avevagroup

ABOUT AVEVA

AVEVA is a world leader in industrial software, providing engineering and operational solutions across multiple industries, including oil and gas, chemical, pharmaceutical, power and utilities, marine, renewables, and food and beverage. Our agnostic and open architecture helps organizations design, build, operate, maintain and optimize the complete lifecycle of complex industrial assets, from production plants and offshore platforms to manufactured consumer goods.

Over 20,000 enterprises in over 100 countries rely on AVEVA to help them deliver life's essentials: safe and reliable energy, food, medicines, infrastructure and more. By connecting people with trusted information and AI-enriched insights, AVEVA enables teams to engineer efficiently and optimize operations, driving growth and sustainability.

Named as one of the world's most innovative companies, AVEVA supports customers with open solutions and the expertise of more than 6,400 employees, 5,000 partners and 5,700 certified developers. The company is headquartered in Cambridge, UK.

Learn more at www.aveva.com

AVEVA