



OSIsoft, Sharepoint, VBA Harmony to Advance Operational Troubleshooting

Beatriz Blanco

Remote Monitoring Center and Development
Manager



Agenda

- MHPS Overview
- Challenge
- Organization Development
- Implementation
- Further Harmonization within the RMC
- Future Growth and Plans
- Questions and Takeaways

About Mitsubishi Hitachi Power Systems

- MHPS is an industry leader in power generation technology including Advanced Class Gas Turbine.
- The MHPS Remote Monitoring Center (RMC) is **using the PI System** to monitor its customers' **power generation assets around the world**.
- Connectivity is tailored for the specific requirements of the customer. The RMC monitors OEM and non-OEM equipment across turbine classes.
- MHPS is currently expanding to Total Plant Monitoring and improving process efficiency through automation is a strong focus.



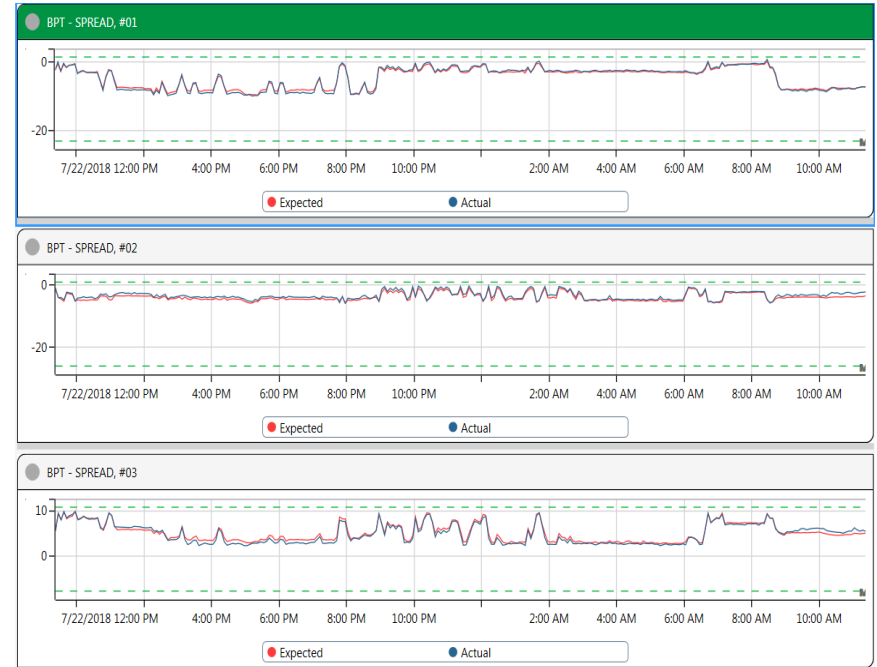
Operations Overview

- 24-Hour OEM Support
- Real-Time Monitoring and Analysis with OSIsoft PI
- Lessons-Learned Fleet Database
- Informative Operational Reports
- Alarm Management
- Web-Based Customer Portal for Remote Data Access



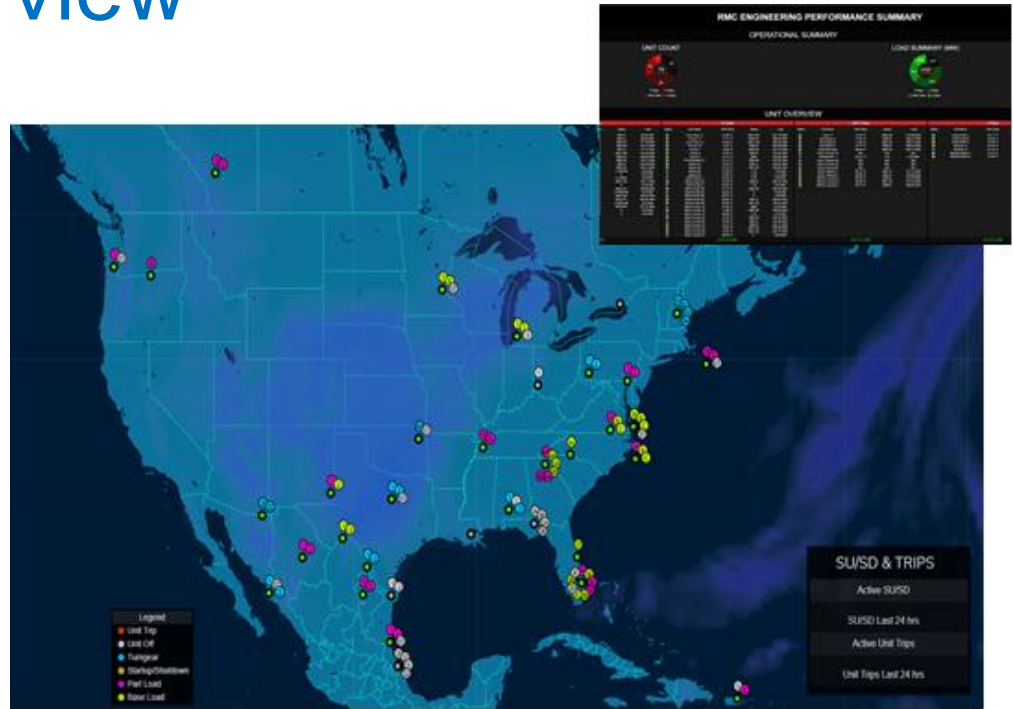
Engineering Overview

- 24-Hour OEM Engineering Support
- Predictive Analysis through Advanced Pattern Recognition software linked to OSIsoft PI
- Performance & Reliability Optimization
- Controls, Combustion Dynamics, Vibration, and Operational Engineering Support

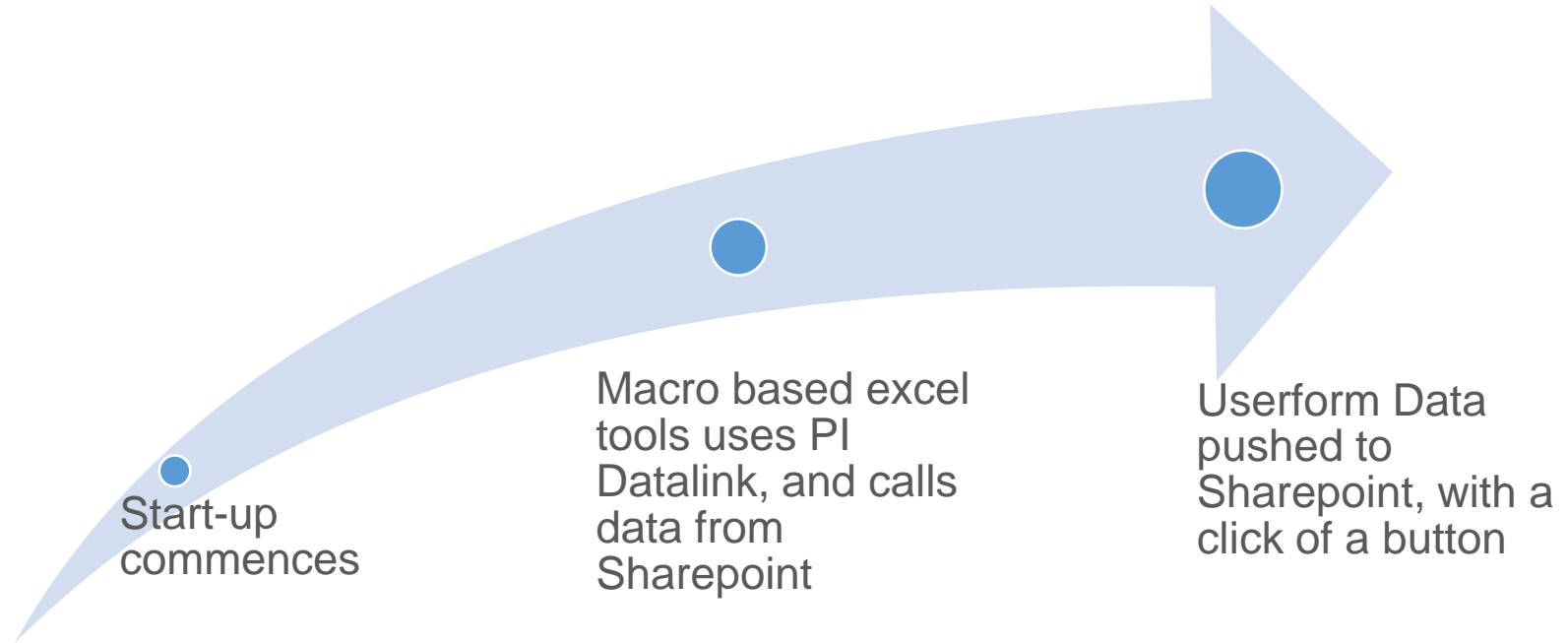


Development Overview

- Total Plant Monitoring Initiative
- Dashboard Development
- Quarterly Report Development
- Custom Coded Tools
- Software Evaluation
- AssetFramework Buildout

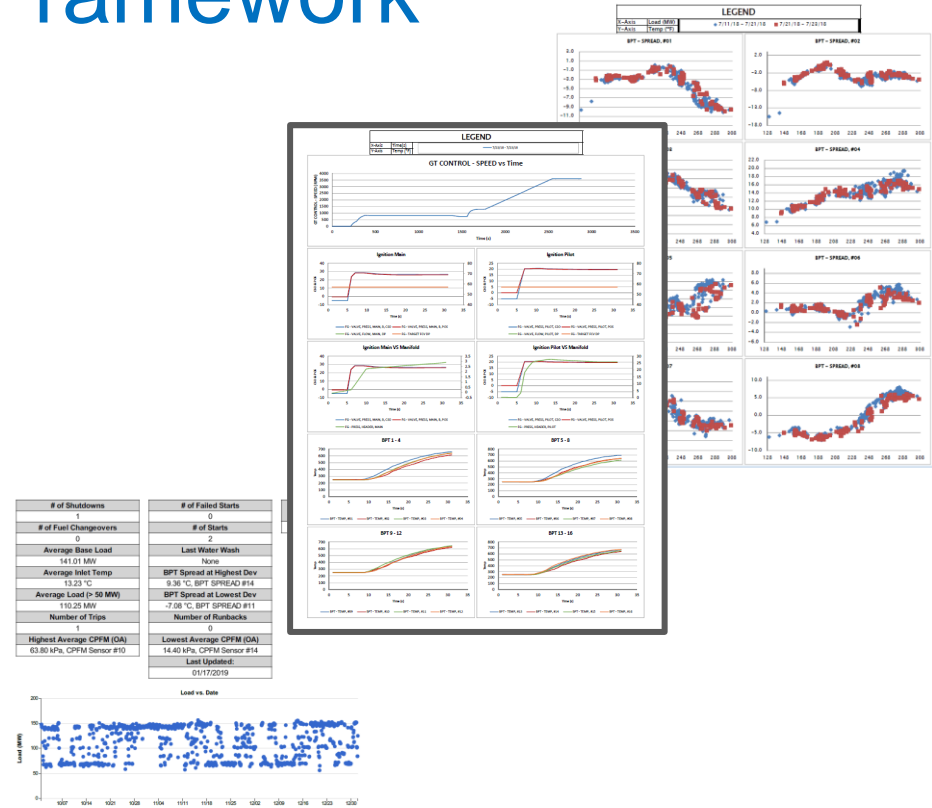


Automating Journal Entries



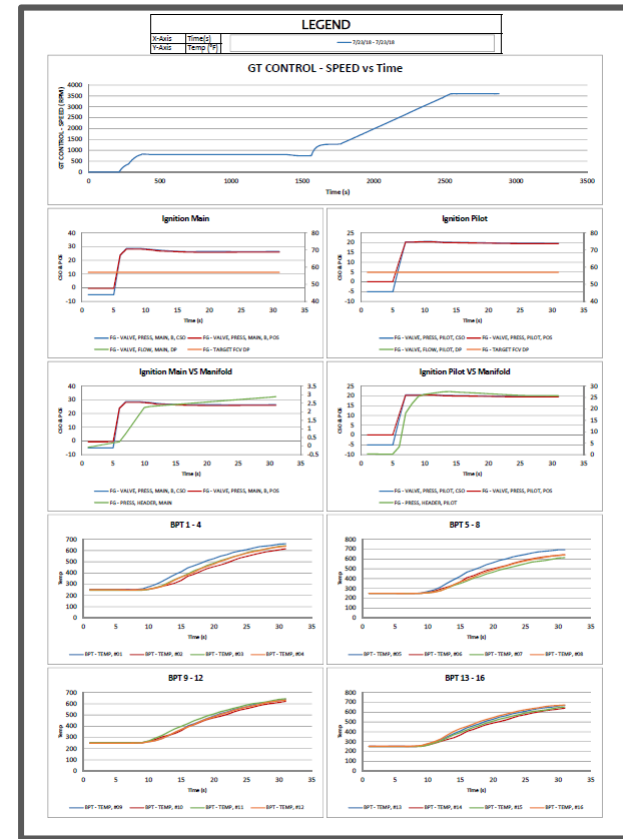
Conversion to AssetFramework

- With the addition of AssetFramework and the subsequent utilization with the RMC of that structure, the majority of our macro based tools were modified to use our AF Fleet structure.
- Custom tools that are built for RMC troubleshooting as well and tools that are created for other MHPS based groups reference this structure.

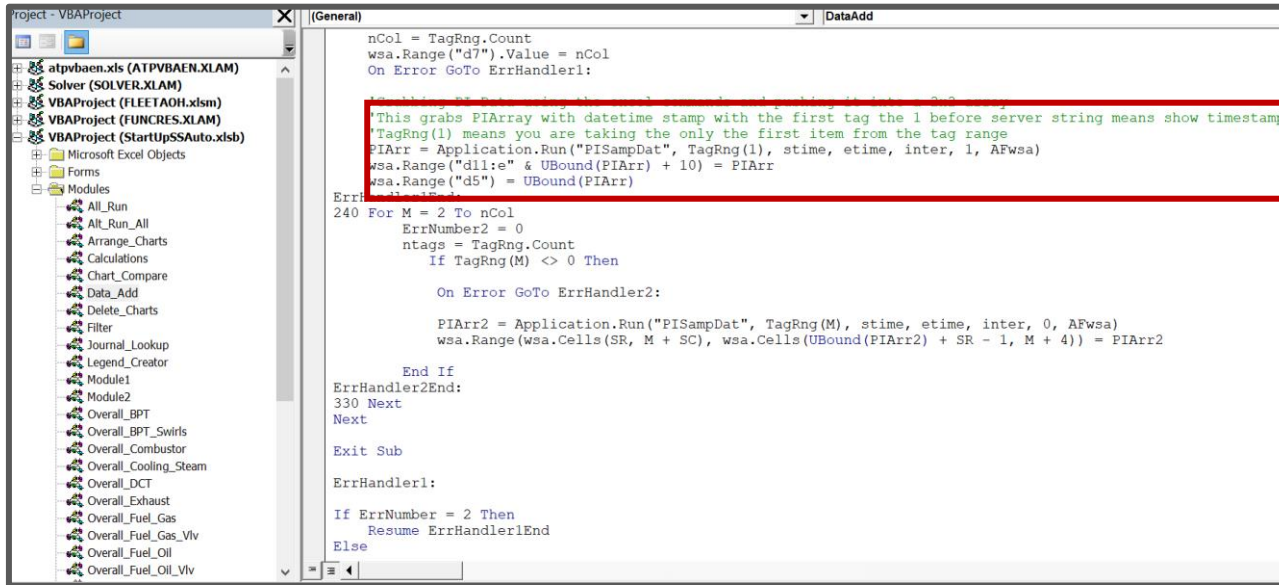


Pulling PI Data

- The startup comparison is an in house custom coded tool that uses Sharepoint information, and PI Datalink to review the profiles of start-ups.
- The RMC uses the tool to watch for degradation of components and to assist in the troubleshooting of failed starts.
- A portion of this tool automates the journal entries created by the operators.

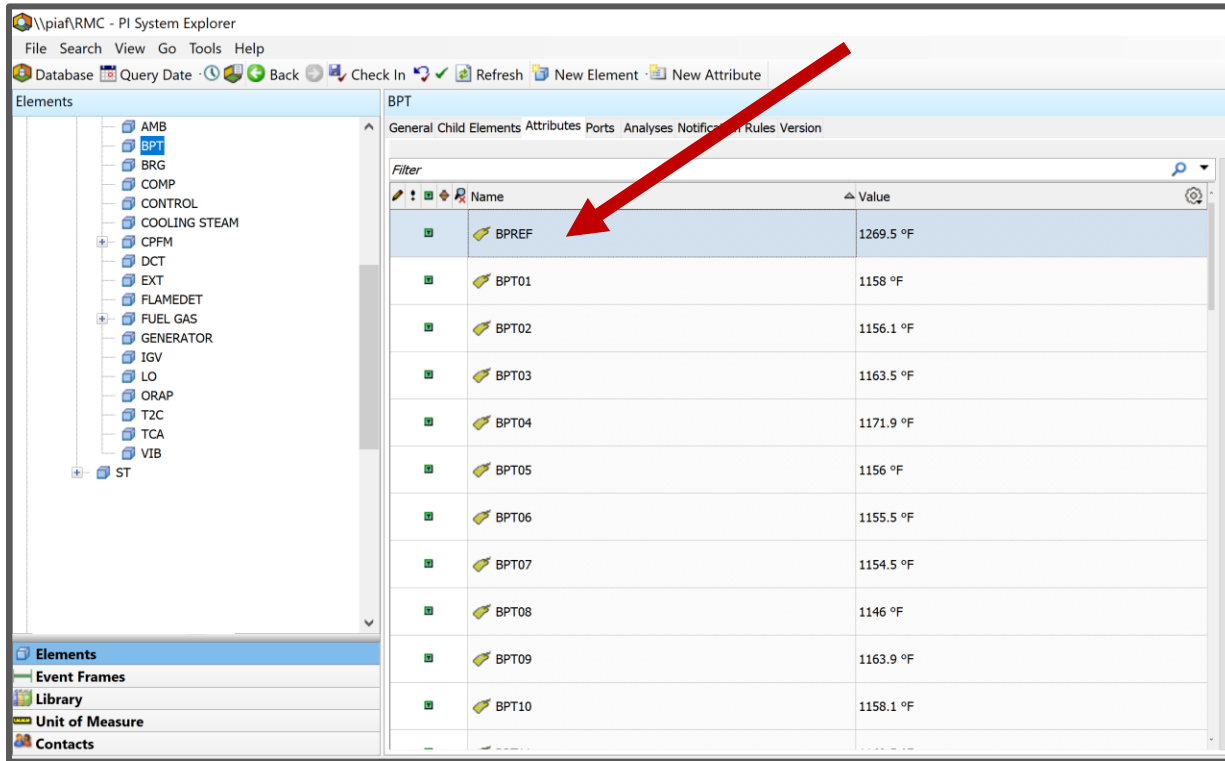


Pulling PI Data



- Automatically pulls data using PI Datalink and references Asset Framework attributes so that it can be used across the fleet.

Referencing Asset Framework

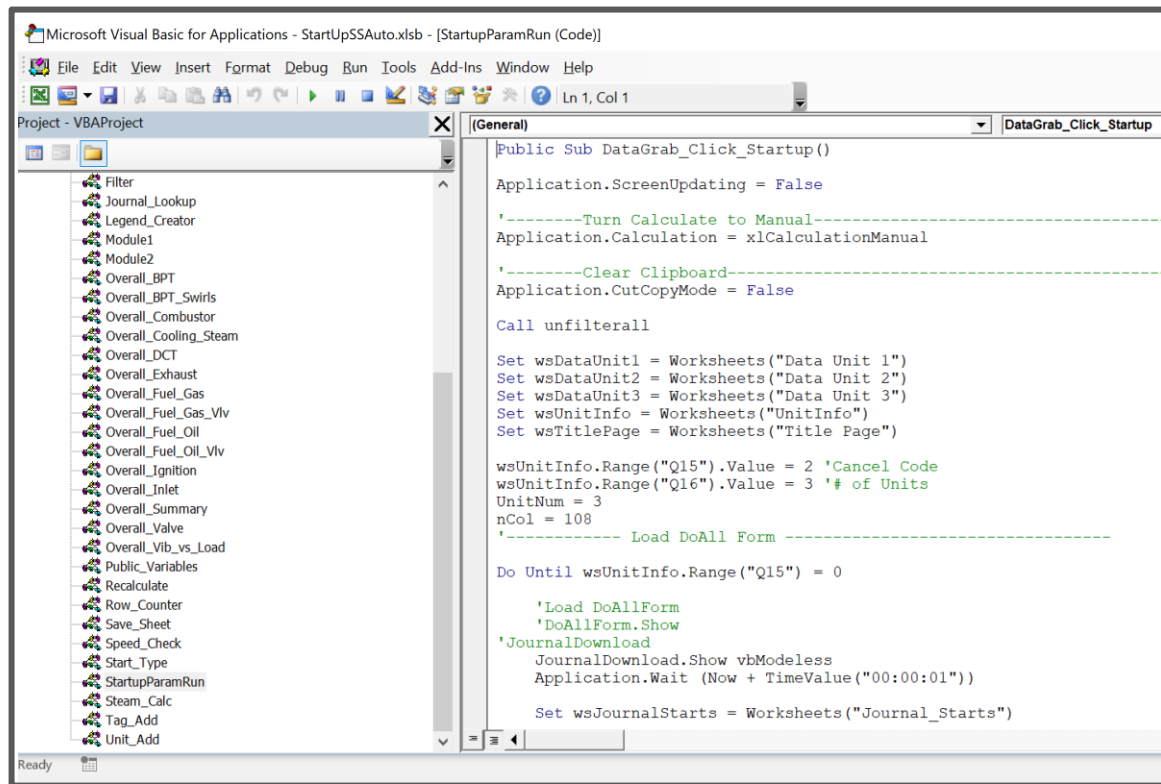


The screenshot shows the OSIsoft PI System Explorer interface. On the left, a tree view lists various elements including AMB, BPT, BRG, COMP, CONTROL, COOLING STEAM, CPFM, DCT, EXT, FLAMEDET, FUEL GAS, GENERATOR, IGV, LO, ORAP, T2C, TCA, VIB, and ST. The BPT element is selected. The main pane displays the 'General' tab for the BPT element, showing a table of child elements. A red arrow points to the 'BPREF' row in the table.

Name	Value
BPREF	1269.5 °F
BPT01	1158 °F
BPT02	1156.1 °F
BPT03	1163.5 °F
BPT04	1171.9 °F
BPT05	1156 °F
BPT06	1155.5 °F
BPT07	1154.5 °F
BPT08	1146 °F
BPT09	1163.9 °F
BPT10	1158.1 °F

- The tool uses the attribute name and path so that the only input the operators needs to make is the selection of the unit in which they are doing the evaluation.

Transferring PI Data to Userform



- The parameters that are required for the startup entry are read and placed into the userform automatically.

Userform

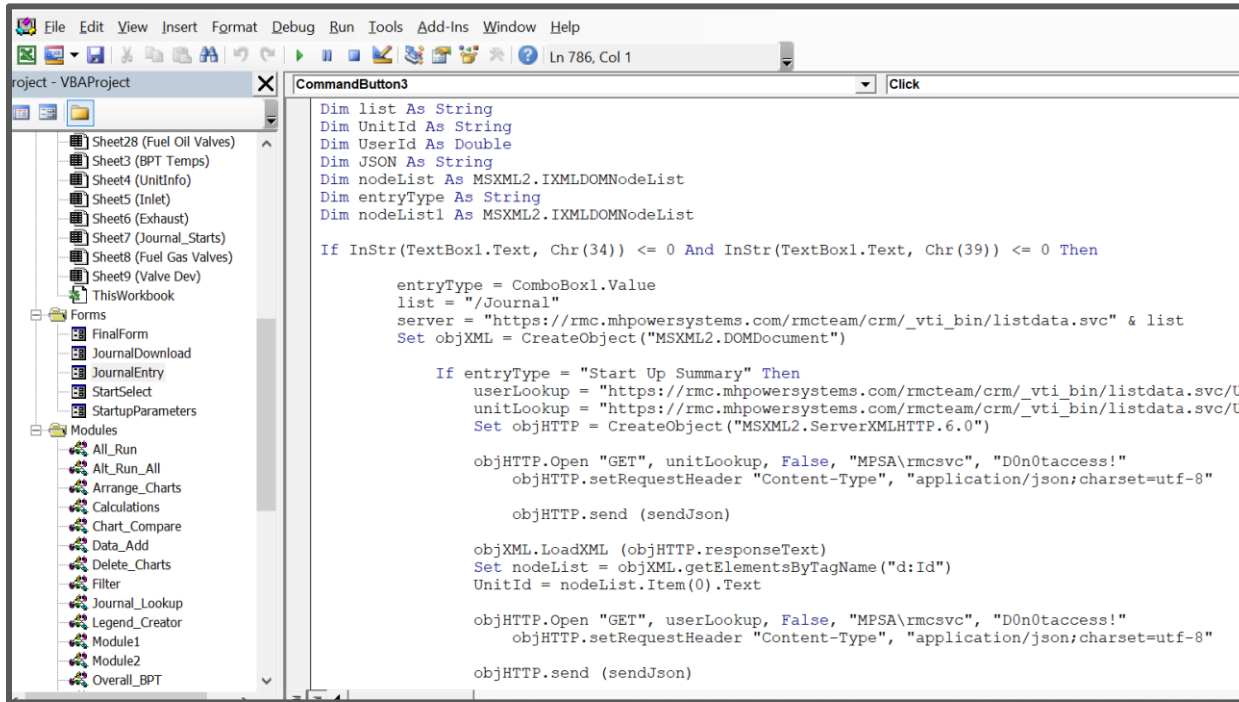
- The final product: A userform that automatically pulls the relevant PI Data via a function that uses PI Datalink, communicates with Sharepoint to pull the related unit issues, and allows for notes to be entered
- With the push of a button publishes all aggregated data to Sharepoint

The screenshot shows a software window titled "Startup Parameters" with a close button (X) in the top right corner. The window is divided into several sections:

- Entry Fields:** A series of input fields on the left, each with a label and a value:
 - Entry Type: Start Up Summary (dropdown)
 - Entry_Date: 12/3/2018
 - Entry_Time: 12:06
 - Work Shift: Day
 - Operator: OPERATOR NAME
 - Unit: UNIT
 - Fuel Type: Fuel Gas
 - Unit Roll_Date: 12/3/2018
 - Unit Roll_Time: 11:24
 - Flame On_Date: 12/3/2018
 - Flame On_Time: 11:46
 - FSNL_Date: 12/3/2018
 - FSNL_Time: 12:03
 - Breaker Closed_Date: 12/3/2018
 - Breaker Closed_Time: 12:06
- Start Up Conditions:** A text area on the right containing:
 - Ambient = 74.0°F
 - T1C = 79.2°F
 - Peak BPT Ave = 1062.0°F @ 1879.4 RPM
 - T1C @ Flame ON = 76.9°F
 - FG Temp @ Flame ON = 115.8°F
 - FG Supply Press @ Flame ON = 565.59
- Disc Cavity Temperature:** A text area on the right containing:
 - DC2 = 462.0°F
 - DC3 = 461.3°F
 - DC4 = 448.7°F
- Highest Vibrations:** A text area on the right containing:
 - 1st Critical = BR2Y: @ 3.152mils @ 1175 RPM
 - 2nd Critical = BR1X: @ 4.493mils @ 2403 RPM
 - Rated Speed = BR2X: @ 4.42mils @ 3603 RPM
- BPT's at FSNL:** A text area on the right containing:
 - Actual Low = BPT #2 @ 582.32°F
 - Actual High = BPT #13 @ 694.73°F
 - Spread Low = BPT #2 @ -33.99°F
 - Spread High = BPT #5 @ 21.17°F
- Notes:** A text area labeled "Operation Notes Entered Here".
- Issue Tracker Status:** A row of radio buttons: Created, Updated, Referenced (selected), No Issue # Needed.
- Issue List:** A table with columns "IR #" and "Description".

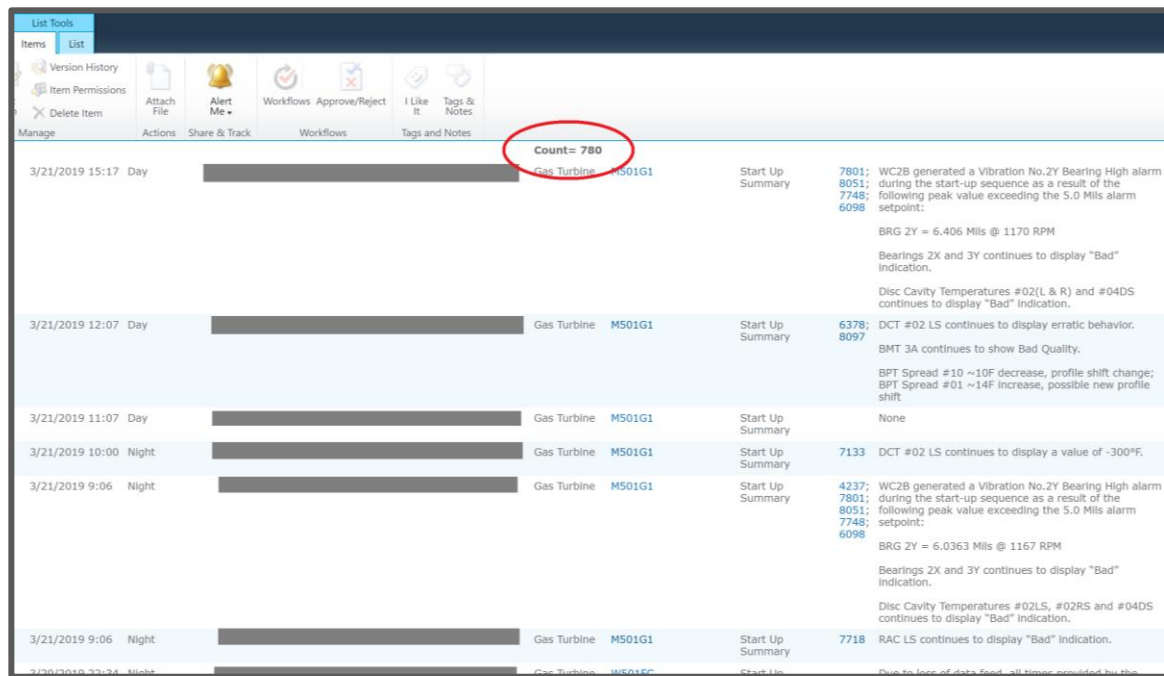
IR #	Description
7707	BPT #4 Erratic Behavior
7593	DCT #2 RS Erratic /Bad Quality Indication
7146	BPT Spread #05 Alarm HL. FO Operation
5380	(A-GPPM) Pressure Sensor Abnormal Low Sensitivity #12
5354	CPFM PS #4 ABN LOW SENSITIVITY
5156	Main-A PCV Deviation Shutdown/Startup House Load
4983	BMT 1A, 1B, 2A, 2B erratic activity, bad input, and trending low.
- Buttons:** At the bottom are three buttons: "Startup SS", "Journal Entry", and "Cancel".

Pushing data back to Sharepoint



- JSON is used to transmit the structured data to the Sharepoint site.

Improving Journal Entry Efficiency

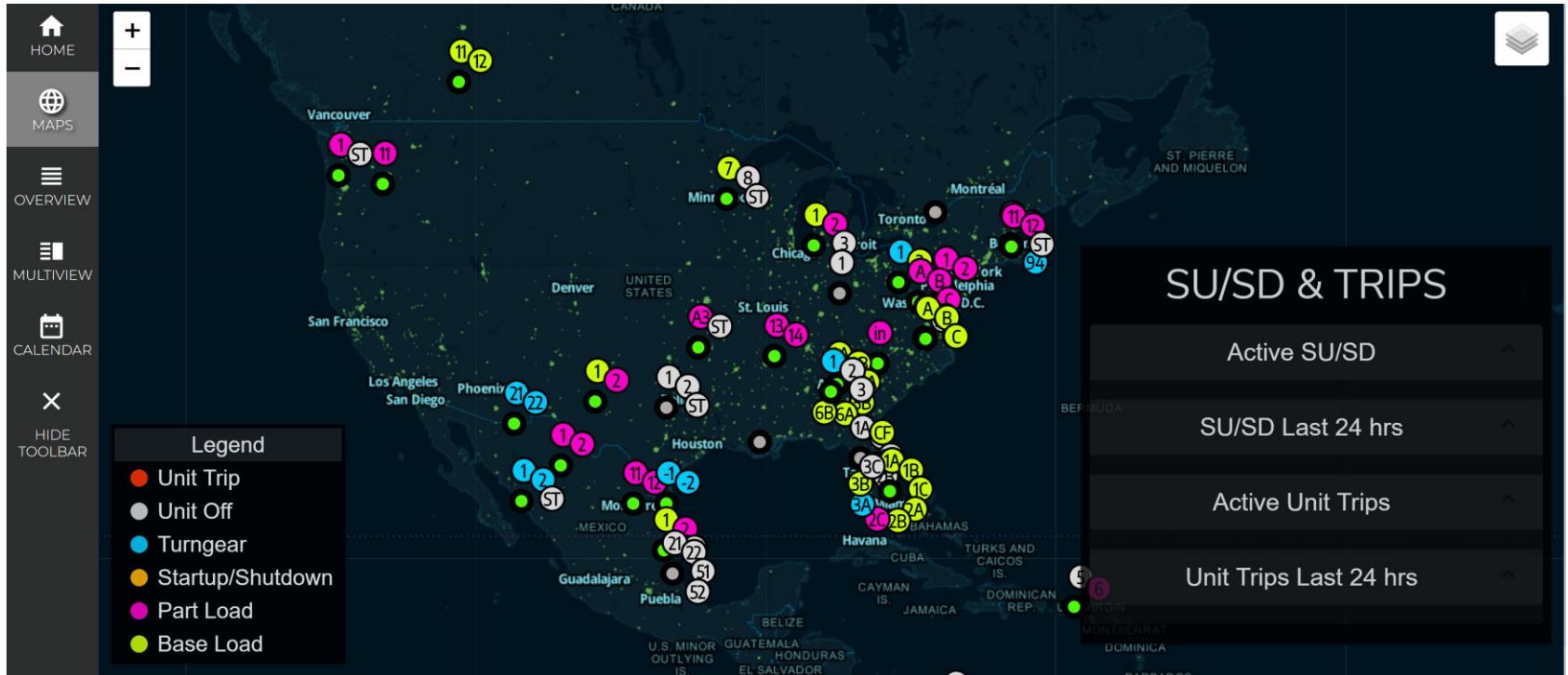


3/21/2019 15:17	Day	Gas Turbine	M501G1	Start Up Summary	7801: WC2B generated a Vibration No.2Y Bearing High alarm during the start-up sequence as a result of the following peak value exceeding the 5.0 Mils alarm setpoint: 8051: 7748: 6098: BRG 2Y = 6.406 Mils @ 1170 RPM Bearings 2X and 3Y continues to display "Bad" indication. Disc Cavity Temperatures #02(L & R) and #04DS continues to display "Bad" indication.
3/21/2019 12:07	Day	Gas Turbine	M501G1	Start Up Summary	6378: DCT #02 LS continues to display erratic behavior. 8097: BMT 3A continues to show Bad Quality. BPT Spread #10 ~10F decrease, profile shift change; BPT Spread #01 ~14F increase, possible new profile shift
3/21/2019 11:07	Day	Gas Turbine	M501G1	Start Up Summary	None
3/21/2019 10:00	Night	Gas Turbine	M501G1	Start Up Summary	7133 DCT #02 LS continues to display a value of -300°F.
3/21/2019 9:06	Night	Gas Turbine	M501G1	Start Up Summary	4237: WC2B generated a Vibration No.2Y Bearing High alarm during the start-up sequence as a result of the following peak value exceeding the 5.0 Mils alarm setpoint: 7801: 8051: 7748: 6098: BRG 2Y = 6.0363 Mils @ 1167 RPM Bearings 2X and 3Y continues to display "Bad" indication. Disc Cavity Temperatures #02LS, #02RS and #04DS continues to display "Bad" indication.
3/21/2019 9:06	Night	Gas Turbine	M501G1	Start Up Summary	7718 RAC LS continues to display "Bad" indication.
3/20/2019 23:34	Night	Gas Turbine	M501EC	Start Up	Due to loss of data feed, all times provided by the

- Year to date we have close to 800 starts across the fleet
- Time of manual entries for start summaries drastically dropped, and continued focus on automation is key for our ever expanding fleet

Other Integration of OSIsoft PI Data

Dashboard Under Development



PIAF connection with Python

```
import sys
sys.path.append('C:\\Program Files
(x86)\\PIPC\\AF\\PublicAssemblies\\4.0\\')
import clr
clr.AddReference('OSIsoft.AFSDK')
from OSIsoft import AF
```

```
system_name = "piaf"
target_database_name = "RMC"
plants_element_name = "Plants"

MySystem = AF.PISystems().get_Item(system_name)
databases = MySystem.Databases
target_database = databases.get_Item(target_database_name)
plant_elements =
target_database.get_Elements().get_Item(plants_element_name).get_
Elements()
```

- The connection to PIAF with python is handled by the OSIsoft AF SDK
- The python package needed is the 'clr' package in order to add the reference to AF SDK this will allow you to import the OSIsoft AF dll as a python package inheriting all of the AF Database functionalities

Getting Information from an Element

- Once an element is accessed you can get the name of the element by using the `get_Name()` function

```
self._plant_name = self._plant_element.get_Name()
```

- To access all attributes in an element you can use the `get_Attributes()` function

```
self._plant_element.get_Attributes()
```

- To get the value of an attribute use the `GetValue()` function [this returns a AF Value object] then this object can use its own `get_Value()` function to return the raw value of the attribute

```
for unit_element in self._plant_element.get_Elements():  
    unit_data_object = UnitDataObject(parent_data_object=self,  
    unit_element=unit_element,  
                                     unit_database=self.unit_database)  
    units_dict[unit_data_object.unit_name] = unit_data_object
```

Getting Information from an Element



- Any values within a PIAF database can be accessed with the functionality noted in the previous slides.
- Once you are connected to the database, the values are all basically broken down into elements and attributes where you just need to find the relationship of child elements of a parent object until you route to the attribute that contains the value you are looking for.

Total Plant Initiative

Operations

Alarm Management

Performance

KPIs

Controls

Assessments

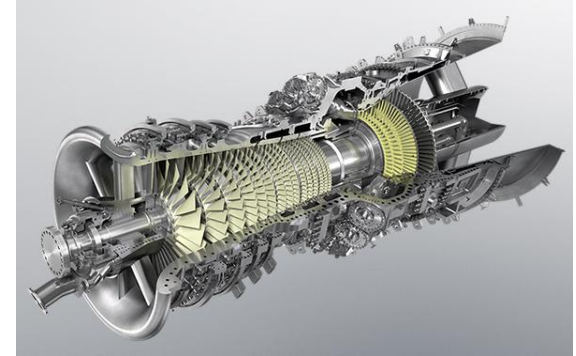
Predictive Models

DIGITAL DOMAIN



Mitsubishi Hitachi Power Systems

Leveraging the PI System to Advance Troubleshooting Techniques and Process Efficiencies



CHALLENGE

Adapting to the demand of digital transformation and the optimization of the use of the data, and the exponential growth of the RMC

- Evolving Customers
- Expanding expectations to total plant and non-OEM equipment
- Expansion of monitored fleet


SOLUTION

Use standardization of Asset Framework, and communication, to development tools to for process improvement

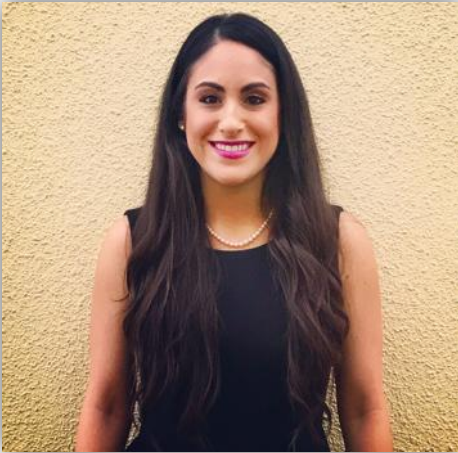
- Automation of startup process evaluation
- Automation of entry to journal
- Future Dashboard currently being used and in development

RESULTS

Reduction of journal entry time, reduction of time for evaluation of startups, improved root cause analysis

 **MHPS** is currently expanding capabilities to offer Total Plant Monitoring and developing in house applications for troubleshooting and performance evaluation

Contact Information



Beatriz Blanco

Beatriz.Blanco@amer.mhps.com

RMC Operations and Development
Engineering Manager

Mitsubishi Hitachi Power Systems America, Inc.

Questions?

Please wait for
the **microphone**

State your
name & company



Please remember

TO DOWNLOAD
APP, SEARCH
OSISOFT



Download on the
App Store



GET IT ON
Google Play



