



Utilizing PI Programming Tools to Support Predictive Model Applications

By: B. Fletcher Ashcraft
Access Computer Solutions



Project Goals

- Utilize technological advances in predictive modeling to optimize product quality.
- Distribute process performance information in real-time to expedite decision making.
- Stabilize the process output by the reduction of initially perceived testing variability.
- Optimize process performance by the reduction of off-standard production.
- Leverage the power of the PI system to automate and execute the system.

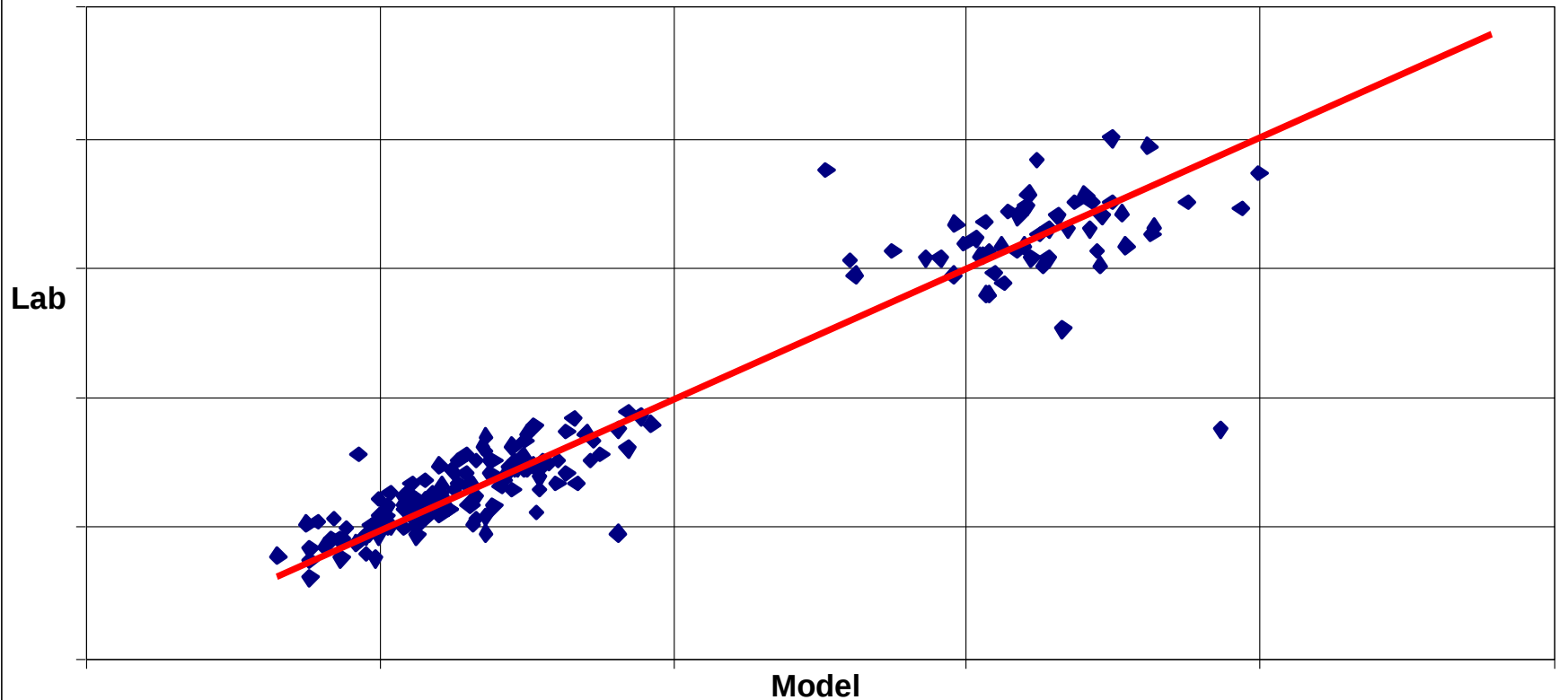
Model Development

- Artificial Neural Network (ANN) software used
 1. Non-linear Process.
 2. Variables are highly interacting.
 3. Abundance of accessible historical data.
 4. Seamless integration with the PI system.
- Models built with Previous 12 months historical data.
 1. Over 70 PI tags per model initially
 2. Pared down to under 20 per model
- Calibration adjustment check system developed.
 1. Run twice per shift
- Batches for paper reels and sets configured.
 1. Ability to track concurrent batches
 2. Programmatically determine start and end times

Model Development

Test Calibration
Jan-03

$$y = 0.9961x$$
$$R^2 = 0.9148$$





Software Development

- **ActiveX technology chosen**
 1. **Portability to a wide variety of applications.**
 2. **Easily converted to web-based delivery.**
 3. **Highly scalable and configurable.**
- **ProcessBook was chosen for the container**
 1. **Designed for real-time data delivery.**
 2. **Additional process information can be displayed**
 3. **No additional licensing required.**
- **Two ActiveX controls were written**
 1. **Paper Tester Data entry**
 2. **Dynamic Batch Grid**

Software Development

Data Entry Control

Reel #	11	Spec Code	L560B	Cull Limit	28.4	Calculate	Submit	Clear
New Timestamp:	22-Apr-03 13:38							
PI Tag: Descriptor:	Front	Front Center	Back Center	Back	Averages			
New Values:	34	33	30	28	31.3			
Model Avgs: Set 4	27.5	28.5	28.7	27.9	28.2			
Difference:	6.5	4.5	1.3	0.1	3.1			

All data included herein is purely fictitious and used as sample data only.

Software Development

Dynamic Batch Grid Control

AutoUpdate Enabled								
Reel & Set	Start	End	Grade	Front	F Center	B Center	Back	Average
PM1_13_1	2:14:32 PM	2:29:00 PM	5600	31.4	32.3	31.6	34.0	32.3
PM1_12_5	2:14:02 PM	2:14:31 PM	5600	31.4	32.3	31.6	34.0	32.3
PM1_12_4	2:05:02 PM	2:14:01 PM	5600	31.0	32.0	31.3	33.5	31.9
PM1_12_3	1:56:02 PM	2:05:01 PM	5600	29.2	30.2	30.0	30.6	29.9
PM1_12_2	1:47:02 PM	1:56:01 PM	5600	27.8	28.8	29.0	28.2	28.5
PM1_12_1	1:38:02 PM	1:47:01 PM	5600	27.4	28.5	28.6	27.8	28.2
PM1_11_4	1:29:02 PM	1:38:01 PM	5600	27.5	28.5	28.7	27.9	28.2
PM1_11_3	1:20:32 PM	1:29:01 PM	5600	27.6	28.7	28.8	28.1	28.4
PM1_11_2	1:11:32 PM	1:20:31 PM	5600	28.1	29.1	29.3	28.6	28.8
PM1_11_1	1:02:32 PM	1:11:31 PM	5600	28.5	29.6	29.7	29.0	29.3

All data included herein is purely fictitious and used as sample data only.

Software Development

Dynamic Batch Grid Control

■ Configurable Properties

1. PI server
2. Batch module and unit
3. Up to five PI Tags
4. Machine positions
5. Timer
6. Significant digits
7. Path to initialization file
8. Column alignment
9. Gridline style
10. All colors

Properties

BatchGridSDK1 BatchGridSDK

Alphabetic | Categorized

(Custom)	
(Name)	BatchGridSDK1
AlarmColor	&H000000FF&
BackColor	&H00C0C0C0&
Backup_Duration	0
Backup_Time	
Batch_Module	
Batch_Unit	
Caution_Limit	0
CautionColor	&H0000FFFF&
Col_Alignment	1 - alCenter
Enabled	True
ForeColor	&H00000000&
GradeTagname	
Grid_Caption	
GridColor	&H00000000&
GridLines	2 - grInset
Height	721
Hi_Limit	100
Left	-14987
Lo_Limit	0
No_of_CalcPoints	1
No_of_Values	10
PI_Server	
PITagname0	
PITagname0asType	5 - astAverage
PITagname1	
PITagname1asType	5 - astAverage
PITagname2	
PITagname2asType	5 - astAverage
PITagname3	
PITagname3asType	5 - astAverage
PITagname4	
PITagname4asType	5 - astAverage
Positionals	False
RowColor	&H00FF80FF&
Sec_Update	0
Selected	True
Sig_Digits	1
Spec_File_Path	



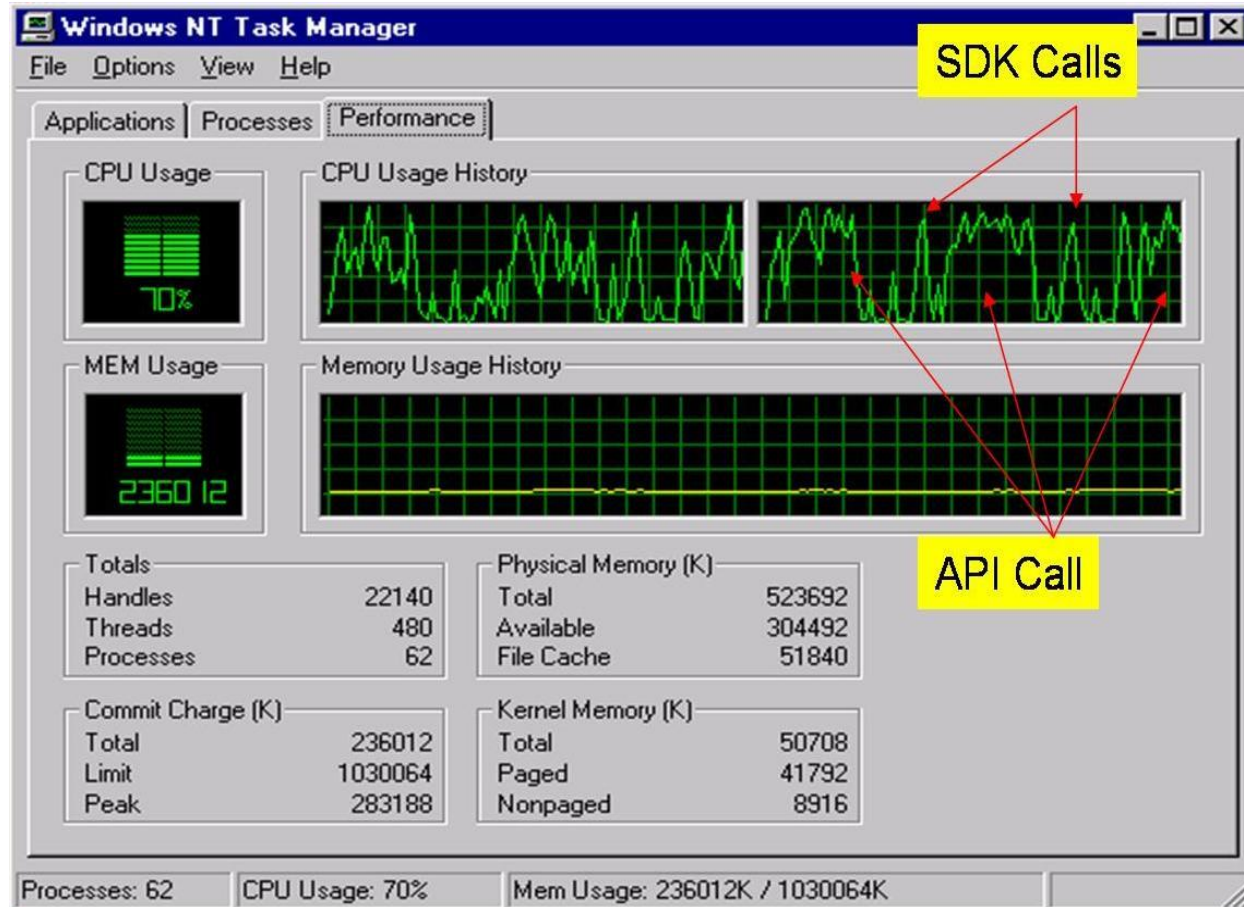
Software Development

■ Learning Experiences

1. Although API calls from the controls while the server is being backed up will cause the control to crash, There is a method in the SDK to check for the server being backed up.
2. Since the batch subsystem is not buffered, it is imperative that the PI server have enough resources to record the batch events.
3. Batch search calls cause a significant strain on processor load

Software Development

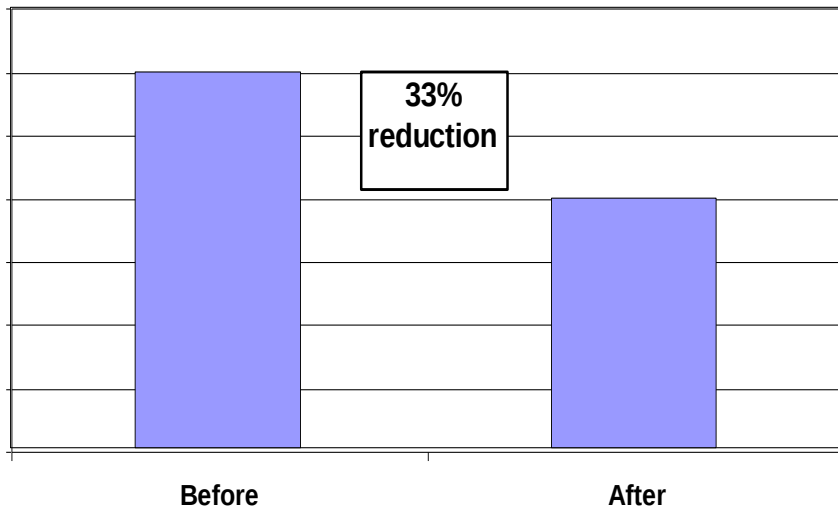
Processor Load



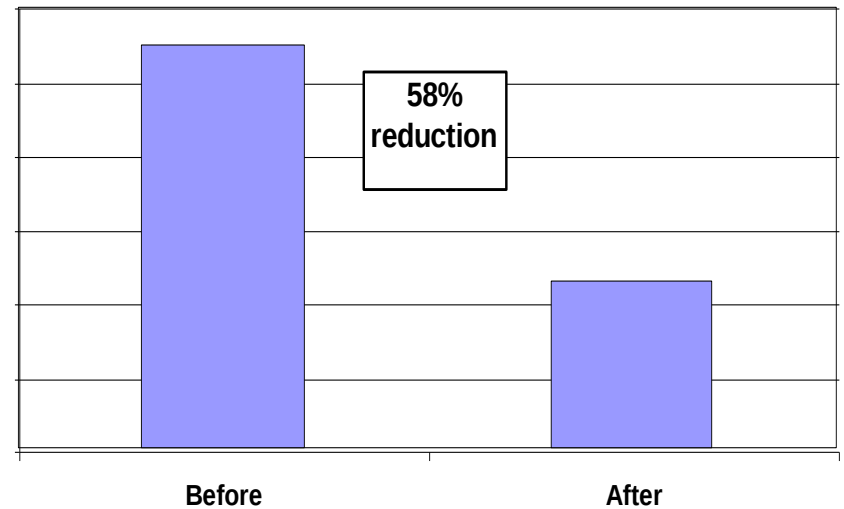
Software Development

System Results

Liner
Off-Standard Reduction



Medium
Off-Standard Reduction





Conclusion

- **The PI system and client applications provided the functionality and versatility to be able to achieve our objectives.**
- **It is imperative to implement a procedure to ensure the model stays in calibration.**
- **It is important to continue periodic testing to provide a feedback to the model for adjusting for long term drifts and subtle changes in the process that were not represented in the original model.**
- **Special attention should be given to make sure that the PI server is operating within it's limitations and that all data is backed up in a timely fashion.**
- **A process model can act as a filter to remove “noise” and allow the machine crew to see the real process variation.**