PI Application Framework Example

Applying the Application Framework

Topics

- Description of the example (Downtime)
- Why the Application Framework
- How the example was developed using Application Framework
- Advantages of applying the Application Framework
- Demonstrate the example

Application Framework Example - Downtime

- The example requirements:
  - Monitor process events and conditions
  - Based on some rules indicate if there has been a violation
  - Assign a process event code for the violation by equipment
  - Enter comments and reason codes by equipment
  - Support the process model structure and changes made to the structure without rewriting the application

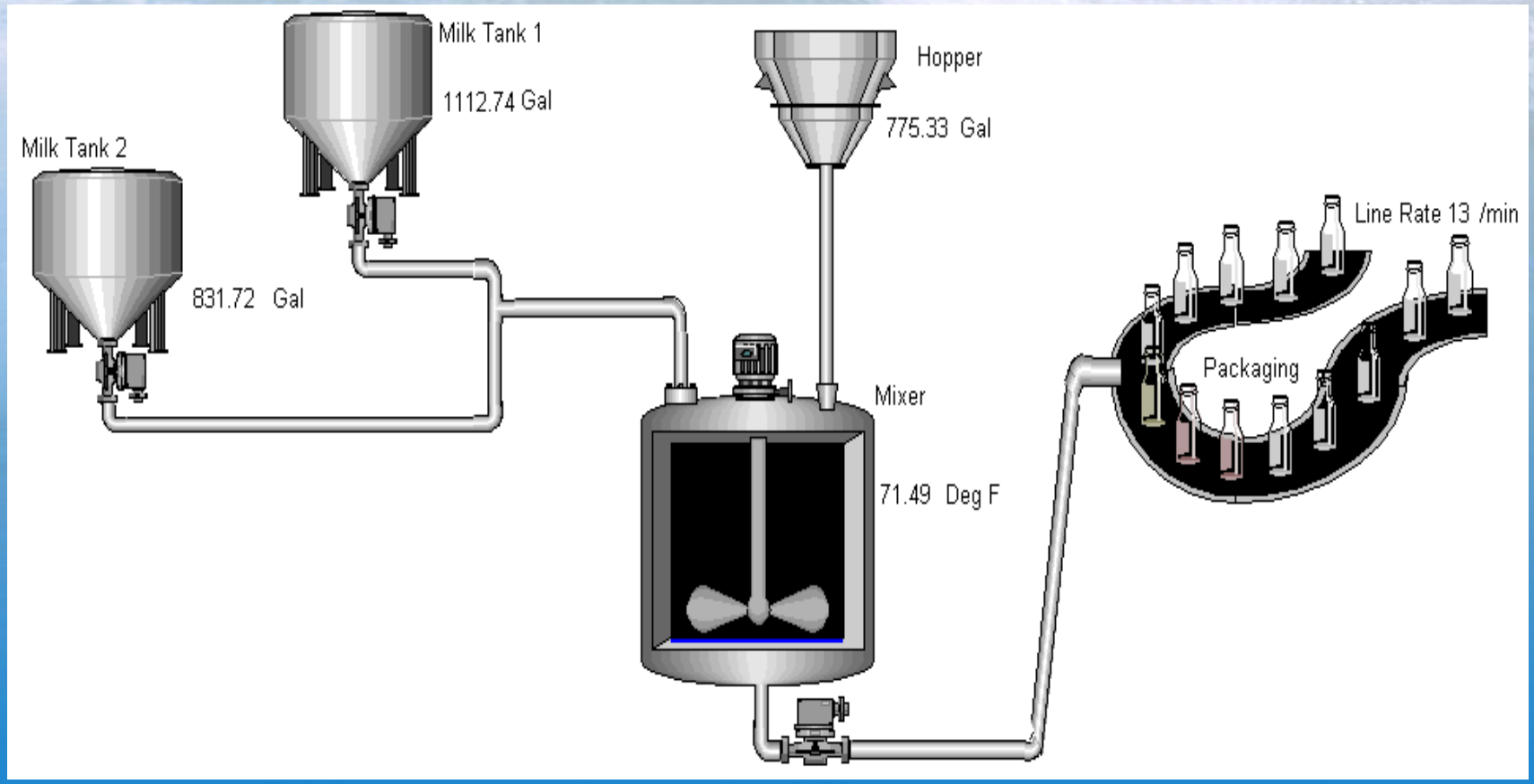
Why the Application Framework

- The Application Framework provides a structured environment for application development:
  - Provides support for reusable applications
  - Supports model structures
  - Provides the development for applications that are easily maintainable
  - Integrates with ProcessBook

Applying the Application Framework

- The following steps were applied in developing the example
  - Define the model
  - Define the element templates
    - Define the attributes for the type of element (volume for a tank)
    - Define categories for grouping and querying capability
  - Define the elements
    - Define the data references (PI tag for volume)
  - Define the analysis plug-in

Application Process



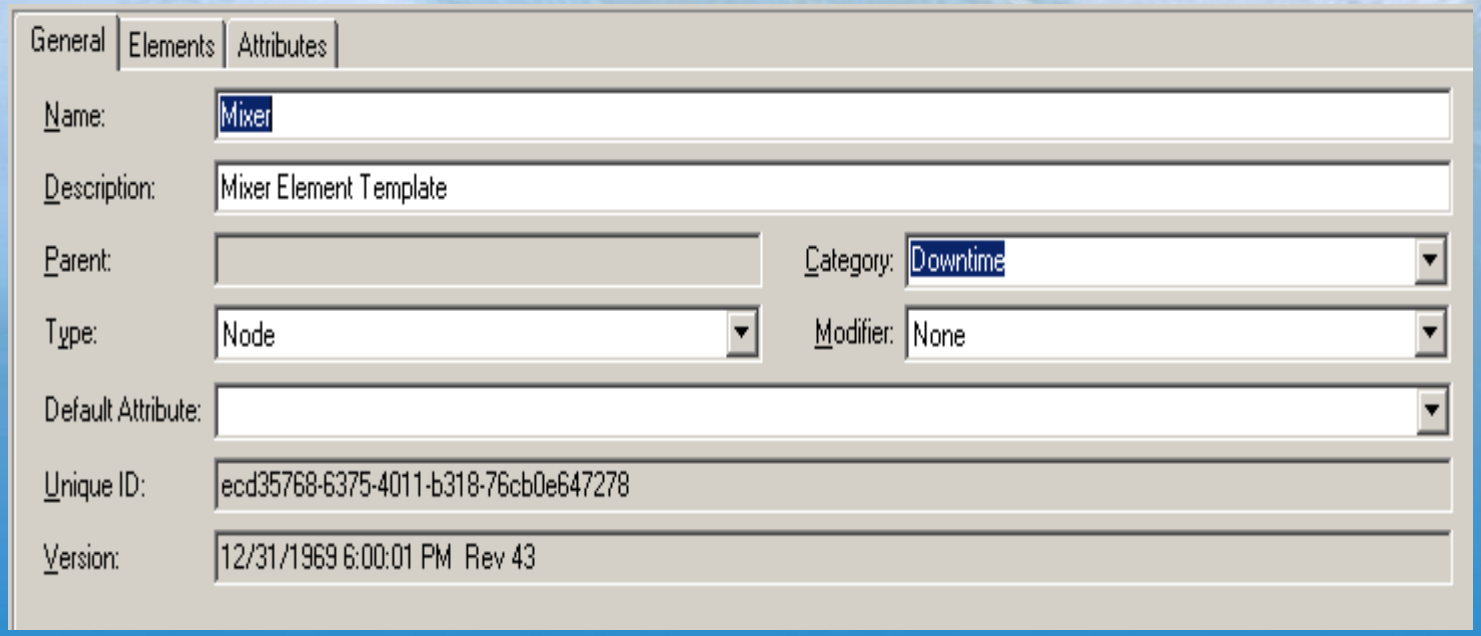
Defining Element Templates

- From the process overview the element templates were defined and created based on the equipment type
  - The element templates are used to define the equipment (tank, mixer)
  - The element templates include attributes
    - PI input tags
    - PI output tags
    - Process limit values
    - Other data sources
  - Supports categories
    - Adds structure and defines grouping

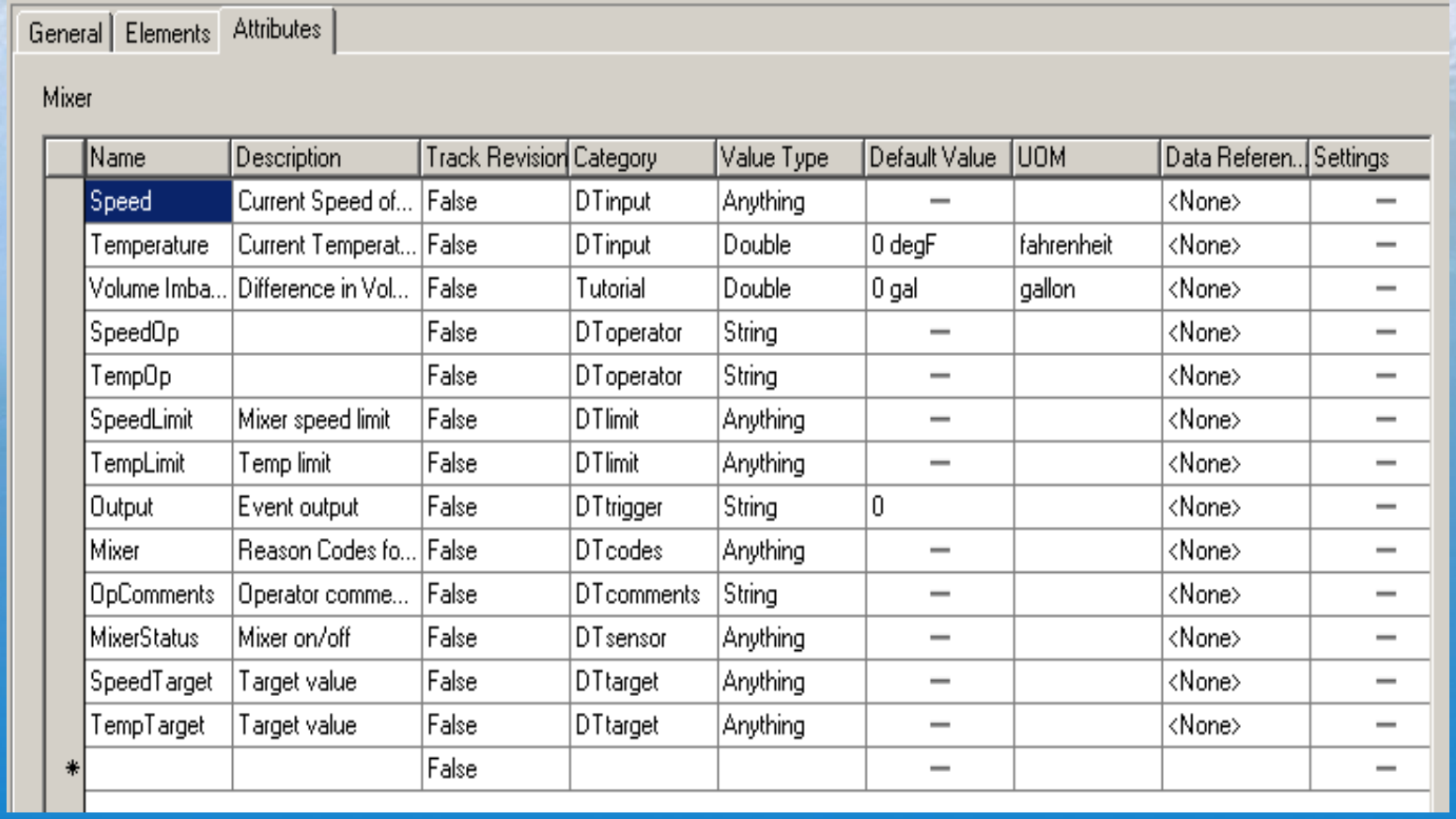
Element Templates

Name	Description	Category	Version	Type	Type Modifier
CartonBottomSealer	Carton Bottom Sealer Element Template		12/31/1969 ...	Node	None
CartonFiller	Cartom Filler Element Template		12/31/1969 ...	Node	None
CartonFolder	Carton Folder Element Template		12/31/1969 ...	Node	None
CartonTopSealer	Carton Top Sealer Element Template		12/31/1969 ...	Node	None
ChocalatePump	Chocolate Pump Element Template		12/31/1969 ...	Node	None
ChocolatePump	Chocolate Pump Element Template		12/31/1969 ...	Node	None
ChocoMilkMixer	Choco Milk Mixer Element Template		12/31/1969 ...	Node	None
ChocoSlurryMixer	Choco Slurry Mixer Element Template		12/31/1969 ...	Node	None
Flow	Flow Element Template	Tutorial	12/31/1969 ...	Flow	None
Hopper	Element Template for a Hopper	Downtime	12/31/1969 ...	Node	Storage
Meter	Meter Element Template	Tutorial	12/31/1969 ...	Measurement	None
MilkPump	Milk Pump Element Template		12/31/1969 ...	Node	None
Mixer	Mixer Element Template	Downtime	12/31/1969 ...	Node	None
ModelDetails	ModelElement Element Template	Tutorial	12/31/1969 ...	Model	None
Node	Node Element Template	Tutorial	12/31/1969 ...	Node	None
PackagingUnit	Packaging Unit Element Template	Downtime	12/31/1969 ...	Node	None
ReceiptTruck	Receipt Truck Element Template	Tutorial	12/31/1969 ...	Node	Reciept Point
ShipTruck	Shipment Truck Element Template	Tutorial	12/31/1969 ...	Node	Ship Point
Tank	Tank Element Template	Downtime	12/31/1969 ...	Node	Storage
Transfer	Transfer Element Template	Tutorial	12/31/1969 ...	Transfer	None

Element Template



Element Template Attributes



General | Elements | Attributes

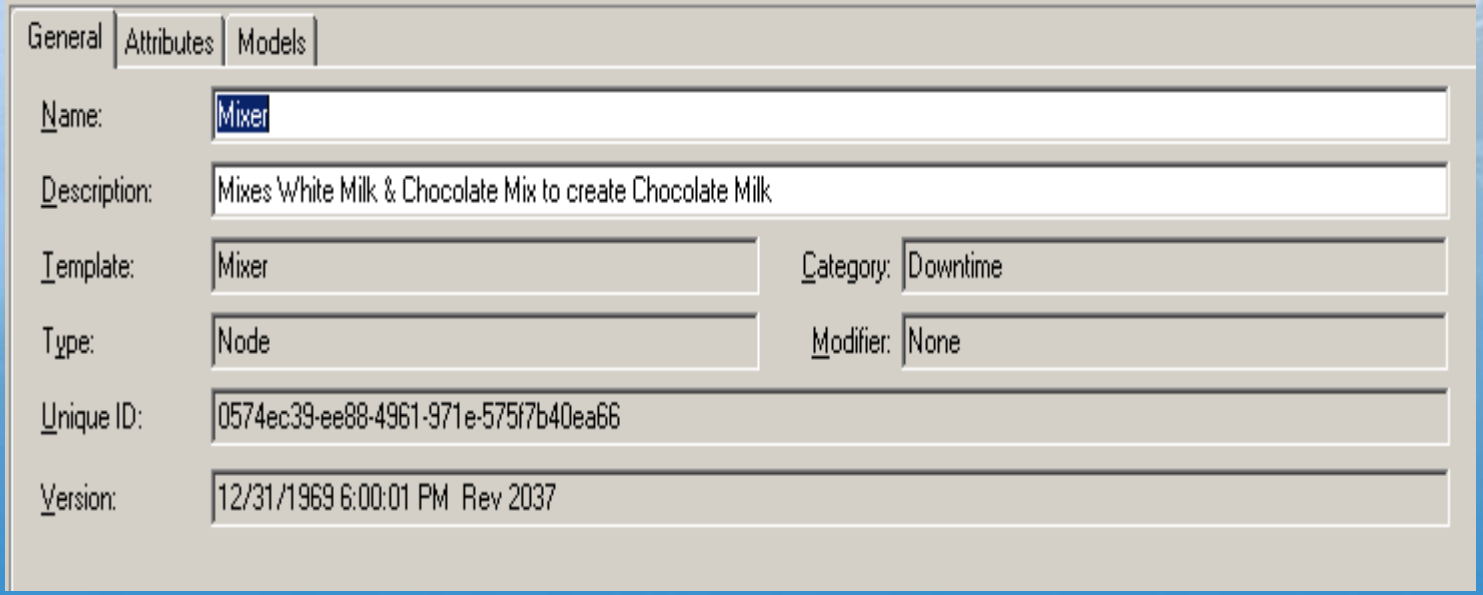
Mixer

	Name	Description	Track Revision	Category	Value Type	Default Value	UOM	Data Referen...	Settings
	Speed	Current Speed of...	False	DTinput	Anything	–		<None>	–
	Temperature	Current Temperat...	False	DTinput	Double	0 degF	fahrenheit	<None>	–
	Volume Imba...	Difference in Vol...	False	Tutorial	Double	0 gal	gallon	<None>	–
	SpeedOp		False	DToperator	String	–		<None>	–
	TempOp		False	DToperator	String	–		<None>	–
	SpeedLimit	Mixer speed limit	False	DTlimit	Anything	–		<None>	–
	TempLimit	Temp limit	False	DTlimit	Anything	–		<None>	–
	Output	Event output	False	DTtrigger	String	0		<None>	–
	Mixer	Reason Codes fo...	False	DTcodes	Anything	–		<None>	–
	OpComments	Operator comme...	False	DTcomments	String	–		<None>	–
	MixerStatus	Mixer on/off	False	DTsensor	Anything	–		<None>	–
	SpeedTarget	Target value	False	DTtarget	Anything	–		<None>	–
	TempTarget	Target value	False	DTtarget	Anything	–		<None>	–
*			False			–			–

Defining Elements

- Elements are created from the element templates to define the process equipment (MilkTank1, MilkTank2, Mixer)
- Element includes:
  - The attribute name defined in the template
    - PI tags
    - Process limit values
  - Data reference
    - PI Point
  - Settings
    - When the data reference is a PI Point the settings is the PI tag

Element Definition



Element Attributes

General | Attributes | Models

Mixer

	Name	Value	Value Type	Data Referen..	Settings
⊟	DTcodes				
	Mixer	Shutdown	String	PI Point	\\piserver2\MixerReasonCodes
⊟	DTcomments				
	OpComments	test ...	String	PI Point	\\piserver2\Mixer
⊟	DTinput				
	Speed	13.13925	Single	PI Point	\\piserver2\mixerspeed
	Temperature	61.78561 degF	Double	PI Point	\\piserver2\tempmixer
⊟	DTlimit				
	SpeedLimit	15	Long	<None>	–
	TempLimit	10	Long	<None>	–
⊟	DToperator				
	SpeedOp	<	String	<None>	–
	TempOp	>	String	<None>	–
⊟	DTsensor				
	MixerStatus	OFF	String	PI Point	\\piserver2\MixerOnOF
⊟	DTtarget				
	SpeedTarget	20	Long	<None>	–
	TempTarget	64	Long	<None>	–
⊟	DTtrigger				
	Output	Mixer Off	String	PI Point	\\piserver2\MixerDownTrigger
⊟	Tutorial				
	Volume Imbalance	0 gal	Double	<None>	–

Analysis Plug-In

- The analysis plug-in is the code that executes the logic or rules and writes the results back to PI that can be viewed through the client applications
- The analysis plug-in developed for the Downtime application example monitors the inputs from each equipment
- Compares the inputs to the target values
- The logic and the rules determines if a violation has occurred

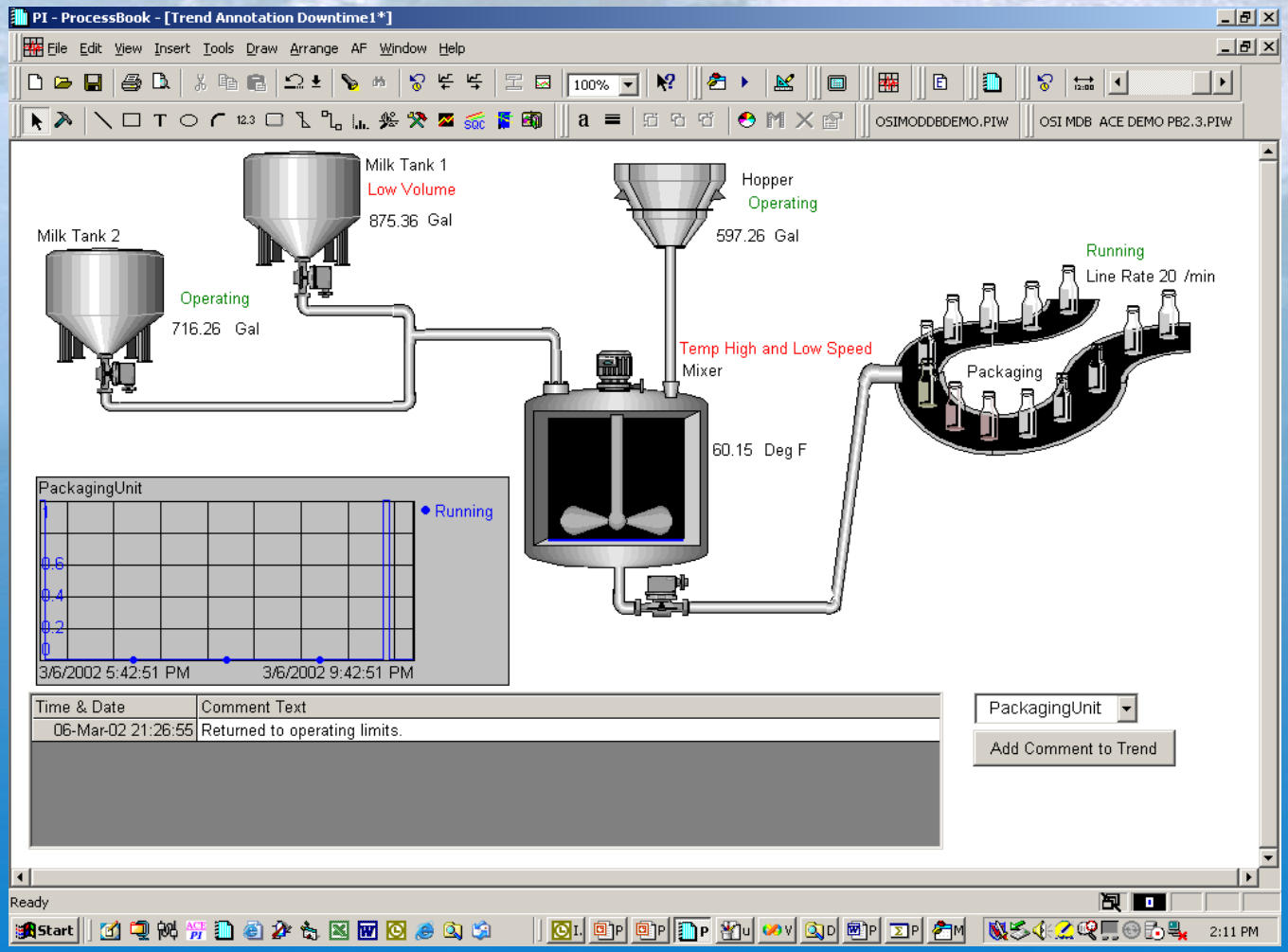
Analysis Plug-In

- The analysis plug-in requires the following:
  - Collect the inputs for the analysis
  - Run the analysis
  - Publish the analysis results
- A tutorial is provided with Application Framework that provides the required code
- The code that is added is specific to the application

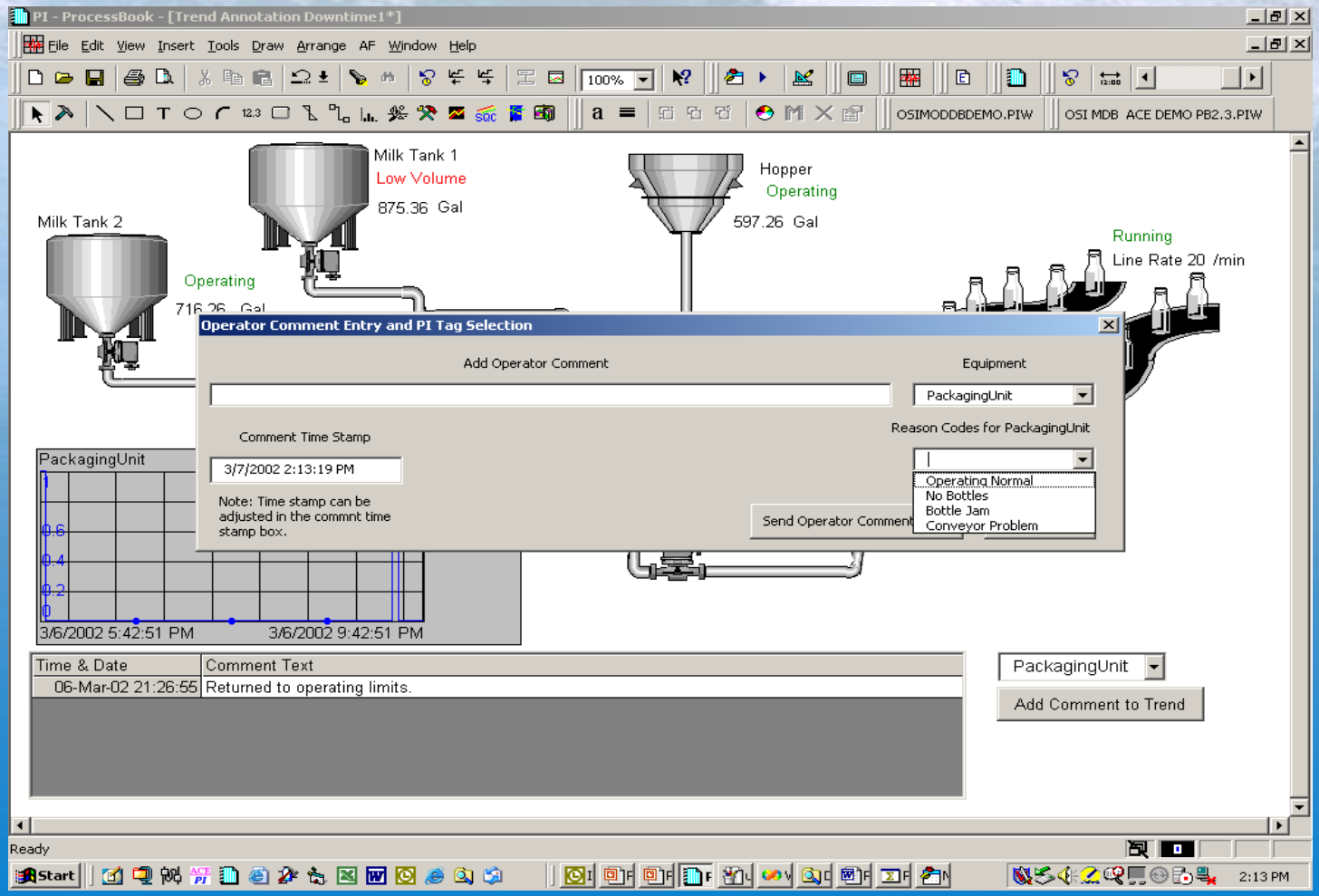
Analysis Plug-In

- Assign a time rule to determine when the analysis is scheduled to run
- In order to apply the analysis it must be registered on the machine where it will run
- The analysis must also be applied to a model that has been defined

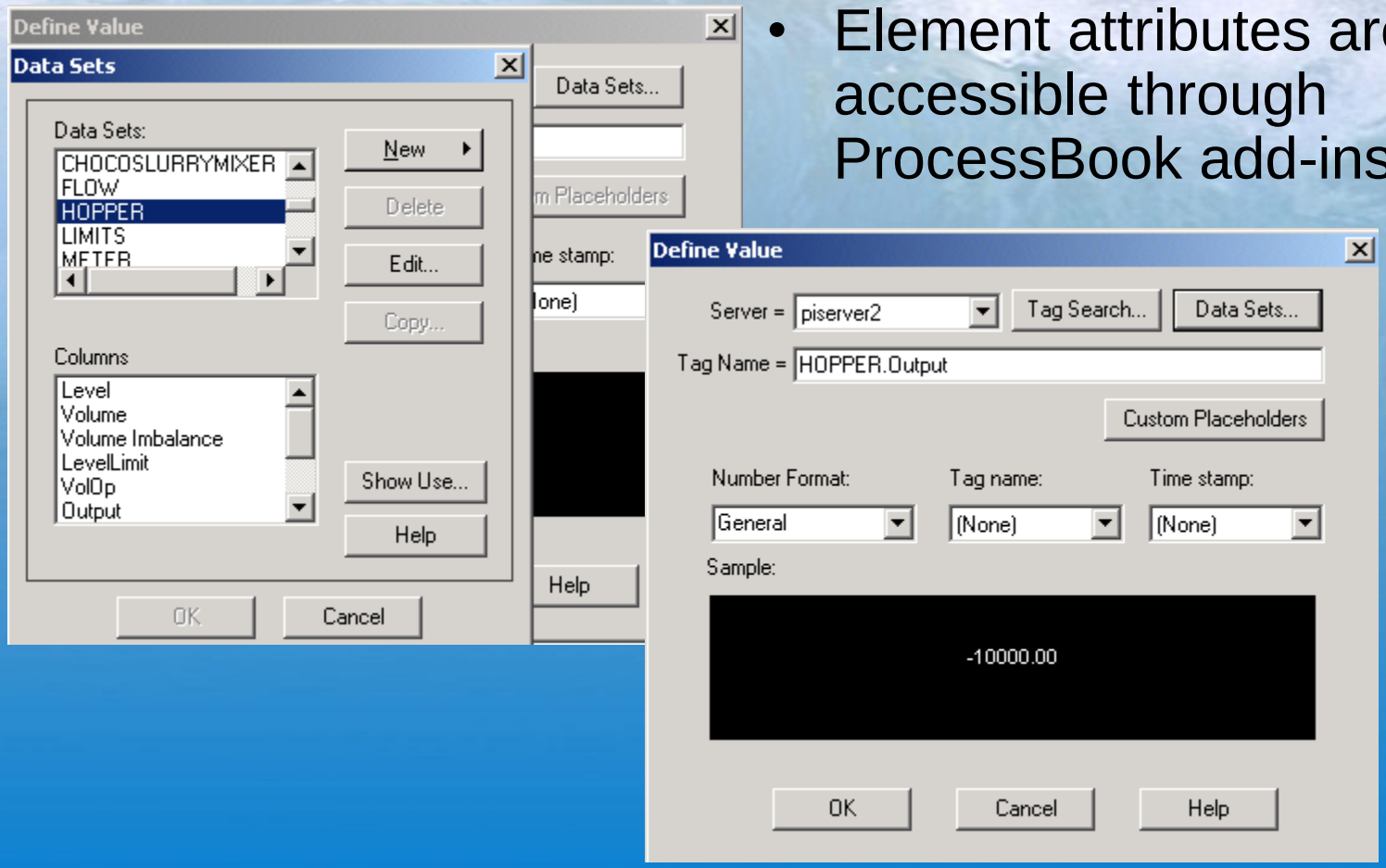
ProcessBook Display



Reason Codes based on Equipment

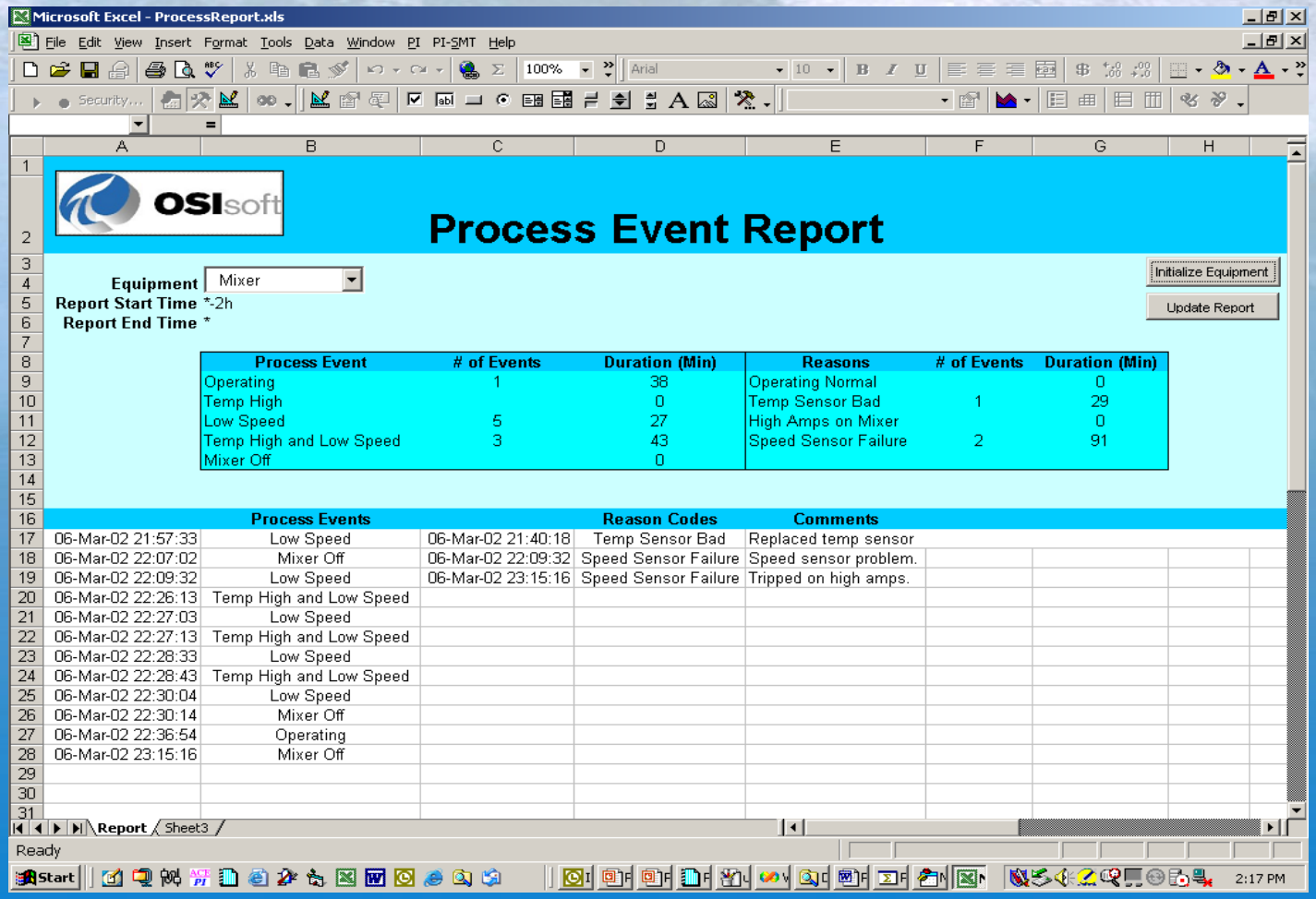


Integration with ProcessBook



- Element attributes are accessible through ProcessBook add-ins

DataLink Report



Summary

- The structured environment of the Application Framework significantly reduces the programming requirements for the analysis plug-in
- The application supports changes to the elements without re-programming
- Applications can be distributed and applied with minimal effort
- Integrates with ProcessBook

Summary

- The Application Framework can be applied to support simple to complex applications
  - Process calculations
  - Material balances
  - Downtime monitoring
  - Scheduling activities
  - Asset and inventory management