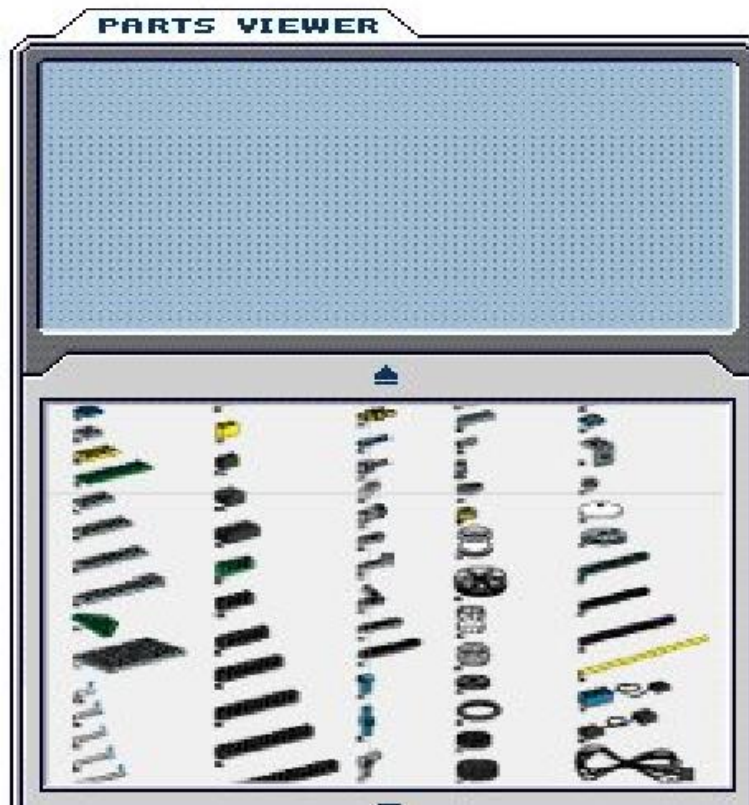


Making use of “LARGE” Models

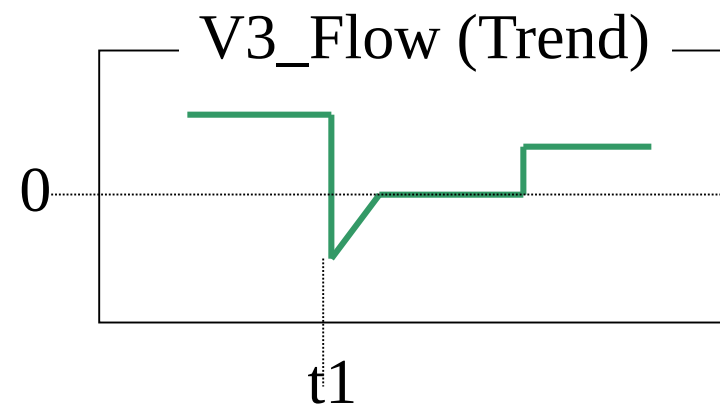
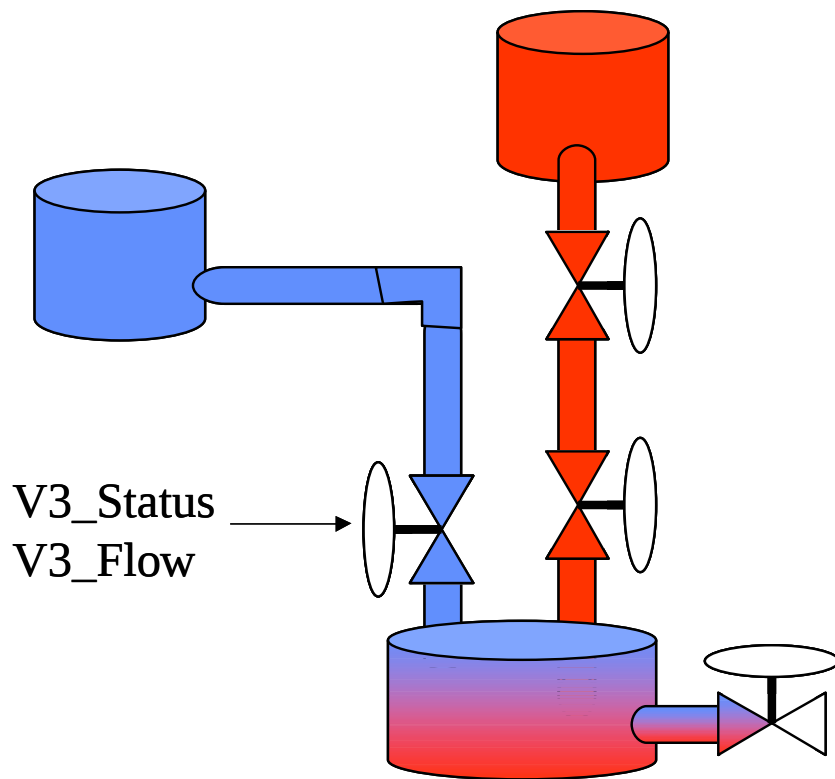


How to benefit from
and use
the Module Database

A little about :



- Have been developing with the MDB for ~1 year.
- Developing products to add value in maintaining models in the MDB.
 - Scalability
 - Large numbers of Modules (e.g. 1.5M)
 - Tested to 100K so far



Information Available

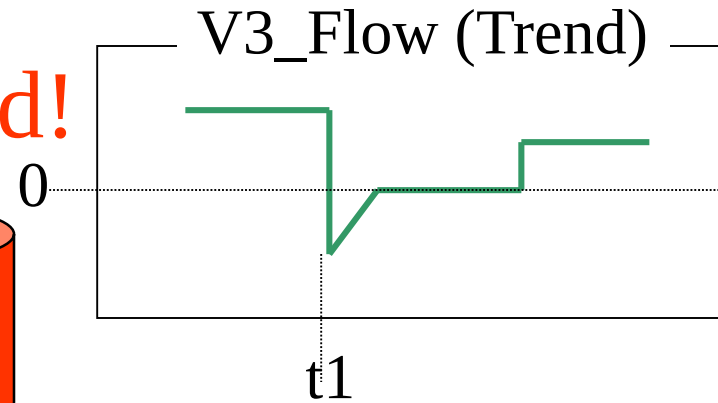
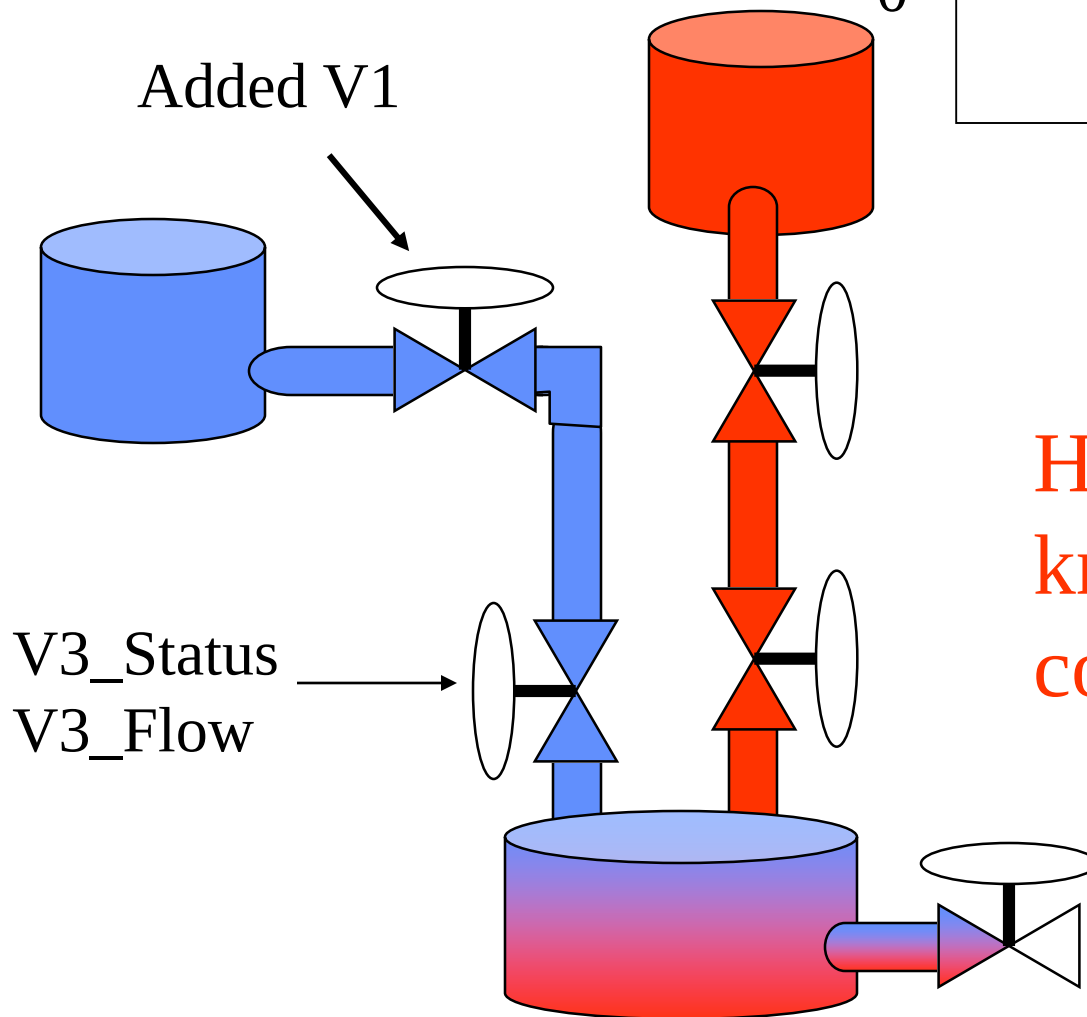
Historized values only
useful within a known
context.

Context embedded in
tag names.

The image shows a 'PI Tag Search' dialog box. It has a 'Search Criteria' section with fields for 'PI Server' (set to 192.168.1.102), 'Tag Mask' (*), 'Descriptor', 'Pt Source' (*), and 'Value' (*). To the right are buttons for 'Connections', 'Search', 'Abort', and 'Reset'. Below the search criteria is a 'Search Results' list box containing a table of tag information. An arrow points from the text 'Context embedded in tag names.' to the first column of the results table. At the bottom, it shows 'List Count: 11921' and 'Percent: 100'. On the right side of the dialog are buttons for 'Select All', 'Pt. Attr.', 'OK', 'Cancel', and 'Help'.

Tag Name	Tag ID	Tag Description
\\192.168.1.102\1611	I1611	IDG
\\192.168.1.102\1611	I1611	MVIAI
\\192.168.1.102\1611	I1611	MWIAI
\\192.168.1.102\1611	I1611 Y	40KVIAI
\\192.168.1.102\1615	I1615	IDG
\\192.168.1.102\1615	I1615	MVIAI
\\192.168.1.102\1615	I1615	MWIAI
\\192.168.1.102\1615	I1615 Y	40KVIAI
\\192.168.1.102\1616	I1616	IDG
\\192.168.1.102\1616	I1616	MVIAI
\\192.168.1.102\1616	I1616	MWIAI
\\192.168.1.102\1616	I1616 Y	40KVIAI

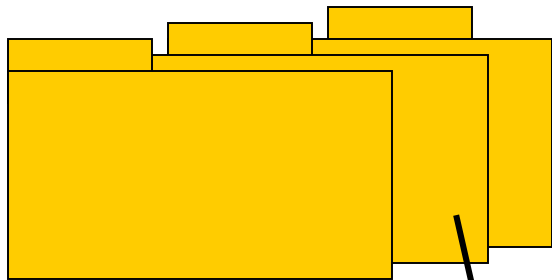
What really happened!



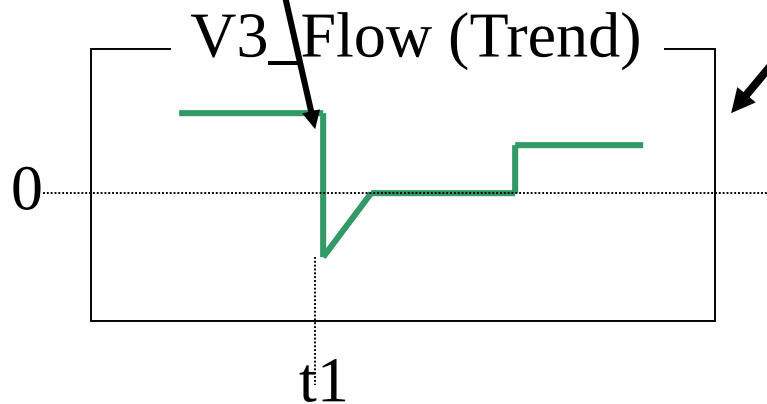
How can this knowledge be conveyed?

Knowledge can be achieved:

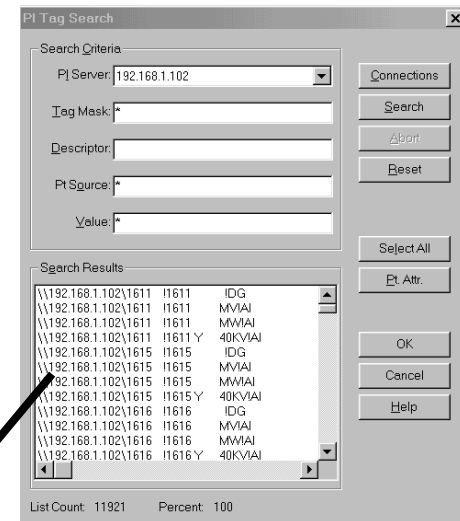
Context



V1 addition shown



Process Variables



Context is more difficult when:

- larger companies
- multiple organizations are involved.
- geographic separation of process from user of information.
- there is a “failure to communicate”
-

Some values are:

- Associated with objects (e.g. a valve).
 - Number of operations
 - Time in service
 -
- As objects are moved, values need to move with object.

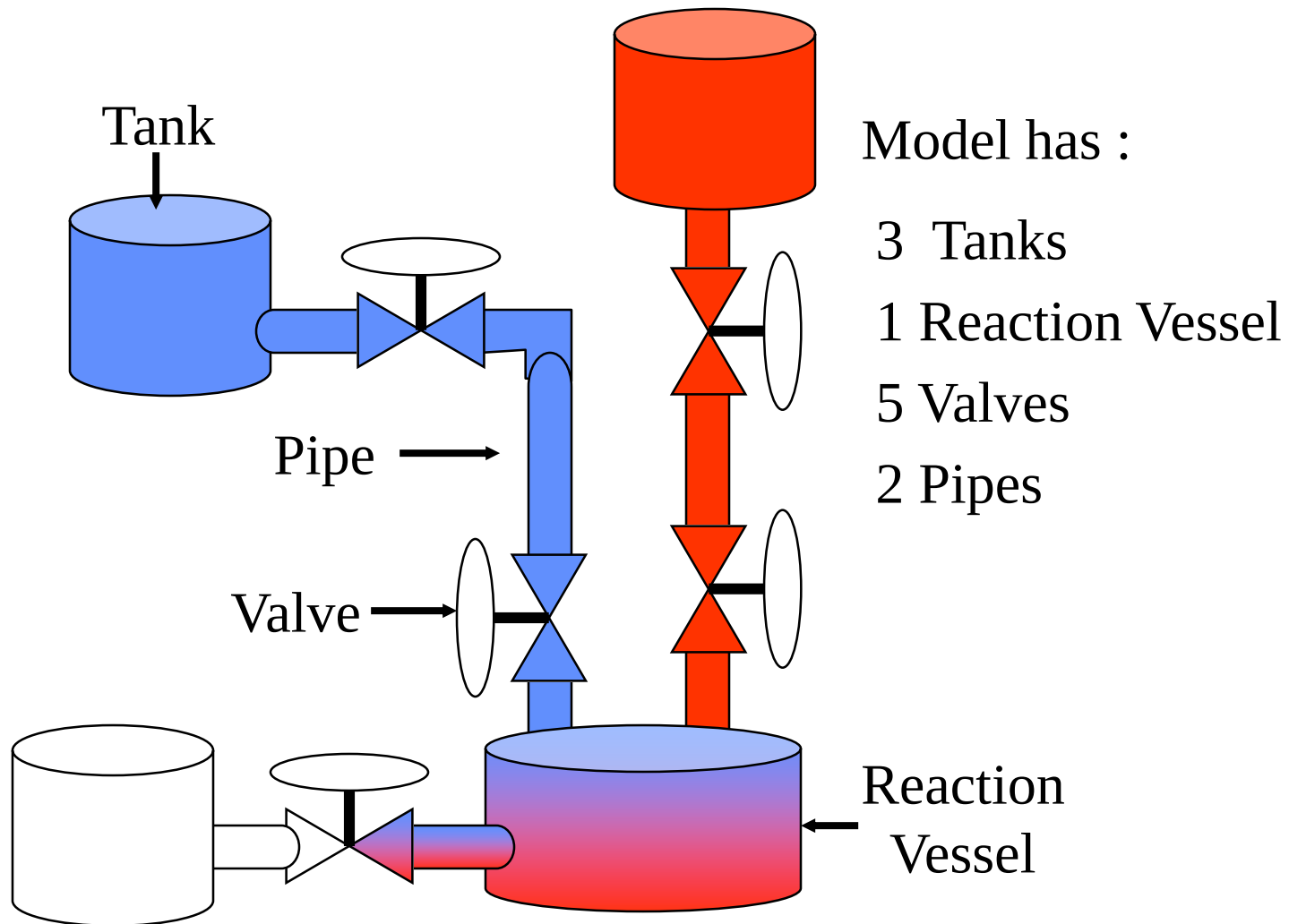
Inherent MDB Benefits

- Ability to provide context for process information.
- Ability to provide long-term history of object operation.
- Ability to minimize interpretation errors.

SISCO saw additional benefits:

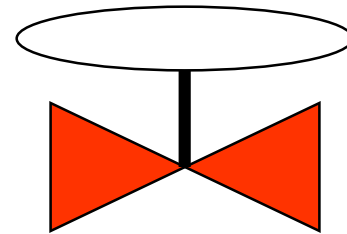
- Ability to synchronize contexts.
- Ability to define programmatic methods to allow association.
- Ability to develop more general and reusable applications.
- Lower risk during deployment.
- Minimize human error.

Classes and Instances of Objects



Class Definition depends on Use

- Asset Management
 - Vendor
 - Purchase Date
 - Commission Date
 - Location....
- Operational
 - Position
 - Differential Pressure
 - Attached to pipe()...
 - ControlledBy()....



How many pipes can be attached to a valve?

Associations Need to be bi-directional

A valve is AttachedTo 0..2 pipes.

Allows users/programs to locate which pipe a valve is attached (knowing valve)

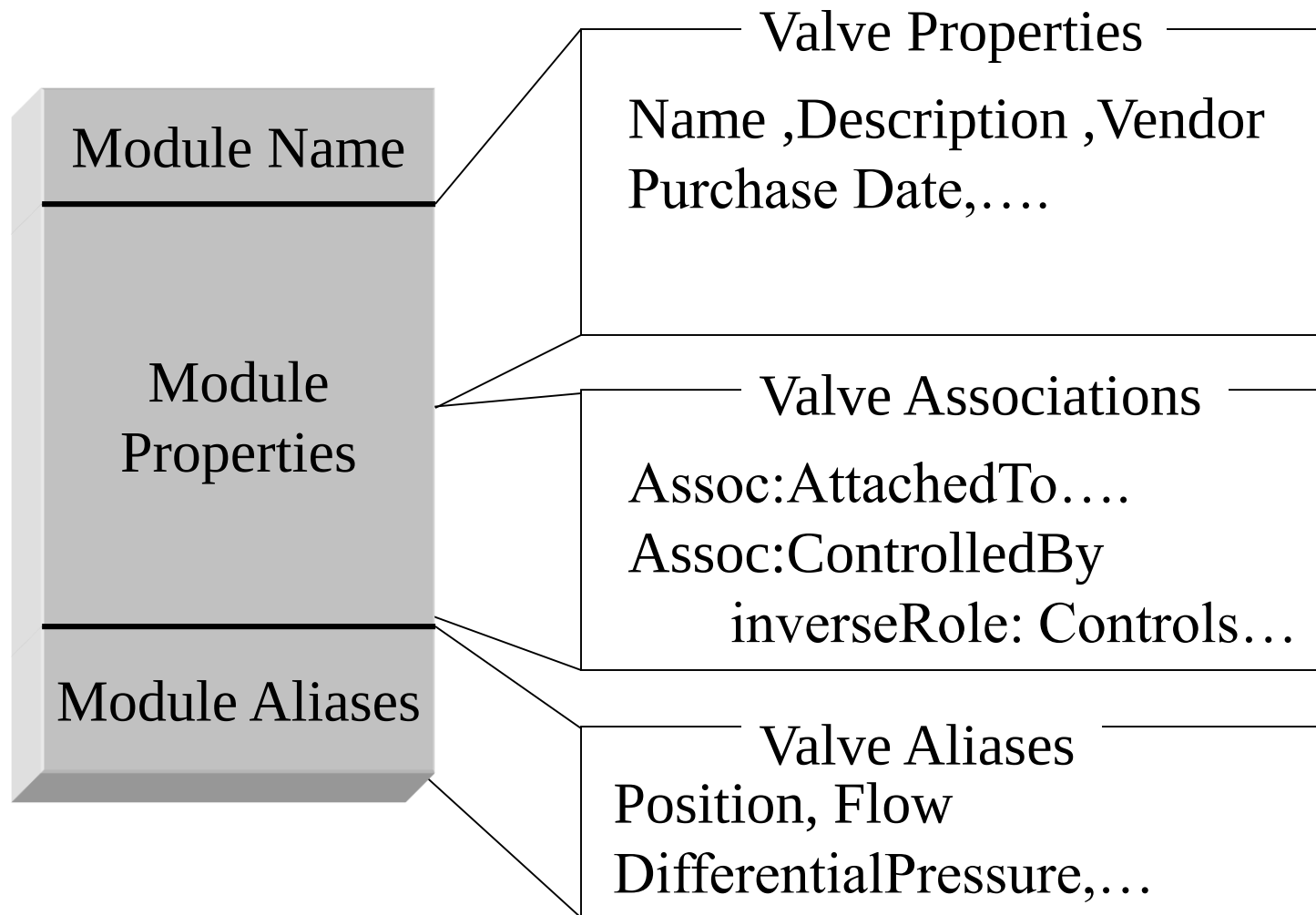
A pipe is AttachedTo 0..n valves.

Allows users/programs to locate which valve(s) are attached (knowing pipe)

Users/Programs Use Model to:

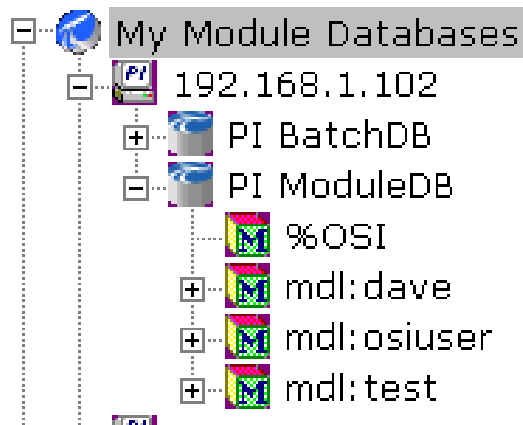
- Locate a particular valve knowing the object being looked for is a valve and its name.
- Locate a particular valve based upon its location in a operational system***.
- Locate a particular pipe based upon its valves***.
- Locate a particular valve based upon what controls it.

Sample valve definition build-out



First Decisions:

Need to plan for multiple models in the MDB.

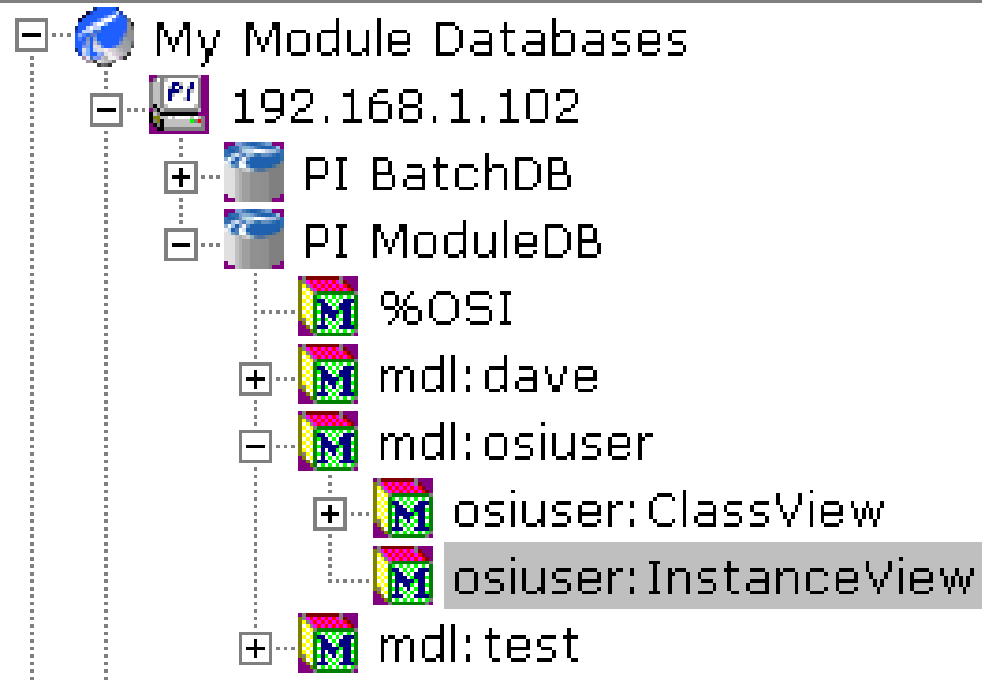


SISCO's design choices:

Need to be able to determine where the start of a model is.

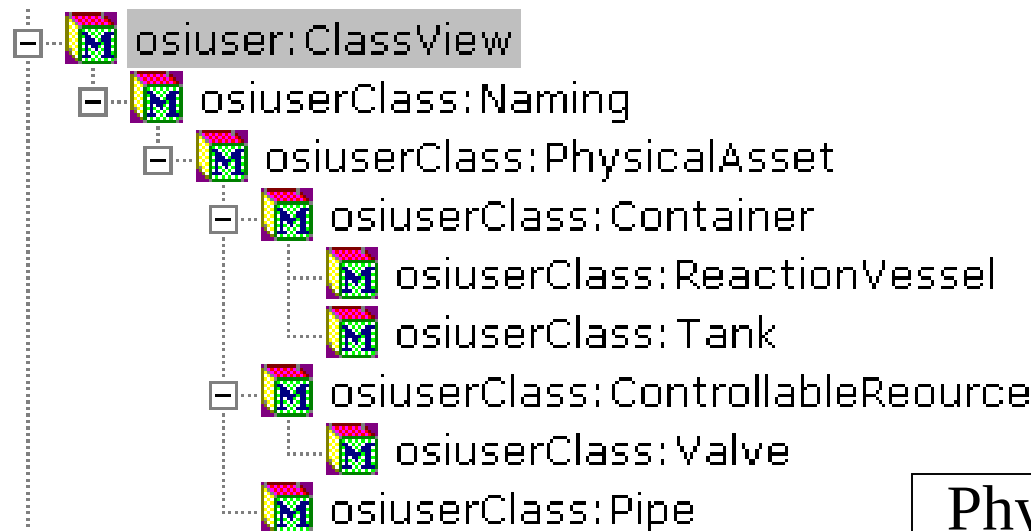
Need to prevent namespace collision.

Next Decide on How users want to use information

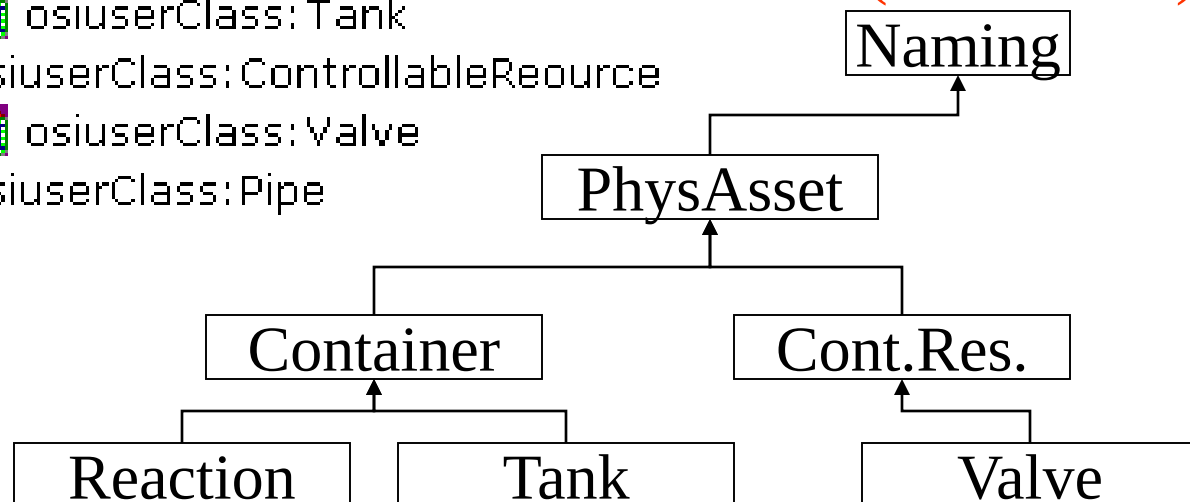


Decide on Model Definition

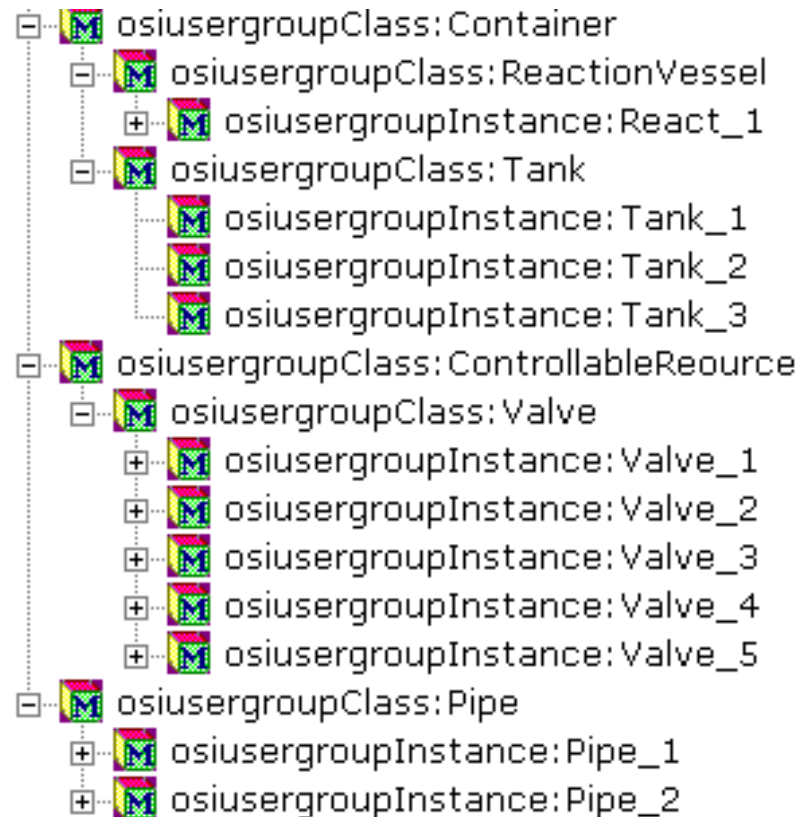
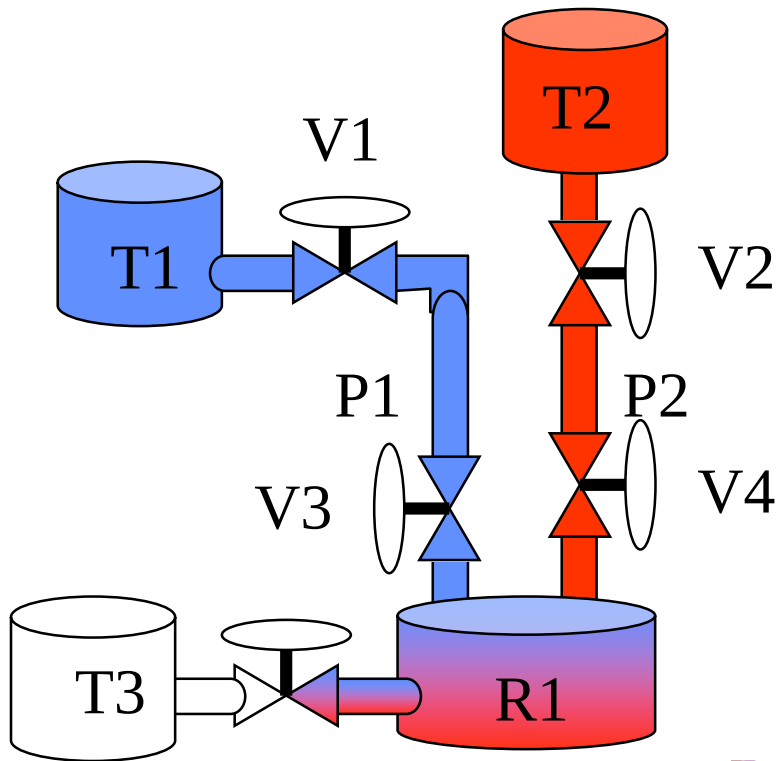
MDB Representation



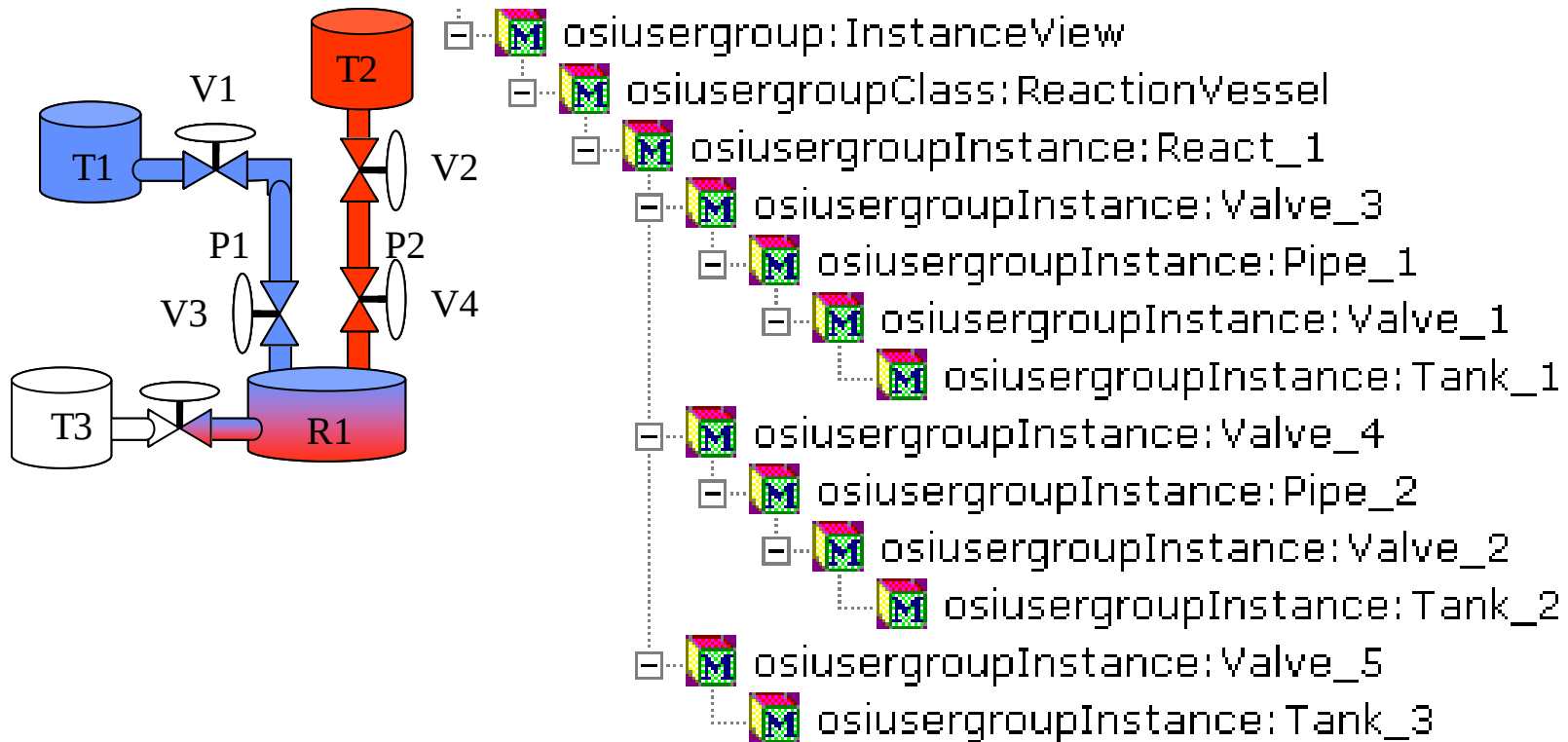
OO Definition (Inheritance)



Objects are Instances of Classes



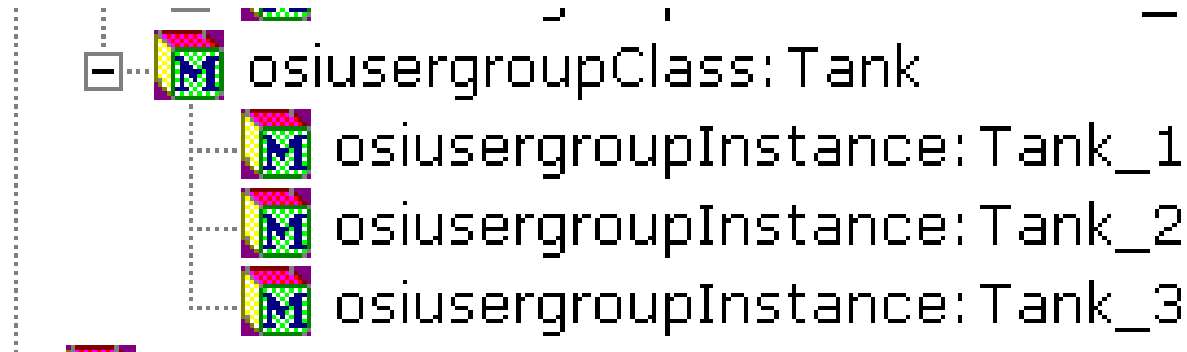
Objects relate to other Objects



What if user wants to start with tanks instead?

What if users wants Tanks?

From: Class View

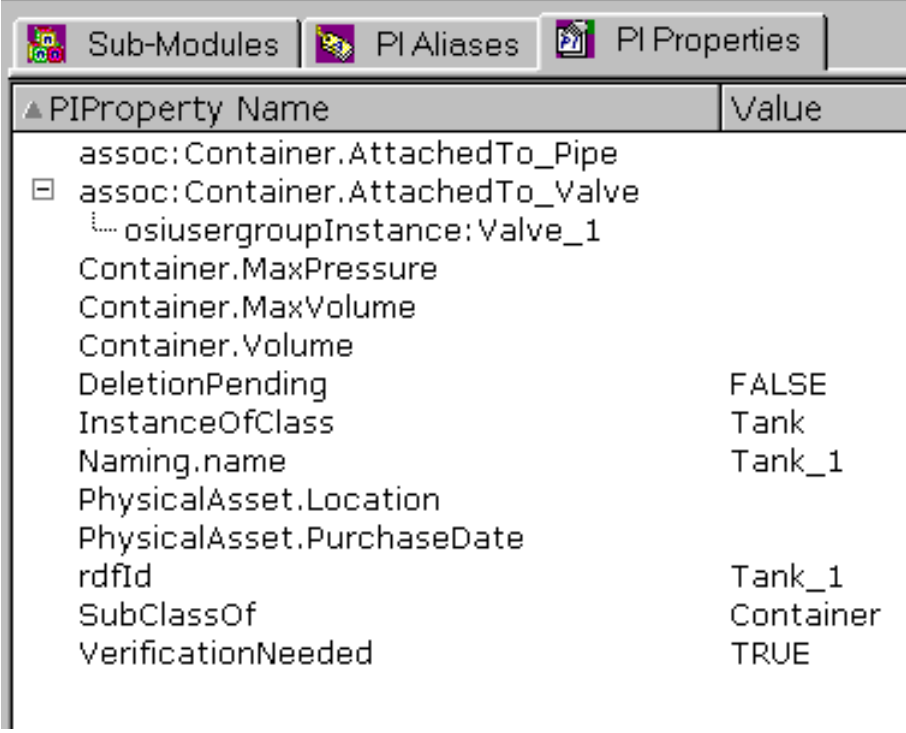


Q:Where are the valves?

A: Circular references are not allowed in the MDB.

So what to do?

A:Embed Object Relationships

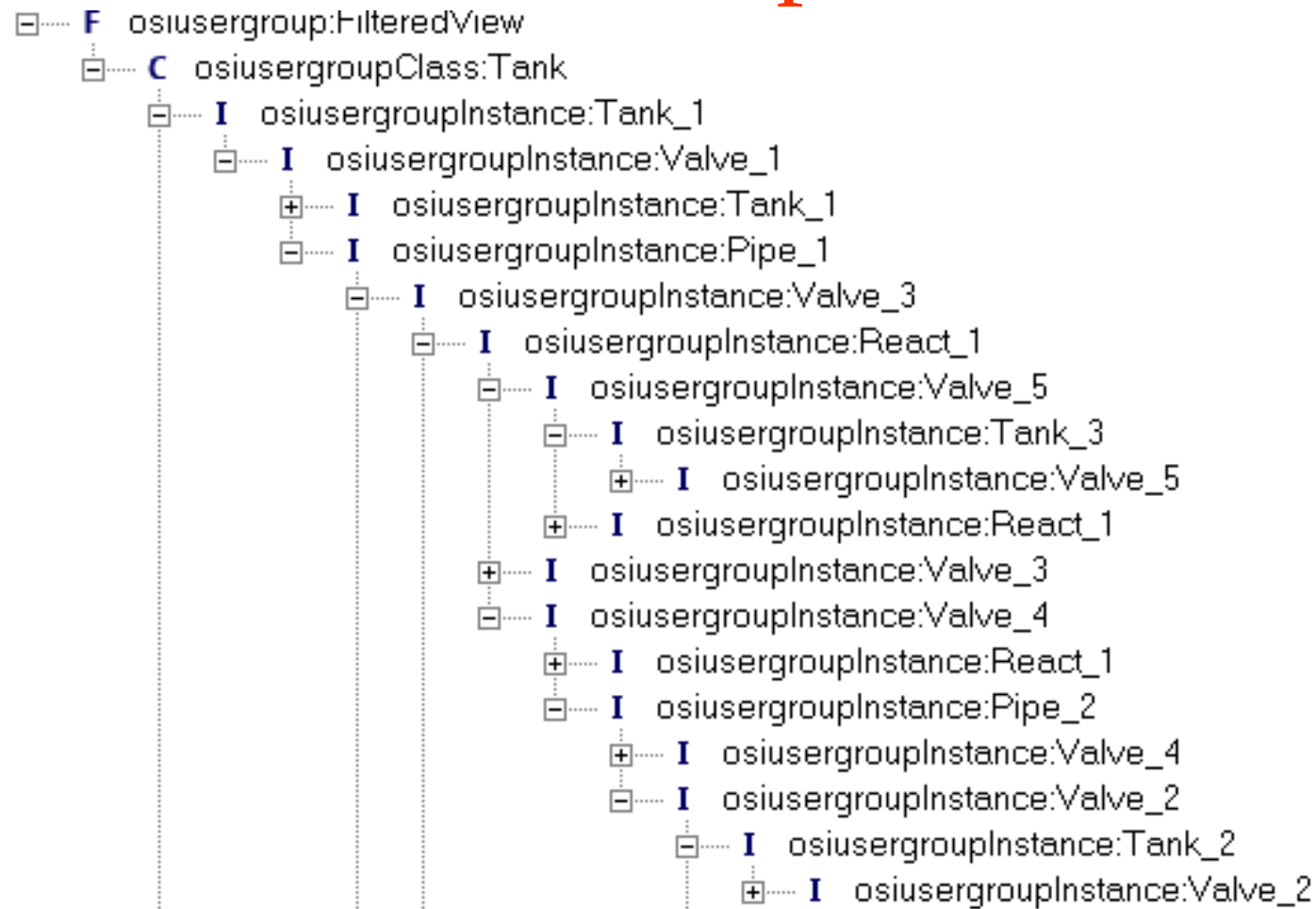


The screenshot shows a software window with three tabs: 'Sub-Modules', 'PI Aliases', and 'PI Properties'. The 'PI Properties' tab is active, displaying a table with two columns: 'PIProperty Name' and 'Value'. The table lists various properties of a 'Tank' object, including its association with a container and valve, its physical characteristics (pressure, volume, location), and its classification (instance of class, sub class of container).

PIProperty Name	Value
assoc:Container.AttachedTo_Pipe	
assoc:Container.AttachedTo_Valve	
osigroupInstance:Valve_1	
Container.MaxPressure	
Container.MaxVolume	
Container.Volume	
DeletionPending	FALSE
InstanceOfClass	Tank
Naming.name	Tank_1
PhysicalAsset.Location	
PhysicalAsset.PurchaseDate	
rdId	Tank_1
SubClassOf	Container
VerificationNeeded	TRUE

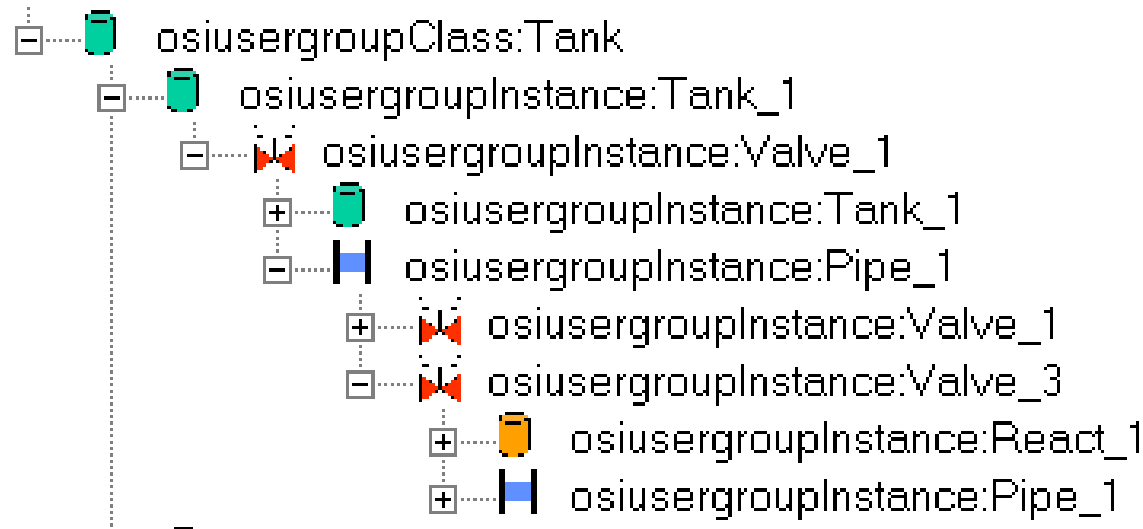
Allows programmatic transversal of object relationships!

Example



Imagine:

- If names were numbers.
- If there were multiple object types at a single level.
- 100 object instances at a single level.



Imagine large models:

- Utility industry models (network)
 - 10K-1.5M Modules (50K Typical)
- Generation facilities (500-10K Typical)
- Process/Batch Facilities (500-10K)

Other truths:

- Models are not static
 - Equipment is added/moved/removed
 - Schedules can be changed
- Models need to be shared.
 - Synchronization becomes an issue
- Models need to be well thought out in order to allow applications to take advantage.

Models can assist applications

Sub-Modules		PI Aliases	PI Properties
PIProperty Name		Value	
assoc:Container.AttachedTo_Pipe			
assoc:Container.AttachedTo_Valve			
osusergroupInstance:Valve_1			
Container.MaxPressure			
Container.MaxVolume			
Container.Volume			
DeletionPending			
InstanceOfClass			
Naming.name			
PhysicalAsset.Location			
PhysicalAsset.PurchaseDate			
rdfId			
SubClassOf			
VerificationNeeded			

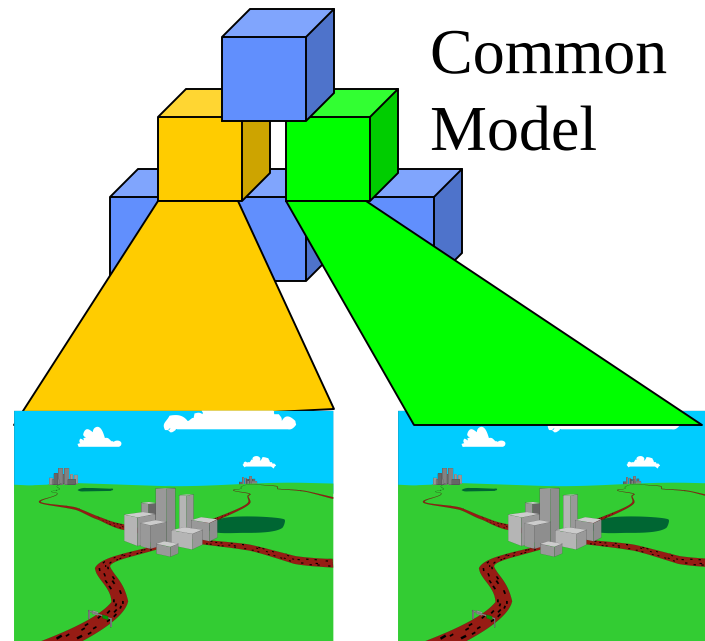
Allows programs
to transverse
relationships

Allows PB apps
to properly choose
display icon.

Allows user
intervention to be flagged

Sharing of Models

- Common Model consists of several subsets.
- Subsets “owned by” different entities.



Model Events change Context

- Adding an Object
 - E.g. a new tank/reactor has been added in a plant.
- Modify of an existing object
 - A valve has been moved
- Deleting an object
 - Valve destroyed.



Why use the MDB

- MDB Benefits
 - Integrates seamlessly with OSI environment.
 - Historizes changes
 - Provides “efficient” methods for hierarchical and browsable models.
 - OSI continues refinements...



Cost Justifications Ease for

- systems requiring contextual tracking
- systems where users desire “context” and real-time merging.
- systems where historian function is needed but unknown numbers/types of databases.



Questions?