

# Process, Power and Marine Division

## SmartPlant 3D Data Model

April 2005

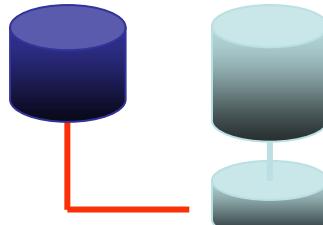


INTERGRAPH

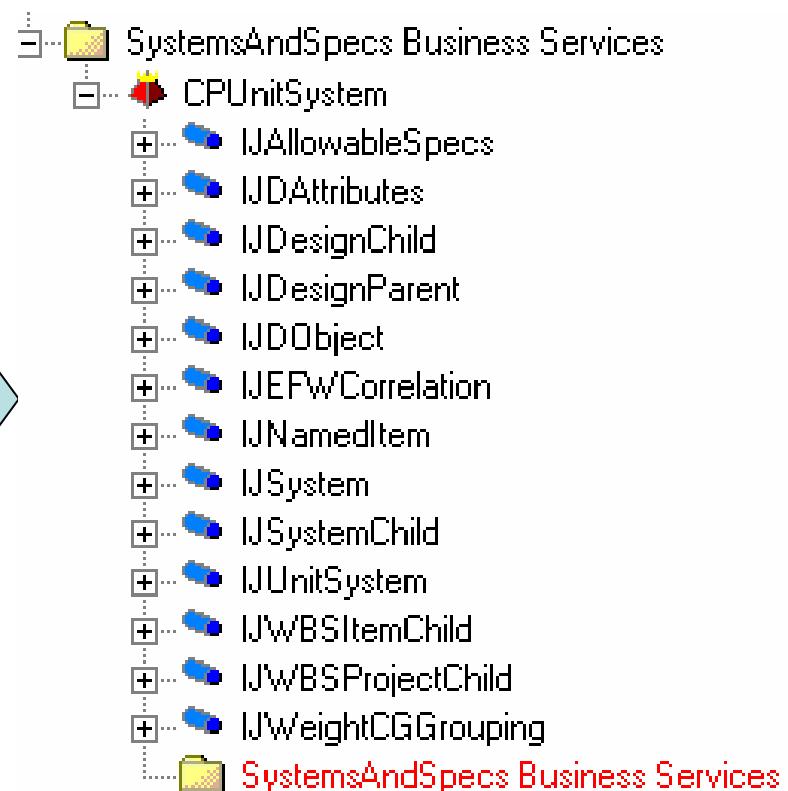
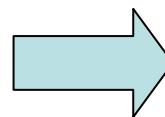
# SmartPlant 3D Data Model

Use the MetaDataBrowser Tool to view the data model

**Model  
Database**



**Catalog Database**  
**Catalog Schema  
Database**



# MetaData Browser Tool

**SmartPlant®  
3D**

The screenshot shows the SP3D Standalone Metadata Browser window titled "SP3D Standalone Metadata Browser - SP3DTrain\_Cat\_SCHEMA". The left pane displays a tree view of the SQL Repository under "SP3D Train\_Cat\_SCHEMA". The right pane shows the "Properties of Interface: IJGeneralNote".

Legend (left side):

- Package
- Class
- Interface
- Property
- Codelist
- Relation Collection
- Relation
- Ancestor

Properties of Interface: IJGeneralNote

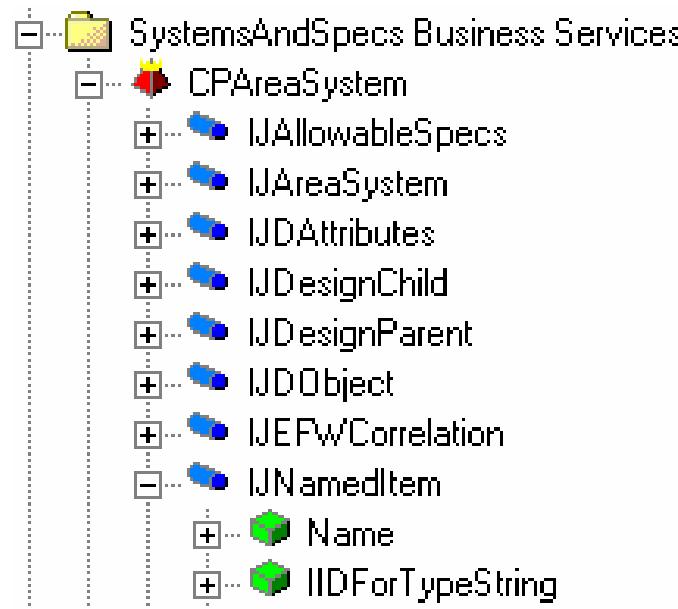
Name	Data
Name	IJGeneralNote
UserName	IJGeneralNote
DBViewName	JGeneralNote
OID	{DC701A4B-27F7-4FFB-95EF-C45DCFE8AD71}
IID	{DC701A4B-27F7-4FFB-95EF-C45DCFE8AD71}
CategoryID	[1] - Standard
Has Attribute	Name
Has Attribute	Diminished
Has Attribute	Purpose
Has Attribute	IJGeneralNote
Has Relation Collection	NotePurpose
Has Relation Collection	Note
Has Relation Collection	ControlPoint
Has Relation Collection	NoteAssociatedToControlPoint
Has Relation Collection	ControlPoint
Is Implemented by First Class Class	
Belongs to Package	CommonApp Business Services

Annotations:

- Metadata Object highlighted in the treeview or found by Find
- Metadata Object Properties
- Metadata Object Relationship

# MetaData Browser Tool

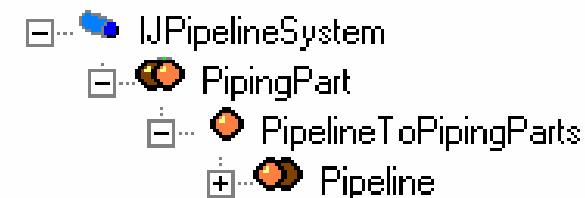
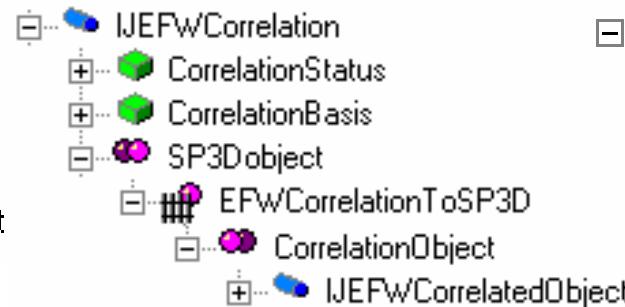
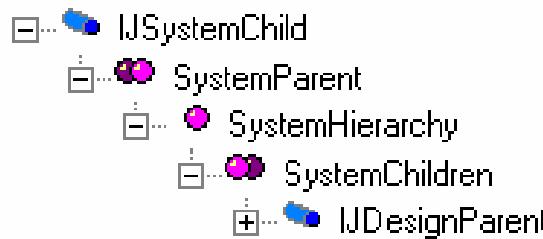
- A regular folder represents a package i.e. the set of metadata, without dependencies.
- A red pyramid represents a class. A red pyramid overlayed by a yellow crown represents a first class, i.e. a class that is a leaf in the BOC (Business Objects Classification) hierarchy.
- A green box represents a property.



# MetaData Browser Tool



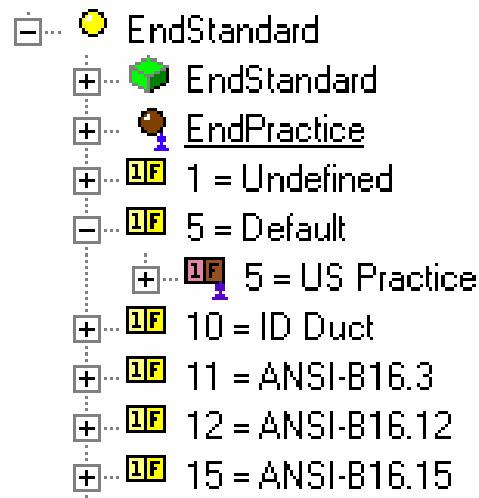
- A single magenta sphere represents a relationship. A single magenta sphere overlayed by a fence represents a private relationship.
- Double magenta/purple spheres represent a relationship collection (a role).
- An orange diamond represents an Edge, i.e a virtual relationship that "bridges" several relationships.
- Double Orange diamonds represent an edge collection (a role).



# MetaData Browser Tool



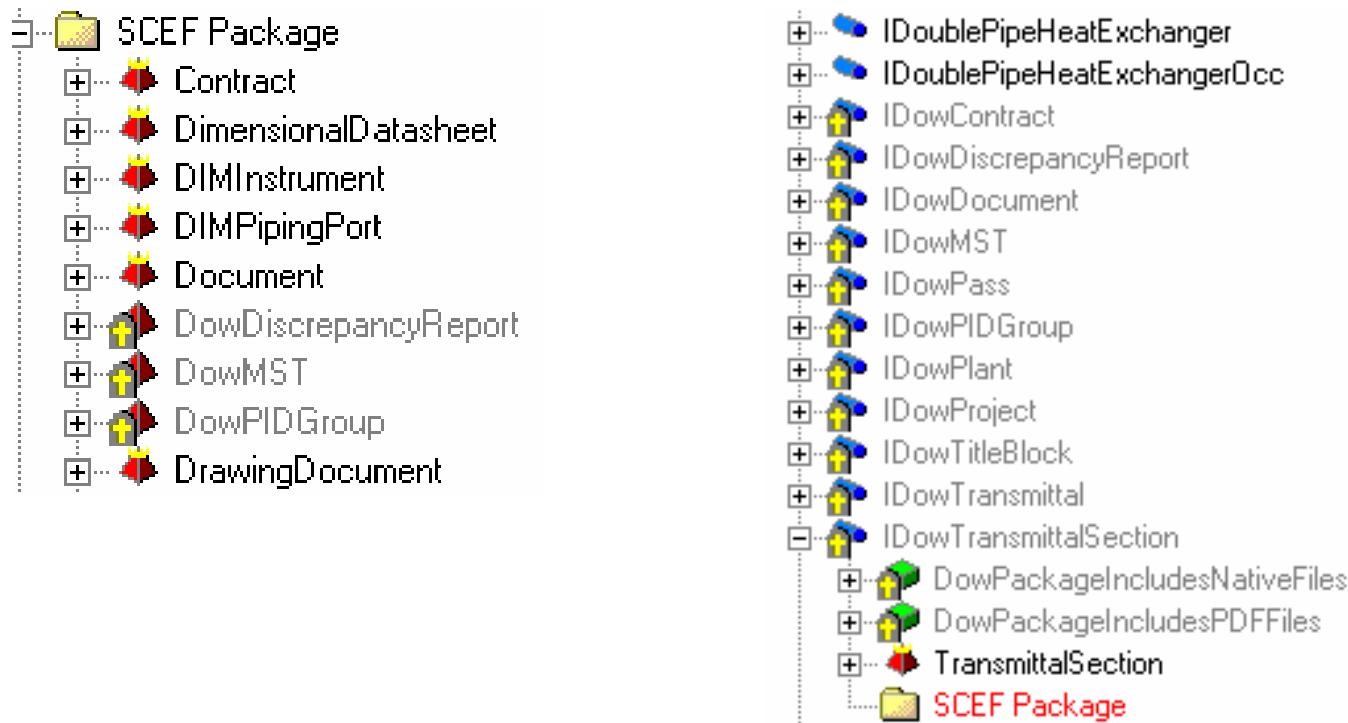
- A yellow sphere represents a codelist table.
- A yellow domino with a number and a letter represents a codelist value.
- A brown sphere overlayed by a blue human outline represents a parent codelist table.
- A brown domino with a number and a letter overlayed by a blue human outline represents a parent codelist value.



# MetaData Browser Tool

**SmartPlant®  
3D**

- Any icon overlayed by a tombstone represents a metadata object marked as deleted (V6).



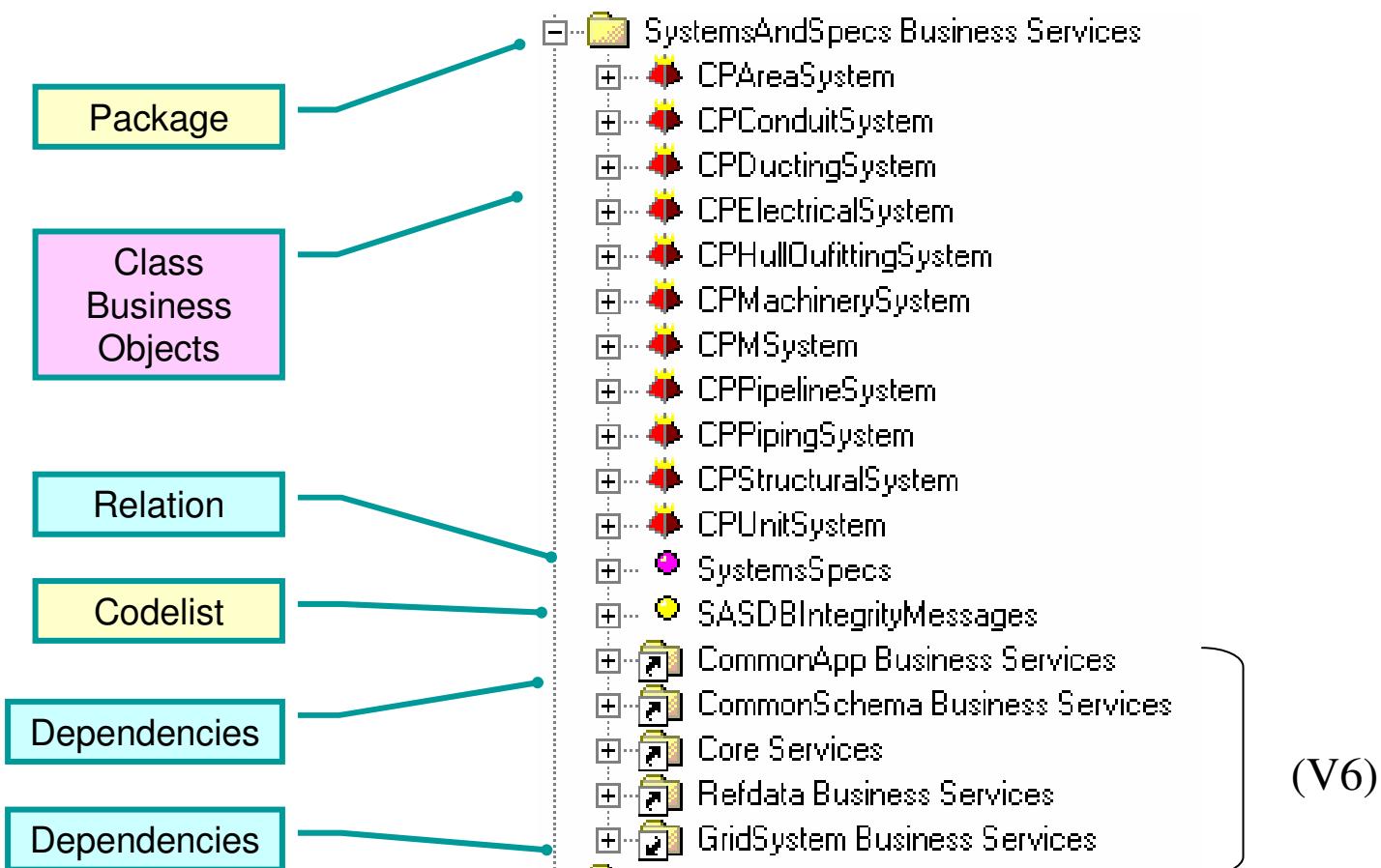
# MetaData Browser Tool



- Under View -> Units of measure, you can see the units for each Unit type. For each Unit type, the blue unit represents the (SI) Database unit (DBU). Right click on any unit allow you to convert a value in DBU.
- Under view -> Enumerations, you can know the different values of all the enumerations used by the metadata.
- If you re-open often the same schema, you should choose "Auto open open DB dialog" in Options.
- You can switch from displaying metadata object internal Names (System) to Usernames (User).

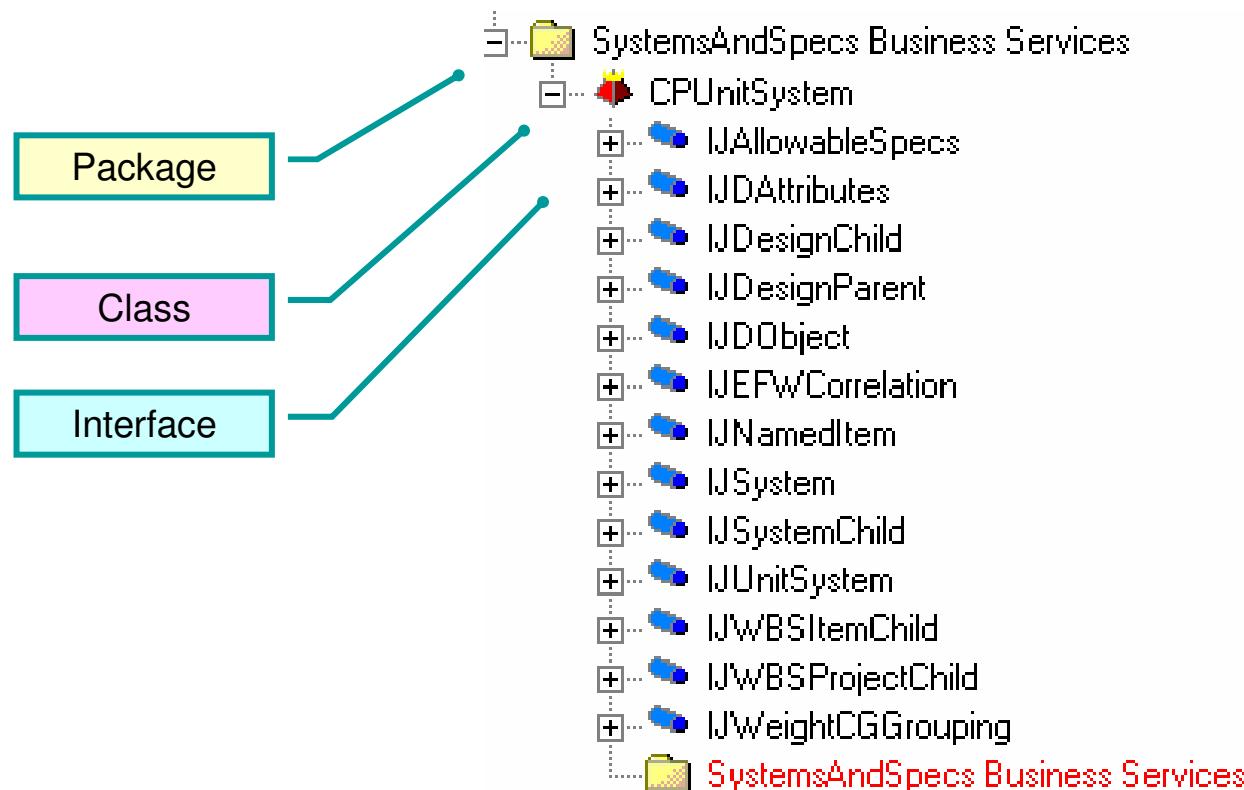
# System Entity Data Model

The system hierarchy is a functional breakdown of the plant model that allows the user easy traversal of the functional components of the design of a plant.



# System Entity Data Model

List of interfaces implemented by the Unit System



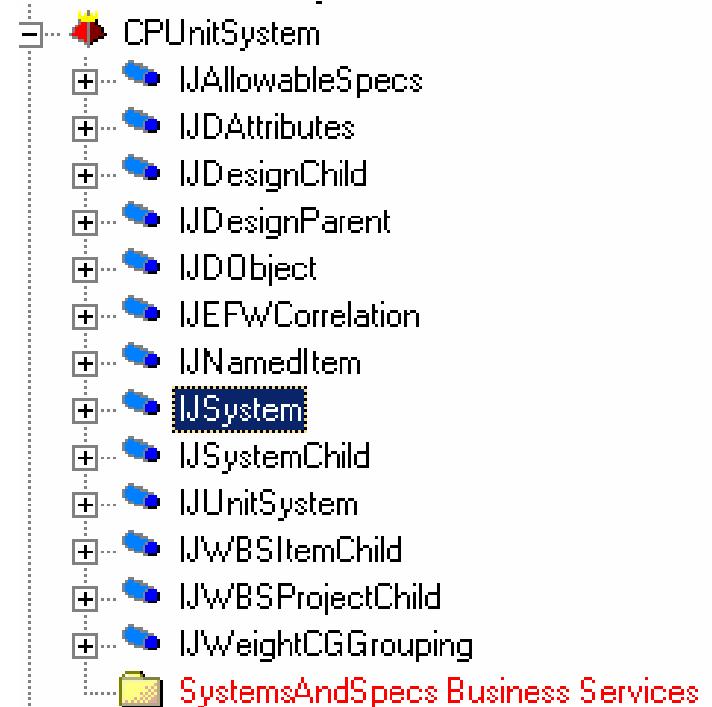
# System Entity Data Model

- IJSystem: This interface is used to provide a common interface for system objects.
- JNamedItem: This interface provides the name and displayable type string of an object.

**Example: Query all systems in the model**

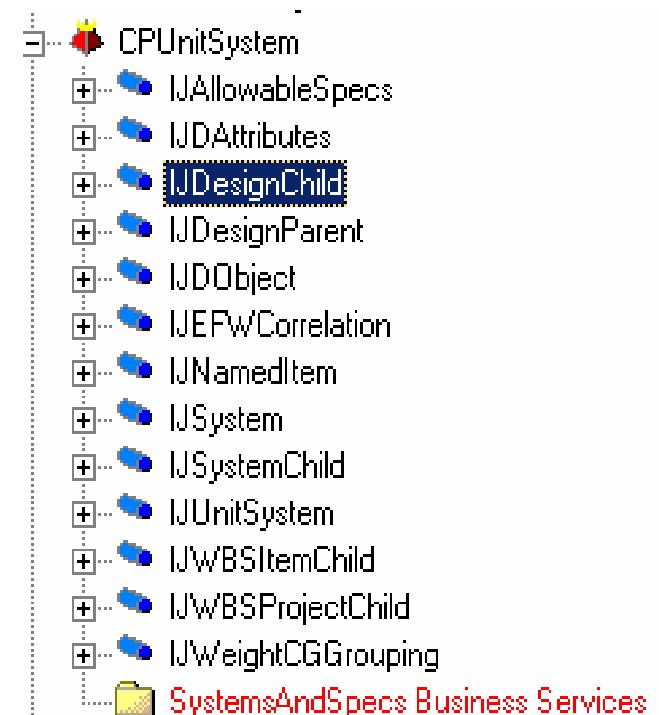
```
Select x3.ItemName
from JSystem x1
Join JNamedItem x3 on x3.oid = x1.oid
```

	ItemName
1	Plant144-P
2	Plant1
3	Unit 300
4	Equipment
5	Structure
6	Columns
7	Main Beams
8	Secondary Beams
9	Braces
10	Mics
11	Footings



# System Entity Data Model

- IJSystemChild: is used to associate an object with its parent by means of the *SystemHierarchy* relationship.
- IJDesignChild: is used to associate an object with its parent.
- IJDesignParent: is used to associate systems and other objects as children of the system.
  - An object in the system hierarchy that can have children must implement the IJDesignParent interface.
  - An object that can be the child of a parent must implement the IJDesignChild interface.

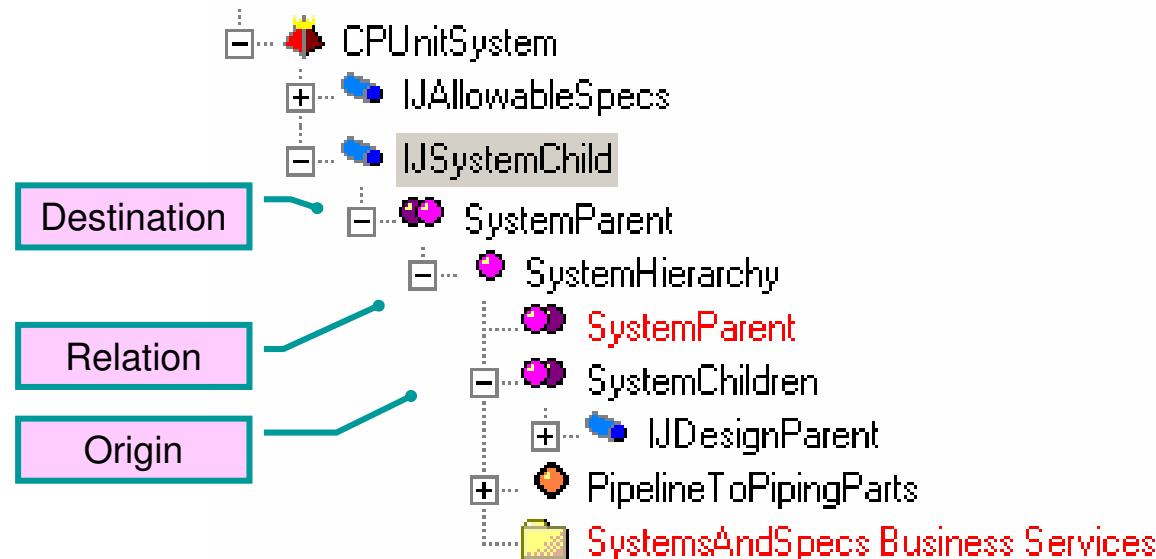


# System Entity Data Model

## Relationships

- Relations are binary
- A relation is between two and only two entities.
- These entities are known as **origin** and **destination** of the relation

Example: system hierarchy relationship (There is a relation between child and the parent system).



# System Entity Data Model

Example: List the child-parent system names

Select x2.ItemName as Child, x4.ItemName as Parent  
from JSystemChild x1

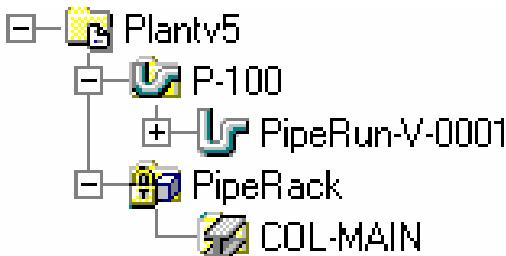
Join JSysteM x5 on x5.oid = x1.oid

Join JNamedItem x2 on x2.oid = x1.oid

Join XSystemHierarchy x3 on x3.oidDestination = x2.oid

Join JNamedItem x4 on x4.oid = x3.oidOrigin

	Child	Parent
1	COL-MAIN	PipeRack
2	P-100	Plantv5
3	PipeRack	Plantv5



# System Entity Data Model

Example: Query piping parts where pipelines are in piping system

Select

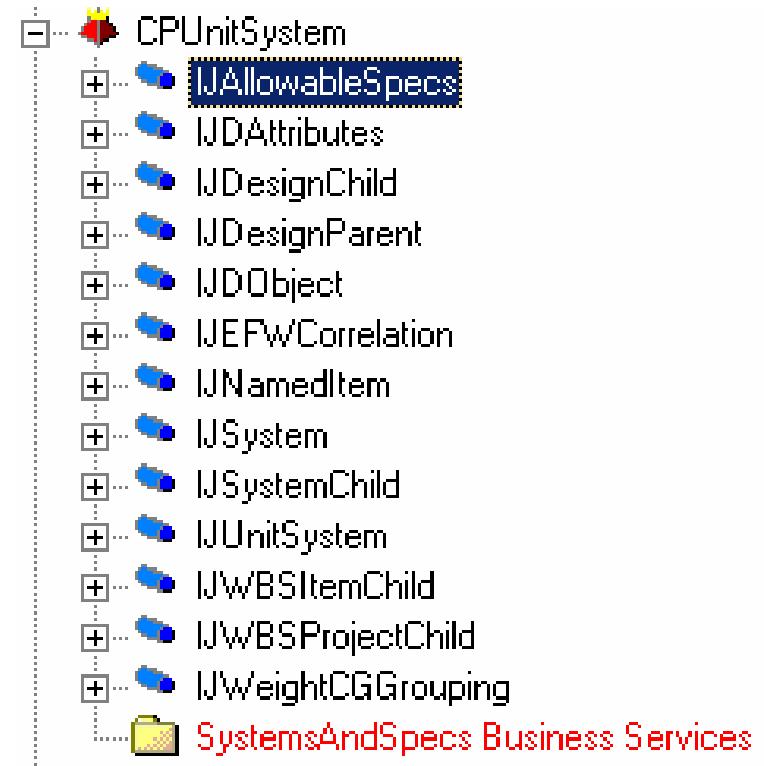
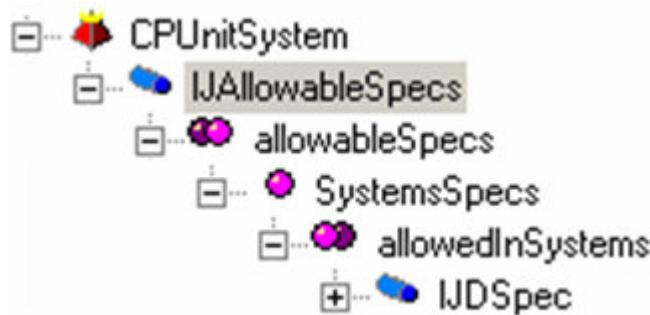
```
x3.ItemName as PipingSystem, x7.ItemName as Pipeline, x6.ItemName as Part  
from JSystemChild x1  
Join JNamedItem x7 on x7.oid = x1.oid  
Join XSystemHierarchy x2 on x2.oiddestination = x1.oid  
Join JNamedItem x3 on x3.oid = x2.oidOrigin  
Join JPipingSystem x4 on x4.oid = x2.oidOrigin  
Join YPipelineToPipingParts x5 on x5.oiddestination = x1.oid  
Join JNamedItem x6 on x6.oid = x5.oidOrigin
```

-- Pipeline system  
-- get pipeline name  
-- get the parent  
-- get parent name  
-- parent = pipsys  
-- get parts  
-- get part name

	PipingSystem	Pipeline	Part
1	P-200 lines	P-200	Pipe
2	P-200 lines	P-200	Component-0401
3	P-200 lines	P-200	Pipe
4	P-200 lines	P-201	Pipe
5	P-200 lines	P-201	VG3-0402
6	P-200 lines	P-201	Pipe
7	P-200 lines	P-201	Flange-0403
8	P-200 lines	P-201	Flange-0404

# System Entity Data Model

- **IJAllowableSpecs:** This interface is required in order to manage specifications assigned to a system.



# System Entity Data Model

## Example: Query all allowable specs per Piping Systems

```

select
x5.SpecName as AllowableSpec,
x6.ItemName as PipingSystem
from JDSpec x1
join JDPipeSpec x5 on x5.oid = x1.oid
-- get all spec in model
-- get only pipe spec

-- return only modified systems using the SystemSpec relation

join XSystemsSpecs x2 on x2.oiddestination = x1.oid
join JNamedItem x6 on x2.oidOrigin = x6.oid
-- get modified system name

-- where modified system is in subquery

where x2.oidOrigin in
(select x3.oid from JPipingSystem x3
 join JNamedItem x4 on x4.oid = x3.oid )
order by x6.ItemName
-- return piping systems

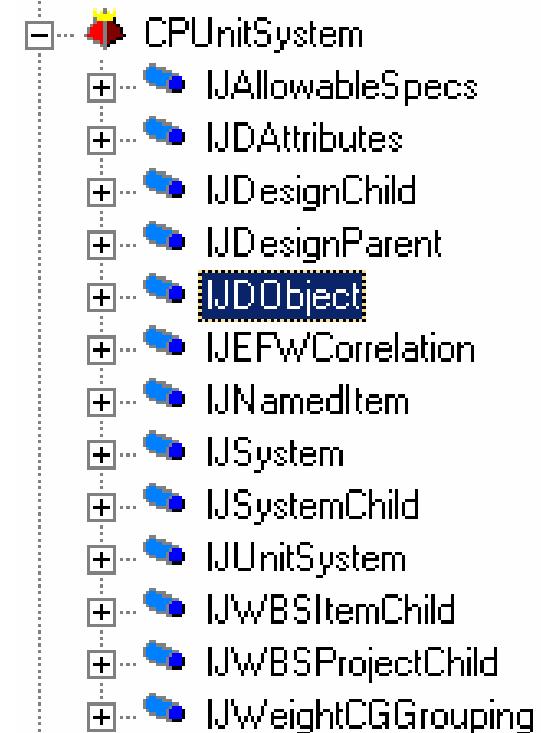
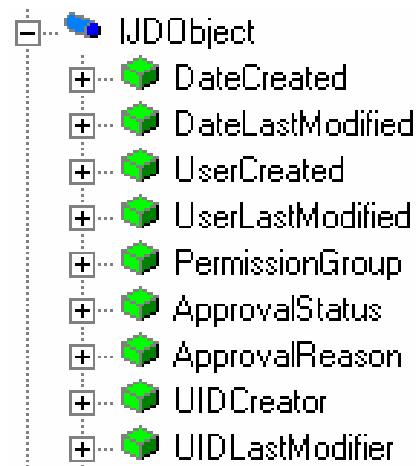
```

	AllowableSpec	PipingSystem
1	1C0031	P-200 lines
2	2C0032	Process Pipelines
3	1C0031	Process Pipelines

# System Entity Data Model

## Generic Object interfaces

- IJDObject interface is a required interface for almost all objects. This interface provides access to the permission group for the object, the status, name of the user who created and last modified the object and the date and time for creation and last modification.
- IJDAttributes interface is required for the system object to support user-defined attributes.



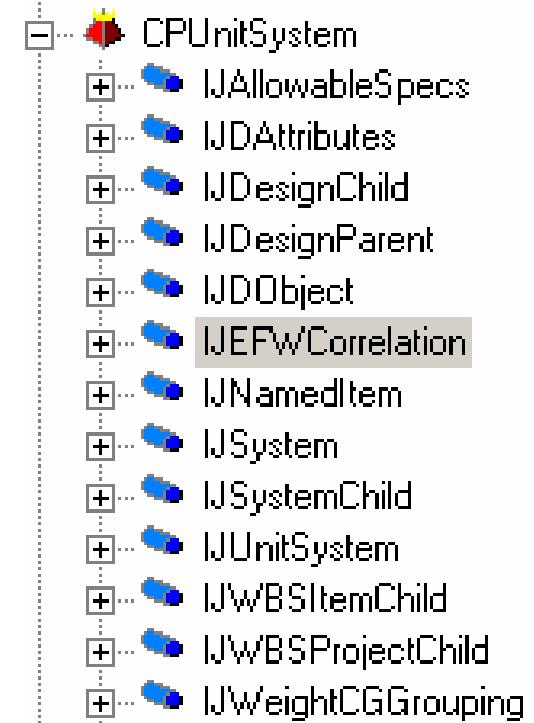
# System Entity Data Model

Example: Query all piping systems and display their permission group

```
select  
x5.ItemName,  
x3.Name,  
x2.DateCreated  
from JSystemChild x1  
Join JNamedItem x5 on x5.oid = x1.oid  
Join JPipingSystem x4 on x4.oid = x1.oid  
Join JDObject x2 on x2.oid = x1.oid  
Join JDPermissionGroup x3 on x3.PermissionGroupID = x2.ConditionID
```

# System Entity Data Model

- IJEFWCorrelation: This interface is required in order to associate the system object to the Engineering Framework Design Basis object and provides the EFW Correlation Properties.
- IJWeightCGGrouping :This IJWeightCGGrouping interface is to manage the weight and center of gravity (Weight&CG) for objects that represent logical groups of parts such as an assembly, or a compartment.



# System Entity Data Model

Example: Query all correlated and non-correlated pipelines

select

```
x3.ItemName as Pipeline,  
x1.CorrelationStatus_ShortValue  
from JEFWCorrelation_CL x1  
Join JPipelineSystem x2 on x2.oid = x1.oid  
Join JNamedItem x3 on x2.oid = x3.oid  
where x1.CorrelationBasis = 0 and x1.CorrelationStatus in (1,2,3)
```

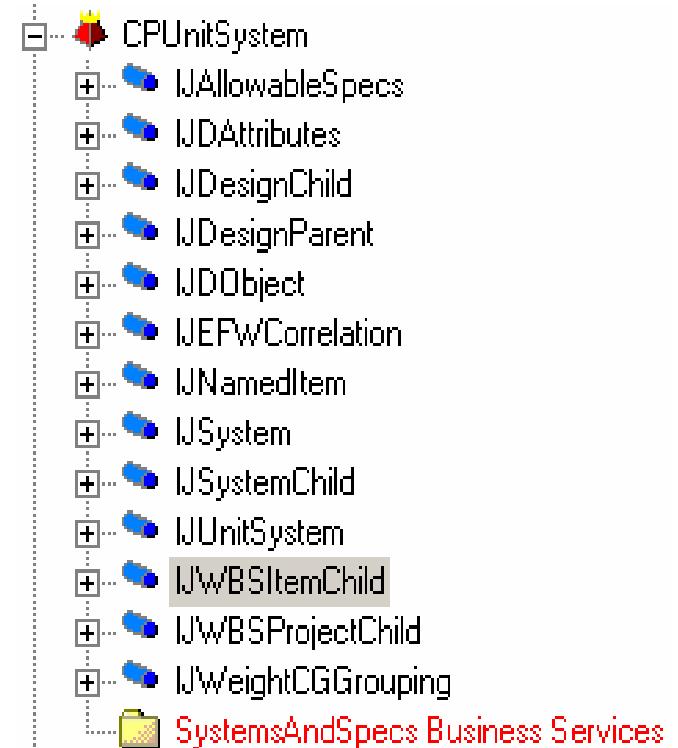
select

```
x3.ItemName as Pipeline,  
x1.CorrelationStatus_ShortValue  
from JEFWCorrelation_CL x1  
Join JPipelineSystem x2 on x2.oid = x1.oid  
Join JNamedItem x3 on x2.oid = x3.oid  
where x1.CorrelationStatus not in (1,2,3)
```

# System Entity Data Model

- IJWBSProjectChild: This interface is used to relate a system to its work breakdown structure project parent by means of the *BelongsToProject* relation.
- IJWBSItemChild :Interface used to relate a system to its work breakdown structure parent by means of the *BelongsToWBSItem* relationship.
- Example: List all systems that belong to a project
 

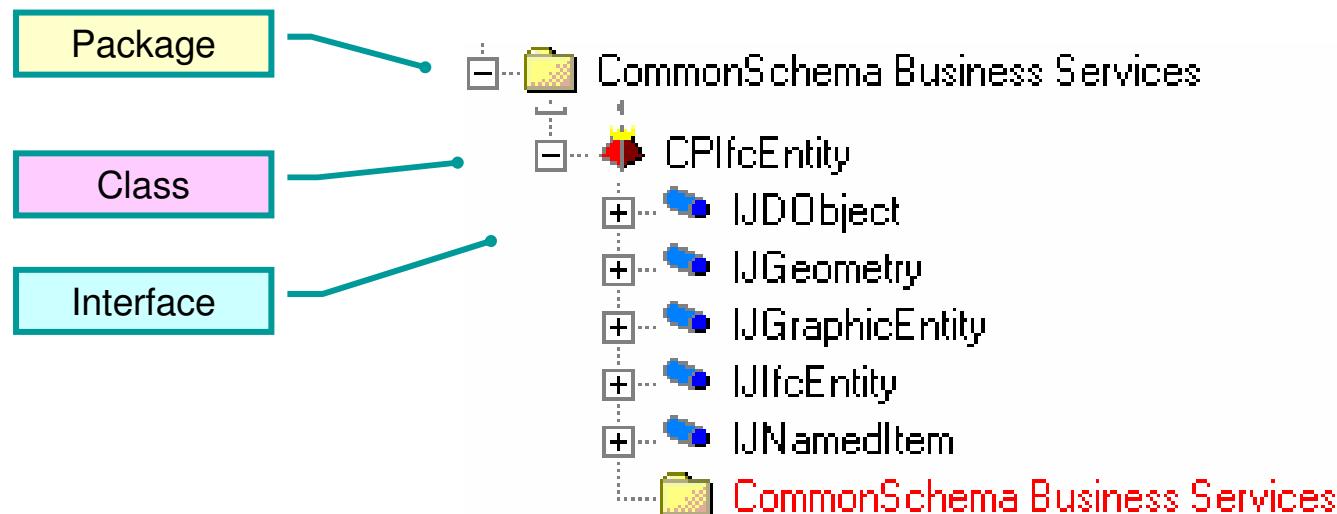
```
select x4.ItemName from
JWBSProjectChild x1
join JSystem x2 on x2.oid = x1.oid
join XBelongsToProject x3 on x3.oidorigin = x1.oid
join JNamedItem x4 on x4.oid = x2.oid
```



# IFC Entity Data Model

**SmartPlant®**  
3D

## CPIfcEntity

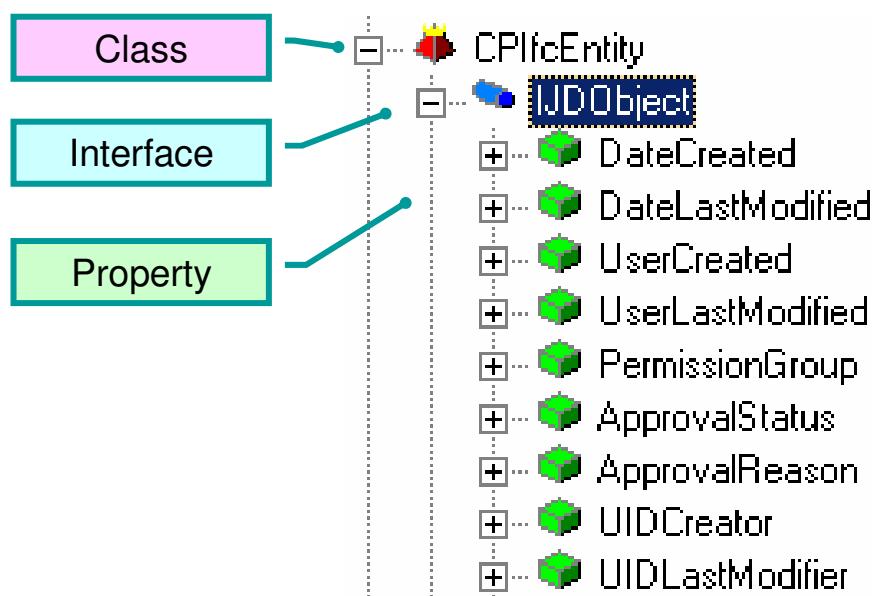


# IFC Entity Data Model



## Generic Object interfaces

IJDObject interface is a required interface for almost all objects. This interface provides access to the permission group for the object, the status, name of the user who created and last modified the object and the date and time for creation and last modification.

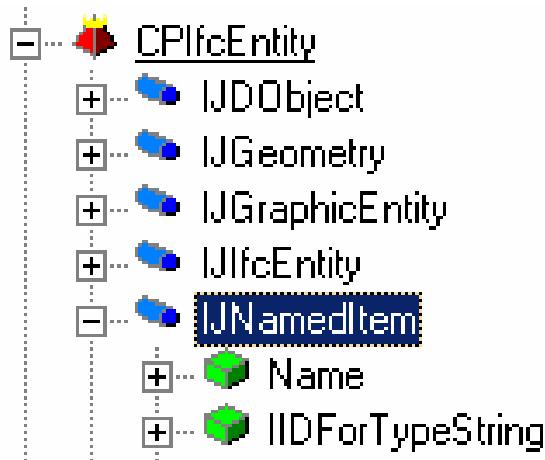


# IFC Entity Data Model



## Generic Object interfaces

IJNamedItem interface provides the name property and a type string for all named objects. This interface is used by any component that needs the name of an object as well as by components that need to display a simple type string for an object.

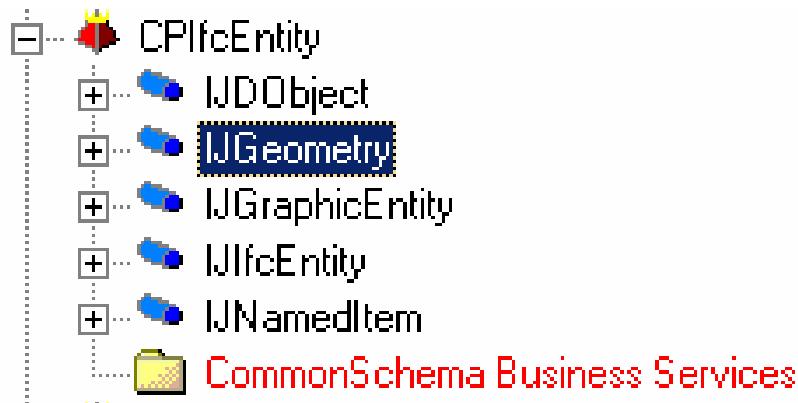


# IFC Entity Data Model



## Geometry interfaces

- IJGeometry interface is required if the BO has geometry.
- IJGraphicEntity interface is required if the BO is displayable

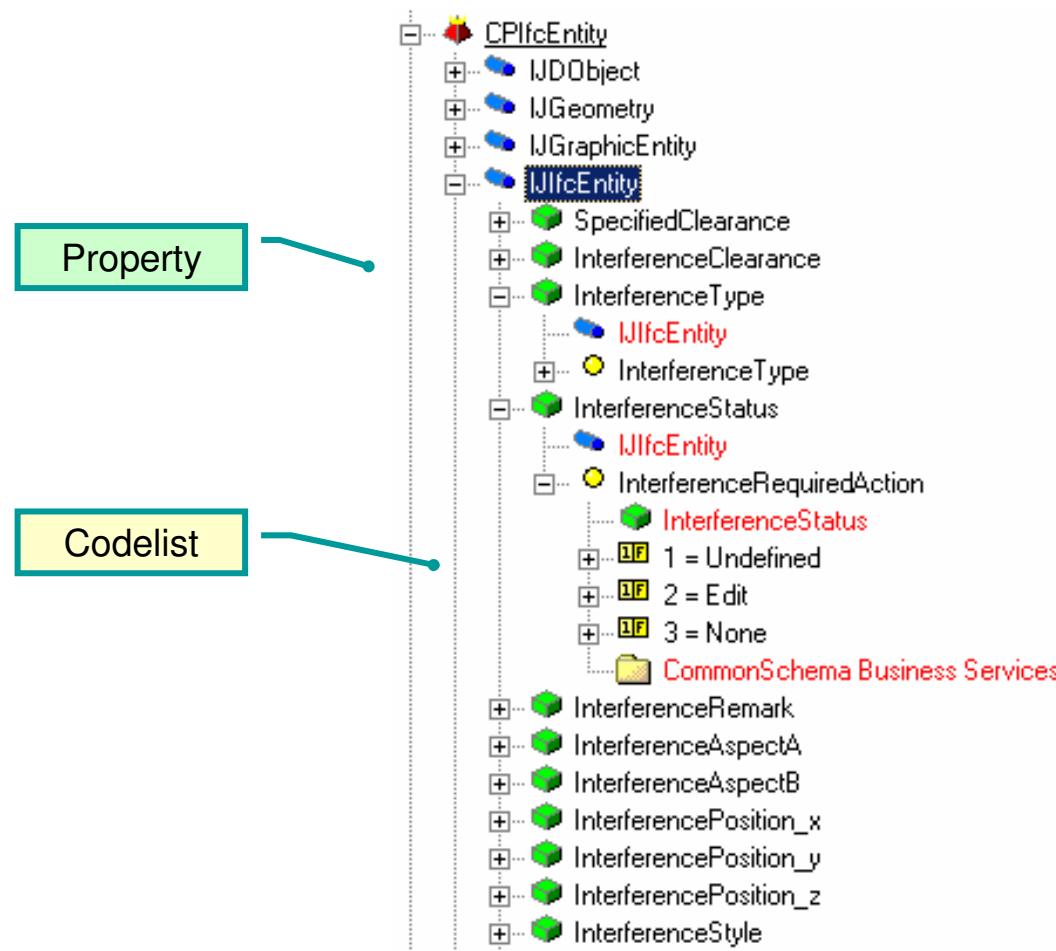


# IFC Entity Data Model



## Interference Interface

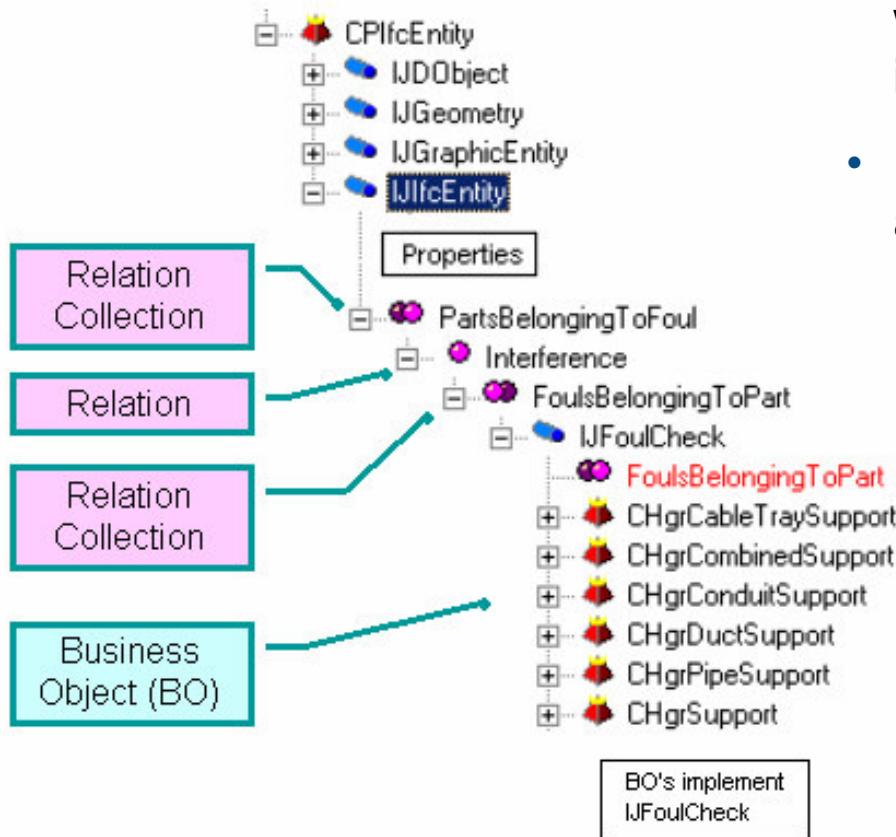
- IJIfcEntity interface keeps track of all the interference properties created by the object.



# IFC Entity Data Model

**SmartPlant®  
3D**

## Interference Relationship



- BO's implement the **IJFoulCheck** interface which is used to identify which objects should be checked for interferences with other object.
- **IJFoulCheck** has a relation with the actual interference object's interface.

# IFC Entity Data Model

**SmartPlant®  
3D**

## Interference Relationship

The screenshot shows the SP3D SQL Repository interface. On the left is a tree view of entities, and on the right is a detailed properties view for a selected entity.

**Properties of Relation: Interference**

Name	Data
Name	Interference
UserName	Object to Interference
DBViewName	XInterference
OID	(30BD2972-BCF0-11D1-A030-080036EF2B03)
RelationGUID	(30BD2972-BCF0-11D1-A030-080036EF2B03)
Properties	[0] - <Intentionally Left Blank>
Ends with Relation	FoulsBelongingToPart
Ends with Relation	PartsBelongingToFoul
Belongs to Pack	CommonSchema Business Services

**Select Properties**

Object type used as the basis for the property identification :

Relationship :

Related object type :

# IFC Entity Data Model



Example: List all the persistent interferences created by equipment objects

Select x1.FoultType as Type,

x2.ItemName as IFCName,

x3.RelationName as Part,

x5.ItemName as EqpName

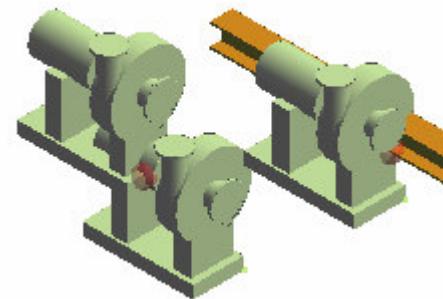
from JIfcEntity x1

JOIN JNamedItem x2 on x2.oid = x1.oid

JOIN XInterference x3 on x3.oidorigin = x1.oid

JOIN JEquipment x4 on x4.oid = x3.oidDestination

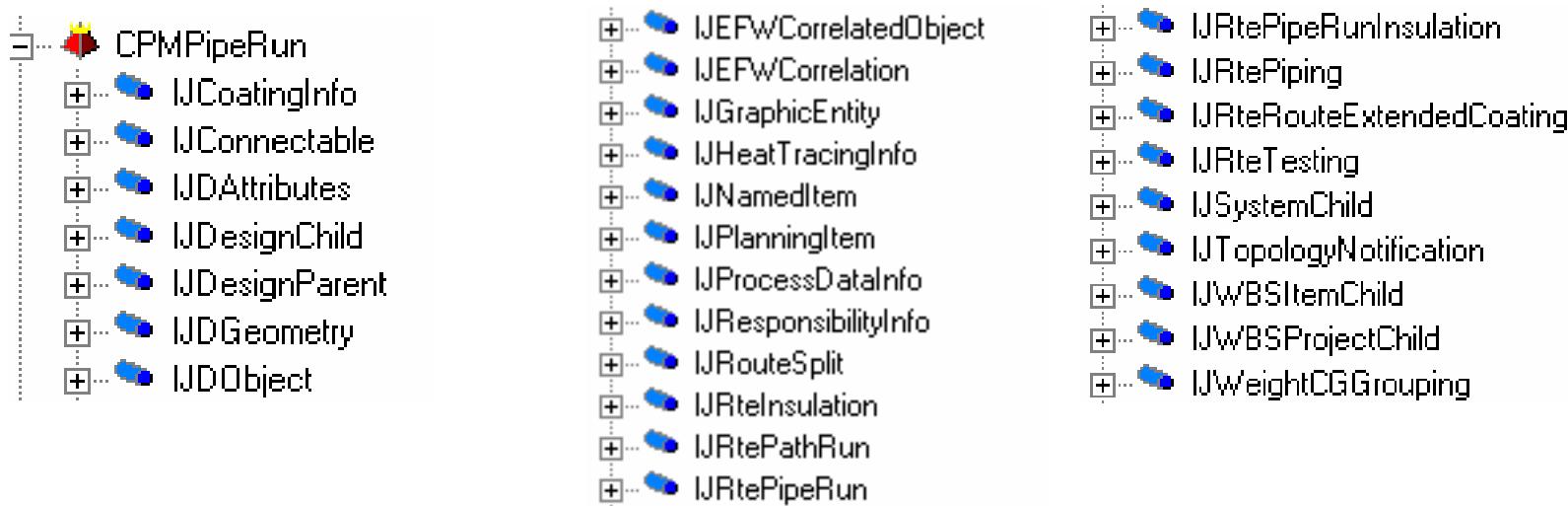
JOIN JNamedItem x5 on x5.oid = x4.oid



	Type	IFCName	Part	EqpName
1	1	I-H-0112-MemberPartPrismatic-H-0106-PUMPO01A_IMP-H-0121	IFC_PartB	PUMPO01A_IMP-H-0121
2	1	I-H-0115-MemberPartPrismatic-H-0107-PUMPO01A_IMP-H-0121	IFC_PartB	PUMPO01A_IMP-H-0121
3	1	I-H-0114-PUMPO01A_IMP-H-0113-PUMPO01A_IMP-H-0114	IFC_PartA	PUMPO01A_IMP-H-0113
4	1	I-H-0114-PUMPO01A_IMP-H-0113-PUMPO01A_IMP-H-0114	IFC_PartB	PUMPO01A_IMP-H-0114

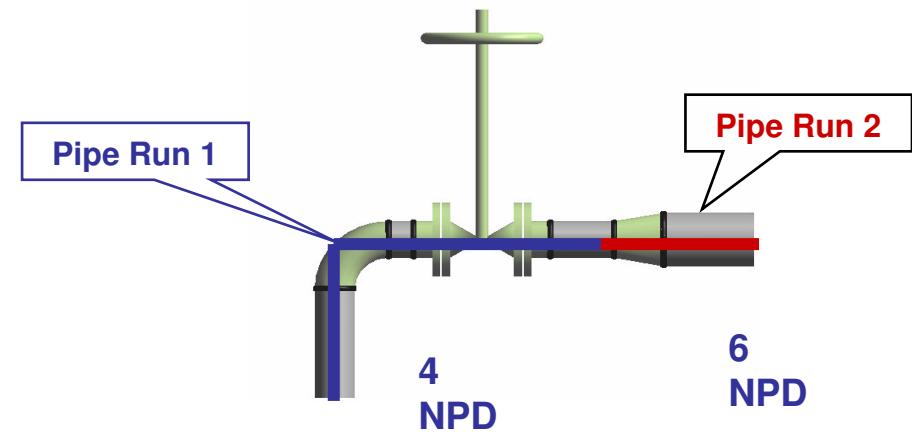
# Piping Entities Data Model

## PipeRun Interfaces



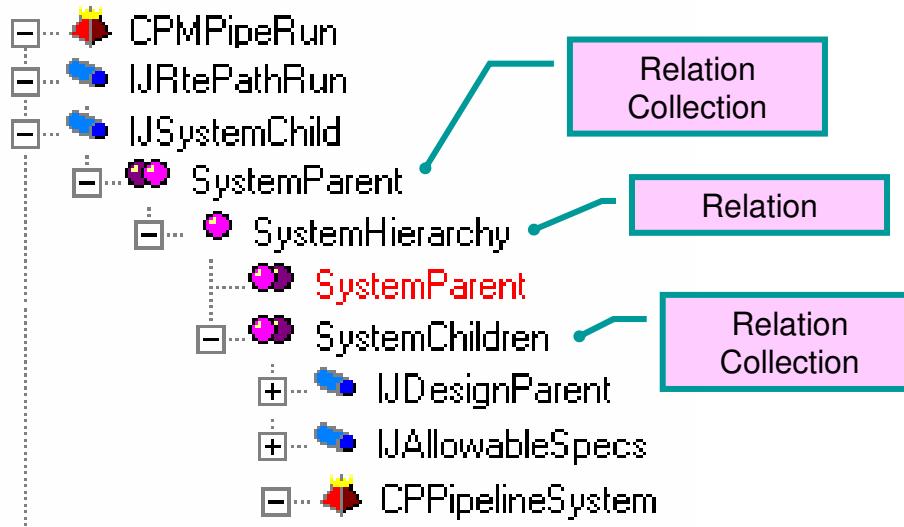
A PipeRun identifies one or more path features that share a common pipe specification, flow direction, size, temperature, pressure, etc....

One or more runs make up a pipeline.



# Piping Entities Data Model

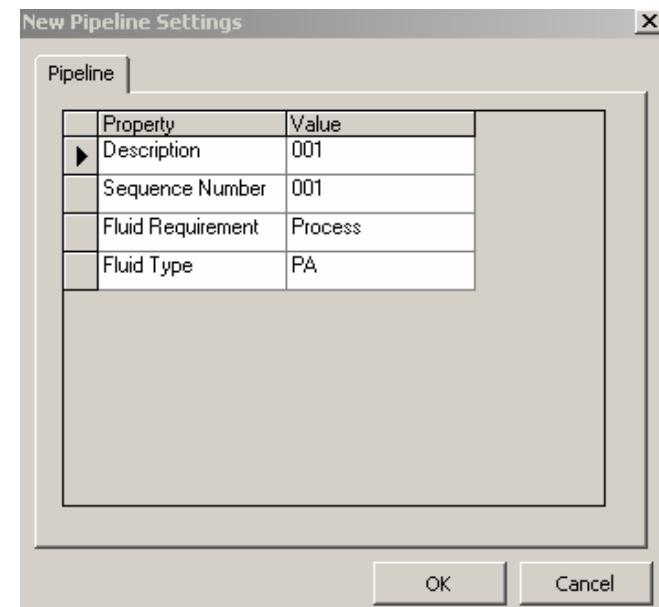
## PipeRun and Pipeline Relationship



Pipeline is a high-level grouping of Pipe Runs that are created in System and Spec Task environment.

## System Hierarchy Relation

The system hierarchy relationship is used to specify that a run (IJSystemChild) has a parent system (JDesignParent)



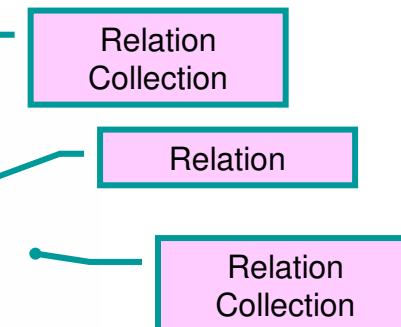
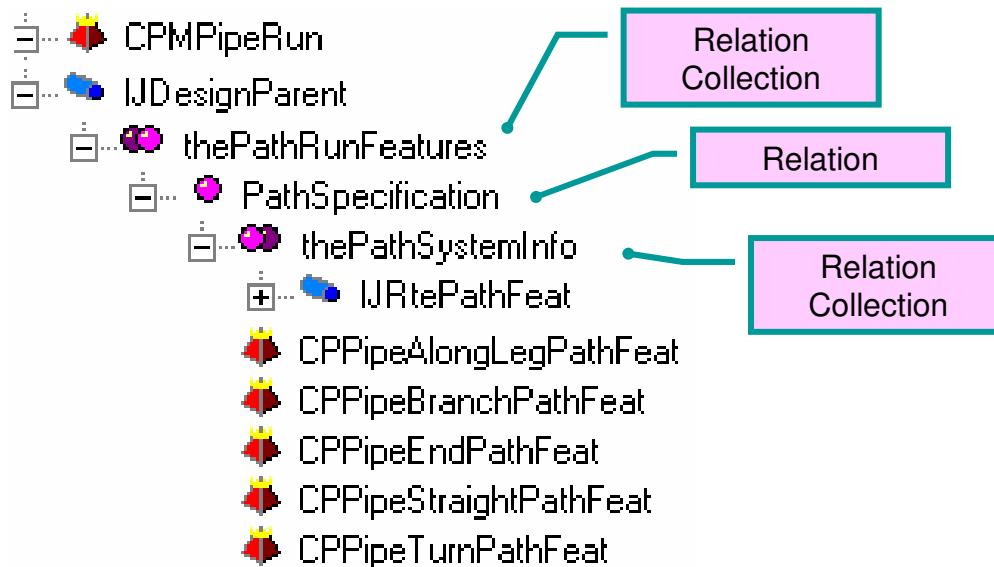
# System Entity Data Model

Example: returns runs located in piping systems

```
select *
from JRtePipeRun x1
join XSystemHierarchy x2 on x2.oiddestination = x1.oid
where x2.oidorigin in
    (select x3.oid from JPipelineSystem x3
     join XSystemHierarchy x4 on x4.oiddestination = x3.oid
     join JPipingSystem x5 on x5.oid = x4.oidorigin)
```

# Piping Entities Data Model

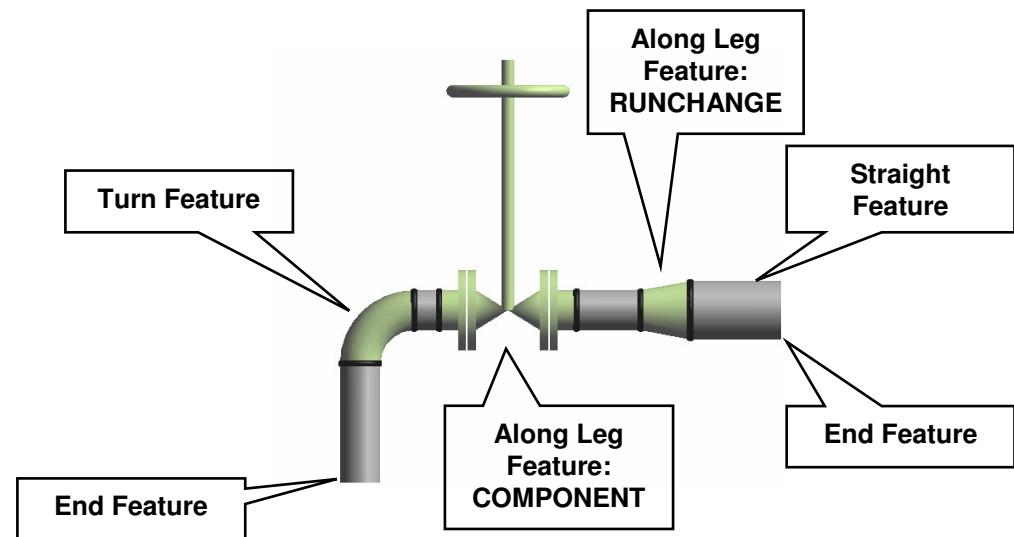
## PipeRun and Features Relationship



Features define the geometry path of the run and your design intent that occurs along the path. When you route a pipe run, you place features.

## Path Specification Relation

All path features are related to a run. This relationship allows the path feature to retrieve the specification and rules that it needs to meet from the run.



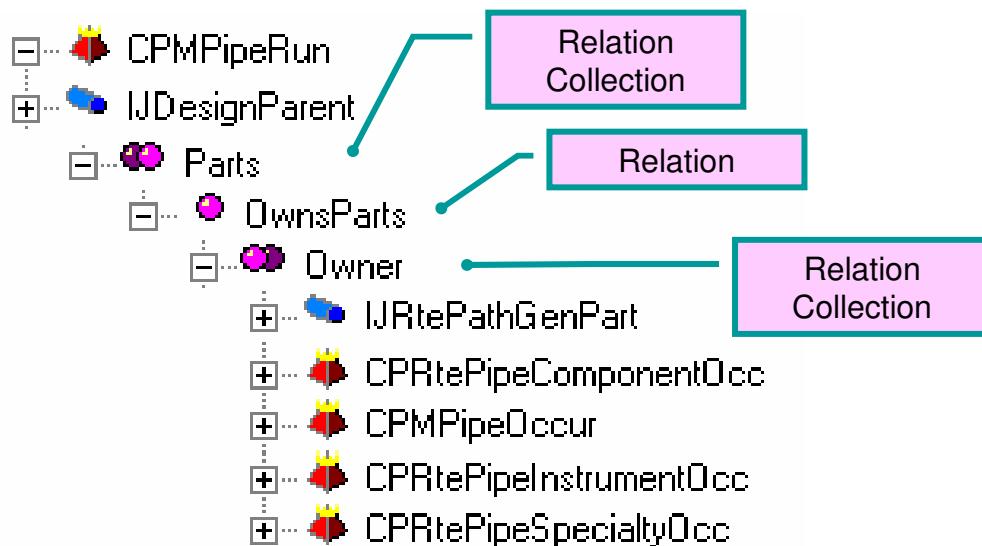
# System Entity Data Model

Example: Select empty runs:

```
select distinct x1.oid from JRtePipeRun x1  
WHERE NOT EXISTS  
(select 1 from XPathSpecification x2 where x1.oid = x2.oidorigin)
```

# Piping Entities Data Model

## PipeRun and Parts Relationship

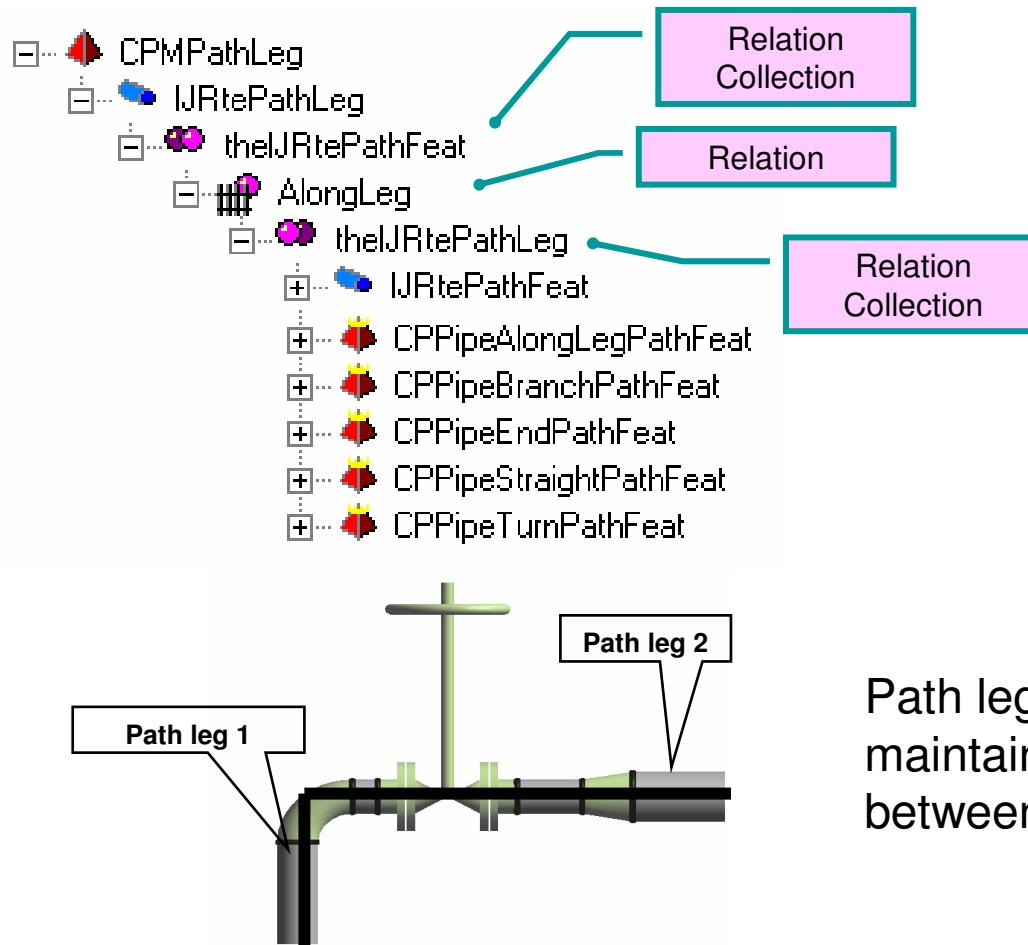


### Owes Parts Relation

The OwnsParts relationship is used to specify that a part occurrence is a child of a run.

# Piping Entities Data Model

## Pathleg and Path Feature Relationship



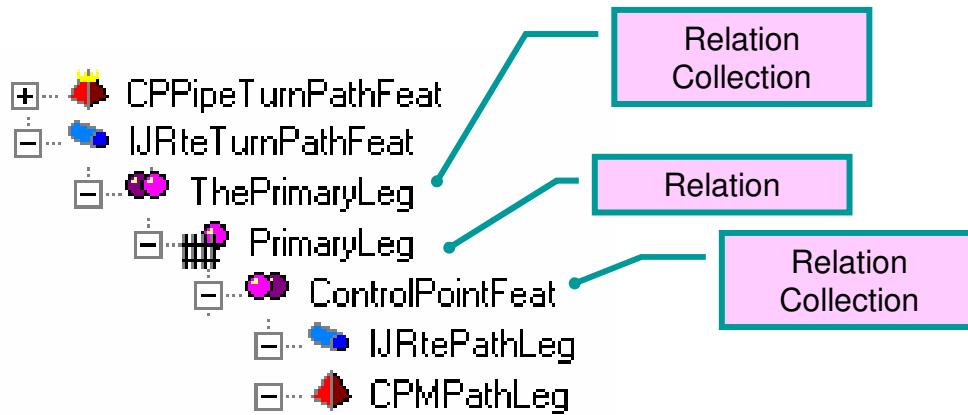
### Along Leg Relation

All path features are related to one path leg. (Turn path features are related to two path legs). The Along Leg relationship is ordered so that from the path leg, each feature can be identified in the order in which it occurs along the leg's path. This defines the logical connectivity.

Path leg is a section of a run maintaining one general direction between turns, branches and ends.

# Piping Entities Data Model

## Pathleg and Path Feature Relationship

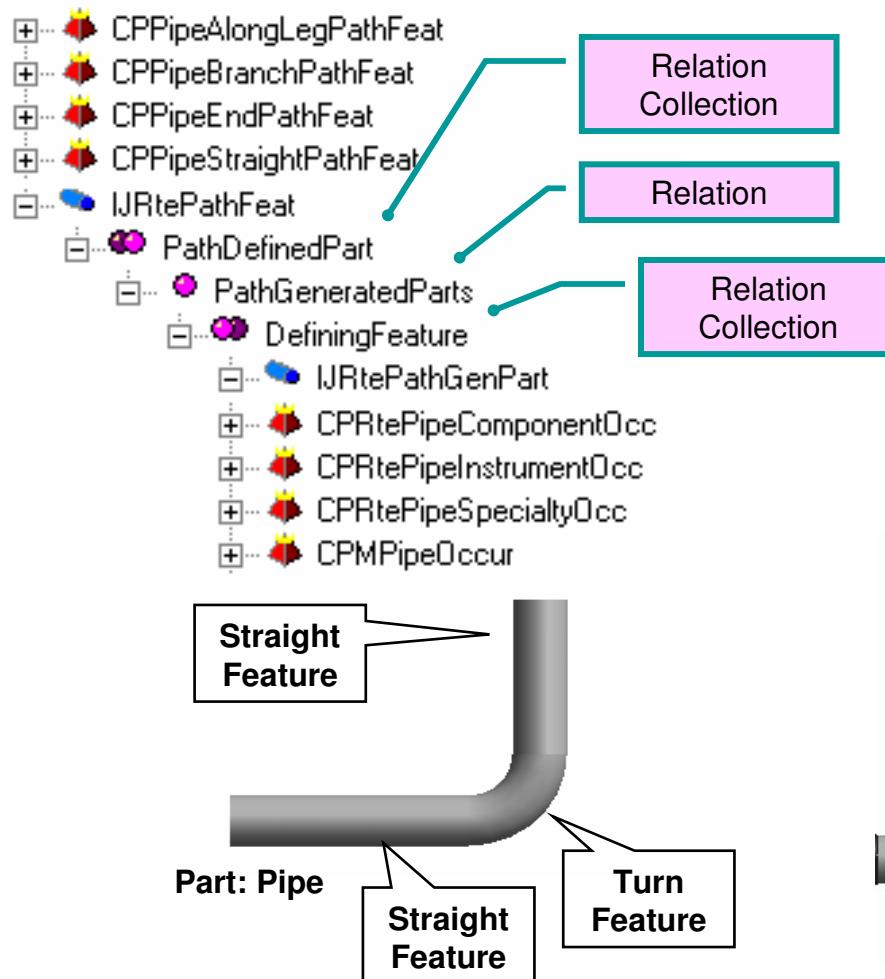


### Primary Leg Relation

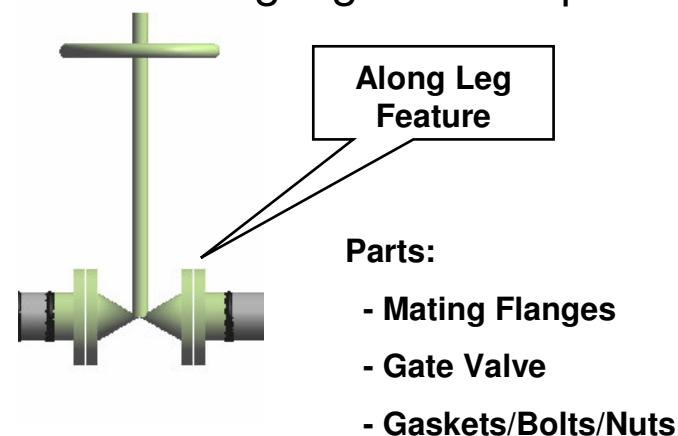
Turn path features are related to two legs in the Along Leg relationship. They also have a Primary Leg relationship that identifies which of the legs begins the turn path feature geometric definition. The turn path feature start position is on the "primary" leg. The end position is on the other leg.

# Piping Entities Data Model

## Feature and Part Relationship

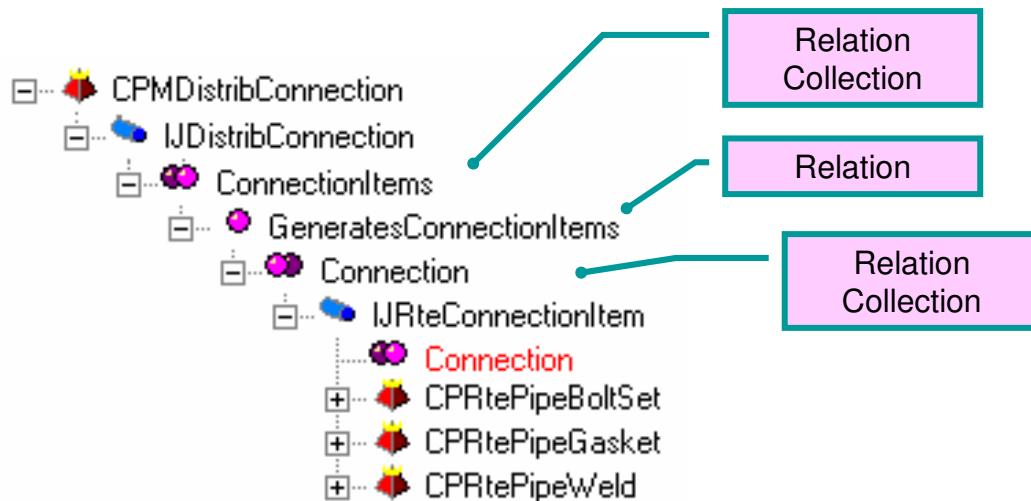


**PathGeneratedParts Relation**  
 This relationship identifies the part occurrences that are generated from a given feature based on specifications. For a feature representing a component, there may be one or more generated parts. For a pipe, there may be multiple features associated with a single generated part.



# Piping Entities Data Model

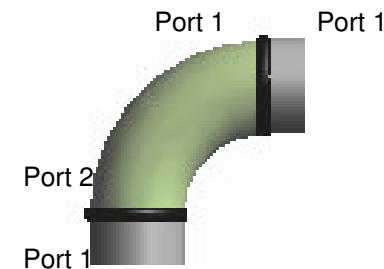
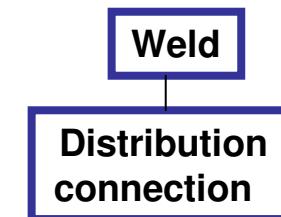
## Distribution Connection Relationship



**GeneratesConnectionItems**

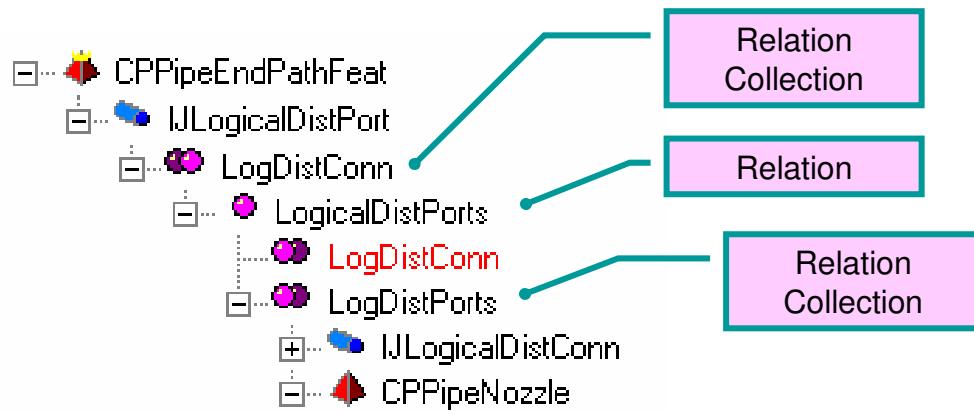
**Relation**

This relationship identifies the connection items that are generated from a given distribution connection.



# Piping Entities Data Model

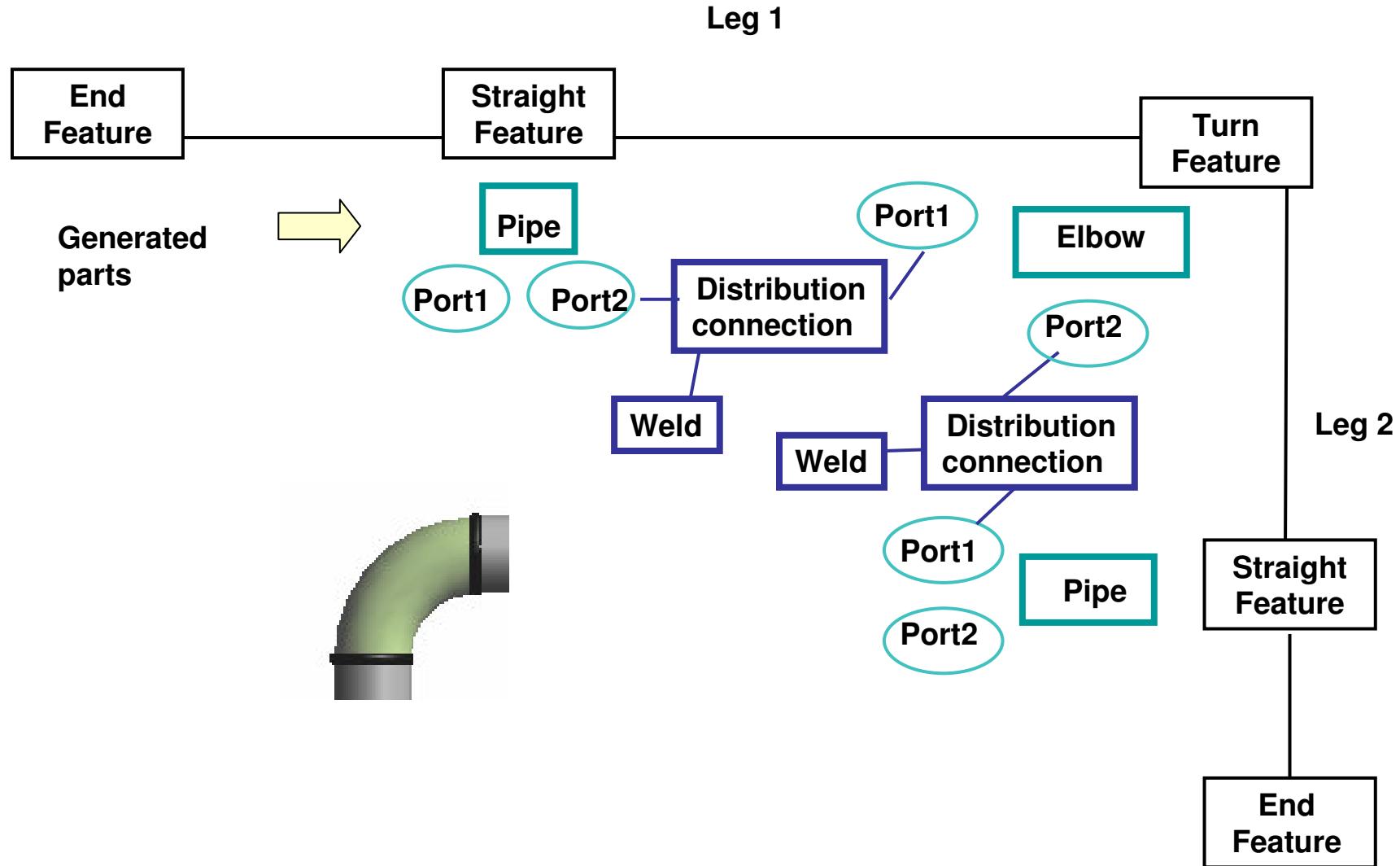
## Logical Distribution Connection Relationship



### LogicalDistPorts Relation

This relationship identifies the connection between an equipment nozzle and the end feature of a path leg.

# Feature Based Model



# Piping Data Model

Example: Query all piping component with part data

select

\*

FROM JRteCompOccur rco

JOIN XMadeFrom rmf ON (rmf.OidOrigin = rco.Oid)

JOIN JDPipeComponent pco ON (pco.Oid = rmf.OidDestination)

JOIN JDPipeComponent2 pco2 ON (pco2.Oid = rmf.OidDestination)

JOIN XDfnMatlCntrlDataForComponent xdmcdfc ON (xdmcdfc.OidOrigin =  
rmf.OidDestination)

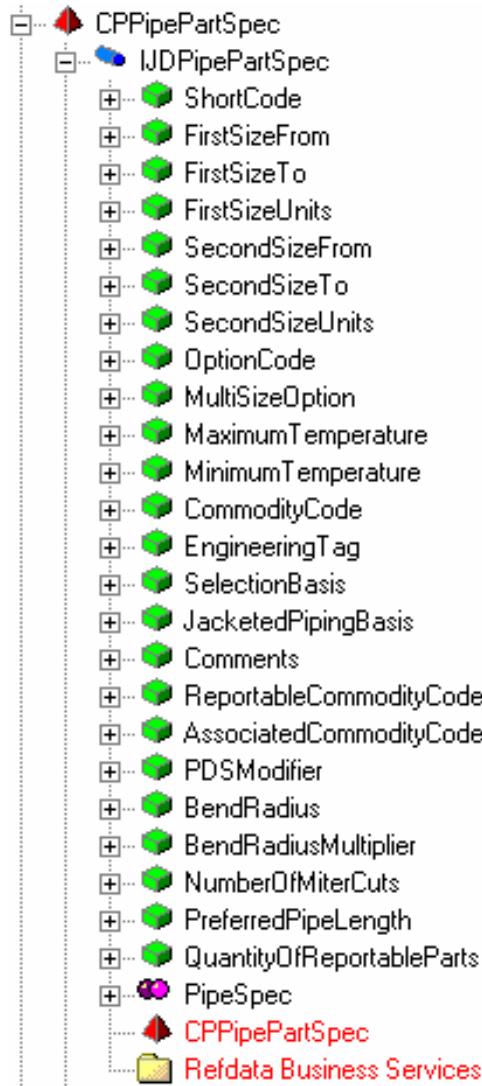
JOIN JGenericMaterialControlData jgmcd ON (jgmcd.Oid = xdmcdfc.OidDestination)

JOIN XOwnsParts rop ON (rop.oiddestination = rco.oid)

JOIN XPipeRunUsesSpec rpus ON (rpus.OidOrigin = rop.OidOrigin)

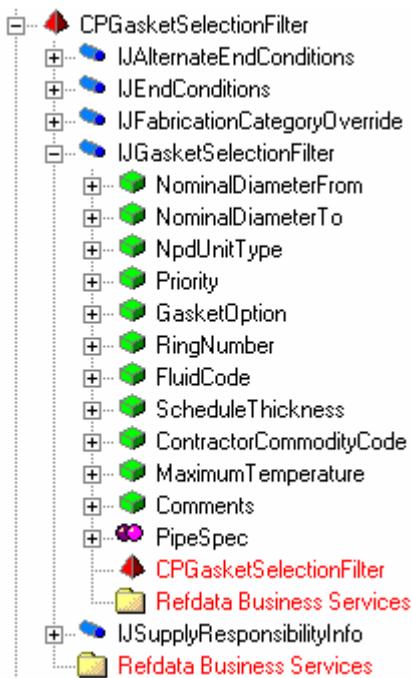
JOIN JDPipeSpec jps\_RUN ON (jps\_RUN.oid = rpus.oiddestination)

# Reference Data Entity Data Model



Select x3.SpecName, x1.ShortCode, x1.FirstSizeFrom,  
x1.FirstSizeTo, x1.FirstSizeUnits, x1.SecondSizeFrom,  
x1.SecondSizeTo, x1.SecondSizeUnits, x1.OptionCode,  
x1.SelectionBasis, x1.MultiSizeOption,  
x1.MaximumTemperature, x1.MinimumTemperature,  
x1.CommodityCode, x1.EngineeringTag,  
x6.ShortStringValue as Sch1, x7.ShortStringValue as  
Sch2, x1.ReportableCommodityCode,  
x1.QuantityOfReportableParts, x1.BendRadius,  
x1.BendRadiusMultiplier from JDPipePartSpec x1 JOIN  
JFirstSizeSchedule x4 on x4.oid = x1.oid JOIN  
CL\_ScheduleThickness x6 on x6.ValueID =  
x4.FirstSizeSchedule JOIN JSecondSizeSchedule x5 on  
x5.oid = x1.oid JOIN CL\_ScheduleThickness x7 on  
x7.ValueID = x5.SecondSizeSchedule JOIN  
XPipeSpecContainsPartSpecs x2 on (x2.oidDestination  
= x1.oid) JOIN JDPipeSpec x3 on (x3.oid = x2.oidOrigin)  
Where x3.SpecName = '1C0031' order by x1.ShortCode

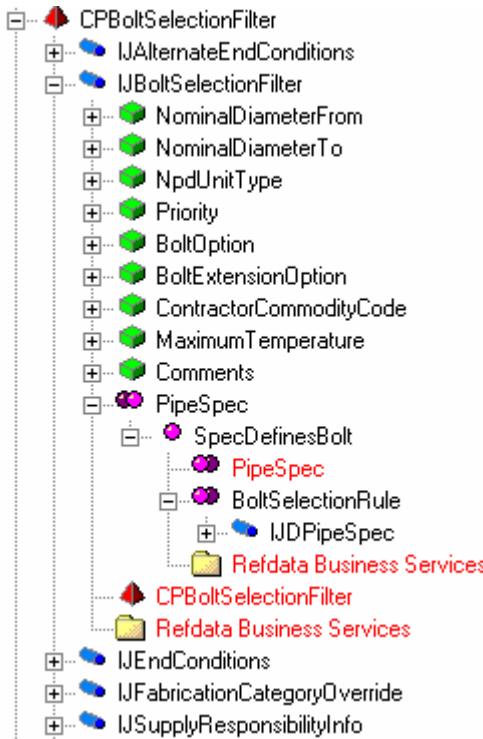
# Reference Data Entity Data Model



```

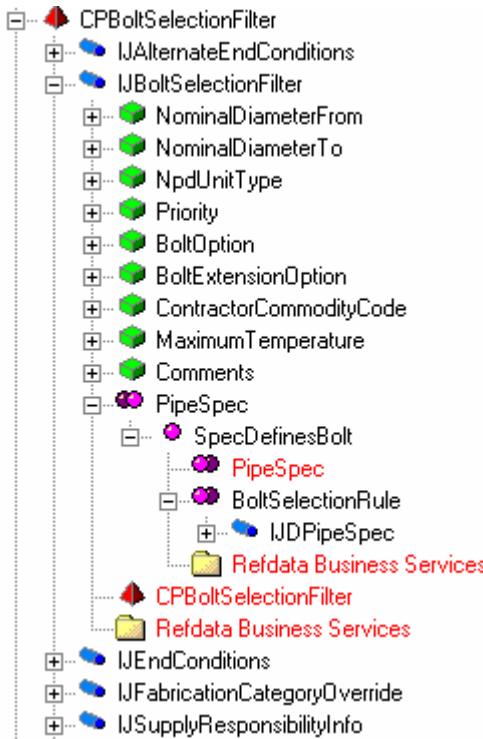
select x1.NominalDiameterFrom, x1.NominalDiameterTo,
x1.NpdUnitType, x4.EndPreparation, x6.ShortStringValue
as PreRating1, x4.EndStandard,
x5.AlternateEndPreparation, x7.ShortStringValue as
PreRating2, x5.AlternateEndStandard, x1.Priority,
x1.GasketOption, x1.RingNumber, x1.FluidCode,
x1.ScheduleThickness, x1.ContractorCommodityCode,
x1.MaximumTemperature from JGasketSelectionFilter x1
JOIN JEndConditions x4 on x4.oid = x1.oid JOIN
CL_PressureRating x6 on x6.ValueID = x4.PressureRating
JOIN JAlternateEndConditions x5 on x5.oid = x1.oid JOIN
CL_PressureRating x7 on x7.ValueID =
x5.AlternatePressureRating JOIN XSpecDefinesGasket x2
on x2.oiddestination = x1.oid JOIN JDPipeSpec x3 on
x3.oid = x2.oidorigin where x3.specname = '1c0031'
  
```

# Reference Data Entity Data Model



```
select x1.NominalDiameterFrom, x1.NominalDiameterTo,
x1.NpdUnitType, x4.EndPreparation, x6.ShortStringValue
as PreRating1, x4.EndStandard,
x5.AlternateEndPreparation, x7.ShortStringValue as
PreRating2, x5.AlternateEndStandard, x1.Priority,
x1.BoltOption, x1.BoltExtensionOption,
x1.ContractorCommodityCode, x1.MaximumTemperature
from JBoltSelectionFilter x1 JOIN JEndConditions x4 on
x4.oid = x1.oid JOIN CL_PressureRating x6 on x6.ValueID
= x4.PressureRating JOIN JAlternateEndConditions x5 on
x5.oid = x1.oid JOIN CL_PressureRating x7 on x7.ValueID
= x5.AlternatePressureRating JOIN XSpecDefinesBolt x2
on x2.oiddestination = x1.oid JOIN JDPipeSpec x3 on
x3.oid = x2.oidorigin where x3.specname = '1c0031'
```

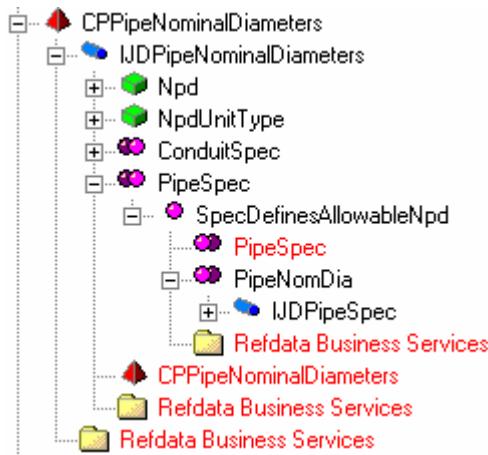
# Reference Data Entity Data Model



```
select x1.NominalDiameterFrom, x1.NominalDiameterTo,
x1.NpdUnitType, x4.EndPreparation, x6.ShortStringValue
as PreRating1, x4.EndStandard,
x5.AlternateEndPreparation, x7.ShortStringValue as
PreRating2, x5.AlternateEndStandard, x1.Priority,
x1.BoltOption, x1.BoltExtensionOption,
x1.ContractorCommodityCode, x1.MaximumTemperature
from JBoltSelectionFilter x1 JOIN JEndConditions x4 on
x4.oid = x1.oid JOIN CL_PressureRating x6 on x6.ValueID
= x4.PressureRating JOIN JAlternateEndConditions x5 on
x5.oid = x1.oid JOIN CL_PressureRating x7 on x7.ValueID
= x5.AlternatePressureRating JOIN XSpecDefinesBolt x2
on x2.oiddestination = x1.oid JOIN JDPipeSpec x3 on
x3.oid = x2.oidorigin where x3.specname = '1c0031'
```

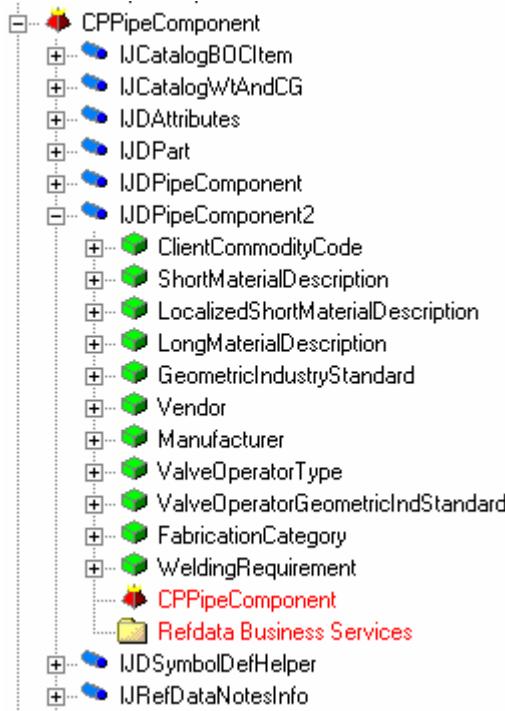
# Reference Data Entity Data Model

**SmartPlant®  
3D**



```
select x1.Npd, x1.NpdUnitType from JDPipeNominalDiameters
x1 JOIN XSpecDefinesAllowableNpd x2 on
(x2.oidDestination = x1.oid) JOIN JDPipeSpec x3 on
(x3.oid = x2.oidOrigin) Where x3.SpecName = '1C0031'
```

# Reference Data Entity Data Model



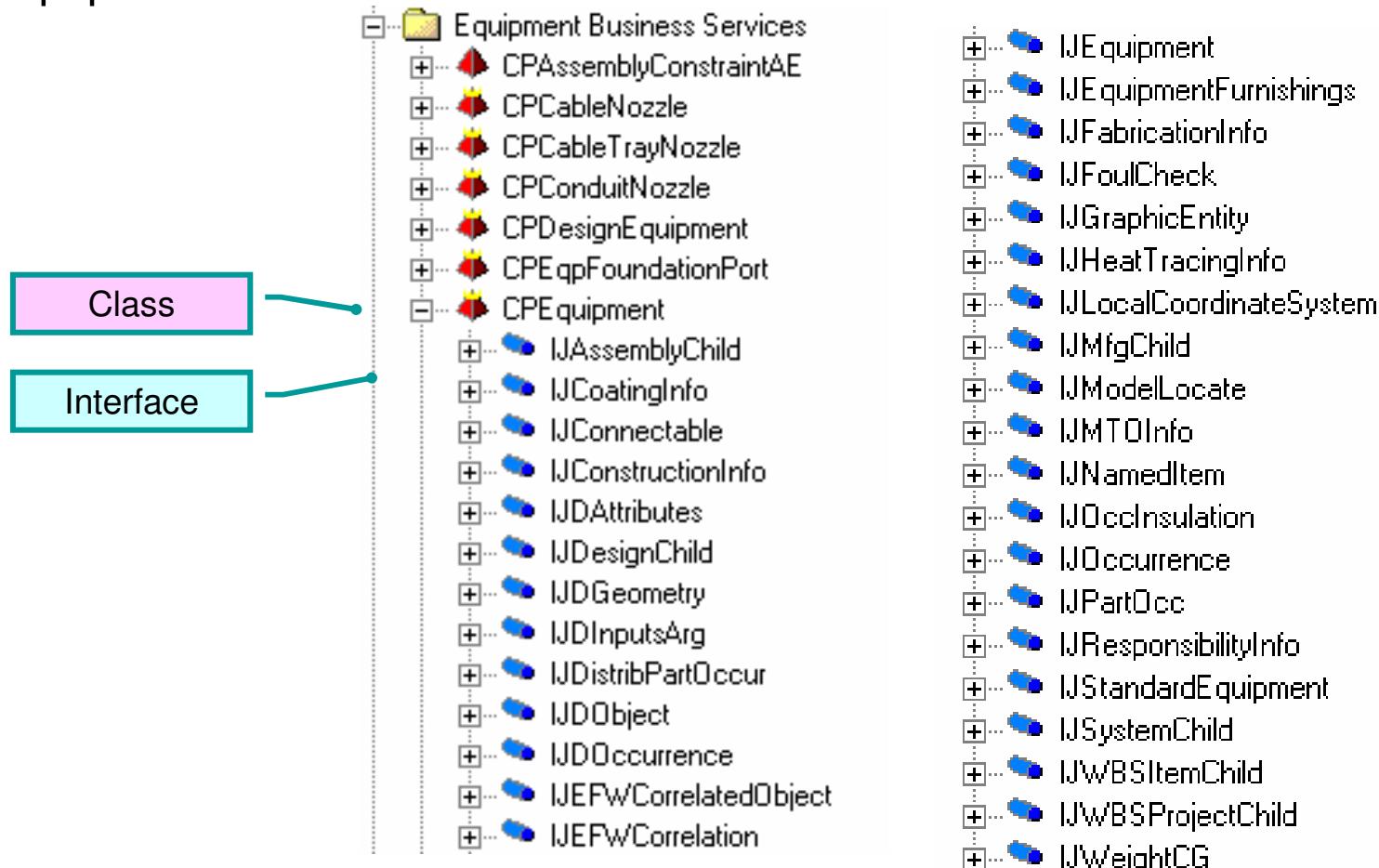
```

select x1.IndustryCommodityCode, x1.FirstSizeSchedule,
x1.PrimarySize, x1.PriSizeNPDUnits, x1.SecondarySize,
x1.SecSizeNPDUnits, x1.CommodityType,
x1.MaterialsGrade, x3.PortIndex, x3.Npd, x3.NpdUnitType,
x3.EndPrep, x3.PressureRating, x3.EndStandard,
x3.ScheduleThickness from JD PipeComponent x1 JOIN
XPartClassContainsParts x4 on ( x4.oiddestination =
x1.oid) JOIN JDPartClass x5 on (x5.oid = x4.oidorigin)
JOIN JDPart x6 on (x6.oid = x1.oid) JOIN
XPartContainsNozzles x2 on (x2.oidOrigin = x1.oid) JOIN
JCatalogPipePort x3 on (x3.oid = x2.oidDestination) where
x5.partclasstype like 'PipeComponentClass' and x5.Name
like 'BlindFlange' and x1.IndustryCommodityCode like
'%MAAAMABZZAADABQZZUS%' order by
x1.IndustryCommodityCode
  
```

# Equipment Entity Data Model

**SmartPlant®**  
3D

## CPEquipment



# Equipment Entity Data Model

## Generic Object interfaces

- The IJDObject interface is a required interface for almost all objects. This interface provides access to the permission group for the object, the status, name of the user who created and last modified the object and the date and time for creation and last modification.
- The IJDAtributes interface is required for the equipment object to support user-defined attributes.
- The IJNamedItem interface provides the name property, a type string and an icon for all named objects.

# Equipment Entity Data Model

## Display/Locate interfaces

- IJGeometry: This interface is required if the BO has geometry.
- IJGraphicEntity: This interface is required if the BO is displayable.
- IJModelLocate: This interface is used to allow client components to graphically locate all of the graphical components of a standard equipment object.
- IJOccurrence: This interface provides a property that allows the matrix to be retrieved or set. A matrix defines the origin and orientation of an equipment object.

# Equipment Entity Data Model

## Equipment specific interfaces

- **IJEquipment:** This interface is used to provide a common interface for both standard equipment and design equipment. This interface also provides methods to get and set the position and orientation of the equipment.
- **IJStandardEquipment:** This interface is used to type only standard equipment that are required to show up under the Business Object Classification.
- **IJEquipmentFurnishings:** This interface is used to type all business objects that are required to show up under the Equipment&Furnishings node.

# Equipment Entity Data Model



- **IJWeightCG:** This interface provides both the wet and dry weights and center of gravity for an object.
- **IJFoulCheck:** The CPEquipment class implements the IJFoulCheck interface which is used to identify which objects should be checked for interferences with other object.
- **IJDesignParent or IJDesignChild:** In order for an object to participate in the System Hierarchy, it must implement either IJDesignParent or IJDesignChild and establish a relationship to a design parent.
- **IJMfgParent and IJAssemblyChild:** Standard Equipment objects can also participate in the Assembly hierarchy. As a part, it can be associated to an Assembly parent object using the Assembly relationship that is established between IJMfgParent and IJAssemblyChild. This relationship allows the equipment part to be associated to one assembly parent object.

# Equipment Entity Data Model



- IJCoatingInfo: This interface provides code-listed properties to describe the coating properties for SP3D parts. These include coating type description and coating color.
- IJConstructionInfo: This interface provides information on the construction status of a part. The status indicates whether the part is new, existing, future, etc.
- IJFabricationInfo: This interface provides information on who is responsible for the fabrication of the part.

# Equipment Entity Data Model



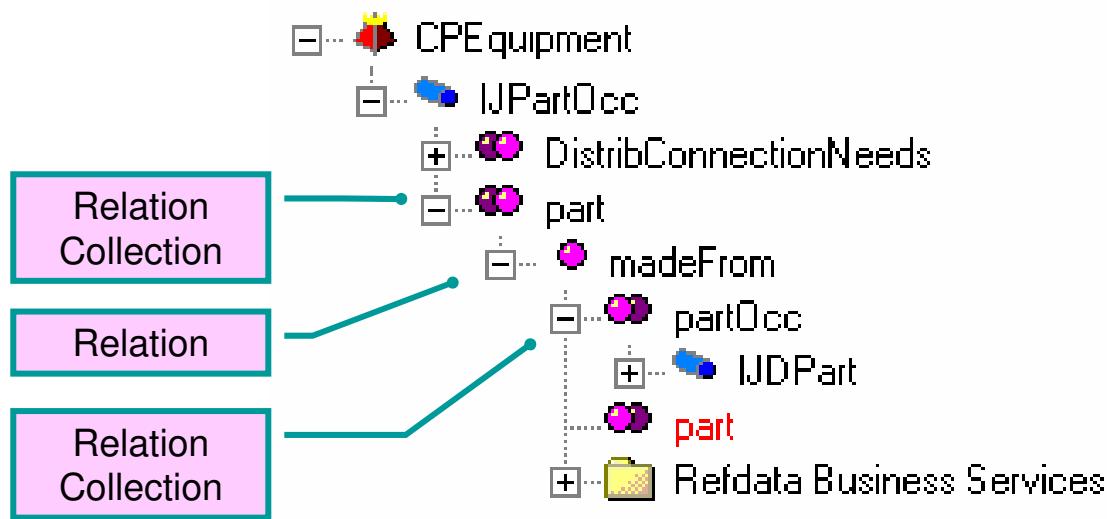
- IJDistribPartOccur: This interface is used to establish a relationship to the distribution ports.
- IJOcclInsulation: This interface is used to define the purpose, material and thickness for insulation for the standard equipment.
- IJConnectable: This interface is used to return a set of equipment foundation ports.

# Equipment Entity Data Model

**SmartPlant®  
3D**

## MadeFrom Relationship

- The part occurrence is related to the part with a relationship named madeFrom.



# Equipment Entity Data Model

Relationships between objects and interfaces

**Related Object <- Relationship**

**Equipment Part <- Part to Catalog Definition**

**WBS Projects <- Object to Project**

**Pipe Port <- Object to Distribution Port**

**Interference <- Object to Interference**

**Equipment System <- System to Equipment**

**Object to Note <- General Notes**

**To Do records <- Object to the To Do List entry**

