

Tagging in Jord

DESIGN DOCUMENT COVER SHEET	
Software Name:	Target Software Version:
Requirement Name: Blues Project	Target Release Date:

FUNCTIONAL REQUIREMENT REVIEW & APPROVAL	
Submitted By: Keith Willshaw	Date: 09 Jan 2013
Approved By: (Dev. Management) Date	Date:

[illegible]

1.0 Table of Contents

1.0	Table of Contents	2
2.0	Associated Documents	3
3.0	Purpose of Document	3
4.0	Definitions	4
	4.1 The P&ID Drawing	4
	4.2 Selected Fragment	4
5.0	Business Objects	4
	5.1 Pumps	5
6.0	Applying the Jord Mapping Methodology	6
	6.1 Selecting my Object Of Interest	6
	6.2 Does the Class Exist	7
	6.3 Declare the Class	7
	6.4 Is the Class Related to another Class with its own information	8
	6.5 Identification	9
7.0	Applying the Jord Whole Life Individual Method	11
8.0	Conclusions	12

2.0 Associated Documents

The Jord Mapping Methodology

<http://www.infowebml.ws/mapping-methodology/index.htm>

The Equipment List and Data identified as being key deliverables for a P&ID by the MIMOSA OGI Pilot SIG available at

<http://www.mimosa.org/?q=webfm>

Under the OGI Pilot Team Utilities >files>Worley Parsons folder

The Jord Endpoint at

<http://posccaesar.org/endpoint/#>

3.0 Purpose of Document

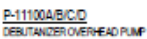
The document is intended to address the following issues

To model a tag name for an object on Mimosa/OGI Pilot project P&ID using Classes, Properties and Templates using the Class methodology outlined in the Jord Mapping Methodology

4.1 The P&ID Drawing

<http://www.mimosa.org/?q=webfm>

The Selected Fragment is defined in the diagram below. Individual sections of this fragment will be analyzed in detail in later sections



Business objects typically represent either single plant assets or collections of such assets. An example of a collection of physical assets is an Instrument Loop. When we transfer data from one system to another we need to specify what the Object is, list its properties, relationships and any constraints. The standard being used in the Plant Industry is ISO 15926 which has several parts which help us define the method by which we accomplish this task.

P-11100A
P-11100B
P-11100C
P-11100D

5.1 Pumps

In ISO 15926 a pump is defined as a physical object that is a driven piece of equipment in which energy is either constantly or periodically added to an amount of pumped liquid in order to increase the pressure required for the process in which the pump is in operation.

The Data modelled in the sample data fragment is as follows:

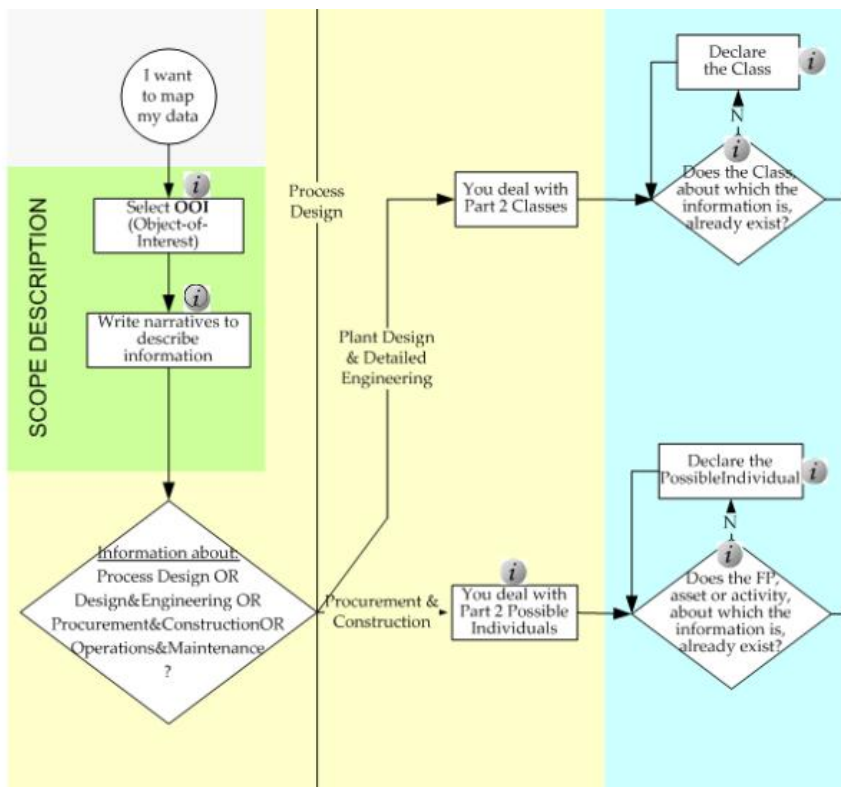
EQUIPMENT LIST										
Equi. No.	Equi. Name	Type	P&ID Name	Const Status	Materials of Constn	Piping Materials Class	Heat Tracing Reqmt	Heat Tracing Temp	Insuln. Purpse	Insuln Thick
P-11100A	DEBUTANIZER OVERHEAD PUMP	Horizontal Centrifugal Pump	11-004	New					N	
P-11100B	DEBUTANIZER OVERHEAD PUMP	Horizontal Centrifugal Pump	11-004	New					N	
P-11100C	DEBUTANIZER OVERHEAD PUMP	Horizontal Centrifugal Pump	11-004	New					N	
P-11100D	DEBUTANIZER OVERHEAD PUMP	Horizontal Centrifugal Pump	11-004	New					N	

6.0 Applying the Jord Mapping Methodology

Lets start with the Tag Number as the first thing we need to do is identify which Pump we are dealing with

6.1 Selecting my Object Of Interest

So looking at the JORD decision tree I see the following



I have selected the Pump and my narrative is fairly straightforward. These are the pumps that will be required to pump the process fluid to the Debutanizer receiver

The next decision step is a little trickier as its asking me to make a hard choice about which of the 4 stages I am dealing with, namely

Process Design
Design&Engineering
Procurement&Construction
Operations and Maintenance

While I am primarily concerned with Plant Design and Detailed engineering this is NOT completely divorced from the procurement phase. The deliverables from the design process , specifically the

Equipment List, Valve List and Instrument Index are key documents used in the procurement process and its common practise to feed information from the procurement process back to the designer.

Examples encountered in the past include equipment weights, power consumption and nozzle sizes.

However we will put this aside for now and follow the decision tree to Deal with Part 2 Classes

6.2 Does the Class Exist

My first problems is that I don't know how to find out if the class exists !

6.3 Declare the Class

Assuming that the class does not exist I click on the box

"Declare the Class"

However what I see is this

Note_Gen_2

Example of declaration of a Part 2 Class

```
<owl:Class rdf:ID="Ce269d540-678a-11e1-b86c-0800200c9a66"> <!-- UUID preceeded by a letter (here a C for owl:Class) -->
<rdfs:label>Pump P-101</rdfs:label>
<rdf:type rdf:resource="&dm;ClassOfInanimatePhysicalObject"/>
<rdfs:subClassOf rdf:resource="&rdl;R4598459832"/> <!-- instance of &dm;ClassOfFunctionalObject called "Pump Function" -->
</owl:Class>
```

NOTE - This is the "Ur-Class" of P-101 and therefore the superclass shall not be more than the applicable instance of the Core ClassOfFunctionalObject. During design and engineering information will be added, changed, and deleted by means of templates, causing the existence of "variants" of this class.

This really isn't very helpful even to those of us who can read OWL, to the average engineer its Gobbledegook. I need to know HOW to do the following

Check if a Class already exists
Declare one if it does not.

The mapping methodology document offers no way to do this !

So lets pass on to the next entry in the mapping methodology

The flowchart illustrates the process of defining relationships between classes and templates. It starts with a decision diamond asking "Does this Class about which the information is, already exist?". If "Y" (Yes), it proceeds to "Define these relationships". If "N" (No), it proceeds to "Declare the Class".

Next, a decision diamond asks "Is the Class related to another Class with its own information?". If "Y", it proceeds to "Define these relationships". If "N", it proceeds to "For each Part2 Class select information aspect(s)".

Then, another decision diamond asks "Does this Class about which the information is, already exist?". If "Y", it proceeds to "Define these relationships". If "N", it proceeds to "Declare the Class".

Finally, a decision diamond asks "Is the Class related to another Class with its own information?". If "Y", it proceeds to "Define these relationships". If "N", it proceeds to "For each Part2 Class select information aspect(s)".

The "FIND TEMPLATE(S)" section lists various templates categorized by "SELECT":

- Structure:** Location, Activity Class, Stream Class.
- Representation:** Identification, Description, Definition, Classific.& Specializ., Structure, Location, Function, Property, ClassOfProperty, IndirectProperty, Shape, Material, (ClassOf)Number, Activity Class, Stream Class, Documents&InfoRep, ClassOfOtherRelship, Relship betw Classes, Miscellaneous.
- Existence:** Representation.

Page 8

6.5 Identification

Unfortunately I got a list of templates I could use rather than just one so lets pick what seems to be the best fit by reviewing what I as an engineer need to say

Unique Identifier of Thing : basically a GUID

The Text String I will use for the identifier : P-11100A

A method of defining the Identifier : Tag Number or Tag Identification Code

I dutifully click on identification and using the constraints that this is a Part2 Class not an individual
See what looks to be an appropriate template - ClassifiedIdentificationOfClass

Template Specification.

TEMPLATE NAME	ClassifiedIdentificationOfClass																				
STATUS	Proposed																				
DESCRIPTION	Identification of a (Part 2) Class with a string, according a given type of identification.																				
EXAMPLE	myLine is identified as RZ17801 according the XYZ Corp. Line Numbering System.																				
NOTES																					
LIFTED AND LOWERED GRAPH																					
LIFTED TEMPLATE ELEMENTS	<table border="1"> <thead> <tr> <th>Ref</th> <th>URI where definition can be found</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>dm:Class</td> </tr> <tr> <td>2</td> <td>dm:Class</td> </tr> <tr> <td>3</td> <td>dm:ExpressString</td> </tr> <tr> <td>4</td> <td>dm:ClassOfClassOfIdentification</td> </tr> <tr> <td>5</td> <td>dm:Specialization</td> </tr> <tr> <td>6</td> <td>dm:Classification</td> </tr> <tr> <td>7</td> <td>dm:ClassOfIdentification</td> </tr> </tbody> </table>	Ref	URI where definition can be found	1	dm:Class	2	dm:Class	3	dm:ExpressString	4	dm:ClassOfClassOfIdentification	5	dm:Specialization	6	dm:Classification	7	dm:ClassOfIdentification				
Ref	URI where definition can be found																				
1	dm:Class																				
2	dm:Class																				
3	dm:ExpressString																				
4	dm:ClassOfClassOfIdentification																				
5	dm:Specialization																				
6	dm:Classification																				
7	dm:ClassOfIdentification																				
DEFINITION OF PROPERTIES OF LOWERED TEMPLATE	<table border="1"> <thead> <tr> <th>Role No</th> <th>Role Name</th> <th>Role Object Type</th> <th>Remarks & Examples</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>hasUrClass</td> <td>dm:Class</td> <td></td> </tr> <tr> <td>2</td> <td>hasIdentified</td> <td>dm:Class</td> <td>ID to be generated automatically</td> </tr> <tr> <td>3</td> <td>validIdentifier</td> <td>dm:ExpressString</td> <td></td> </tr> <tr> <td>4</td> <td>hasIdentificationType</td> <td>dm:ClassOfClassOfIdentification</td> <td>use ID from RDL or elsewhere</td> </tr> </tbody> </table>	Role No	Role Name	Role Object Type	Remarks & Examples	1	hasUrClass	dm:Class		2	hasIdentified	dm:Class	ID to be generated automatically	3	validIdentifier	dm:ExpressString		4	hasIdentificationType	dm:ClassOfClassOfIdentification	use ID from RDL or elsewhere
Role No	Role Name	Role Object Type	Remarks & Examples																		
1	hasUrClass	dm:Class																			
2	hasIdentified	dm:Class	ID to be generated automatically																		
3	validIdentifier	dm:ExpressString																			
4	hasIdentificationType	dm:ClassOfClassOfIdentification	use ID from RDL or elsewhere																		
SPECIFICATION IN FIRST-ORDER LOGIC																					

My first reaction is confusion as only 2 of the roles have any entries in the column Remarks & Examples and I don't have any firm guidance on how to proceed

Looking at the Lowered Template roles in detail I see

Role 1 : I don't know how to get this value if it already exists or how to create it if it doesn't.

Role 2 : OK I know how to get an identifier for this

Role 3 : OK the value of the Identifier P-11100A fits here

Role 4 : In the RDL we used TagIdentificationCode which was defined as
'An identification code for a 'functional_physical_object'.

Unfortunately this all falls apart when I look for Tag Identification Code in the Jord Endpoint because I find its been flagged as RETIRED and there's an instruction to use 'tagged object code string class'

When I look at this I find it's NOT a ClassofClassOfIdentification but is defined as [RepresentationForm](#) which in part 2 is defined as

A <representation_form> is a <class_of_class_of_information_representation> that distinguishes the form of representation.

EXAMPLE Hexadecimal, text, script, symbol, picture, diagram, semaphore, Morse code, music score, MIDI file format, and XML can each be represented by instances of <representation_form>.

So we take another look and I find something called TAGGED OBJECT CODE ASSIGNMENT CLASS
Now this is a ClassofClassOfIdentification but frustratingly has no notes or description.

So the best I can do is guess that we need to create a member of ClassOfClassOfIdentification, probably in the Project RDL, such as OGI PILOT PROJECT TAGGED OBJECT CODE SO we end up with the following

Role No	Role Name	Role Type	Typical Value
1	hasUrClass	Class	Who knows ?
2	hasIdentified	Class	Unique identifier of UrClass ?
3	valIdentifier	ClassOfExpressInformationRepresentation	P-11100A
4	hasIdentificationType	ClassofClassOfIdentification	OGI Pilot Project Tagged ObjectCode (see Notes 1 & 2)

Notes

1) I assume that an example would an entry in the project RDL of entity type ClassOfIdentification OGI PILOT PROJECT TAGGED OBJECT CODE – is this correct ?

2) This appears very useful as it gives me a way of properly classifying alternative tags for a functional object such as OldTagName, BechtelTagName etc

7.0 Applying the Jord Whole Life Individual Method

I now turned my attention to the 'simpler' WholeLifeIndividual

Following the same steps as for the Class Model I end up with the template - Classified Identification of Individual

Template Specification.

TEMPLATE NAME			
ClassifiedIdentificationOfIndividual			
STATUS			
Proposed			
DESCRIPTION			
A PossibleIndividual is identified according a given type of identification			
EXAMPLE			
The person, who (later) is identified as John Doe, is lifelong identified with "First Son of Jack Smith" (other identifications, except DNA, are invariably applicable for temporal parts, even his name). OR: myPump is identified with a tag number 'P-101'.			
NOTES			
1) Normally use ClassifiedIdentificationOfTemporalPart. 2) This template shall normally be used for WholeLifeIndividuals. It can be used for simple data transfer reasons that do not require lifecycle information integration.			
LIFTED AND LOWERED GRAPH			
LIFTED TEMPLATE ELEMENTS			
Ref	URI where definition can be found		
1	dm.PossibleIndividual		
2	dm.ExpressString		
3	dm.ClassOfClassOfIdentification		
4	dm.Classification		
5	dm.ClassOfIdentification		
DEFINITION OF PROPERTIES OF LOWERED TEMPLATE			
Role No	Role Name	Role Object Type	Remarks & Examples
1	hasIdentifier	dm.PossibleIndividual	use ID from RDL
2	validIdentifier	dm.ExpressString	
3	hasIdentificationType	dm.ClassOfClassOfIdentification	
SPECIFICATION IN FIRST-ORDER LOGIC			

As with the Class Model the entry in Remarks & Example is a little cryptic so once again I have to guess.

The Current iRING mapping uses the initial set template ClassifiedIdentification and looks like this

Role No	Role Name	Role Type	Typical Value
1	hasIdentificationType	ClassOfClassOfIdentification	TAG IDENTIFICATION CODE
2	hasObject	PossibleIndividual	Unique identifier of Thing
3	valIdentifier	Express String	P-11100A

This has some serious problems as TAG IDENTIFICATION CODE is the wrong entity type being a DocumentDefinition and has been Retired and as before there's an instruction to use 'tagged object code string class'

My assumption is that with the JORD Template ClassifiedIdentificationOfIndividual I would seem something like

Role No	Role Name	Role Type	Typical Value
1	hasIdentified	PossibleIndividual	Unique identifier of Pump
2	valIdentifier	RepresentationForm	P-11100A
3	hasIdentificationType	ClassOfClassOfIdentification	OGI Pilot Project Tagged ObjectCode (see Notes 1 & 2)

Notes

- 1) There are no satisfactory examples of ClassOfIdentification in the RDL however there is a ClassOfClassOfIdentification called TAGGED OBJECT NAME ASSIGNMENT CLASS. It appears that the intent is that the ClassOfIdentification should be added to the Project RDL. Thus for the Mimosa/OGI project we would add new Class 'OGI PILOT PROJECT TAG NUMBERING SYSTEM'. Is this correct ?
- 2) If so this allows more than one tag numbering schema to be around and allows support for entries such as 'Old Tag Number' , 'Vendor Tag Number' etc

8.0 Conclusions

We need to clear this up ASAP, Tagging of Plant Items is the essential pre-requisite for any data exchange and there are multiple industry exchange projects being fast tracked. Examples include the Mimosa OGI Pilot and DECHEMA initiatives. Additionally people are already using the modelling methodology rolled out in the iRING initiative so time is of the essence here.