NTNU – Trondheim
Norwegian University of
Science and Technology

Process design; Ensuring consistency between a piping and instrument diagram and the 3D model.

Ole Magnus Urdahl

Master of Science in Engineering and ICT
Submission date: June 2014
Supervisor: Bjørn Haugen, IPM
Co-supervisor: Petter Nilsson, Aker Business Services

Norwegian University of Science and Technology
Department of Engineering Design and Materials

THE NORWEGIAN UNIVERSITY
OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF ENGINEERING DESIGN
AND MATERIALS

MASTER THESIS SPRING 2014
FOR
STUD.TECHN. OLE MAGNUS URDAHL

PROCESS DESIGN; ENSURING CONSISTENCY BETWEEN A PIPING AND INSTRUMENT DIAGRAM AND THE 3D MODEL
Prosess design; Konsistens sjekking mellom "Piping and Instrument Diagram" og 3D modell

The process and piping system for a process plant is normally represented in two different ways:

1. Schematic (Logical)
   This is a 2D logical representation called P&ID, and is used by the process engineer to design the process system. The P&ID contains objects like valves, reducers, branches and equipment, but not elbows and length of pipes.
2. 3D (Physical)
   This is the representation used to fabricate the piping system. The 3D model includes all objects with position, orientation and size.

In Aker Solutions, two different systems are used for the schematic and 3D representations of the process and piping system, Comos for P&IDs and PDMS for 3D design, and both systems have their own proprietary database. This means that the same object is represented in two different databases.

It's important to ensure consistency between the two representations of the piping system. Some vendors of plant design systems offer solutions for automatic consistency checking, but Aker Solutions has not implemented such solutions. Manual or semi-manual checking is costly and can potentially reduce the quality.

The work should encompass the following

1. Analyze tools for consistency checking between a P&ID and 3D models, such as Aveva Schematic 3D Integrator, by setting up a user environment with genuine process system project for testing.
2. Analyze usability, and construct software application or customized add-on software for increased usability for consistency checking.
3. Analyze possible data integration between Comos and PDMS based on ISO 15926, and constructing any necessary adaptions for the process to work efficiently.
4. Recommend actions for improving the efficiency and quality of the consistency checking in Aker Solutions, and create a business case which lists the costs and savings for the recommended actions.

Three weeks after start of the thesis work, an A3 sheet illustrating the work is to be handed in. A template for this presentation is available on the IPM's web site under the menu

“Masteroppgave” (http://www.ntnu.no/ipm/masteroppgave). This sheet should be updated one week before the Master’s thesis is submitted.

Performing a risk assessment of the planned work is obligatory. Known main activities must be risk assessed before they start, and the form must be handed in within 3 weeks of receiving the problem text. The form must be signed by your supervisor. All projects are to be assessed, even theoretical and virtual. Risk assessment is a running activity, and must be carried out before starting any activity that might lead to injury to humans or damage to materials/equipment or the external environment. Copies of signed risk assessments should also be included as an appendix of the finished project report.

The thesis should include the signed problem text, and be written as a research report with summary both in English and Norwegian, conclusion, literature references, table of contents, etc. During preparation of the text, the candidate should make efforts to create a well arranged and well written report. To ease the evaluation of the thesis, it is important to cross-reference text, tables and figures. For evaluation of the work a thorough discussion of results is appreciated.

The thesis shall be submitted electronically via DAIM, NTNU’s system for Digital Archiving and Submission of Master’s thesis.

Contact person: Petter Nilsson, Aker Business Services AS

Torgeir Welo
Head of Division

Bjørn Haugen
Supervisor

NTNU
Norges teknisk-naturvitenskapelige universitet
Institutt for produktutvikling og materialer

NTNU

Fastsatt av Rektor 20.01.2012

STANDARDAVTALE

om utføring av masteroppgave/prosjektoppgave (oppgave) i samarbeid med bedrift/ekstern virksomhet (bedrift).

Avtalen er ufravikelig for studentoppgaver ved NTNU som utføres i samarbeid med bedrift.

Partene har ansvar for å klarere eventuelle immaterielle rettigheter som tredjeperson (som ikke er part i avtalen) kan ha til prosjektbakgrunn før bruk i forbindelse med utførelse av oppgaven.

Avtale mellom

Student:	Ole Magnus Urdahl	født:	24.08.89

Veileder ved NTNU:	Bjørn Haugen
Bedrift/ekstern virksomhet:	Aker Business Services

og

Norges teknisk-naturvitenskapelige universitet (NTNU) v/instituttleder

om bruk og utnyttelse av resultater fra masteroppgave/prosjektoppgave.

1. Utførelse av oppgave

Studenten skal utføre

Masteroppgave	X
Prosjektoppgave	

(sett kryss)

i samarbeid med

Aker Business Services

bedrift/ekstern virksomhet

14.01.14 – 10.06.14

startdato – sluttdato

Oppgavens tittel er:

Process Design; Ensuring Consistency between a Piping and Instrument Diagram and the 3D Model

Ansvarlig veileder ved NTNU har det overordnede faglige ansvaret for utforming og godkjenning av prosjektbeskrivelse og studentens læring.

2. Bedriftens plikter

Bedriften skal stille med en kontaktperson som har nødvendig veiledningskompetanse og gi studenten tilstrekkelig veiledning i samarbeid med veileder ved NTNU. Bedriftens kontaktperson er:

Petter Nølsson – Aker Business Services

Formålet med oppgaven er studentarbeid. Oppgaven utføres som ledd i studiet, og studenten skal ikke motta lønn eller lignende godtgjørelse fra bedriften. Bedriften skal dekke følgende utgifter knyttet til utførelse av oppgaven:

3. Partenes rettigheter

a) Studenten

Studenten har opphavsrett til oppgaven. Alle immaterielle rettigheter til resultater av oppgaven skapt av studenten alene gjennom oppgavearbeidet, eies av studenten med de reservasjoner som følger av punktene b) og c) nedenfor.

Studenten har rett til å inngå egen avtale med NTNU om publisering av sin oppgave i NTNUs institusjonelle arkiv på internett. Studenten har også rett til å publisere oppgaven eller deler av den i andre sammenhenger dersom det ikke i denne avtalen er avtalt begrensninger i adgangen til å publisere, jf punkt 4.

b) Bedriften

Der oppgaven bygger på, eller videreutvikler materiale og/eller metoder (prosjektbakgrunn) som eies av bedriften, eies prosjektbakgrunnen fortsatt av bedriften. Eventuell utnyttelse av videreutviklingen, som inkluderer prosjektbakgrunnen, forutsetter at det inngås egen avtale om dette mellom student og bedrift.

Bedriften skal ha rett til å benytte resultatene av oppgaven i egen virksomhet dersom utnyttelsen faller innenfor bedriftens virksomhetsområde. Dette skal fortolkes i samsvar med begrepets innhold i Arbeidstakeroppfinnelsesloven[1] § 4. Retten er ikke-eksklusiv.

Bruk av resultatet av oppgaven utenfor bedriften sitt virksomhetsområde, jf avsnittet ovenfor, forutsetter at det inngås egen avtale mellom studenten og bedriften. Avtale mellom bedrift og student om rettigheter til oppgaveresultater som er skapt av studenten, skal inngås skriftlig og er ikke gyldig inngått før NTNU har mottatt skriftlig gjenpart av avtalen.

Dersom verdien av bruken av resultatene av oppgaven er betydelig, dvs overstiger NOK 100.000 (kommentert i veiledningen[2] til avtalen), er studenten berettiget til et rimelig vederlag. Arbeidstakeroppfinnelsesloven § 7 gis anvendelse på vederlagsberegningen. Denne vederlagsretten gjelder også for ikke-patenterbare resultater. Fristbestemmelsene i § 7 gis tilsvarende anvendelse.

c) NTNU
De innleverte eksemplarer/filer av oppgaven med vedlegg, som er nødvendig for sensur og arkivering ved NTNU, tilhører NTNU. NTNU får en vederlagsfri bruksrett til resultatene av oppgaven, inkludert vedlegg til denne, og kan benytte dette til undervisnings- og forskningsformål med de eventuelle begrensninger som fremgår i punkt 4.

4. Utsatt offentliggjøring

Hovedregelen er at studentoppgaver skal være offentlige. I særlige tilfeller kan partene bli enig om at hele eller deler av oppgaven skal være undergitt utsatt offentliggjøring i maksimalt 3 år, dvs. ikke tilgjengelig for andre enn student og bedrift i denne perioden.

Oppgaven skal være undergitt utsatt offentliggjøring i

ett år	
to år	
tre år	

(sett kryss bak antall år hvis dette punktet er aktuelt)

Behovet for utsatt offentliggjøring er begrunnet ut fra følgende:

De delene av oppgaven som ikke er undergitt utsatt offentliggjøring, kan publiseres i NTNUs institusjonelle arkiv, jf punkt 3 a), andre avsnitt.

[1] Lov av 17. april 1970 om retten til oppfinnelser som er gjort av arbeidstakere
http://www.lovdata.no/all/hl-19700417-021.html
[2] Veiledning til NTNUs standardavtale om masteroppgave/prosjektoppgave i samarbeid med bedrift
http://www.ntnu.no/studier/standardavtaler

Selv om oppgaven er undergitt utsatt offentliggjøring, skal bedriften legge til rette for at studenten kan benytte hele eller deler av oppgaven i forbindelse med jobbsøknader samt videreføring i et doktorgradsarbeid.

5. Generelt

Denne avtalen skal ha gyldighet foran andre avtaler som er eller blir opprettet mellom to av partene som er nevnt ovenfor. Dersom student og bedrift skal inngå avtale om konfidensialitet om det som studenten får kjennskap til i bedriften, skal NTNUs standardmal for konfidensialitetsavtale benyttes. Eventuell avtale om dette skal vedlegges denne avtalen.

Eventuell uenighet som følge av denne avtalen skal søkes løst ved forhandlinger. Hvis dette ikke fører frem, er partene enige om at tvisten avgjøres ved voldgift i henhold til norsk lov. Tvisten avgjøres av sorenskriveren ved Sør-Trøndelag tingrett eller den han/hun oppnevner.

Denne avtale er underskrevet i 4 - fire - eksemplarer hvor partene skal ha hvert sitt eksemplar. Avtalen er gyldig når den er godkjent og underskrevet av NTNU v/instituttleder.

Trondheim 10.6.14 Ole M. Urdahl

sted, dato student

Bjørn Haugen

sted, dato veileder ved NTNU

10/06/14.

sted, dato instituttleder, NTNU institutt

sted, dato for bedriften/institusjonen stempel og signatur

Preface

This thesis is based on Aker Solutions wish to enhance the interface and consistency between their logical and physical process plant design software applications. The work has taken place at NTNU, and at Aker Solutions offices at Fornebu, in close cooperation with Petter Nilsson, Technical Service Manager for PDMS.

Trondheim, 06-10-2014

Ole Magnus Urdahl

Acknowledgment

First off, I would like to thank Aker Solutions for the opportunity to write this thesis. I would also like to thank Rune Vassdal, Kottage Deshaka, Svein Erik Tårland, Chris Woodland, Fredrik Skepstett and Therese Wolff for giving me assistance, understanding and general sharing of their knowledge, through dialogs and discussions.

Special thanks goes out to Petter Nilsson and Professor Bjørn Haugen for their roles as Supervisors, for help throughout the semester, and the feedback they have given me.

O.M.U.

Summary

The focus of this thesis is looking into different ways of ensuring consistency checking between piping and instrument diagrams and the 3D model in process plant design. The current interface between Siemens Comos and Aveva PDMS is inadequate when it comes to interoperability, and especially when it comes to efficient consistency checking between Comos and PDMS databases. Increasing the knowledge of available systems and procedures, will allow Aker Solutions increase efficiency by automating parts of the interface that is, today, handled manually by engineers. By automating consistency checking, Aker Solutions will reduce delivery time of projects and reduce errors in their process plant designs, thus saving money.

This thesis starts with analysing the term "consistency" and what consistency means in a process plant context, before taking a closer look at ISO 15926, which is the ideal standard for the representation of process plant life-cycle information, and data transfer between different process plant CAD tools.

Further the thesis looks into the two software applications that are currently being used by Aker Solutions for designing schematic logical P&IDs and physical 3D models, Comos and PDMS respectively. An immersive look at the interface between the two software applications will give an understanding of how the two correlate with each other, to help understand what is necessary to develop a better interface. Tag assigning, batch jobs and the Comos Construction Assistant are all parts of the interface today, even though much of the work towards consistency checking done by Aker Solutions' engineers is done manually. The thesis will therefore look into the use of applications such as Comos PDMS Integrator, Aveva Schematic Integrator, Aveva Diagrams and the possibility of an in-house ISO 15926 solution, trying to find an automated way of securing consistency between P&IDs and 3D models.

To justify expanding the interface between Comos and PDMS, a business case will evaluate how improving the consistency checking in Aker Solutions process plant projects affects Aker Solutions. The business case is based on implementing an interface where schematic diagrams are exported from Comos as .xml files based on ISO 15926, which are then imported into Avevas schematic databases using the Schematic Model Manager. Aveva Schematic 3D Integrator then being used for consistency checking should be a viable solution for automating consistency checking. Unfortunately, due to contract negotiation with Aveva, Aker Solutions were not able to acquire the necessary software applications for testing during the development of this thesis.

It is important to mention that the Comos PDMS interface should, for future

projects, always be under review. The solutions this thesis' business case is based on, is currently best way of ensuring consistency between P&IDs and 3D models. This is not necessarily the case in the future, as it is subject to change as the CAD software market is under constant development. Going forth starting to document the Comos PDMS interface and any cost related to consistency checking and inadequate software application interoperability is crucial for making any significant changes to the Comos PDMS interface. A complete review of the Schematic Model Manager and Schematic 3D Integrator should be done to implement a better interface for consistency checking.

Sammendrag

Fokuset i denne oppgaven tar for seg de ulike måtene det er mulig å sikre konsistens mellom rør- og instrumentdiagrammer og 3D-modellen i design for prosessanlegg. Det nåværende grensesnittet mellom Siemens Comos og Aveva PDMS har mangler når det gjelder interoperabilitet, og spesielt når det kommer til effektiv konsistenssjekking mellom Comos og PDMS sine databaser. Å øke kunnskapen om tilgjengelige datasystemer og prosedyrer, vil gi Aker Solutions økt effektivitet ved å automatisere deler av grensesnittet som i dag er, manuelt utført av Aker Solutions sine ingeniører. Ved å Automatisere konsistenssjekking, vil Aker Solutions redusere leveringstid av prosjekter og redusere feil i design av prosjektene, dermed spare penger.

Denne avhandlingen vil analysere begrepet "konsistens" og hva konsistens betyr i sammenheng med et prosessanlegg. Deretter vil det bli sett nærmere på ISO 15926, som er den ideelle standarden for representasjon av prosessanleggs livssyklusinformasjon, og dataoverføring mellom ulike CAD-verktøy.

De to programmene som er i bruk av Aker Solutions for å designe skjematisk logisk prosess- og instrument diagrammer og fysiske 3D-modeller, er henholdsvis Comos og PDMS. En omsluttende titt på grensesnittet mellom de to programmene vil bli gjort for å gi en bedre forståelse for hvordan de to programmene korrelerer med hverandre, for å forstå hva som er nødvendig for å utvikle et bedre grensesnitt. Tagging, batch-jobber og Comos Construction Assistant er alle deler av dagens grensesnitt, men mye av arbeidet mot konsistens som er gjort av Akers Solutions sine ingeniører, er utført manuelt. Oppgaven vil derfor se nærmere på bruk av applikasjoner som Comos PDMS Integrator, Aveva Schematic Integrator, AVEVA Diagrams og muligheten for en ISO 15926 løsning, laget av Aker, i et ønske om å finne en automatisert måte å sikre konsistens mellom P&IDer og 3D-modeller.

For å rettferdiggjøre det å utvide grensesnittet mellom Comos og PDMS, vil en business case vurdere hvordan bedre konsistens i Aker Solutions prosess og pipingprosjekter kan utføres, og hvordan det kan påvirke Aker Solutions sine prosedyrer. Business casen er basert på å implementere et grensesnitt hvor skjematiske diagrammer eksporteres fra Comos som .xml-filer basert på ISO 15926, som deretter importeres til Avevas skjematiske databaser ved hjelp av Schematic Model Manager, og som tilslutt sammenlignes med 3D-modellen ved hjelp av Aveva Schematic 3D Integrator. Denne løsningen burde være en levedyktig løsning for automatisering og konsistenssjekke. Dessverre, på grunn av kontraktsforhandlinger med Aveva, var Aker Solutions ikke i stand til å tilegne seg de nødvendige programmer for testing av denne prosedyren for oppgaven.

Det er viktig å nevne at Comos PDMS grensesnittet alltid burde vurderes i fremtiden. Løsningen denne business casen er basert på, er den beste løsning for å sikre konsistenssjekkening i dette øyeblikket, mellom P&IDs og 3D modeller. Dette betyr ikke at det kommer til å være den beste løsningen for fremtiden, siden CAD programmer stadig utvikles og forbedres. Videre arbeid for Aker Solutions vil være å dokumentere det fullstendige Comos PDMS grensesnittet og alle kostnader som kan kobles til konsistenssjekkening. Dette er avgjørende for å starte og underbygge fremtidige betydelige endringer i Comos PDMS grensesnittet. En fullstendig vurdering av «Schematic Model Manager» og «Schematic 3D Integrator» burde gjøres som første steg i implementering av et forbedret grensesnitt for konsistenssjekkening.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The introduction chapter will give an explanation of the issues with consistency between a Piping and Instrument Diagram and the 3D model of process design. By knowing the background, objectives and the approach taken in this paper, the thesis will give a better understanding of how the complex problem of process design consistency can be solved.

1.1 Background

When designing a process plant there are two essential ways of representing the way the pipe should be structured.

1. Schematic (Logical): The 2D logical representation is called P&ID (Piping and Instrumental Diagram), and is used by the process engineer to design the process system. The P&ID contains objects like valves, reducers, branches and equipment, but not elbows and length of pipes.

2. 3D (Physical): The representation that is used to fabricate the piping system. The 3D model includes all objects with position, orientation and size.

Particularly in the process industry, often planning and realisation of complex industrial plants span a period of several years. During this period of time, a huge amount of engineering data from different sources and in different file formats accumulate, which constantly change due to frequent revisions. This engineering data has to be integrated into PDMS 3D systems often used in the process industry. Using individual engineering database solutions without automated bidirectional data exchange, so-called isolated applications, this can at best be realised by investing manpower, time and costs that cannot be economically justified. [7]

Generally, the information models of domain-specific software tools suffer from a lack of a well-structured, standardised information representation [8]. Thus, the exchange of data between the different tools is often hindered by the inherent heterogeneities of the underlying data sources [9]. This is the essence of the issue when it comes to consistency checking between P&ID and the 3D model of a process plant. The basic data storage, class construction and product modulation is different between the two ways of representing a pipe. This includes the two different systems used by Aker Solutions, for the schematic and 3D representations of the process and piping systems. Comos for P&IDs and PDMS for 3D design are both systems represented by their own proprietary database. This means that every part of a pipe is represented in each of the two different databases, in their own semantically separate way.

Unresolved inconsistencies can cause incorrect decisions, design rework or even worse problems [4], such as design flawes in the finished process plants. As a result, the lack of appropriate tool interoperability and data integration are major cost drivers in the design phase. In this respect, the National Institute of Standards and Technology (NIST) in the U.S. recently reported that the lack of interoperability costs the U.S. capital facilities industry 15.8 billion dollars per year [10], compared to a hypothetical scenario where the exchange of data and the access to information are not restricted by technical or organizational boundaries [6]. Hence, there is an enormous potential for productivity gains, and consequently for cost reduction and quality improvement.

1.2 Problem Formulation

In Aker Solutions the piping discipline work is an overlapping, concurrent work process. Team members working downstream, such as piping, have to rely on preliminary information, often given by the process discipline. Since the design processes in chemical engineering typically are of a creative and evolutionary nature, this preliminary information may frequently change substantially. Consequently, team members working downstream must readjust their work if colleagues from upstream disciplines change the design requirements or specifications in an unexpected way. Under these conditions, upstream engineering changes may cause significant downstream rework, potentially delaying the whole project [11]. Given the evolutionary nature of the design process, there can be, and often is, substantial changes to the piping design. This is why effective consistency checking is of such great importance. The following tasks will therefore be the focus of the study:

1. Analyse tools for consistency checking between a P&ID and 3D models, such as Aveva Schematic 3D Integrator, by setting up a user environment with genuine process system project for testing.

2. Analyse usability, and construct software application or customized add-on software for increased usability for consistency checking.

3. Analyse possible data integration between Comos and PDMS based on ISO 15926, and constructing any necessary adaptions for the process to work efficiently.

4. Recommend actions for improving the efficiency and quality of the consistency checking in Aker Solutions, and create a business case which lists the costs and savings for the recommended actions.

1.2.1 Objectives

Focusing on these four points, the thesis will approach Aker Solutions schematic and 3D process plant design, as an outsider, looking for issues and improvement opportunities.

To give a closer understanding of the main focuses in this thesis, the prestudies in chapter three regards both the term "consistency" and the ISO 15926 that is relevant for process system design. It will describe the software applications in use by Aker Solutions, the interface that exists between the software applications, and how developing the interface further will benefit the company.

1.2.2 Limitations and Scope

Looking for improved functionality, software applications, and/or processes will only mind piping design between Comos and PDMS. There are other engineering areas that use the same software and connectivity as piping, such as instruments and equipment. These disciplines will not be the focus of this thesis.

During the given time of this thesis, Aker Solutions ended up in contract discussion with Aveva regarding the software applications surrounding Aveva PDMS, which were of major focus of the thesis. The limited access to software applications such as Aveva Schematic Model Manager and Aveva Schematic 3D Integrator halted the possibility of developing a customized solution or any software application or add-on. The scope of the thesis altered in the direction of any available software solution, and the development of the market.

Chapter 2

Definitions and Abbreviations

2.1 Definitions

Autocad a commercial software application for 2D and 3D CAD and drafting.

Aveva PDMS 3D design software developed by AVEVA (System) [12]

Aveva P&ID AVEVA P&ID is a P&ID drafting program that allows a user to create intelligent, project-wide data as the P&ID is designed using the familiar AutoCAD drafting system.

COMOS Process, automation and electrification design software by Siemens [13]

CWA The closed world assumption (CWA) is the presumption that what is not currently known to be true is false, thus if it is not defined, it does not exist.

Diagram A description that represents information using the topology of symbols

ISO 15926 An open international standard for integration of life-cycle data for process plants including oil and gas production facilities. This representation is specified by a generic, conceptual data model designed to be used in conjunction with reference data: standard instances that represent information common to a number of users, process plants, or both (ISO 15926:2003)

PML A domain specific language developed by Aveva to enable customisation of their plant and marine design products.

RDL A standard proposed by Microsoft for defining reports

XML Markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. (XML 1.0 Specification)

XMpLant A brand name owned by Noumenon Consulting out of the UK

XMpLant Schema A XML schema using ISO 15926 Part 4, released into the public domain by Noumenon in 2001. (Schema example can be found at http://www.fiatech.org/specauto/ProteusSchema_3.3.3.xsd)

2.2 Abbreviations

3D 3 Dimensions

ABS Aker Business Services

AET Aker Engineering & Technology

CAD Computer-Aided Design

CAE Computer-Aided Engineering

COMOS COMponent Object Server

CWA Closed World Assumption

DCR Design Change Request

EPC Engineering, Procurement and Construction

FEED Front End Engineering Design

HTML Hyper Text Markup Language

HVAC Heating, Ventilation, and air conditioning

IDF Intermediate Data File

ISO Piping Isometric drawing

PDMS Plant Design Management System

PED Process Engineering Diagram

PFD Process Flow Diagram

PML Programmable Macro Language

P&ID Piping and Instrumentation Diagram

RAM Random Access Memory

RDL Report Definition Language

SMS Short Message Service

STEP STandard for the Exchange of Product Model Data

UDA User Defined Attributes

UDET User Default Element Types

URG User Reference Group

XML Extensible Markup Language

Chapter 3

Prestudy

3.1 Consistency

There are several ways of describing consistency, and there are also varying levels of consistency. By understanding what consistency means in this specific case, one can build an understanding of how to build consistency for P&IDs and 3D models.

3.1.1 Definition

1. Steadfast adherence to the same principles, course, form, etc.

2. Agreement or harmony between parts of something complex; compatibility.

In classic deductive logic, a consistent theory is one that does not contain a contradiction [14]. Since both P&IDs and 3D models operate under CWA, the very simplest form of consistency that is concluded between a P&ID and a 3D model is that if there exist an object in one of the process plant representations, there should exist a similar object in the other.

3.1.2 Degree of consistency

The degree of consistency correlates directly with the level of detail of a design. When looking into a process system, certain "rules" for consistency becomes

#	Rule/Check (Based on the existence of an object in one accompanying design drawing)	Degree of consistency
1	A similar exclusive object can be found in the other design.	1
2	A similar object can be found in the other design, with connected ID tag (4.3).	2
3	A similar object can be found in the same area, in the other design (4.3.2).	1
4	A similar exclusive object can be found in the other design, with connected exclusive ID tag	3
5	An accompanying object is connected to the same objects as the original object.	2
6	An accompanying object has the same connectivity points as the original.	3
7	An accompanying object has the same size, attributes, dimensions as the origianl.	2

Table 3.1: Rules for degree of consistency

clear. Some of the rules can be used in general cases, while others become somewhat unique to the particular dataset or data system.

Taking into account the classic deduction (3.1.1), in the very simplest way, if there is an object in one drawing there should be a similar object in the other drawing. Regarding the fact that there can be several objects in one drawing that correlates with only one object in the other, the first rule becomes a key to solving over populating connectivity by introducing exclusivity. Rule number two on the other hand, opens up for surjective data sets, where several objects in one drawing, is pointing to the same object in the other drawing. This does not necessarily mean that you have over populated connectivity, but rather that the information logic, representing the P&IDs and 3D models, is different. Take a 3D model of a pipe for example, where the pipeline is put together of bends, t-joints and straight lines, while in the schematic/logical drawing the whole pipeline is defined as one singular object. By using ID Tags, some objects that are supposed to be defined separately will be defined separately, while other objects like pipes can be defined by which pipeline they are a part of.

Rule number three is a necessity based on area allocation of objects. Making sure that the right equipment is in the vicinity of the objects that interact with that specific object. The next step in the same type of consistency checking is to introduce connectivity, not only checking if the objects are in the right area, but also making sure that the objects that are next to each other, or interact, are connected.

The third degree of consistency is extremely detailed, and also takes into account certain assumptions of the different types of system designs, such as area allocation, or specified coordinate data. The rule can be quite useful for making sure that the equipment is located in the right place, depending on if it is a P&ID or 3D model. Rule four can only be useful between system designs that are bijective, where every object has a one-to-one representation between drawings, making sure that every object has an identical representation in the accompanying design. Rule six and seven are only applicable if an object has been given certain rules for connectivity, and if each system design has a set of attributes that correlate with the other. The third degree of consistency therefore comes closer to making sure that the design drawings are identical rather than consistent.

3.1.3 Adapted to engineering

To be able to efficiently use consistency checking as an engineer, some traits become clear. Looking into P&IDs and 3D models, there are limits to data structure, computational power and general efficiency. Finding the optimal degree of consistency for consistency checking becomes vital for a fast and reliable consistency checking software tool.

Regarding the first and third degree of consistency, checking if the right amount of equipment exists in the given area becomes somewhat an elementary approach, but can be quite beneficial. Such a check can be done by simply counting the number of equipment in each area location. Though it can be used as a quick check before any major consistency check begins to make sure that the quite necessary objects and equipment are present. It becomes rather unnecessary to do major data calculations on objects that surely are not present.

For the rest of the rules, their usability is determined by data structure. Rule number five and six becomes obsolete if the system design drawings does not contain connectivity information, but can be extremely useful for a detailed consistency check if the necessary data is present. It is also important to make a clear decision between using rule number two or four as it is determined by the data set being surjective or bijective, respectively. Lastly consistency checking for any available attribute that the system design drawings contain can be extremely time consuming as there are huge amounts of data. When trying to make an efficient consistency check, smaller attribute sets can be used to check that the most important data is set as according to the system drawings.

3.2 ISO 15926

ISO 15926 is an International Standard for the representation of process plant life-cycle information. The representation is specified by a generic, conceptual data model that is suitable as the basis for implementation in a shared database or data warehouse [1].

3.2.1 Description

Teijgeler [15] best described the standard when he said, "ISO 15926-2 can, in a way, be compared with a natural language". It has 201 entity types ("characters"). From that templates ("words") can be built, and from linked templates the user can build a "story", that of what happened in our plant during its lifetime ("cradle to grave"). The standard essentially is a computer understood language that can describe every part of an oil platforms life cycle.

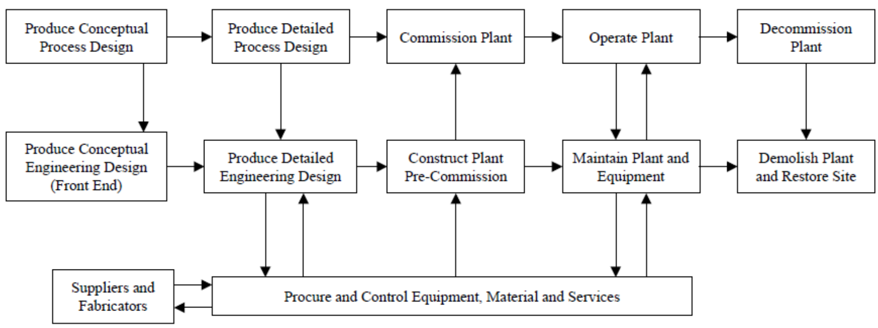


Figure 3.1: Activity model of the process plant life-cycle [1]

The parts of ISO 15926 are like the parts of human speech. Part 2 is the data model equivalent to the rules of grammar, and Part 4 is the reference library, equivalent to the dictionary. When any two people use the same rules of grammar and use the same dictionary, they can communicate freely [2]. A rather good metaphor would be if a Norwegian person met a German, a conversation in their native language would not yield great communication. But if they both translated their thoughts and words into English (i.e. ISO 15926), their ability to communicate would increase drastically. In a way, the ISO 15926 becomes a standard of communication. Each part of the standard representing a different part of natural language:

Part 2 (Core Data Model) Natural Language Grammar (Basic Rules)

Part 4 (Reference Data) Dictionary and Thesaurus (Words & Terms, extensible)

Part 7 (Templates) Phrase Sentence Paragraph (Useful semantic structures, extensible)

Part 8 (RDF/OWL) Paper, File, Stone tablet (Representation technology)

Part 9 (Façades) Website, Postal Service (Read, write, query, service technology)

There are several other standards that have the same role in communicating as ISO 15926. HTML and SMS both are standard for communicating webpages and short texts, respectively. All kinds of devices can read and send SMSes, and different web browsers all conceive web pages that are written in HTML, even if they do not have the same source code.

3.2.2 Reason for use

Even agreement on the skeleton for a life-cycle data model on a coarse-grained scale can be difficult to achieve due to the divergent objectives of the various stakeholders [16]. What separates ISO 15926 from other life cycle standards like STEP (ISO 10303) and CAEX (IEC 62424), is the advantage of representing both an abstract model which represents the technical necessities of a unit, and the actual functional unit/device which fulfils the functional necessities of the unit. Hildre et al. [17] emphasize that, models seek to represent empirical objects, phenomena, and physical processes in a logical and objective way. All models are in simulacra, that is, simplified reflections of reality, but despite their inherent falsity, they are nevertheless extremely useful. Building and disputing high level models is fundamental to the enterprise. What ISO 15926 does, is connecting the value of the representation of the empirical object and connects it to an actual object, with a lifespan. As one can see in figure 3.2, the object "thing" is separately connected to an abstract object, and a possible individual that fulfils the criteria given by the abstract object. When applied correctly, it leads to many possibilities for automated design based on given parameters. Therefore ISO 15926 gives advantages such as reduced production time and increased consistency. However, except in the oil and gas industry, the ISO 15926 has not found broad acceptance, due to the extreme complexity of the data model on the one hand [18], and its rather narrow scope on the other hand [19].

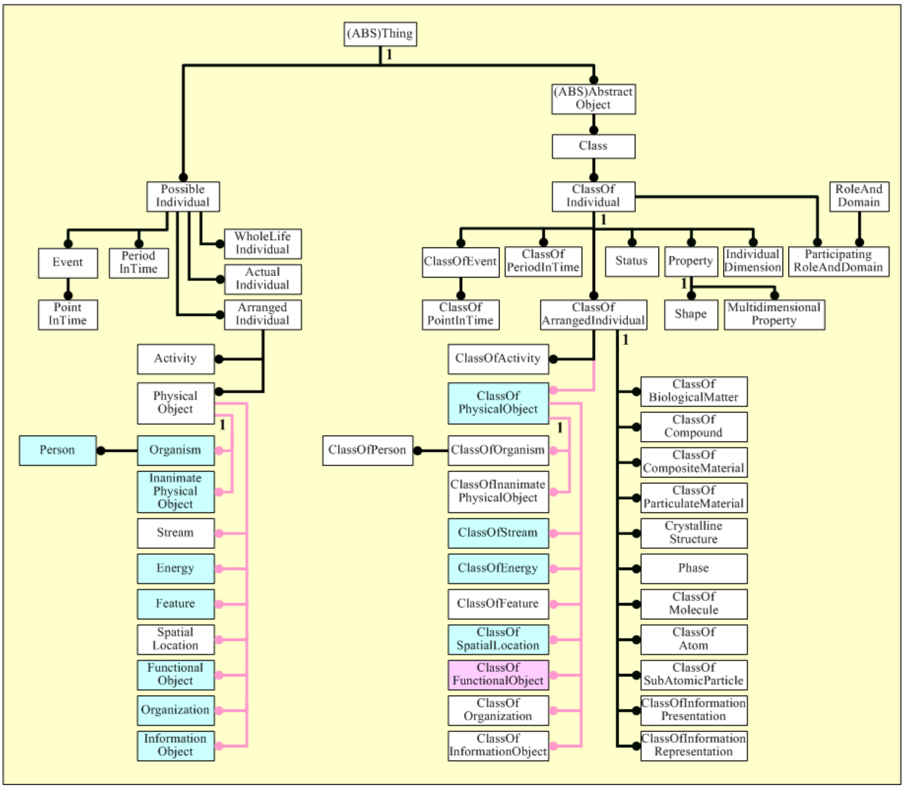


Figure 3.2: Proposed class schema for PossibleIndividual and ClassOfIndividual

A good example for why one would want to use ISO 15926 can be seen in figure 3.3. Two seperate data sheets represent a centrifugal pump, but with very different ways of displaying and grouping the data. When trying to compare data between the two sheets, several issues appear, firstly by one sheet utilizing

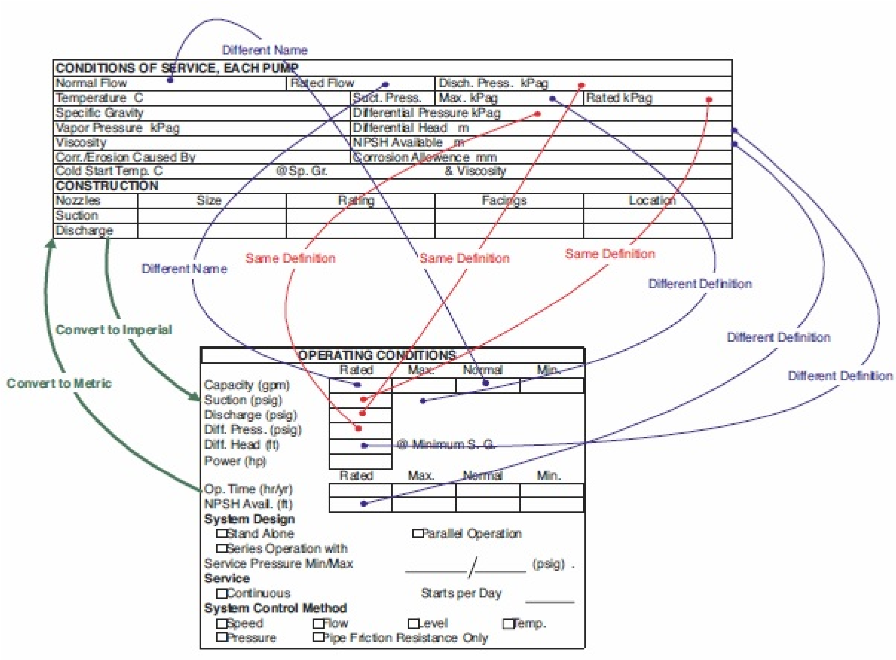


Figure 3.3: Comparison of two data sheets [2]

metric units, while the other is utilizing imperial units. While some data is presented by the same or close to comparable definitions, for example "Different pressure (kPag)" and "Diff. Press. (psig)", other attributes are presented more ambiguous, such as "Rated Flow", which by an experienced engineer would be connected to the "Capacity (gpm) Rated". As many of the attributes are presented differently, not only visually, but also fundamentally, a standard for what the necessary attributes for a process plantis quite useful. Since it becomes too hard to justify that all software must change their fundamentals to accommodate the standard, in the same way that you cannot force every human being to speak the same language, ISO 15926 becomes a second language to assist any communications or translations necessary to execute efficient data transfer between different interfaces.

Chapter 4

Software tools in use by Aker Solutions

Comos and PDMS are the two current tools used by Aker Solutions to design process plant P&IDs and 3D models respectively. The two software applications are built upon two separate databases that do not communicate directly in any way. That is why Aker Solutions have built an interface around the two applications to help with both attribute comparing and updating. Building a greater understanding of how the two databases interact in this particular interface will give a clearer picture of how a consistency checking solution may or may not be completed.

4.1 Comos

Comos is a multidiscipline tool for systems engineering developed by Siemens [20]. Comos is used for a complete logical design, and as projects data hub when Aker Solutions design process systems. In other words, all key information is imported from other sources and logged in Comos. Approximately 80% of Aker Engineering and Technology are Comos users [21].

4.1.1 Data Structure

The data structure that is set for pipelines in Comos is a three-level pipe structure. The three levels are [13]:

Pipe: A pipe is created at the top level. This object collects and administers pipe branches and segments. Usually, one pipe object corresponds to an entire pipe run.

Pipe branch/equipment: Pipe branches are created underneath the pipe object, and are the actual data holders. They encapsulate data changes within a pipe. You also have different equipment such as valves and tees and flange objects at this data level.

Pipe segment: The pipe segment form the lowest level of the pipe structure. They do not have their own tabs.

In other words, when constructing a pipeline, there are pipes, split into pipe branches that again are represented by pipe segments.

4.1.2 Pipe Attributes

As any other data object, each object in Comos is characterized by a set of attributes. The attributes can be grouped into two main tabs, technical data and operating data. Since Comos is the main data hub for system projects at Aker Solutions the different groups may contain a large variation of data, but for piping, the technical data will mainly consist of descriptive attributes such as diameter and material specifications. On the other hand you have operational data that for pipes will have attributes describing the operational functionality of the pipes, such as volume flow, and temperature.

When creating a new object in Comos, there are a set of attributes that have to be specified to meet the minimum requirements for creating an object. There are several reasons for this, but mainly because without these attributes it can be hard for downstream work processes to be executed efficiently. The minimum required attributes that needs to be completed when creating a new object in Comos are:

- Tag number
- Tagged object state (active/voided)
- Discipline identification
- Area code (may be corrected later on by either Comos or PDMS if equipment is moved to a new area location)
- Service description
- Procurement package number

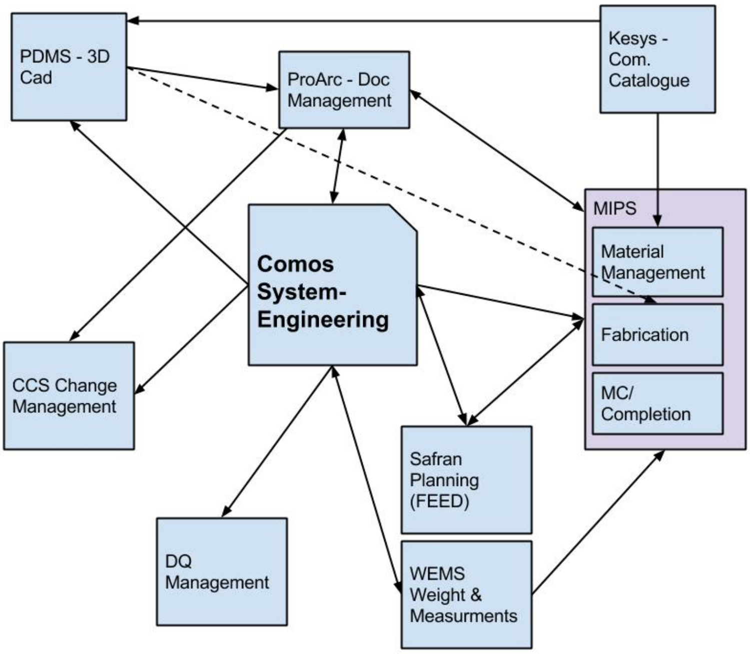


Figure 4.1: Aker Solutions' application model, displays how Comos is a key part in the process plant project development. Acting both as software application for P&ID development, and as a data hub.

- Line connection

4.2 AVEVA PDMS

For 3D modelling of piping and equipment in a process plant system, Aker Solutions use PDMS, a multidiscipline 3D CAD software for engineering, design and construction projects distributed by AVEVA [12].

4.2.1 Data Structure

The data structure in PDMS is split into three different levels.

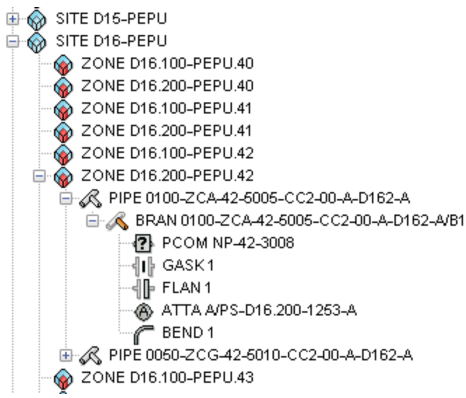


Figure 4.2: Example of pipe, branch, components in PDMS

Pipes: Pipes may be considered as lines on a flow sheet. They are created at the top level, can run between several end connection points and are usually grouped by a common specification and process.

Branch: Branch objects are created as sections of a pipe, and thus have known start and finish points. Branch share the same level name as the second object level in Comos, but differs in the fact that in PDMS branches are complete pipe segments that consist of both pipe segments and equipment.

Piping component: A branch can own a wide variety of components that include gaskets, flanges, straights, tees and valves. These are the different specific components that make up the pipeline, and are obtained from item catalogues that contain an assortment of vendor components.

4.2.2 Pipe Attributes

Every element in a PDMS database has a fixed set of properties known as its attributes. Some attributes are common throughout the range of elements while others differ according to the type element involved [12]. There are general attributes that are common to most of the elements in PDMS as they define space positioning, owner and orientation. But there are also special attributes that are custom to each element type, depending on the subclass of the element. For a pipe straight, there are pipe diameter and thickness, or for pressure valves there are equipment specialized maximum pressure attributes. All pipe elements have the following attributes as a minimum:

- Name
- Type
- Lock
- Owner
- Members

4.3 Tags

One of the most important attributes used by Aker Solutions for the process plant system interface are tags. Tagging is done in both by using Aveva Tags, and Comos UIDs, each being unique to Aveva and Comos objects respectively. They have several purposes including [22]:

- Create, manage and view user-defined lists of elements from an Aveva PDMS project database.
- Use these lists to add, edit and delete elements.
- Navigate the hierarchical structure of a database using an explorer. Use the explorer to view, add and delete elements.
- View and change the life-cycle status of elements.
- Highlight changes to elements between selected database sessions.
- Manage local and extract claims of elements and attributes.
- Compare attributes in other Aveva product databases and update and link them if required.
- Export and import data to and from Excel spreadsheets.
- Publish data to Aveva NET.
- View schematic diagrams, schematic models and the files, URLs and e-mail addresses associated with elements.
- Produce reports from list data
- Update the project life-cycle status

Essentially the tags are attributes that act like links between the connected objects in Comos and PDMS. The attributes do not have any functionality other than maintaining knowledge of which objects are coupled, which gives it a diverse use. By accessing an object in one application, and using the tag attributes, one can find the same object in all the other applications Aker Solutions use if a tag is assigned. There are several rules for how an object is tagged, from where it is located in the process plant, to what kind of equipment the object is. Since Comos and PDMS do not represent objects in the same way, especially pipelines, not all objects are tagged in bijection. Rather the pipeline objects are tagged in surjection where there are fewer than, or even one, object in Comos, that represent a pipeline of many objects in PDMS. Since a pipeline in Comos, in its simplest form, only represents connectivity between equipment, it is often represented by a single object. While in PDMS, the same pipe line may be put together by straights, flanges and tees, all with tags assigned to one particular object pipeline in Comos.

Considering that most of the different objects represented in either Comos or PDMS have object tags linking one Comos object to a PDMS object and vice versa, the task of consistency checking becomes exponentially more complex when regarding the fact that the database structure and object data sets result in a one-to-several, or one-to-none object link. If all tags had bijective connectivity, one could simply go from object-to-object and check that all the data in each object was similar, but that is unfortunately not the case.

4.3.1 Naming Convention

Naming Convention is an approach that in many data coordination tasks become an important part of the solution. Unfortunately a consistent naming convention cannot be the focus of coordinating Comos and PDMS attributes. Since Aker Solution takes on projects given by companies like Conoco Philips and Statoil, they are seldom in control of the naming convention that is determined for the projects. Not only do the project owning companies have different naming conventions, in many cases even the same company have unique conventions depending on the projects. Some parts of the naming convention for tags have a tendency of being prominent every time there is a new project. There is the subclass that defines which data fields (attributes) that will be transferred between Comos and PDMS during nightly batch jobs (section 5.2). The tag will contain a name extension describing which subclass it belongs to; “EP” or “EQ” for Mechanical equipment, or “EI” or “EP” for Instruments. There are also area allocations that are present in the tag names, depending on which part of the

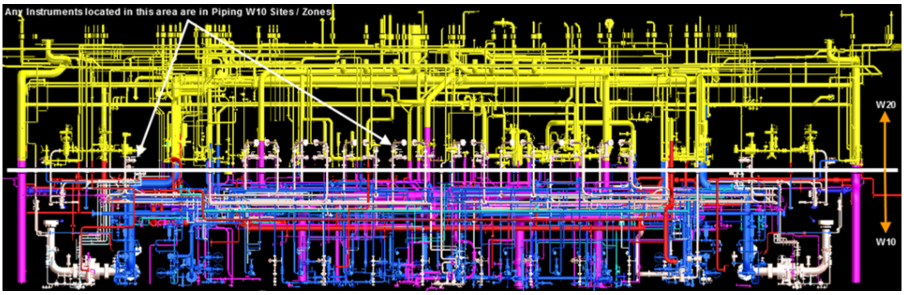


Figure 4.3: An example of area allocation, and how lines/pipes and instruments that are in Zone W10 also have physical coordinates in area W20.

platform the object is placed.

4.3.2 Area allocation

Area allocation is a big part of process system design, divided both by physical space and system functionality. As living quarters and instrumentation and piping have different functionality and are therefore given different area codes.

There are some issues with how the area allocation works [3]. As can be seen in figure 4.3 a PDMS-tag picks the Area or location code, from the zone it is allocated to, and not from the datum coordinates. The figure is a good example of this where several of the objects in W10 have physical coordinates located in W20. This may lead to issues when updating UDAs and UDETs (section 4.4), since the area code does not match the area code used in Comos.

Another important issue that should be mentioned, is if a tag is moved in PDMS from one area to another area, e.g. P10 to P20, then the tag between Comos and PDMS must be re-assigned. The reference number in PDMS is changed when the tag is moved, which does not automatically update in Comos. If the tag is not re-assigned there will be a loss in connectivity and information between the correlating objects will no longer be updated, which is a grave error and can lead to a lot of complications.

4.4 Updating UDA UDET

In Aveva applications the UDAs and UDETs can be accessed and set through the command line, which gives admittance to pushing attribute updates. By setting up grid sheets, such as an excel sheet, you can add a list of attribute names with updated attributes to the command line and that way update all of the project attributes. Ultimately this way of approaching attribute connectivity should only be considered for work arounds and is in no way a solution for bigger projects. In large projects there are a huge amount of attributes that are constantly being changed and updated in the data sets. When updating through the command window the users are prone to human error, which may result in slow or even faulty updating of the attributes. Human error can consist of several things, but mainly typing errors, or inability to handle text or attribute exceptions. These errors may seem like small irregularities, but when working with large datasets, such errors can result in large data mismatches. It is even worse if some attributes are updated containing the wrong values without the mistakes being noticed, which may lead to grave mistakes later in the production process. Even though this way of updating attributes is discouraged, it is used on several process plant projects.

The other way the UDAs and UDETs are updated by Aker Solutions is through nightly batch jobs, which is explained in section 5.2.

4.5 PML

Programmable Macro Language is the customisation accessible by Aveva Plant and is used by Aker Business Services for making add-ins for PDMS. It provides a mechanism for users to add their own functionality to the Aveva Plant software family [23]. It is a coding language specific to Aveva products based on the command syntax that is used to drive PDMS. This essentially means that any command or function that can be executed in Aveva is available through PML, such as retrieving attributes, changing the colour representation of objects, or making an expanded view control in PDMS. Since the macro language includes IF statements, variables, DO loops and error handling, the use of PML becomes vast.

When moving forth with the Comos - PDMS Interface, PML may be used as an important tool, to help bulk handling in the Aveva software applications. Bulk handling is crucial for efficient data handling of large data sets, which is typical for Aker Solutions.

Chapter 5

Attempted implemented approaches

As of now, there has been taken steps into assuring dataflow between Comos and PDMS. There are integrated tagging functionality and UDA and UDET updates executed through nightly batch jobs. Though the whole data exchange process is somewhat “a hidden secret” [24], as very few engineers know how the complete transactions work. There are no official working procedures or instructions for data exchange and interface, and as Tillmann[25] puts it, the procedures are not aligned with Aker Solutions’ execution model. This section will take a closer look at the solutions in use, and why they are insufficient as engineering tools for future projects. Data consistency between Comos and PDMS is a challenge, and on the recently ended projects Eldfisk - Edvard Grieg there were need of a lot of manual checking and labour due to weak interoperability between the P&ID and 3D model software.

Current solution

The current Comos and PDMS interface exists of three parts:

- A Comos - PDMS Tag assigning
- Nightly batch transfer that will update Comos with PDMS access to the data fields in the xml-file transfer, independent of the tag’s subclass.
- The Construction Assistant - Comos

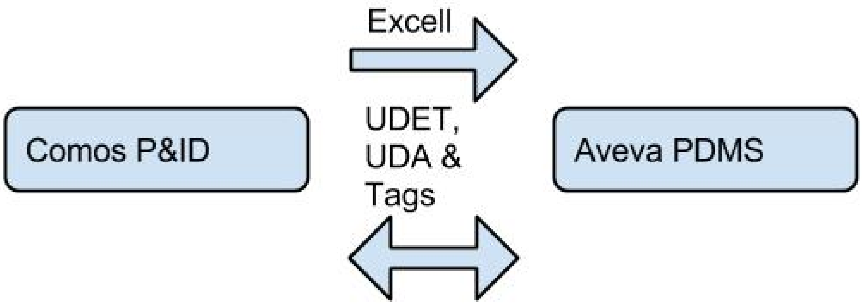


Figure 5.1: Currently used data exchange between Comos and PDMS used by Aker Solutions.

5.1 Comos - PDMS Tag assigning

The Comos - PDMS tag assigning is initiated when an object that exist in Comos, is created in PDMS. The pipeline designer navigates in the 3D model to the place where the pipeline should start and then use a function that is integrated between the applications to "import" the pipeline from Comos into PDMS. The import function assigns ID tags in the software databases, but does not apply pipeline design, connectivity or any equipment on the pipeline such as valves, thermostats, etc. After establishing a connection between the Comos and PDMS objects, the tag's subclass will define which data attributes will initially be transferred between the two applications, by the nightly batch transfer [3].

The information logic of tag assigning between Comos and PDMS can unfortunately be considered feeble when it comes to information preservation. Tags set between object in Comos and PDMS, changing the name of either object in any database results in a name change in the correlating database. The name change takes place in the overnight update that is run every night. Unfortunately, the nameID and the tagID in Comos are rather similar, and it has happened, that an engineer has altered the tagID when he really meant to change the nameID. Changing the tagID makes the same grievous database flaw as to delete an object in Comos, and then remaking the same object with a different name, without ensuring connectivity between the object affected by the change. Essentially, the results being two objects that exist in the databases, but they do not have connectivity with each other. This leads to data that is no longer being updated by the night jobs, and attributes getting outdated, and even promoting critical errors. Being able to do credible and continuous consistency checks may limit or even stop such errors affecting the system in the long run. By applying stricter master slave functionality for the object attributes in the databases, you reduce the possibilities of constructing inconsistencies when designing process systems iteratively.

5.2 Comos - PDMS Nightly batch job

Currently the data transfer is based on a combination of export/import of XML files, and direct inputs based on excel spreadsheets. There is no direct link between the two proprietary databases, which is the reason the attribute update batch job takes place every night. The work flow of the batch job was first developed as a simple XML file exported from PDMS into Comos, updating all

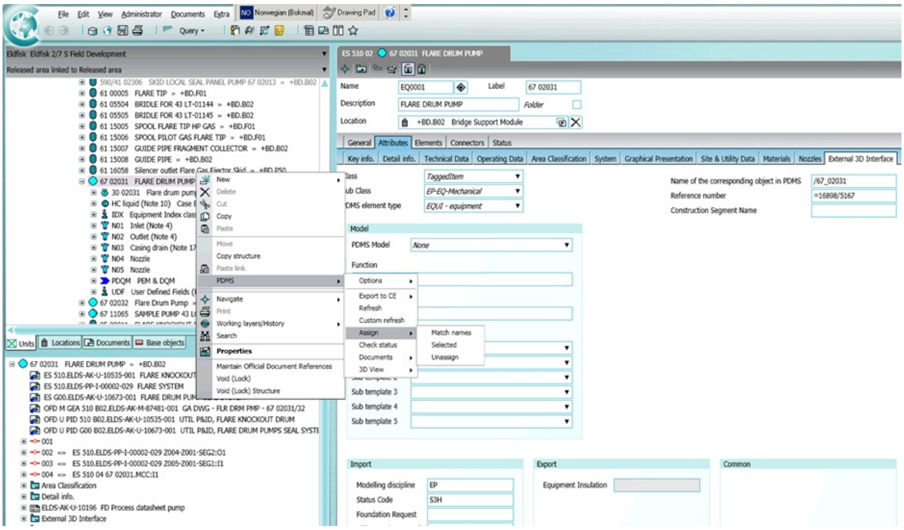


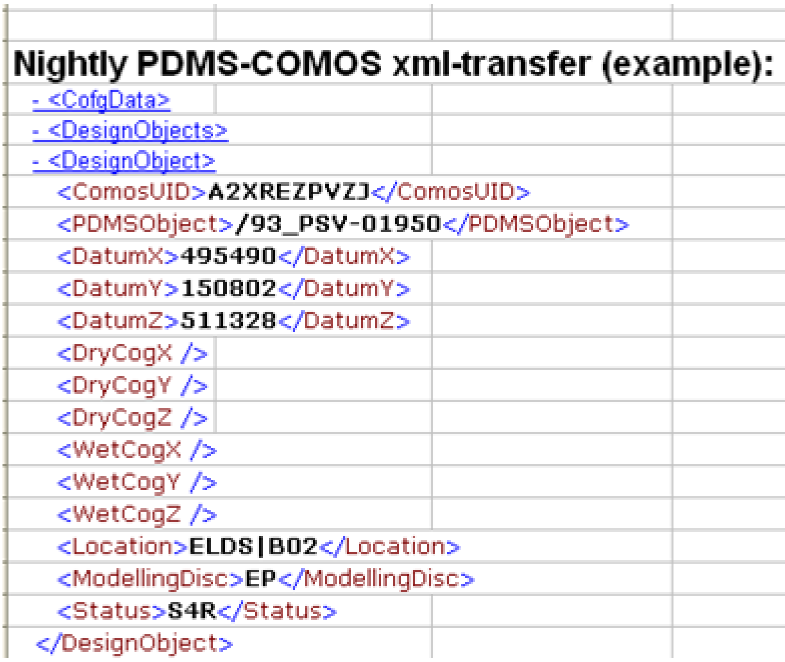
Figure 5.2: Tag assign in Comos done manually

the attributes that were given new values, example in figure 5.3, and updating the object location from where it is located in PDMS into Comos. Later a bidirectional attribute update was developed which gave the users the option to determine which of the attributes in the proprietary databases would be updated. For each attribute the system administrator can set one of the attributes as the master value, which will update the corresponding attribute in the other database if it is inconsistent with the master value.

The batch can also be initialized by using the PDMS "Refresh" commands, in Comos, at wanted level/tag to refresh all data fields defined in the subclass transfer. Though noted the location code is not updated when doing a Refresh. It is also possible to run Unassign and Assign (Match Names) if PDMS Zone is changed for the tag. This will update the area code of the object in Comos.

5.3 The Construction Assistant - Comos

The Construction Assistant is a separate, stripped down version of the Comos product. It displays all the relevant Comos objects such as equipment, process lines and instruments, and allows users to export objects and attributes between



Nightly PDMS-COMOS xml-transfer (example):

```
- <CofgData>
- <DesignObjects>
- <DesignObject>
    <ComosUID>A2XREZPVZJ</ComosUID>
    <PDMSObject>/93_PSV-01950</PDMSObject>
    <DatumX>495490</DatumX>
    <DatumY>150802</DatumY>
    <DatumZ>511328</DatumZ>
    <DryCogX />
    <DryCogY />
    <DryCogZ />
    <WetCogX />
    <WetCogY />
    <WetCogZ />
    <Location>ELDS|B02</Location>
    <ModellingDisc>EP</ModellingDisc>
    <Status>S4R</Status>
  </DesignObject>
```

Figure 5.3: An example of nightly data transfer between Comos and PDMS [3]

Figure 5.4: Aker Solutions also tried hiring in Noumenon Consulting LTD as an external contractor to assist development of the Comos PDMS interface

Comos and PDMS [5]. The software application has a set of functions made to make the process to piping engineering tasks easier, as functions seen in table 5.1.

Having a closer look at table 5.1, The Construction Assistant gives the impression of being a great tool for consistency checking and interoperability between Comos and PDMS, but talking to some of its users, there seems to be some issues with the actual functionality of the software. The first hinder being that The Construction Assistant is only functional if communication between the two systems have been initiated, which has to be done by starting up the Construction Assistant and finding the right project database and then, through PDMS' own separate action, initialize the connection. It should also be mentioned that Comos thereby including the Construction Assistant is run through a citrix server, which again makes the whole process more cumbersome. Moreover, the functionality of exporting items, checking statuses, and general usage of the application is said to be so slow that many of the functions are plainly not ever used. A veteran in the piping discipline proclaimed that the only functions that were in use, were the 3D view and the "Assign" functions. When creating a pipeline, instead of exporting equipment or pipelines from Comos, the objects were created, and then assigned to the corresponding object in Comos.

On the other hand, administrating the application also becomes somewhat of a nuisance. Since most businesses have their own rules and regulations for PDMS modelling and work flows, the Construction Assistant relies upon a custom and specific configuration setup for every project [26]. This resembles configuring a self-distributed database system solution, since data connection maps are made for every new project.

Function	Implementation
Export to CE > TaggedItem	Will attempt to create the selected tagged Comos object below the current element in PDMS
Export to CE > Pipe	Will attempt to create the selected Comos pipe object below the current element in PDMS
Refresh	Will transfer all attributes bidirectionally, so that the two systems are synchronized
Custom refresh	Will display all the attributes that are different in the two systems, the user may choose to update the two systems or not
Assign > Selected	Will attempt to assign the selected tagged Comos object to the current element in PDMS, essentially linking two object counterparts
Assign > Match names	Will attempt to assign the element in PDMS which has a matching name, based on the project naming rules
Check status	Will check the consistency status w.r.t PDMS for the selected Comos object(s)
3D View > Add / Remove / Mark / Zoom / Selected	Manipulates the PDMS 3D view directly from Comos

Table 5.1: The Construction Assistant function overview[5]

5.4 External contractor

Data flow and consistency has always been an important part of the IT solution at Aker Solution. Back in 2011-2012, they realized that there was a need for improvement in the system, and contacted an external contractor to help achieve the necessary interoperability between Comos and PDMS. The acquired firm was Noumenon Consulting LTD, the leading consulting company when it comes to process plant system interoperability. A User Reference Group was put together to look into what Noumenon Consulting LTD were able to offer with their XMpLant solution. In February 2012 the following comments were presented by the URG [27]:

- We see errors on the graphics (objects not showing up right) in e.g. .SVG files (has been identified in .DWG/.DGN files as well)
- Error according to schema
- Schema out of date, even on 9.2 released autumn 2011
- Not well formed <XML>

Aker Solutions noted that if the data transfers were in any way incomplete, developing into errors both in the graphics and the diagrams. Since the data became inconsistent while transferring, the need to manually check the forms opened up for human error. Aker Solutions also had limited resources to "fix" issues on new P&IDs, efficiently nullifying the gain of a decent data transfer. The accountability for the shortcomings was given to the need for a deep knowledge of PDMS to be able to finalize the data transfer. As Noumenon failed to deliver on expectations, the contract was terminated.

Chapter 6

Possible future approaches

Looking into what is already known, trying to take knowledge from previous experiences, this section will look into future possible approaches for data interoperability and consistency checking.

6.1 Custom database / Middle Ware

6.1.1 Expanding current solutions

The first step Aker Solutions should make if they are going to continue using the already implemented system solution, is to expand it to incorporate ISO 15926. The fastest and definitely easiest way to do this is to use the free XMpLant Schema that is developed by Fiatech. The XMpLant Schema can in many ways be called a template for setting up the ISO 15926 logic, and was developed to encourage companies to start using the ISO standard. Noting that setting up an interoperable system using the XMpLant Schema is no easy feat, Aker Solutions do have experience with setting up interoperability, and implementing a middle ware solution will make later data assigning faster, and less time will be used on expanding the data sets.

When constructing a software application that needs to access two different data sources, such computer support has to fulfil two major requirements: (i) it has to provide a single point of access to the miscellaneous, heterogeneous data sources such that they appear to the user as a single, homogeneous data set, and (ii) it must determine the relations between the contents of these data sources

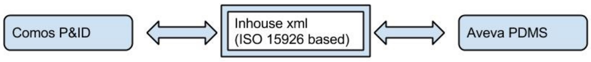


Figure 6.1: Internal interface solution including ISO 15926

in order to detect and reconcile possible inconsistencies in the distributed information. These relations can be both temporal and logical. Temporal relations result from versioning, where information item X is replaced by Y. Logical relations exist, if there is any conditional dependency between information items A and B[28]. The key point being able to represent information based on different data sources in a way that is consistent with what one might expect of an application with a single data source. When broadening the knowledge scope, one can see that this also applies to the different disciplines needed to make a good consistency checker. Not only is the need for good software programming skills necessary, but also a general knowledge of both piping and process engineering. The amount of people with knowledge within all three disciplines is very limited, thus making development of a consistency checker application difficult.

If Aker Solutions was to go forth with expanding their current solution, looking at appendix B and C to assist the implementation of ISO 15926 would be a good start. Appendix B discusses the information model for P&IDs for ISO 15926, such as "plant item", "catalogue", "equipment" and several objects for piping-relations. It also includes TagID and cross-page-connectivity, which is crucial for quality data/diagram exchange between different system interfaces. For the 3D-model, appendix C is moderately more item specific, including extended object types for cable, HVAC and piping. Using the attribute connectivity list for the Comos - PDMS interface, found in appendix A a new interface that includes ISO 15926 can be developed.

6.1.2 Benefits

There are some clear advantages with an approach towards in-house middle ware. When expanding a solution that the company already have knowledge of, there will be employees that have a lot of experience with the software applications at hand, and that who are likely to know of many of the data sets and shortcuts that the company might be able to use. Fast and fluid communication within the company may help the developers understand what the users want, and with insight into what the company does, they are more likely to find good solutions in the system development process. The result being an immediate stability in information exchange, customized in a way the company wants.

6.1.3 Cautions

The software applications that are connected to the middle ware can, and most of the time will, change over time. Both Comos and Aveva PDMS are under constant development, and small and large system changes will take place sooner or later. The effects of such changes depends very much of what development process and future scope the separate companies Aveva and Siemens have for this software. The result being that the custom solution developed by Aker Solutions has a limited time span before maintenance is needed to either expand or adapt the system solution. The unfavourable outcome is that the system requires high maintenance and is high cost, depending on the an unpredictable future development of Comos and PDMS the system can be regarded as fragile, and may have a short lifespan, depending on how well the system can adapt. Unfortunately it is hard to make an efficient database that is adaptable to frequent change, and an unfortunate consequence of high maintenance and cost is leapfrogging.

When a company is affected by frequent change, there becomes a desire to repress unnecessary changes that may cost the company money and man-hours. When faced with the option of upgrading to the latest software edition and upgrading the middle ware, or delaying to a moment where the software upgrade becomes an necessity, the company is prone to delay any changes. The leapfrogging may end up in a complete stop in software development, due to a wish of not wanting to upgrade and develop the middle ware yet again. A risk of becoming ancient due to the resist in wanting to change is prevalent in the ever-changing society we live in today.

Company dynamics can also be affected by an in-house system solution. If something goes wrong with the middle ware and data flow becomes halted or unreliable, and the development/maintenance team struggle to find a solution, tension within the company may grow. Unnecessary tension build up often result in reduced or hostile communication, ending in a less efficient company dynamic.

Taking the benefits and cautions into account, expanding the current solution, based on how future outcomes look today, this solution is not recommended.

6.2 Comos PDMS Integration

Aker Solutions are currently running test on upgrading Comos P&ID to the latest version of the software, Comos Generation 10. The latest version of Comos opens up for the opportunity to implementing Comos PDMS Integration, a Comos PDMS interface, built and maintained by Comos.

6.2.1 Extended solution by Comos

The Comos PDMS Integration promises to solve several of the issues with Aker Solutions current Comos and PDMS Interface, and is in many ways an extension of the Construction Assistant. Among the things Comos PDMS Integration promises there are Interface operations with functionality such as [29]:

- Export of construction objects from Comos to PDMS
- Import of construction objects from PDMS to Comos
- Definition of rules for the owner restriction
  Rules that define precisely under which owner an object must be located in PDMS. The rules are the applied during the interface operations.
- Navigate between the Comos objects and the corresponding PDMS objects.
- Use various 3D view operations: Zoom, Add, Remove, Mark
- Carry out a status check for the objects in both applications.
- If a status check shows that new objects have been created or the attributes of objects that have already been connected have changed: Synchronize the data by importing missing objects or synchronizing existing objects.
- Assign objects to one another specifically or remove the assignment.

Not only being an extension of the already used Construction Assistant, the Comos PDMS Integration application encapsulates many of the internal data solutions that Aker Solutions have developed themselves and use resources on

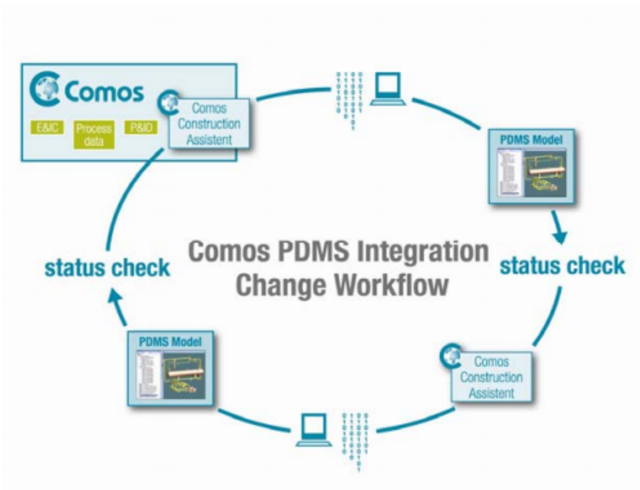


Figure 6.2: Comos PDMS Integration

maintaining for it to run efficiently. By implementing the application, further development of the Comos PDMS interface will become Siemens responsibility since they now have to make sure that the application runs as well as it should. As todays businesses have to become market oriented to obtain or maintain clients, by bringing into focus the needs of the customer. As a big client/customer, Aker Solutions has the ability of getting assistance from Siemens, and pushing the development of the Comos PDMS Integration software in a direction that benefits Aker Solutions.

It should also be mentioned that since the interfaces data-exchange is built around the data hub, there is little chance that there will be loss in data because of I/O or import/export data transactions.

6.2.2 Issues

Looking into what Siemens have delivered before, and talking to some people that have experience with Comos and Comos extensions, there are some worries with the interface. If the interface is built upon the same principles as Comos Construction Assistant, the chances are that the slow execution of the different functionalities makes the application inefficient. Running big calculations on data sets that have long loading times limits the efficient use of the interface to overnight jobs only. Furthermore, there is a problem with how the system does consistency checks, comparing pipe for pipe and not area for area, limiting the wanted checking scope of Aker Solutions.

Since the piping discipline is downstream of the process discipline, they have the main responsibility when it comes to consistency between the P&ID and the 3D model, which opens up for some interaction inconveniences. Since both the Comos PDMS Integration software and PDMS has to be open at the same time to execute, the piping engineers have to use an application wide from what they normally use, and are accustomed to. As well as having to run several different application simultaneously reducing effective use, Aker Solutions execute Comos through citrix, which exposes them to server timeouts. Essentially the Comos PDMS Integration is at the most a decent extension of the tools already in use by Aker Solutions, but cannot be called a solution to the data flow and consistency issue that the company has. That is why Comos PDMS Integration is not a recommended solution, but Aker Solutions should pay attention to the future development of the software, as it may improve greatly over time.

6.3 P&ID import into PDMS

Aveva have for some years, and is, currently working with developing applications and interfaces to expand their plant design management system. Their development towards schematic process system software has resulted in several different applications that revolve around different fundamentals. Aveva Diagrams is based on Microsoft Visio, a diagramming and vector graphics application which is part of the Microsoft Office suite. Aveva Schematic Model Manager is a software extension used to import P&ID data into Schematic databases, and to view and manage the imported data. Lastly there is Aveva P&ID which is based upon Autocad, a CAD and drafting tool available since 1982. Aveva P&ID is does not import P&ID data in any way, and is not feasible as a substitution to Comos due to the fact that Comos is such an integrated software system in Aker Solutions' projects (figure 4.1), and also works as a data hub. Further research will therefor only be carried out on Aveva Diagrams and Schematic Model Manager.

6.3.1 Benefits of Aveva tools

Before looking into which of the Aveva schematic tools that is better suited to use in the Aker Solutions environment, there are some clear benefits of using a tool developed by Aveva to operate within Avevas own interface. When Aker Solutions cooperated with Noumenon using XMpLant, one of the key factors that halted the development of a useable interface was the Aveva PDMS part of the interface. Whenever Aker Solutions have expanded their Aveva software suit, Aveva offers to send consultants to assist the setup off the system. By getting the right people with vast knowledge of the interface and limitations of PDMS, it will be a lot easier to set up a functional interface.

PML (4.5) is another factor that makes Aker Solutions customization noticeably easier. With easy access to data and attributes, batch running is much more approachable and development friendly. Aker Solutions have several engineers that are active PML programmers and develop add-ins and customizations in the PDMS interface. Since bulk data handling is not a part of several Aveva functionalities, PML can be used to increase bulk handling and therefore efficiency.

Being able to work with data sets that are native to the Aveva interface makes calculations and data transfer faster than exporting and importing data between Comos and PDMS, which is crucial when working with the huge data sets that are in production at Aker Solutions' plant system projects. By saving project data on the same server eliminates data transfer time, and issues with unstable connections. Unfortunately, due to the lack of time for testing due to software contract negotiations, there has not been done any testing on the improved speed of a Aveva native interface, though this should be looked into by Aker Solutions when deciding what the future of plant system interfaces they are going to use.

6.3.2 Aveva Schematic 3D Integrator

The Schematic 3D Integrator is a provided add-in to Aveva Plant DESIGN [4]. The Integrator add-in is provided by Aveva to assist in comparing and updating schematic and 3D model data, in an integrated environment, enabling quick and easy browsing and reporting on that data.

Avevas comparing solution

The Schematic 3D Integrator software extension is a crucial part of what makes importing schematic data into PDMS a beneficial way of executing consistency checking. By importing schematic diagrams and data into the Aveva interface, three crucial functions, for consistency checking, is made available to the user:

Compare 3D Design against Schematic can compare 3D objects against their corresponding schematic objects and report any inconsistencies in connectivity or attributes to easily configured rules.

Build 3D Model Data from Schematic Data can create 3D objects using data from their corresponding schematic objects, connect objects such as equipment and Pipework, and set key attributes such as tags and process data using easily configured rules.

Link 3D Model Data from Schematic Data can link existing 3D objects with their corresponding schematic objects, enabling verification of consistency of 3D design against schematic design.

These functions are already available through Siemens software, but are inefficient in such a manner that it becomes of no benefit to the engineers at Aker Solutions. By importing the schematic data, the schematics and the 3D-model will be located on the same server, which makes for increased efficiency when it comes to running checks and compares on the data. The export/import of data from the external source into the Aveva interface can be done automatically and subsequently as changes are made to the source diagrams, as well as initializing overnight jobs.

Since the current Aker Solutions interface is built around a combination of self-implemented and Comos solutions, implementing the Schematic 3D Integrator into the interface is no simple task. But with assistance from Aveva, the software application should be implemented without greater issues. That is why an implementation of the Schematic 3D Integrator is recommended and a part of the business case.

Testing

When testing the add-in with Aveva Diagrams data files, there were some issues that came to light. The Diagram Viewer, that can be seen in figure 6.3, had some issues when trying to display the .SVG diagram files. Only after updating from Aveva PDMS 12.0.SP2 to latest version Aveva PDMS 12.0.SP4, the diagrams would display correctly in the Schematic 3D Integrator. It is uncertain how the Diagram Viewer interacts with schematic data that has been imported through ISO 15926, but this shows that the applications are currently in development, and that they are still prone to bugs and malfunctions. There were also issues when attempting to use the "Build 3D Model Data from Schematic Data" functionality, where the 3D model object would be created, but would not be displayed in the 3D graphical view. Probably there were issues with definitions in the "configured rules", where it had not been determined what kind of model the 3D view was to display, when given the schematic object. These kind of issues are surely in Avevas focus when continuing developing the Schematic 3D Integrator, and would also possibly be resolved with assistance from Aveva consultants when building a test project.

6.3.3 Aveva Diagrams

First viable solution

Aveva Diagrams was the first software application that was mentioned when discussing the future of Comos PDMS interface with Aker Solutions engineers. It is the diagram application that Aveva is promoting the most actively at the moment, by sending consultants to teach and test the software with several engineers in different companies. The lengthy feature list advertised on Avevas home website page, include:

- Efficient dedicated diagram design

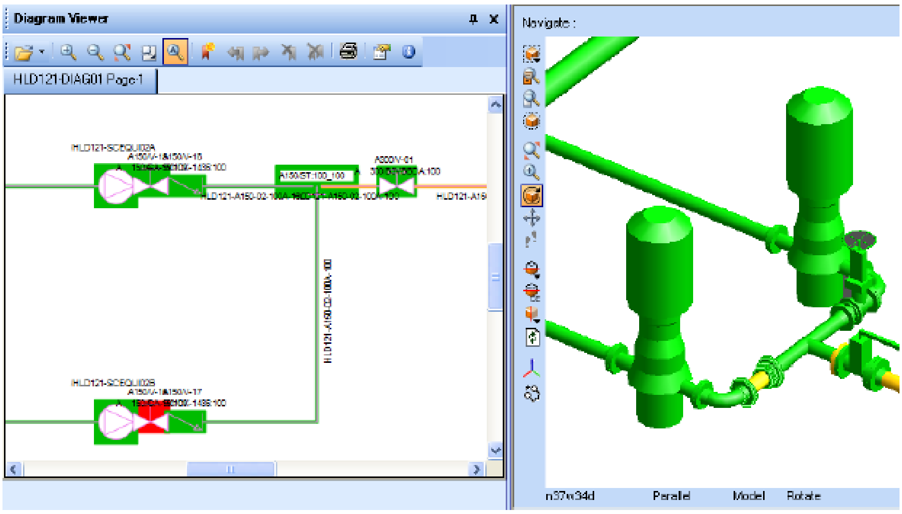


Figure 6.3: Compare schematic diagram and 3D model using Schematic 3D Integrator

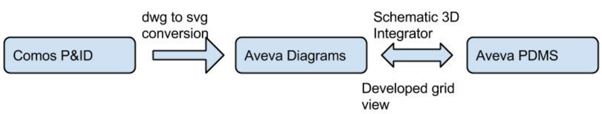


Figure 6.4: The Aveva Diagrams import/compare data flow, with conversion before import

- Powerful, rule-based auto-formatting functions
- Several automated, intelligent solutions

In online reviews [30] and on forums Visio is promoted as an easy to use, easy to learn cheaper solution to the more market familiar Autocad. But, when talking to the engineers that have gotten their hands on the software, the feedback is not as optimistic. There were concerns with how the software has to be adapted to every project, that there were very little out of the box, ready to use functionality. The symbols that are made to represent objects in the schematic diagrams have to be made or imported one by one through the "Import Shape Wizard" [31]. The wizard is not made for bulk creation and was found inconvenient by admins that had to set up several symbols or shapes before handing the shape stencils over to process engineers. It is clear that the customizability in Aveva Diagrams is vast, but it is not necessarily needed. The application is not out of the box adapted to do advanced P&IDs which makes it less useful for Aker Solutions which needs it for a specific task, and not the option of expanding the use to other areas as well.

Testing

To be able to use Aveva Diagrams for the Comos PDMS interface, there are some default frustrations that need to be considered. The easiest way to transfer P&IDs from Comos into Aveva Diagrams, is to export the P&IDs from Comos as .DWG files, convert the files from .DWG to .SVG files using 3rd party software, before importing the .SVG files into Diagrams. There are several major complications with the export/import process:

- Risk of meta data loss in every import/export/conversion
- Time-consuming data conversion
- Unreliable 3rd party software or 3rd party software cost

When designing new diagrams, the interface was firstly a graphic diagram designing tool. None of the automated P&ID development tools that can be found in established software application were present. There was, simply put, a need for a process systems admin to implement all the necessary functions and symbols, to make an efficient functioning tool for the everyday user. To develop the functionality that is needed for Aker Solutions to use the software, completely removes the efficiency gain of making an interface for Comos PDMS interoperability.

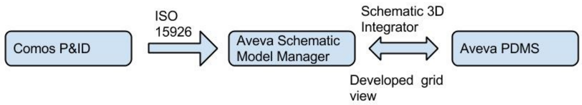


Figure 6.5: Exporting schematic data from Comos using the ISO 15926 standard, importing the data with the Schematic Model Manager and the executing consistency checking with Schematic 3D Integrator.

There was also an issue where deleting a diagram would remove the diagram, but leave the database connected to the diagram unaffected. This is a grievous error that can cause trouble with the data structure, and become a major nuisance for the whole project if not corrected before the unconnected database does any harm.

Lastly the loss of meta data makes the Comos to Aveva Diagrams approach fruitless to what Aker Solutions wants out of the software. Even if engineers were able to secure the conservation of meta data (UDETs and UDAs) for a specific case, the achievement would almost definitely be case specific, and customization would be needed for every new project that was to use this specific approach.

6.3.4 Aveva Schematic Model Manager

Unfortunately, Aker Solutions were not able to get a hold of Aveva Schematic Model Manager for testing purposes. This was largely due to stagnant contract negotiations regarding software application, delivered by Aveva , built around PDMS. On the other hand Aveva Schematic Model Manager seems to be the application that assists the assurance of automated consistency checking between Comos P&IDs and the Aveva PDMS 3D model. Aveva are committed to the use of open standards for the interchange of plant data between Aveva and 3rd-party products [32]. The Schematic Model Manager can, according to Aveva, import schematic diagrams based on ISO 15926 that maintain UDETs and UDAs. Since both Comos and Aveva have been a part of the companies that have contributed to the development and use of ISO 15926 [33], the export/import should be carried out without the loss off meta data. When the schematic diagrams have been imported into the Aveva interface, fast consistency checking can be done using integrated software applications, such as Aveva Schematic 3D Integrator6.3.2.

The path of the schematic drawing can is visualized in figure 6.6, where 3rd Party P&ID can be imported with the Schematic Model Manager, into the Schematic Model Database, where it again can be compared with the 3D Model Data, using the Schematic 3D Integrator.

Much of what have made the Schematic Model Manager so hard to come by, is the fact that Aveva has changed its name several time over the last couple of years. When trying to find information about standard schematic import into PDMS, there are application that fits the mould, but most of the time you cannot find the application in Avevas application list. The Schematic Model Manager has gone from being called P&ID Manager, which should not be confused with Aveva P&ID. The Schematic Model Manager must not be mistaken for the Schematic 3D Integrator, which is another useful tool reviewed in this thesis. The Schematic Model Manager cannot be found in Avevas application list today since it has in the latest update become a part of Aveva Engineering, which is a software group that is used to manage the evolving multi-discipline engineering data for tagged items such as lines and equipment [34].

A deeper analysis of how the implementation will affect Aker Solutions will be assessed in the business case 7.

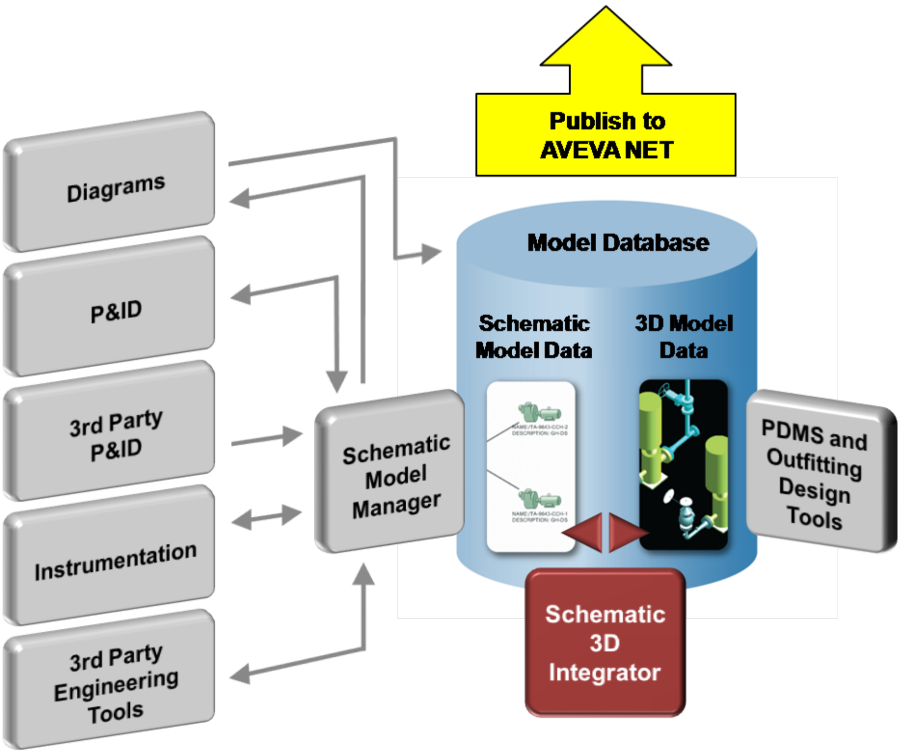


Figure 6.6: The complete Aveva Model Database Interface [4]

Chapter 7

Business case

7.1 Executive Summary

This business case outlines how implementing an automation of consistency checking will affect Aker Solutions current process concerns, the benefits of the project, and recommendations and justifications of the project. The business case is based on implementing a solution based on schematic diagrams exported from Comos as .xml files based on ISO 15926, imported into Avevas schematic databases using the Schematic Model Manager, and comparing and executing consistency checking between the schematic data imported into Aveva and the 3D model, native to the Aveva database.

7.1.1 Issue

The issue which is being addressed in this business case is consistency checking between P&IDs and 3D models, represented in Siemens Comos and Aveva PDMS respectively. It has been noticed that a significant amount of resources and man hours are used on manual consistency checking and data follow-up. The current way of executing consistency checking is done with yellow line mark-up, using P&ID diagrams on paper, the 3D model, the data specs and a yellow marker. The procedure consists of drawing up yellow lines for each pipeline or piece of equipment that has been checked by an engineer. This not only leads to time unnecessarily spent, but also opens up for human error, which can lead to grave errors downstream in a process plant project.

Aker Solutions therefore wishes to automate and streamline the way consistency checking is being done, by utilizing the process plant standard ISO 15926, and integrated software applications delivered by Siemens and Aveva.

7.1.2 Anticipated Outcomes

Automating the consistency checking between P&IDs and 3D-models will reduce the amount of time spent on consistency checking, increase the stability of data transfer, and reduce the risk of human error when updating and maintaining process plant databases. If the consistency checking automation is developed as hoped, correlation errors in the databases will present themselves. That way the engineers do not have to look for errors like needles in a haystack, increasing the quality of projects produced by Aker Solutions.

The solution fit the Norwegian engineering way, where implementing an increased cost system eventually reduces cost by both reduced man-hours and errors. When you can make sure of a consistent data checking with reduced accessibility for human error, the result is a decreased amount of necessary iterations, due to project errors. You end up with a shorter delivery time, and a product with increased quality and value.

7.1.3 Recommendation

Various options and alternatives were analysed to determine the best way to improve consistency checking between P&IDs and the 3D models. The approaches were based on the already developed interface between Siemens Comos and Aveva PDMS, looking at ways of altering and expanding the interface. The selected approach described in this business case was seen as the most efficient solution, and the solution that was the most flexible to future software and market changes. By utilizing ISO 15926, Aveva Schematic Model Manager, and Schematic 3D Integrator, Aker Solutions will share the responsibility of future software development with Aveva. This will ensure that Aker Solutions will not be kept back, by relying on having to develop the interface themselves, and removing the risk if leapfrogging on the ever-changing software market. The approach will reduce time spent on excessive tasks that take focus and time away from more advantageous tasks. The technology will give the engineers the ability to:

- Easily import the schematic diagrams of a process plant into the Aveva interface

- Visualize both the P&ID and the 3D model without printing out paper diagrams
- Automatically seek out errors between the P&ID and the 3D model
- Access fast attribute editing in PDMS

7.1.4 Justification

The automation of consistency checking will result in greater efficiency, reduced amount of grave errors in projects, and workers will be happy they do not have to carry out repetitive menial tasks.

It is difficult to overestimate the value of being able to exchange information with anyone without fear of transcription error, while maintaining the precise meanings of all terms, even though you know nothing at all about your partner's internal work processes and methods of data storage [2].

There has not been recorded any time spent on executing consistency checking in the projects, and it becomes hard to justify the approach based on cost and numbers. When talking to piping engineers at Aker Solutions there is a consensus that the way consistency checking is being done today, is inefficient and outdated compared to the other tools they have access too. Automating consistency checking is the next logical step in assuring that Aker Solutions stays in a market and business position as a quality producer of process plant projects.

A good rule of thumb for software development for engineering purposes is, the time it takes to develop the application should be half the amount of time that the application recover when a task is executed using the software. Seeing that Aker Solutions will not stop producing process plant systems in the near future, it is secure to say that the implementation of Aveva software to help consistency checking is justified.

7.2 Problem Definition

7.2.1 Problem Statement

When Aker Solutions decided to start using Comos as a data hub for process plant systems, there became an issue to ensure that all data representing the objects of the process plant were consistent with each other. Since Aveva PDMS and Siemens Comos had two separate databases, there was no automated way of ensuring that the data set in one database was correlating with the data set in the other. On paper this was not an issue, since the piping discipline designing the 3D model using PDMS, was downstream from the process discipline using Comos, the PDMS database would simply be updated when the process discipline were finished creating the P&IDs. This is a good work flow model in theory, but since producing process plants have become an iterative procedure, the two databases develop simultaneously. The result of the iterative process was that the connectivity between correlating objects in Comos and PDMS had to be assigned after they were created, and attributes and information had to be updated while the plant was being developed as new iterations were concluded.

Aker Solutions started executing nightly batch jobs using xml to update the databases, and Comos and PDMS to assign the correlating objects using tags. Aker Solutions have realized that the current interface is insufficient; taking the latest Eldfisk project that had 7549 individual piping elements that needed monitoring as an example. Forcing engineers to monitor thousands of objects manually is unjustifiable when the interface can and should be automated, and inconsistencies can be presented to the engineers, instead of having the engineers look for them.

7.2.2 Organizational Impact

The Comos PDMS Interface project will impact Aker Solutions in several different ways, from project execution to company dependency. The following provides a high-level explanation of how tools, processes and company relations will change due to the new implementation.

Processes: Less time will be used on ensuring consistency between P&IDs and 3D models, and better consistency checking will make sure that process plant project delivery will be of a higher quality.

Roles and Responsibilities: The roles and responsibilities will not change drastically, but the execution of consistency checking will be much more efficient, giving administrators and engineers more time for working on design.

Hardware/Software: The export of schematic diagrams from Comos, and the implementation of Aveva Schematic Model Manager and Schematic 3D Integrator will in the beginning expand the number of licenses needed per engineer. While in the long run the implementation may reduce the number of licenses for Comos Construction Assistant.

Small server cost to Aveva database servers may come due to duplicate representation of P&IDs in both Comos databases and Aveva Schematic databases.

Company: The implementation of additional Aveva software applications will make Aker Solutions more dependent on Aveva, the software they deliver and future development.

7.2.3 Technology Migration

To efficiently implement the new software applications and interface, a phase approach has been developed in order to minimize the disruption of day to day operations. This high level overview shows how the phase approach will carried out:

Phase I: Software and licenses will be purchased, and a test project will be set up by software administrators with assistance from Aveva.

Phase II: The test project will be assessed and tested by a small user group of piping engineers that have experience with consistency checking.

Phase III: All employees that benefit or is affected by the new interface will receive training on the new ISO 15926 and Aveva Schematic software interface.

Phase IV: The interface will be implemented in a real time project, were feedback from user groups will be logged continuously. Back up of the old interface will be available for unforeseen complications but will not be available for active use.

Phase V: Research for optimizing the interface will go on continuously as new technology is available and Aveva issues updates.

To maximise the efficiency of the new interface, Aker Solutions engineering will only need a few licenses for Aveva Engineering. As the export/import functionality between Comos and PDMS should be executed every time there is an update in the P&IDs located in the Comos database, Aker Solutions can expand the interface to automatically push ISO 15926 P&IDs from Comos, into a “conversion que”, that is continuously running, thereby removing the need for manually import the schematic diagrams into PDMS. By doing so, the updated schematics will be available to the piping discipline through the Schematic 3D Integrator shortly after the P&IDs are “saved” in Comos.

7.3 Project Overview

The Comos PDMS interface project overview provides details for how this project will address Aker Solutions business problem regarding consistency checking between Comos P&IDs and PDMS 3D models. The overview is based on this master thesis, Logical and Physical Process Design. If the project moves forward based on Aker Solutions decision on future Comos PDMS interface, the project overview should expand based on their knowledge base, and should include a greater level of detail in working towards the project plan.

7.3.1 Project Description

The Comos PDMS interface project will, based on the Logical and Physical Process Design thesis, expand the current Comos PDMS interface to include automated consistency checking between Comos and PDMS. The project will execute this automation by implementing the software application Schematic Model Manager, as well as start using Schematic 3D Integrator and schematic ISO 15926 export by Comos.

The implementation will be executed by Aker Solutions software administrators, where a setup of Comos schematics export will be executed every time a new updated version of the P&ID is saved, and the converted into Aveva Schematics using the Schematic Model Manager. The set up should be similar to the PDMS to NavisWorks conversion, where the software application that executes the conversion is running continuously. The piping engineers will get access to Aveva Schematic 3D Integrator, which uses the Aveva schematic data to execute consistency checking and object creator to ease the load of manual tagging and consistency checking.

.

7.3.2 Goals and Objectives

The goals and objectives of the Comos PDMS interface supports the Aker Solutions corporate objective of delivering quality results, as ensuring consistency is crucial for quality and efficient delivery of process plant projects. Table 7.1 lists the business goals and objectives of the project.

Business Goal/Objective	Description
Improve staff efficiency	Automating consistency checking will allow piping engineers especially to use their time on piping design and development, instead of doing manual attribute and consistency checking.
Reduce long term costs	Since the implementation is an increased cost solution, the profits are found in long term increased quality, and shorter time to delivery, for process plant projects.
Reduce risk of human error	Automating consistency checking will reduce the reliability of human monitoring of attributes and data, which will reduce risk of human error.
Reduced blowback	By increasing quality of consistency checking, less database errors pass through to later stages of process plant projects, where such error can cause huge amount of time consuming rework. In the long run the improved interface will reduce time to market.

Table 7.1: List of the business goals and objectives that the Comos PDMS Interface project supports and how it supports them.

7.3.3 Project Performance

The actual measure of project performance is hard to measure based on how projects operate today since Aker Solutions does not record any data related to consistency checking. Since consistency checking is a part of a bigger work process, Aker Solutions do not record specific numbers connected to that specific work task. The following table 7.2 should be used as a starting point for Aker Solutions when it comes to recording data connected to consistency checking. The performance measure table should be expanded and have greater detail as the Comos PDMS Interface project progress.

7.3.4 Project Assumtions

The following assumptions apply to the Comos PDMS Interface project. Further assumptions that are identified during the execution of the project must be added to the list:

- The export/import of schematic diagrams between Comos and PDMS using ISO 15926 is without metadata loss and retains connectivity

Key Resource/Process/Service	Performance Measure
Consistency checking	The amount of inconsistencies the checking finds over the project time period. Acquiring the share amount of errors is a good picture for success.
Errors capable of blowback	The amount of errors that pass the consistency checks which results in blowbacks and rework.
Software and System Maintenance	Man hours spent on software and system maintenance, the amount of work for software admins will increase in a short term period. Important measure that the final solution is an improvement over the old interface.
Staff Resources	The amount of time spent on consistency checking, as this is manual labour, this should reduce greatly from the current situation.

Table 7.2: Definitions for measuring performance for consistency checking between P&IDs and 3D models in process plant projects.

- Aker Solutions software administrators are available for test project setups
- There are Aveva consultants available for assisting in setting up test project
- Sufficient funding for software licenses and training
- Project has executive-level support and backing

7.3.5 Project Constraints

The following constraints apply to the Comos PDMS Interface project. Further constraints that are identified during the execution of the project must be added to the list:

- Project owner constraints, whether the project should be in a collaboration with an upcoming Aker Engineering project, or if the project is developed by Aker Business Services independent of future projects
- Constrained by Aveva development, how far the development of the software applications in question has come. Any further development of the interface is connected to the development direction Aveva takes for the software applications.

- Constrained by server and export/import speed

7.3.6 Major Project Milestones

Since there are no project dates specified by Aker Solutions, the project should be executed before the next major process plant project starts. With major milestones listed:

1. Project charter
2. Project plan review and completion
3. Project kick-off
4. Phase I - IV complete
5. Project completion

Phase V is a continuous task that must have follow-up as long as there is need for the Comos PDMS Interface.

7.4 Cost Benefit Analysis

To make a thorough cost benefit analysis, there is a certain need for numbers correlating to cost and saving. Unfortunately, Aker Solutions project administrators do not keep records of the amount of time is used on consistency checking or interface implementation or upkeep. Even though, this analysis wants emphasize the needs for good consistency checking and software interoperability. The need to increase the cost efficiency of process plants is leading to business practice that depend on the efficient integration and sharing of plant information in a computer processable form [1].

Over time, process plant systems have grown a lot in both size and complexity, and the numbers of tagged items in projects are not becoming smaller in any way. The following numbers are recorded from two of Aker Solutions recent process plant projects [27]:

- Ekofisk Zulu: 200 P&IDs, 50 000 Tags
- Eldfisk South: 300 P&IDs, 110 000 Tags

A software administrator responsible for consistent tagging estimated that checking consistency of a tag took approximately between 30 to 60 seconds per tag resulting in over 1000 man hours of repetitive consistency checking on Eldfisk South alone. The measurement is not regarding that every tag has to be checked several times, due to the iterative nature of process plant development. The results should therefore be multiplied by the number iteration the project is subjected to. The result is a excessive amount of time spent on manual labour that is exposed to mistakes and errors occurring. By automating the process, the amount of time needed for making sure that the P&IDs and 3D model is consistent becomes a fraction of the time spent checking consistency manually. In an attempt to show how much inadequate interoperability between CAD software cost engineering companies, table 7.3 show the total cost of inadequate interoperability for General Contractors in the US year 2002. Totalling in on an incredible 1.8 Billion dollars, it is easy to understand that there is great room for improvement. Especially when it has been pointed out by Aker Solutions engineers that the current interface is inadequate, and should be improved.

Life-Cycle Phase	Cost Category	Cost Component	Average Cost per Square Foot	Average Cost per Square Meter	Inadequate Interoperability Cost Estimate ($Thousands)
Planning, Engineering, and Design	**Avoidance Costs**	Inefficient business process management costs	0.14	1.55	163,674
		Redundant CAx systems costs	—	—	—
		Productivity losses and training costs for redundant CAx systems	—	—	—
		Redundant IT support staffing for CAx systems	—	—	—
		Data translation costs	—	—	—
		Interoperability research and development expenditures	0.0006	0.006	630
	Mitigation Costs	Manual reentry costs	0.16	1.74	184,028
		Design and construction information verification costs	0.006	0.06	6,302
		RFI management costs	0.12	1.24	131,299
	Subtotal	Avoidance costs	0.14	1.55	164,304
		Mitigation costs	0.28	3.05	321,629
		Subtotal	**0.43**	**4.59**	**485,933**
Construction	**Avoidance Costs**	Inefficient business process management costs	0.82	8.78	927,487
		Redundant CAx systems costs	—	—	—
		Productivity losses and training costs for redundant CAx systems	—	—	—
		Redundant IT support staffing for CAx systems	—	—	—
		Data translation costs	—	—	—
		Interoperability research and development expenditures	0.003	0.03	3,571
	Mitigation Costs	Manual reentry costs	0.11	1.19	126,047
		Design and construction information verification costs	—	—	—
		RFI management costs	0.16	1.74	183,818
		Construction site rework costs	0.01	0.11	11,356
	Delay Costs	Idle employees costs	0.01	0.12	12,988
	Subtotal	Avoidance costs	0.82	8.78	931,059
		Mitigation costs	0.28	3.04	321,221
		Delay costs	0.01	0.12	12,988
		Subtotal	**1.11**	**11.94**	**1,265,268**
Operations and Maintenance	**Mitigation Costs**	Post construction redundant information transfer costs	**0.04**	**0.48**	**50,419**
Total Cost					**1,801,620**

Table 7.3: Costs of Inadeqaute Interoperability for General Constractors [6]

7.5 Alternative Analysis

Alternative Option	Reason For Not Selecting Alternative
Keep the current interface	• Not aligned with Aker Solutions' execution model • Prone to human errors • Cumbersome work process for integrated software applications • Lack automation
In-house middleware solutions	• Short life span • Vulnerable to leapfrogging • Risk of becoming ancient
Implement the Siemens developed Comos PDMS Integrator	• Slow interface • Cumbersome work process • Limiting consistency checking scope

Table 7.4: The alternative options have been considered to address the business problem, but not selected due to a number of reasons.

Chapter 8

Conclusion

Particularly in the process industry, often planning and realisation of complex industrial plants span a period of several years. During this period of time, a huge amount of engineering data from different sources and in different file formats accumulate, which constantly change due to frequent revisions. Using individual engineering database solutions without automated bidirectional data exchange, so-called isolated applications, can at best be realised by investing manpower, time and costs that cannot be economically justified [7]. Given the evolutionary nature of the design process for process plant projects, there often is substantial changes to the piping design over the course of the project. This is why effective consistency checking is of such great importance, to ensure a quick and high quality deliver of the process plant project.

To get a better understanding of how consistency checking is executed at Aker Solutions the thesis has taken a closer look at Siemens Comos, Aveva PDMS and the interface between the two applications. The current interface is a combination of application add-ins and an in-house solution, which is not aligned with Aker Solutions' business model, and is inadequate for the task it is set to carry out. That is why it is crucial for Aker Solutions to take actions to further develop the Comos PDMS interface, thereby ensuring suitable consistency checking between P&IDs and 3D models.

What makes creating a functional consistency checker between P&IDs and 3D models so difficult at Aker Solutions, is that a lot of the Comos PDMS interface are "hidden secrets" and one man solutions that have failed to be documented correctly. Aker Solutions should document the interface more thoroughly, making the interface more transparent and available for discussions.

Only by realising that there is room for improvement and opening a forum for discussion, will the Comos PDMS interface improve and keep up with the future changes of CAD software development.

Taking knowledge from previous experience with procedures and software applications, the thesis takes a closer look at what available options Aker Solutions have for developing an adequate interface between Comos and PDMS. By understanding the benefits and disadvantages of the different system solutions, the research proved implementing a solution based on the Aveva Schematic Model Manager, would be the immediate most beneficial course of action. By exporting the P&IDs from Comos based on ISO 15926, the Schematic Model Manager can import the Schematics into the Aveva interface. Within the Aveva interface, consistency checking and data comparing can be done with the 3D model in PDMS, using the Schematic 3D Integrator. Based on what can be read from the administrator and user guides, this should be done without loss of UDAs, UDETs and meta data.

It is important to mention that the interface should, for future projects, always be under review. The solutions this business case is based on, is the current best way of ensuring consistency between P&IDs and 3D models. This is not necessarily the case for the future, as it is subject to change as the CAD software market is under constant development. The Comos PDMS Integrator may one day be a much more efficient and integrated application, which makes it a definitive contender as a consistency checking solution.

The preservation of consistency between P&IDs and 3D models has great cost benefits. In many ways it is difficult to measure the actual cost of ineffective consistency checking since it affects several cost areas in smaller ways. Inadequate software applications cost money to maintain and engineers use more time on repetitive menial tasks that allows for human errors to occur. Human errors may cause rework and unforeseen cost downstream from where the actual consistency checking takes place. To have Aker Solutions use some effort trying to estimate the value of good consistency checking and software system interoperability is the first step in making any significant changes to the Comos PDMS interface.

Bibliography

[1] ISO15926. Industrial automation systems and integration. *Integration of life-cycle data for process plants including oil and gas production facilities*, 2004.

[2] Fiatech. An introduction to iso 15926. pages 5 –, 2011.

[3] Erik Tillmann. Comos and pdms interface - how it works. *Aker Solutions. PowerPoint Presentation*, 2013.

[4] AVEVA Solutions Ltd. Schematic 3d integrator user guide. 2010.

[5] Gaute Schjerve. User guide - the construction assistant. *Aker Solutions, Internal Documentation*, 2011.

[6] M. P. Gallaher, A. C. O'Connor, J. L. Dettbarn Jr., and L. T. Gilday. Cost analysis of inadequate interoperability in the u.s. capital facilities industry. *NIST GCR 04-867*, 2004.

[7] Manager Public Relations Evelyne Kadei. Comos pdms integration - bridging of 3d software pdms and comos p&id world. *Press release*, 2009.

[8] B. Bayer and W. Marquardt. Towards integrated information models for data and documents. *Computers and Chemical Engineering*, pages 1249–1266, 2004.

[9] S.M. Embury, S.M. Brandt, J.S. Robinson, I. Sutherland, F.A. Bisby, and W.A. Gray. Adapting integrity enforcement techniques for data reconciliation. *Information Systems*, pages 657–689, 2001.

[10] Michael P. Gallaher, Alan C. O'Connor, John L. Dettbarn Jr., and Linda T. Gilday. Cost analysis of inadequate interoperability in the u.s. capital facilities industry. *NIST*, 2004.

[11] R. M. Eastman. Engineering information release prior to final design freeze. *IEEE Transaction on Engineering Management, 27*, page 37, 1980.

[12] AVEVA Solutions Ltd. Getting started with pdms. *User manual*, 2007a.

[13] Siemens. Comos p&id process operations manual. 2011.

[14] Alfred Tarski. Introduction to logic and to the methodology of deductive sciences, second edition. 1946.

[15] Hans Teijgeler. Iso 15926 data modeling topics. 2013.

[16] R. Schneider and W. Marquardt. Information technology support in the chemical process design life cycle. *Chemical Engineering Science*, pages 1763–1792, 2002.

[17] Hans Petter Hildre, Ola Jon Mork, and Vilmar Aesoey. The maritime innovation factory. *Modular Architectures -the third industrial revolution*, pages 69–80, 2010.

[18] B. Bayer. Conceptual information modeling for computer aided support of chemical process design. page 787, 2003.

[19] B. Bayer and W. Marquardt. A comparison of data models in chemical engineering. *Concurrent Engineering Reaserch and Application*, pages 129 – 138, 2003.

[20] Siemens. Highest consistency for process plant engineering. 2013.

[21] Erik Tillmann. Comos urg and methods, system and area engineering method. *Aker Solutions. PowerPoint Presentation*, 2013.

[22] AVEVA Solutions Ltd. Tags user guide. 2012.

[23] AVEVA Solutions Ltd. Programmable macro language. *Training guide*, 2009.

[24] Erik Tillmann. Experience transfer eldfisk - edvard grieg, comos-pdms 3d model layout engineering development. *Aker Solutions. PowerPoint Presentation*, 2013b.

[25] Erik Tillmann. Comos urg kick off meeting. *Aker Solutions. PowerPoint Presentation*, 2012a.

[26] Gaute Schjerve. Administrators guide - configuring the construction assistant. *Aker Solutions, Internal Documentation*, 2007.

[27] Jann Slettebakk CIO. Xmplant and aker solutions, iso 15926 and akso, a view from the outsidel. *Comos ->XMpLant->SP P&ID*, 2012.

[28] Andreas Weisner, Jan Morbach, and Wolfgang Marquardt. Information integration in chemical process engineering based on semantic technologies. *Computers and Chemical Engineering, Volume 35, Issue 4*, pages 692–708, 2011.

[29] Siemens AG. Pdms comos interface 3d integration operation. pages 2–19, 2014.

[30] Radica Software. Is visio a sensible alternative to cad systems/? 2014.

[31] AVEVA Solutions Ltd. Diagrams user guide. *User manual*, 2010.

[32] AVEVA Solutions Ltd. Schematic model manager user guide. 2012.

[33] Neil McPhater. Pca presentation for kl. 2009.

[34] AVEVA Solutions Ltd. Aveva engineering. 2013.

[35] Paul Polwin AVEVA Solutions Ltd. Administrator user guide. *User manual*, 2009.

[36] AVEVA Solutions Ltd. Aveva p&id manager, user guide. 2008.

[37] AVEVA Solutions Ltd. 3d design software for accurate and clash-free plant design. 2013.

[38] AVEVA Solutions Ltd Ben Tilbury. Aveva p&id user guide. 2011.

[39] Aker Business Services. Pdms interface documentation. *Aker Solutions*, 2013.

[40] Amec. Detailed engineering design phase. 2013.

[41] Herve Baron. The oil & gas engineering guide. *Editions Technip*, 2010.

[42] Thomas Holm, Lars Christiansen, Markus Goring, Tobias Jager, and Alexander Fay. Iso 15926 vs. iec 62424 - comparison of plant structure modeling concepts. *Helmut Schmidt University, Hamburg and Siamens AG, Erlangen*, 2012.

[43] P. Gu and S. Sosale. Product modularization for life cycle engineering. *Robotics and Computer Integrated Manufacturing 15*, 1999.

[44] Erik Tillman. Comos and pdms interface. *Assign, Refresh, Re-assign, Batch Transfer and Data Exchange*, 2012.

[45] Subhendu Moulik. Project engineering & management textbook. *AuthorHouse*, 2012.

[46] W. Marquardt and M. Nagl. Workflow and information centered support of design processes - the improve perspective. *Computer and Chemical Engineering*, pages 65–82, 2004.

[47] J. Pennock. Piping engineering leadership for process plant projects. *Gulf Professional Publishing*, 2001.

[48] Phil Robins. Posc-caesar fiatech ids-adi projects. *Intelligent Data Sets Accelerating Deployment of ISO15926, Realizing Open Information Interoperability*, 2012.

[49] Rune Vassdal. Information about pdms interface documentation. *Aker Solutions, internal documentation*, 2010.

[50] Erik Tillmann. Eldfisk experience transfer, comos - pdms 3d model layout engineering development (rev. 2). *Aker Solutions. PowerPoint Presentation*, 2012.

[51] P. Sauder. Design for assembly and schedule optimization for large structure construction. *Unpublished Msc Thesis, Department of Mechanicle Engineering, University of Saskatchewan, Saskatoon Canada*, 1998.

[52] G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computers*, pages 38–49, 1992.

Appendix A

Comos - PDMS Interface for building XML interface

Source application details			Destination application details	
Field no	**Descriptive name**	**COMOS attribute**	**PDMS attribute**	**Direction (import = to COMOS)**
1	PipeRunSuffix	ENG903.ENGA0456	N/A,!!ComosSuffix	import
2	PipeSpec	Owner.Unit.Spec("ENG001.ENGA0920").Value	N/A,!!CGetPipeSpec(),!!CSetPipeSpec()	export
3	NominalDiameter	Owner.Unit.Spec("E3D.PDMSA0036").DisplayValue	bore	export
4	InsSpec	Owner.Unit.Spec("E3D.PDMSA0043").DisplayValue	Ispec	export
5	TrSpec	Owner.Unit.Spec("E3D.PDMSA0018").DisplayValue	Tspec	export
6	ProcFluid	Owner.Unit.Spec("ENG001.ENGA0921").Value	!!CapComosGetFluref(),!!CapComosSetFluref()	export
7	ChemClean	Owner.Unit.Spec("ENG001.ENGA5148")	:CHEMCLEAN	export
8	DesignPressure	Owner.Unit.Spec("ENG003.ENGA0231").GetXValue(1)	:DESPRESS	export
9	OpertempMIN	Owner.Unit.Spec("ENG003.ENGA0938").GetXValue(0)	:MINOPTMP	export
10	Opertemp	Owner.Unit.Spec("ENG003.ENGA0938").Value	:OPTEMP	export
11	OpertempMAX	Owner.Unit.Spec("ENG003.ENGA0938").GetXValue(1)	:MAXOPTMP	export
12	DesignTempMIN	Owner.Unit.Spec("ENG003.ENGA0229").GetXValue(0)	:MINDESTMP	export
13	DesignTempMAX	Owner.Unit.Spec("ENG003.ENGA0229").GetXValue(1)	:MAXDESTMP	export
14	PedCategory	Owner.Unit.Spec("ENG001.ENGA0480").Value	:PEDCAT	export
15	PidNum1	Owner.Unit.spec("ENG903.ENGA0001.ENGA0001b")	:PIDNUM	export
16	PidNum2	Owner.Unit.spec("ENG903.ENGA0001.ENGA0001b")	:PIDNUM2	export
17	LineLabelP&ID	Owner.Unit.Spec("ENG001.ENGA0910").DisplayValue	:LINELABEL	export
18	PipeTestMedium	Owner.Unit.Spec("ENG003.ENGA0946").DisplayValue	:TESTFLUID	export
19	Testpressure	Owner.Unit.Spec("ENG003.ENGA2559").DisplayValue	:TESTPRESS	export
20	TracingIdentificator	Owner.Unit.Spec("E3D.PDMSA0028").Value	:TRACE	export
21	ProcessLineStatus	Owner.Unit.Spec("E3D.PDMSA0046").DisplayValue	:PROCSTAT	export
22	NDTGroup	E3D.PDMSA0021	:NDTGROUP	import
23	StressIsoNumber	owner.Unit.Spec("ENG909.ENGA0550").DisplayValue	:STRESSNO	export
24	StressRemark	owner.Unit.Spec("ENG909.ENGA0551").DisplayValue	:STRESSNOTE	export
25	MaterialType	E3D.PDMSA0019	:MATERIAL	import
26	SecondaryMaterial	E3D.PDMSA0020	:MATERIAL-SEC	import
27	PMI	E3D.PDMSA0022	:PMI	import
28	PaintCode	E3D.PDMSA0023	:PAINTCODE	import
29	TopcoatSpecification	E3D.PDMSA0029	:TOPCOAT	import
30	Capstatus	E3D.PDMSA0013	:capStatus	import
31	ModellingDisc	E3D.PDMSA0039	Owner.Owner.purp	import
32	InsulationThickness	owner.Unit.Spec("ENG001.ENGA0927").Value	:INSTHICK	export
33	InsulationCode	owner.Unit.Spec("ENG001.ENGA0924").Value	:INSTYPE	export
34	PipeTag	owner.Unit.label	:KESYSTAG	export
35	Area Line			import

Appendix B

ISO 15926 PID model

ISO 15926 P&ID model

http://ids-adi.org

POSC Caesar – FIATECH

Intelligent Data Sets - Accelerating Deployment of ISO15926

Realizing Open Information Interoperability

Rev	Date	Description	Author	Check
1.1	24/6/2008	First public draft	Adrian Laud	
1.2	24/7/2008	Minor edits for Instrument Loops	Adrian Laud	
1.3	15/8/2008	Changes for InstrumentLoops and SignalLines	Adrian Laud	
1.4	9/9/2008	Changes following London Workshop	Adrian Laud	
1.5	1/10/2008	Incorporate comments from Bentley	Adrian Laud	
1.6	9/10/2008	Incorporate comments from Dow	Adrian Laud	
2.0	30/10/2008	Matrix 1-3 Boston Workshop edits and Release	Adrian Laud	
2.1	12/12/2008	Edits following feedback from project participants	Adrian Laud	
2.2	10/1/2009	Edits following Technical Workshop discussions	Adrian Laud	
2.3	13/1/2009	Update for PropertyBreak	Adrian Laud	
2.4	19/1/2009	CrossPageConnection edit	Adrian Laud	
2.5	27/1/2009	Add additional AnnotationItems	Adrian Laud	
2.6	28/1/2009	Edits following TGW 27/1/2009	Adrian Laud	
2.7	4/2/2009	Add special use of PipeConnectorSymbol	Adrian Laud	
2.8	18/2/2009	Add definition for use of ComponentType	Adrian Laud	
2.9	26/3/2009	Expand on Connection and IDs for OPCs	Adrian Laud	

Executive Summary

This document discusses the information model for Process and Instrumentation Diagrams (P&IDs) for ISO 15926.

The document describes the main information objects that are used for a P&ID and uses examples implemented as XML exchange files conforming to the XML Schema used by XMpLant - a Yellow category implementation.

The model is described using the classes of ISO 15926-4 Reference Data Library (RDL) or classes proposed for addition to the RDL where no appropriate class was found. The classes of the RDL and the current proposed classes are held in the RDS/WIP which is where new classes will also be submitted.

The model takes into account the requirement for it to be compatible with the model for 3D.

Table of Contents

Purpose

This document is to form the basis for discussion and refinement of the definition of the P&ID model for ISO 15926. This document identifies an XML exchange level category with local Schema or referencing the RDS/WIP.

The information objects and Schema can also be used as a requirements specification for the Templates and OIM's required to support P&ID's for Template based exchange.

Definitions from ISO 15926-4

Definitions that are in the RDS/WIP are in quotes. Those that are not are not quoted are the additional classes that are to be will be submitted to the RDS/WIP.

PlantModel	A container for a set of process plant information - Not yet in the RDS/WIP.
PlantInformation	A container for the meta data about the PlantModel - Not yet in the RDS/WIP.
PlantItem	An abstract super-type for any physical asset of the plant.
AnnotationItem	An abstract super-type for any object that needs an identifier that is not a physical asset of the plant.
System	"A functional object which is an assembly of functional objects forming a network to provide a type of service or serving a common purpose". Whilst this spans more than process plant in that context it covers a PipingSystem and the instrument Loops that effect control of it.
Specification	"A definition of one or more aspects of one or more physical objects or activities".
Catalogue	"A document containing information of systematic arrangements of enumerated items giving descriptive details, a list or register in alphabetical or other methodical order."
ShapeCatalogue	A collection of geometric definitions of the physical dimensions of a process plant component using either explicit or parametric dimensions - Not yet in RDS/WIP.

Component	"A physical object that is only used as a part of a larger physical object".
Equipment	"An artefact class that contains classes of artefacts or physical resources required for a purpose." A generic class covering all types of equipment.
Nozzle	"A physical object that has a protruding part through which a stream of fluid is directed."
PipingSystem	"A system that is designed for conveyance of fluids by use of pipes, tubes, pipe fittings, valves etc. and connected process equipment".
PipingNetworkSystem	"A fluid system of interconnected piping network branches limited by Unit Operation Inlet/Outlet and Piping Network Terminators."
PipingNetworkSegment	"The piping limited by a Node and a Break, Node and a Connector, two Nodes, two Breaks, two Connectors or a Break and a Connector. The last five providing there are no Breaks or Connectors in between. In the last three cases the Segment will coincide with a Piping Branch."
PipingNetworkBranch	"A fluid transport device connecting piping network connections, terminators and units".
PipingComponent	"A process piping equipment class that contains classes of equipment used in or in connection with a piping system".
InstrumentLoop	A combination of 2 or more instruments or control functions arranged so that signals pass from 1 to another for the purpose of measurement and/or control of a process variable - Not yet in the RDS/WIP.
InstrumentComponent	"An instrument component that is not inline or connected to a PipingNetworkSegment".
InstrumentConnection	"A connection where an instrument is connected to an object" For the P&ID it is treated as the topological location where a ProcessInstrument is connected to a PipingNetworkSegment.
ProcessInstrument	"A physical object that detects an aspect of something; records, modifies and/or displays such an aspect or performs a combination of these activities".
Datasheet	"A description that is a set of defined fields within which information is supplied".
Drawing	"A document containing graphic representation of shape or layout".

PropertyBreak	A break in a process line where one or more key properties if the line changes. This covers changes such as Specification, Insulation etc. - Not yet in the RDS/WIP. PipingSpecificationBreak is in the RDS/WIP but applies to Specification change only.
PipeConnectorSymbol	An on/off page connector symbol for a PipingNetworkSegment. - Not yet in the RDS/WIP.
SignalConnectorSymbol	An on /off page connector symbol for an InstrumentLoop – connects to a SignalLine. - Not yet in the RDS/WIP.
PipeFlowArrow	A graphical arrow symbol that is used to visually identify the flow direction for a PipingNetworkSegment.

Process and Instrumentation overview

The model contains three independent representations

- Flow
- Topology
- Materials definitions

The first two of these are closely linked and the latter is largely independent but may impose constraints on the first two.

Flow representation

This is the graphical presentation of the model depicting the PipingNetworkSystems and their structure on a Drawing. This contains symbols to represent the key piping and instrumentation symbols and their connectivity is graphically depicted. The representation is drawing oriented and uses special symbols to identify cross page connectivity. Nozzles are explicit on Equipment but may be implicit for inline Equipment. Whilst there will be no graphics for an implicit Nozzle it will exist in the Topology.

There are some items that are mainly for graphical representation but that can also carry information and be referenced within the model. These items include PropertyBreak, PipeConnectorSymbol, SignalConnectorSymbol and PipeFlowArrow.

PipingComponents and ProcessInstruments are represented by symbols, which for a given standard, are a defined graphical representation. Each definition will be held in a ShapeCatalogue and identified by its ComponentName. The ShapeCatalogue can be stand alone, or contained in the XML file for each P&ID in which case only the symbols for that diagram will be present.

Each instance of a symbol will define the Position of the symbol, the ComponentName and the attributes specific to that instance (eg Tag). The ConnectionPoint Nodes will be located relative to the origin of the Drawing.

On the drawing, a PipingNetworkSystem is a set of symbols connected to each other by graphical lines – CenterLines which belong to a PipingNetworkSegment. Where there is a junction in the PipingNetworkSystem (PipingNetworkBranch), there will be the junction of three PipingNetworkSegments each of which will have a Centerline that starts or ends at the location of the PipingNetworkBranch.

The exception to this is where the branch connection is small bore tube where the connection is for the purpose of enabling the fluid to be accessed by a ProcessInstrument that will convert a given property to a signal, which signal is conveyed via the InstrumentLoop. In this case the main PipingNetworkSegment will not contain a PipingNetworkBranch but a special symbol InstrumentConnection instead. The CenterLine of the main PipingNetworkSegment need not terminate at these special symbols and it will not terminate the PipingNetworkSegment as a PipingNetworkBranch would. Such small bore PipingNetworkSegments for connection to ProcessInstruments will have the subclassification of InstrumentationFluidConductor.

On a P&ID the Tag names of Equipment, ProcessInstruments etc as well as intelligent links in many cases are data driven from the attributes of the plant items that they are annotating. The Text object has the capability to reference the attribute(s) that make up the string to be displayed.

Annotation that does not belong to a plant item (eg. Drawing border, notes etc) will be contained in a Drawing object.

Topology

This view is concerned with the structure and connectivity of the model. Connectivity involves PipingNetworkSegments, InstrumentLoops and the connection of PipingNetworkSegments that connect between InstrumentLoops and normal PipingNetworkSegments.

The key top level topology for piping is a PipingNetworkSystem which is a connected set of PipingNetworkSegments. The second level of topology here is that of the PipingNetworkSegment itself which is an ordered sequence of PipingComponents, ProcessInstruments and special symbols.

Each PipingComponent and ProcessInstrument contains a set of Nodes which are the points at which CenterLines or SignalLines can connect to the symbol. These Nodes are contained in a ConnectionPoints object. Equipment will contain Nozzles which in turn will contain a ConnectionPoints object for the Nodes. The Node is required even if the Nozzle has no graphics.

An InstrumentConnection symbol is used to locate the connection point of a PipingNetworkSegment that is of subclass InstrumentProcessConnection in the sequence of the Components in a PipingNetworkSegment.

InstrumentLoops are also represented on the P&ID as a collection of instruments that are either graphically connected with a SignalLine or an implied InstrumentLoop. The rules for the grouping of ProcessInstruments in a Loop needs to be identified so that these can be expressed in the mapping files to enable them to be collected into an InstrumentLoop. The initial proposal is to use the LoopNumber.

Because an InstrumentLoop is an unordered collection, the ProcessInstruments will not be contained in the InstrumentLoop but will have an Association to it of type "is a part of". Where a ProcessInstrument such as a Control valve is physically part of the flow then it is contained within the PipingNetworkSegment. However, it is also part of the InstrumentLoop which controls it and so will also have an Association to the InstrumentLoop.

Topology for Equipment is concerned with containing the Nozzles that belong to the Equipment and the connections between the Nozzles and the PipingNetworkSegments that connect to them.

A PipingSystem is a collection of PipingNetworkSystems, Equipment, ProcessInstruments and possibly InstrumentLoops. A PipingSystem can be formed for many purposes. It can be as a parent for a given fluid system or for a purpose such as a commissioning package. The same plant items can belong to many PipingSystems at any one time.

Materials definition

For the P&ID this is mostly specification driven and the Specification attribute is the link to the entry in the Specification information which will in turn reference a Datasheet.

For the P&ID the Specification is a key attribute the value of which is the name of the Specification. The name is also usually part of the line label.

Model Overview

Whilst there are many differences between the total model for a P&ID and a 3D model the core topology model for process and instrumentation is the same. The P&ID is a functional specification and the 3D is the spatial implementation of this model.

The coordinate system used is right handed Cartesian with the origin at the bottom left of the Drawing. The positive X is along the bottom and the positive Y is up the left hand side.

There are three areas of interest (and these relate to ADI Matrix projects 1, 2 and 3) P&ID exchange, 3D model exchange and P&ID – 3D model exchange and comparison. Indeed model comparison is valuable for P&ID to P&ID and 3D to 3D as well as between them.

All Elements contain a Presentation Element that allows for layers, colour, line types and text fonts.

The main hierarchy of the model is as follows:

PlantModel
- PlantInformation
- ShapeCatalogue
- System
- PipingSystem
- Equipment
  - Nozzle
- InstrumentLoop
- SignalLine
- ProcessInstrument
- InstrumentComponent
- SignalConnectorSymbol

- PipingNetworkSystem
  - PropertyBreak
  - PipingNetworkSegment
    - Equipment
    - CenterLine
    - PipingComponent
    - ProcessInstrument
    - PipeConnectorSymbol
    - PipeFlowArrow

Annotation Elements are also part of the Drawing

- Drawing
  - Component
  - Curve
  - Text

PlantItem

A PlantItem is an abstract super-type for all objects that represent physical assets in the plant. Eg. Equipment, PipingComponents etc.

The Schema defines this object such that it has eight Attributes and contain any of the geometric classes, some key engineering attributes, History, Associations as well as itself. Ie. It can be nested to any level enabling the hierarchy of the plant to be represented. It can also contain any number of GenericAttributes Elements each of which is a container for Attributes of the plant item. These should all be classes from the RDS/WIP and allows for all of the information from the source to be retained.

Most Attributes are optional except ID and are:

- ID The transient identifier for the object – unique in the file
- TagName The engineering Tag name (A code intended to reference an item.)
- Specification The specification to which the plant item conforms
- StockNumber A unique part reference to identify it – commodity code
- ComponentClass The fine grained class from the RDL
- ComponentName The name of the symbol or component shape
- ComponentType The definition type - Explicit or Parametric
- Revision The revision number of the object
- Status The status of the object. Eg. Deleted

NB. ComponentType is also use to identify whether the shape definition of the Component is in the ShapeCatalogue or not. If the Attribute does not exist then the definition is with the instance and if it is present then it is in the ShapeCatalogue.

In addition to the other Elements a PlantItem may also contain a PersistentID Element which is used where the source system supports the concept of a persistent ID. This has the form

<PersistentID Identifier=”EEFA1234567” Context=”Project ABC”/>

The identifier is a string and so can be whatever the source system uses and if the Element exist then the Identifier is mandatory. Context is optional but is required if needed to make the Identifier unique.

A PlantItem can also contain a Presentation Element which allows the retention of source information for example where specific Layers are used to indicate the type of PlantItem. Whist the Presentation Element is present on the graphical Elements it may be useful at this level but is optional.

Presentation has the following Attributes:

- Layer Name or number
- Color Colour index or Name
- LineType Name of the font or glyph for the line (eg. Dashed)
- LineWeight Number defining a line thickness or ratio for line thickness (eg. 0.35 mm)
- R G B Three Attributes to define the RGB components of the colour (0.0 – 1.0)

AnnotationItem

An AnnotationItem is an abstract super-type for objects that are used and referenced in the P&ID but that do not represent a physical asset. Eg. PipeConnectorSymbol.

The Schema defines this object such that it has six Attributes and can contain any of the 2D curves but it cannot contain itself.

The Attributes are optional and are:

- ID The transient identifier for the object – unique in the file

- ComponentClass The fine grained class from the RDL
- ComponentName The name of the symbol or component shape
- ComponentType The definition type - Explicit or Parametric
- Revision The revision number of the object
- Status The status of the object. Eg. Deleted

NB. ComponentType is also use to identify whether the shape definition of the Component is in the ShapeCatalogue or not. If the Attribute does not exist then the definition is with the instance and if it is present then it is in the ShapeCatalogue.

In addition to the other Elements a PlantItem may also contain a PersistentID Element which is used where the source system supports the concept of a persistent ID as well as a History – see PlantItem.

An AnnotationItem can also have a Presentation Element – see PlantItem.

PlantInformation

This Element is a container for the metadata for the model and contains amongst other aspects the Schema version (it conforms to), originating system, time and date, default units of measure and a flag to identify this is a P&ID.

ShapeCatalogue

The ShapeCatalogue is the container for the definition of symbols used on the P&ID for PipingComponents, ProcessInstruments, InstrumentComponents, PipeConnectors and SignalConnectorSymbols and can include Equipment.

The graphical definition of the symbol can be Line, PolyLine, Circle, Ellipse, BsplineCurve, TrimmedCurve or CompositeCurve. It can also contain Text that is fixed for the symbol. The graphics are defined with respect to the origin of the component which in most cases is its centre. The main flow direction is along the X axis of the component.

This can also be a standalone ShapeCatalogue containing a full library of symbols for a given standard or company.

Equipment

Equipment is usually one off, but can be made up of a collection of standard symbols. This contains the graphics of the Equipment, some key attributes and nested symbols for the Nozzles it contains. It can also contain Equipment symbols for sub equipment.

The Key attributes are:

TagName
ComponentClass – type of equipment – value as per RDS/WIP

Nozzle

A Nozzle contains the key Attribute TagName and the geometry as well as a ConnectionPoints Element that will contains two or three Nodes. The mandatory ones are at the origin and the point where the CenterLine or the Node of a PipingComponent or ProcessInstrument connects. This is for consistency with PipingComponents.

Instrument Loops and Signal Lines

An InstrumentLoop is a collection of ProcessInstruments, InstrumentComponents and SignalLines. The InstrumentLoop may contain the SignalLines that belong to it but not the ProcessInstruments or InstrumentComponents.

ProcessInstruments are instruments that are either inline with the process (eg: Control Valve) or connected to the process (eg: Pressure Indicator). A Control Valve is physically part of the PipingNetworkSegment it controls the flow of, but is also part of the InstrumentLoop that controls it. The association to the InstrumentLoop is using an Association of type "is a part of".

InstrumentLoops are not necessarily concerned with the connectivity of instruments and it cannot be used as a definition of an order. The representation of these is less explicit where the drawing can contain implicit InstrumentLoop by virtue of the Loop Number (TagName). Some inline components such as Valves are also allowed and off-page connectors (InstrumentConnectorSymbols) can also be present where a loop spans P&IDs.

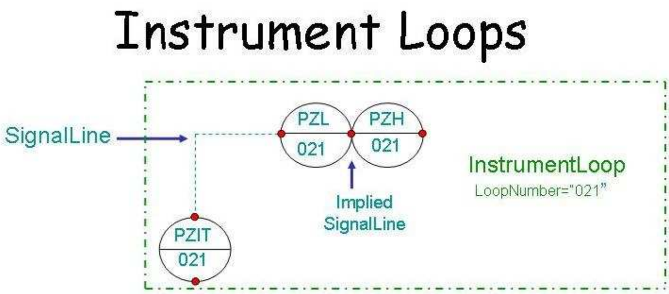
An InstrumentLoop may or may not have an overall Connection – it should only be present where it makes sense.

The SignalLine contains the graphical representation of a connection between instruments with its main purpose being to represent the type of connection. It can be simply a CentreLine or a collection of CentreLines and Symbols (in a Component) that define the graphical representation of the signal. A SignalLine will have a ComponentClass that defines the type of the signal (Hydraulic, Pneumatic, Electric etc).

There can be junctions where multiple SignalLines meet, where these are shown on a P&ID, they are only for layout purposes and don't signify a physical item. The connection needs to be from the implied source and the Instrument. A SignalLine may have a Connection which will contain the Tag or ID of the Element at the start and end of the SignalLine but may not contain any instruments. Where the start or end of a SignalLine is a SignalConnectorSymbol then, if the Connection is used. the ID of the SignalConnectorSymbol will be the FromID and ToID respectively.

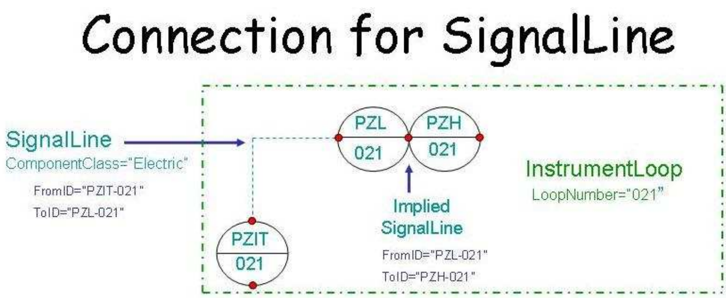
The graphical lines that connect an Instrument to an InstrumentConnection are represented by a Conductor.

For off or on page connections the SignalConnectorSymbol is used.



An InstrumentLoop is a collection of instruments

A SignalLine is the connection and graphical representation of the type of connection between the instruments



SignalLine Connection Element
From is the ID the object it starts at
To is the ID the object it ends at

SignalConnectorSymbol

This is a special symbol for on or off page connections. It optionally contains the name of the drawing that the connection is going to / from and the Identifier that will be used to match the other end of the connection and optionally a Context. The SignalLine will have a Connection to or from the SignalConnectorSymbol.

The connectivity is formed using the CrossPageConnection Element which is also used by the PipeConnectorSymbol.

CrossPageConnection

This Element uses either the “LinkLabel” attribute or the “PersistentID” of the matching connector. For LinkLabel this can be the concatenation of a number of actual attributes of the line but will be treated simply as a text string that needs to match at both ends.

If the PersistentID is used then each end will hold the PersistentID of the other and the Context will ensure that the PersistentID’s are unique in that Context.

Eg.

```
<SignalConnectorSymbol>
        <CrossPageConnection DrawingName=”Drawing002” LinkLabel=”ABC185”/>
</SignalConnectorSymbol>
```

Would match

```
<SignalConnectorSymbol>
        <CrossPageConnection DrawingName=”Drawing001” LinkLabel=”ABC185”/>
</SignalConnectorSymbol>
```

or

```
<SignalConnectorSymbol>
        <PersistentID Identifier=”AA1987E344” Context=”Project ABC”/>
        <CrossPageConnection DrawingName=”Drawing002”>
                <LinkedPersistentID Identifier=”D001234CBA” Context=”Project ABC”/>
        </CrossPageConnection>
</SignalConnectorSymbol>
```

Would match

```
<SignalConnectorSymbol>
        <PersistentID Identifier=”D001234CBA” Context=”Project ABC”/>
        <CrossPageConnection DrawingName=”Drawing001”>
                <LinkedPersistentID Identifier=”AA1987E344” Context=”Project ABC”/>
        </CrossPageConnection>
</SignalConnectorSymbol>
```

PipingSystem

A PipingSystem is a stand alone Element that has no direct children but may have many Associations. Whilst it is possible, it is not envisaged that the Plant items involved will have Associations to the PipingSystems to which they belong.

Note that these are lifecycle information objects in that they may be required for a specific part of the lifecycle but have little meaning elsewhere (eg. For a Commissioning Package).

PipingNetworkSystems

PipingNetworkSystems are a connected set of piping that may have multiple sources and multiple destinations. They are concerned with the arrangements of interconnected PipingNetworkSegments which in turn contain PipingComponents, ProcessInstruments that take part in the flow of the fluid that they contain and special symbols. The P&ID is a functional definition of the spatial layout and represents a one to one relationship with the 3D model at the PipingNetworkSegment level with the exception of the special symbols. Also there are items in the 3D model that are not represented on the P&ID (pipe supports, elbows etc).

A key here is that the topology for both is the same as this is what can drive from, or validate the 3D model against the P&ID. Some of the attributes that each contain will be different as will the geometry. The definition of the PipingNetworkSegment is a vital part of this which enables the PipingNetworkSystem to be decomposed into single flow sequences that directly relate. Where a key engineering parameter changes or the flow splits then the segment ends.

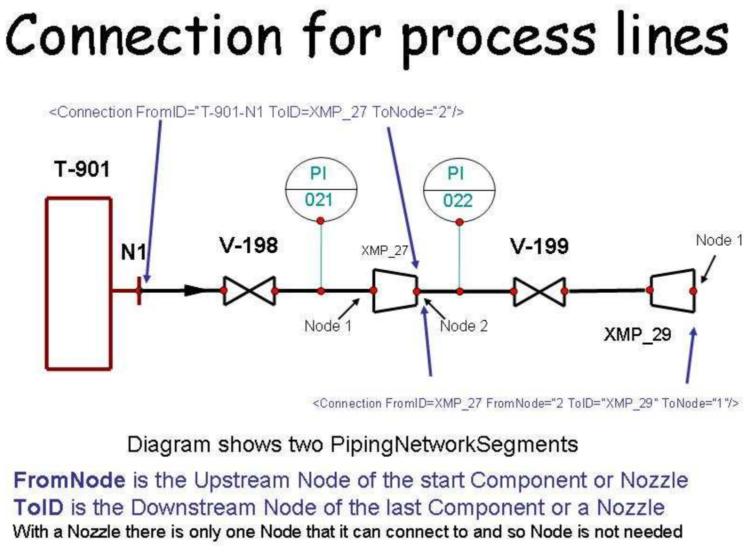
If the segments ends where parameters change a PropertyBreak can be used, especially if this occurs in the middle of a section of CenterLine, rather than at a PipingComponent. A PropertyBreak is not required for diameter change by a reducer.

PropertyBreak

The PropertyBreak is a special symbol that is a specialisation of AnnotationItem and will occur in the PipingNetworkSystem. This is a general class of break that can identify the property being changed using the ComponentClass Attribute (eg. Insulation). The PropertyBreak can be in the middle of a CenterLine or at a Node of a PipingComponent. If the break is not coincident with a PipingComponent then the PropertyBreak itself will be referenced as the termination of a PipingNetworkSegment and the start of the connecting one. If the break is coincident then the PipingComponent will be the end / start of the PipingNetworkSegments and there will be Associations between the PipingComponent and the PropertyBreak “is associated with”.

PipingComponent

A PipingComponent a generic class for all piping components. These represent physical components that are usually catalogue items. The actual class of Component is identified by its ComponentClass attribute.




```
<PipingNetworkSystem>
    <PipingNetworkSegment  ID=”XMP12” Tag=”200-CC110-01234-B1”>
        <Connection FromID=”T-901-N1” ToID=”XMP_27” ToNode=”2”/>
        <CenterLine …/>
        <PipingComponent ID=”XMP_15” Tag=”V-198” ComponentClass=”Valve” …/>
        <CenterLine …/>
        <InstrumentConnection ID=”XMP_23”  …/>
        <PipingComponent ID=”XMP_27” ComponentClass=”Reducer” …/>
    </ PipingNetworkSegment>
    <PipingNetworkSegment  ID=”XMP_13” Tag=”200-CC110-01234-B2”>
        <Connection FromID=”XMP_27” FromNode=”2” ToID=”XMP_29” ToNode=”1”/>
        <CenterLine …/>
        <InstrumentConnection ID=”XMP_35”  …/>
        <PipingComponent ID=”XMP_36” Tag=”V-199” ComponentClass=”Valve” …/>
        <CenterLine …/>
        <PipingComponent ID=”XMP_29” ComponentClass=”Reducer” …/>
    </ PipingNetworkSegment>
</PipingNetworkSystem>
```

PipingNetworkSegments

The key to a PipingNetworkSegment is that it has a single start and end. Components in it are ordered from head to tail. The connectivity of PipingNetworkSegments themselves is through the Connection Element which identifies the plant item, and Node of that item, that the segment is connected from (upstream) and to (downstream). The plant item can be identified by its Tag or

ID if it has no Tag. Where the PipingNetworkSegment starts or ends at a PipeConnectorSymbol then the ID of the PipeConnectorSymbol will be the FromID and ToID respectively.

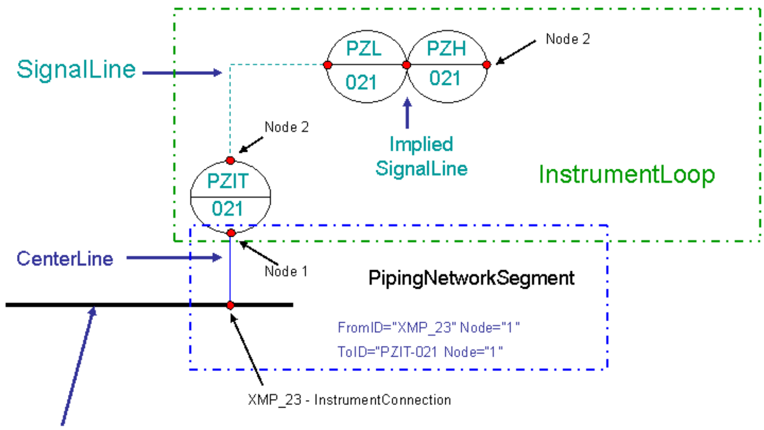
Each PipingComponent and ProcessInstrument will have Attributes Flowin and Flowout which identify the Node of the Component where the upstream and downstream connections are made respectively.

The graphical lines that represent the piping are CenterLine Elements which may or may not have an ID or a PersistentID. These lines are mainly graphical but are contained in the PipingNetworkSegment in sequence head to tail along with the other components. Where a CenterLine has complex graphics then these will be contained in a Component that has the ComponentClass of “CenterLine” and the CenterLine will not be present as such.

The topology for PipingComponents and ProcessInstruments (inline instruments) in the PipingNetworkSegment is by their order in the PipingNetworkSegment whereas the connections to ProcessInstruments, that are not part of the flow (eg. Pressure indicator), are not. The special symbol InstrumentConnection resolves this, as it defines the location where the ProcessInstrument will connect to the PipingNetworkSegment with respect to the inline components.

The graphical line (CenterLine) representing the process line to which the connection is made may or may not be broken at the connection and if not broken precedes the InstrumentConnection and geometrically will end after the InstrumentConnection, matching the 3D pipe it represents. The topology has a clear order. Where the flow of a PipingNetworkSegment is both ways, then there will be a DualFlow attribute set to True and the topology defines the order from one end to the other. In this case either end can be the head.

Connecting Loops and Process



PipingNetworkSegment

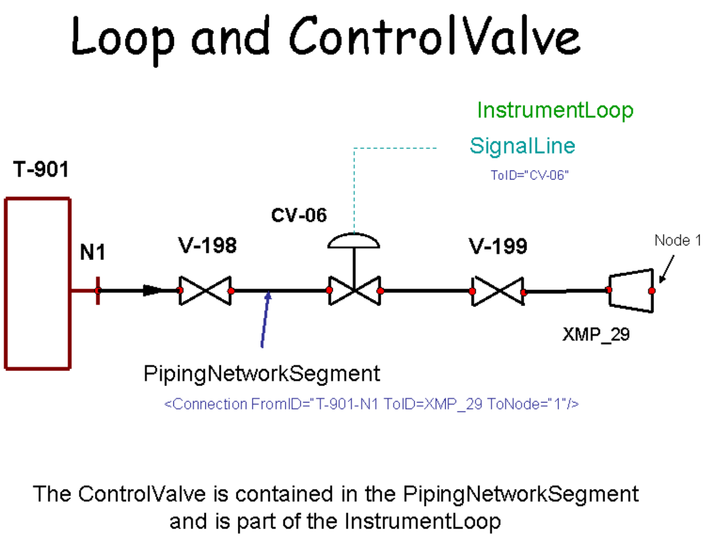
CenterLine may or may not be broken at the InstrumentConnection but the segment does not end

XML Fragment

```
<PipingNetworkSegment  ID=”XMP12” Tag=”200-CC110-01234”>
        <Connection FromID=”T-901-N1” ToID=”XMP_27” ToNode=”2”/>
        <CenterLine …/>
        <PipingComponent ID=”XMP_15” Tag=”V-198” ComponentClass=”Valve” …/>
        <CenterLine …/>
        <InstrumentConnection ID=”XMP_23”  …/>
        <PipingComponent ID=”XMP_25” Tag=”CV-06” ComponentClass=”ControlValve” …/>
</ PipingNetworkSegment>

<PipingNetworkSegment ID=”XMP_32” ComponentClass=”InstrumentProcessConnection”>
        <Connection FromID=”XMP_23” FromNode=”1” ToID=”PZIT-021” ToNode=”1”/>
        <CenterLine>
                <Coordinate X=”100” Y=”60”/>
                <Coordinate X=”100” Y=”65”/>
        < CenterLine >
</ PipingNetworkSegment>
```

The InstrumentLoop is a collection ProcessInstruments, InstrumentComponents and contains the SignalLines

```
<InstrumentLoop ID="XMP_55" >
<Association Type="is a collection including"  ItemID="XMP_70"/>
<Association Type="is a collection including"  ItemID="XMP_71"/>
<Association Type="is a collection including"  ItemID="XMP_72"/>
</InstrumentLoop>

<InstrumentComponent ID="XMP_70" Tag="PZIT-021">
        <Association Type="is a part of"  ItemID="XMP_55"/>
</ InstrumentComponent >

< InstrumentComponent ID="XMP_71" Tag="PZL-021">
        <Association Type="is a part of"  ItemID="XMP_55"/>
</ InstrumentComponent >
< InstrumentComponent ID="XMP_72" Tag="PZH-021">
        <Association Type="is a part of"  ItemID="XMP_55"/>
</ InstrumentComponent >
<SignalLine ComponentClass="Electric" >
        <Connection FromID="PZIT-021"ToID="PZL-021"/>
        <CenterLine …/>
</SignalLine>
<SignalLine ComponentClass="Electric" >
        <Connection FromID="PZL-021"ToID="PZH-021"/>
        <CenterLine …/>
</SignalLine>
```



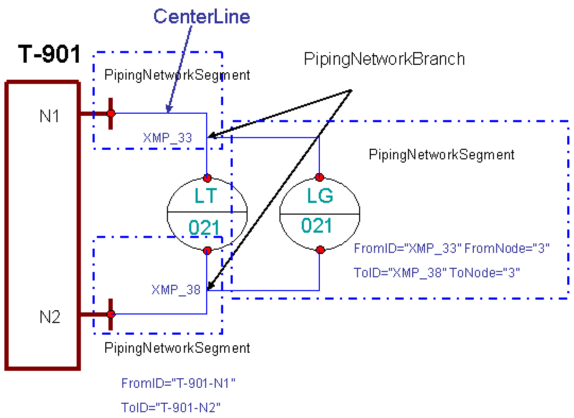
XML Fragment

```
<PipingNetworkSegment  ID="XMP12" Tag="200-CC110-01234">
        <Connection FromID="T-901-N1" ToID="XMP_27" ToNode="2"/>
        <CenterLine …/>
        <PipingComponent ID="XMP_15" Tag="V-198" ComponentClass="Valve" …/>
           <ConnectionPoints Flowin="1" Flowout="2">
                <Node>
                    <Position>
                        <Location X=" " Y=" " Z=" "/>
                        <Axis  X=" 0" Y="0" Z="1"/>
                        <Reference X=" 0" Y="0" Z="1"/>
                    </Position>
                <Node ../>
                <Node ../>
        <CenterLine …/>
        <InstrumentConnection ID="XMP_23"  …/>
        <ProcessInstrument ID="XMP_25" Tag="CV-06" ComponentClass="ControlValve">
                <Association Type="is a part of" ItemID="XMP_55"/>
        </ ProcessInstrument>
        <CenterLine …/>
        <PipingComponent ID="XMP_29" Tag="V-199" ComponentClass="Valve" …/>
        <CenterLine …/>
        <PipingComponent ID="XMP_35" ComponentClass="ConcentricReducer">
</ PipingNetworkSegment>

<InstrumentLoop ID="XMP_55">
        <Connection FromID="" ToID="CV-06"/>
        <SignalLine .../>
        <Association Type="is a collection including" Tag=" CV-06"/>
</ InstrumentLoop>
```

Vessel trim piping is piping that is fitted to the Equipment by the Equipment manufacturer before delivery and it may or may not have line labels (Tag).

Vessel trim is made up of PipingNetworkSystems and PipingNetworkSegments which can contain PipingComponents and ProcessInstruments. They will also use Conductors rather than CenterLines.

Vessel Trim piping



PipingNetworkSegments for Vessel trim the components are connected with PipingNetworkSegments with a ComponentClass of "InstrumentProcessConnection "

XML Fragment

```
<PipingNetworkSegment  ID=”XMP12” ComponentClass=” InstrumentProcessConnection”>
        <Connection FromID=”T-901-N1” ToID=”XMP_33”/>
        <CenterLine …/>
        <PipingComponent ID=”XMP_33” ComponentClass=”PipingNetworkBranch” …/>
</ PipingNetworkSegment>
<PipingNetworkSegment  ID=”XMP13” ComponentClass=” InstrumentProcessConnection”>
        <Connection FromID=”TXMP_33” ToID=”XMP_38”/>
        <CenterLine …/>
        <ProcessInstrument ID=”XMP_41” Tag=”LT-021” ComponentClass=”OfflineInstrument” …/>
        <CenterLine …/>
        <PipingComponent ID=”XMP_38” ComponentClass=”PipingNetworkBranch” …/>
</ PipingNetworkSegment>
<PipingNetworkSegment  ID=”XMP14” ComponentClass=” InstrumentProcessConnection”>
        <Connection FromID=”XMP_38” ToID=”T-901-N2”/>
        <CenterLine …/>
</ PipingNetworkSegment>
<PipingNetworkSegment  ID=”XMP15” ComponentClass=” InstrumentProcessConnection”>
        <Connection FromID=”TXMP_33” ToID=”XMP_38”/>
        <CenterLine …/>
        <ProcessInstrument ID=”XMP_42” Tag=”LG-021” ComponentClass=”OfflineInstrument” …/>
        <CenterLine …/>
</ PipingNetworkSegment>
```

PipeConnectorSymbol

This is a special symbol for on or off page connections. It optionally contains the name of the drawing that the connection is going to / from and the Identifier that will be used to match the other end of the connection and optionally a Context.

The PipeConnectorSymbol is a specialisation of the abstract class AnnotationItem and as such can have an ID and a PersistentID.

The connectivity is formed using the CrossPageConnection Element which is also used by the SignalConnectorSymbol. The CrossPageConnection Element is described in that section.

There is another special use for this symbol where there is an open pipe such as a vent or drain.

This use can be identified as the ComponentClass will be “FluidConnection”. The Description will contain the text that is to be accessible for Line Lists etc. This Description will be referenced by a Text item in the PipeConnectorSymbol so that it will appear on the Drawing.

ProcessInstrument

A ProcessInstrument is a component that is part of the PipingNetworkSegment or is connected to it via a conductor. Eg. Control Valve or Pressure indicator.

PipeFlowArrow

A PipeFlowArrow is a symbol to indicate visually the flow direction in a PipingNetworkSegment. It may or may not be part of the PipingNetworkSegment itself and is purely a graphical object that is driven by the actual flow direction of the PipingNetworkSegment.

For two way flow there will be PipeFlowArrows in each direction where used.

This is a specialisation of the abstract class AnnotationItem and as such can have an ID and a PersistentID.

InstrumentComponent

An InstrumentComponent is a component that is not connected directly to the process.

Annotation

Annotation is concerned with any graphics or text that is not concerned directly with the P&ID model. It can be Drawing borders, tables and notes etc. Annotation can be grouped using Components – eg the Drawing border.

Any Curve can be used (Line, PolyLine, Circle, Ellipse, BsplineCurve, TrimmedCurve, CompositeCurve) as well as Text.

The container for annotation is the Drawing.

Drawing

A Drawing is a document that contains a graphical representation of the model. In this case the model is the P&ID and much of the model graphics are a part of the definition of the plant items themselves (eg Symbols). The Drawing contents will be the annotation information such as Text for Labels, Notes etc. There are also some special symbols that can be contained in the Drawing that have a significance for information processing that are AnnotationItems which as detailed above can take part in Associations.

A Drawing can contain:

DrawingBorder
Component
Curve (any 2D curve)
Text

AnnotationItem specialisations

Label	- a textual label that will be associated with a PlantItem
InsulationSymbol	- a symbol to indicate that a PipingNetworkSegment has insulation
ScopeBubble	- a collection of curves to surround a number of PlantItems for a given purpose

DrawingBorder

A DrawingBorder is a container for the contents of border for a drawing. This can also be represented as a Component if the source system does not enable it to be specifically identified.

Text

Text on a P&ID can be explicit text or can be the representation of one or a combination of Attributes of a plant item in the plant model. Eg. Tag name.

For the former there will simply be a String attribute that contains the text string.

For text that is presenting the values of Attributes, then the DependantAttribute Attribute contains the definition of the Attributes to be presented in the form:

[Name1]<explicit text>[Name2]]<explicit text>[Name3] ……

Where the [ ] brackets mean "the vale of the Attribute with the name in the [ ] brackets and < > means interpret the text in the < > angle brackets. In the above case it means any explicit text string. eg "-" or "/" – typical delimiters.

The resulting text will be the value of the Attribute Name1 followed by the explicit text (eg. -) followed by the value of the Attribute Name2 etc.

If the Attributes are contained in the parent Element of the Text (eg. Equipment) then this will be sufficient. If not then the ItemID attribute is used to specify the ID of plant item where they are.

Text presentation parameters are simple as we are dealing with the requirements for a P&ID not a draughting system.

Parameters are:

Font	Name of the Text font
Justification	- 9 grid points around string (eg BottomLeft)
Width	Width of the String
Height	Height of the String
TextAngle	Angle of the String
SlantAngle	Slant angle of each character

Appendix C

ISO 15926 3D model

ISO 15926 3D model

http://ids-adi.org

POSC Caesar – FIATECH

Intelligent Data Sets - Accelerating Deployment of ISO15926

Realizing Open Information Interoperability

Rev	Date	Description	Author	Check
1.0	24/06/2008	First public draft	Adrian Laud	
1.1	6/10/2008	Add more discipline definitions	Adrian Laud	
1.2	14/10/2008	Expand on StructuralSteelSection	Adrian Laud	
2.0	30/10/2008	Matrix 1-3 Workshop edits and Release	Adrian Laud	
2.1	16/3/2012	Add description for PipeBend	Adrian Laud	

Executive Summary

This document discusses the 3D spatial model of Process Plant for ISO 15926.

The document describes the main objects that are used for 3D and uses examples implemented as XML exchange files conforming to the XML Schema used by XMpLant - a Yellow category implementation.

The model is described using the classes of ISO 15926-4 Reference Data Library (RDL) or classes proposed for addition to the RDL where no appropriate class was found. The classes of the RDL and the current proposed classes are held in the RDS/WIP which is where new classes will also be submitted.

The model takes into account the requirement for it to be compatible with the model for P&ID where there are common objects – Equipment, Piping and Instrumentation.

Whilst this document covers all disciplines, the main ones to be addressed first are Equipment, Piping and Instruments. This will include Catalogues and Specifications.

Details for HVAC, Structural Steel, Civil and Electrical will be added later.

Table of Contents

Purpose

This document is to form the basis for discussion and refinement for the definition of the Process Plant 3D model for ISO 15926. This document identifies an XML exchange level category with local Schema or referencing the WIP / RDL.

The objects and Schema can also be used as a requirements specification for the Templates and OIM's required to support Process Plant 3D for Template based exchange and sharing.

Definitions from ISO 15926-4

Definitions that are in the RDS/WIP are in quotes. Those that are not are not quoted are the additional classes that are to be will be submitted to the RDS/WIP.

PlantModel	A container for a set of process plant information - Not yet in the RDS/WIP.
PlantInformation	A container for the meta data about the PlantModel - Not yet in the RDS/WIP.
System	"A functional object which is an assembly of functional objects forming a network to provide a type of service or serving a common purpose". Whilst this spans more than process plant in that context it covers a PipingSystem and the instrument Loops that effect control of it.
Specification	"A definition of one or more aspects of one or more physical objects or activities".
Catalogue	"A document containing information of systematic arrangements of enumerated items giving descriptive details, a list or register in alphabetical or other methodical order."
ShapeCatalogue	A collection of geometric definitions of the physical dimensions of a process plant component using either explicit or parametric dimensions. - Not yet in RDS/WIP.
CableTray	A collection of interconnected CableTraySections. - Not yet in RDS/WIP.
CableTrayBend	"A cable tray section that turns the direction of the cable tray run through an angle in the plane of the cable tray".

CableTrayComponent	A Cable Tray class that contains classes of Cable Tray components used within a CableTray. - Not yet in the RDS/WIP.
CableTraySection	"A physical object consisting of a continuous base, raised edges and no covering".
CableTrayStraight	"A cable tray section that is straight".
Component	"A physical object that is only used as a part of a larger physical object".
Equipment	"An artefact class that contains classes of artefacts or physical resources required for a purpose." A generic class covering all types of equipment.
Nozzle	"A physical object that has a protruding part through which a stream of fluid is directed".
HVAC	"An equipment class that contains classes of artefacts normally used in heating, ventilation and air condition (HVAC) installations".
HVACBend	A physical object that is used to change the direction of an HVAC segment with a non-standard radius. - Not yet in the RDS/WIP.
HVACComponent	"A heating, ventilation and air conditioning (HVAC) equipment class that contains classes of equipment used as parts of HVAC equipment".
HVACDuct	"A large sheet-metal duct or casing through which air is passed for forced-draught, ventilation or air-conditioning purposes".
HVACSegment	HAVC limited by two Nodes. - Not yet in the RDS/WIP.
PipingSystem	A system that is designed for conveyance of fluids by use of pipes, tubes, pipe fittings, valves etc. and connected process equipment - Not yet in the RDS/WIP.
PipingNetworkSystem	"A fluid system of interconnected piping network branches limited by Unit Operation Inlet/Outlet and Piping Network Terminators".
PipingNetworkSegment	"The piping limited by a Node and a Break, Node and Connector, two Nodes, two Breaks, two Connectors or a Break and a Connector. The last five providing there are no Breaks or Connectors in between. In the last three cases the Segment will coincide with a Piping Branch".
PipingNetworkBranch	"A fluid transport device connecting piping network connections, terminators and units".

PipingComponent	"A process piping equipment class that contains classes of equipment used in or in connection with a piping system".
Pipe	"A physical object that is a long tube or hollow body intended for conduction of liquid, gaseous or finely divided solid materials, or used for structural purposes".
PipeBend	"A physical object that is used to change piping direction with a non-standard radius".
InstrumentComponent	"Classes of instrument components that forms parts of an instrument". An instrument that is not inline or connected to a PipingNetworkSegment.
ProcessInstrument	"A physical object that detects an aspect of something; records, modifies and/or displays such an aspect or performs a combination of these activities." – Not yet in the RDS/WIP (taken from ISA).
Weld	"An artefact which is made by applying weld material between the artefacts that are joined".
Structure	A collection of mechanical, civil or structural elements for a purpose. In the RDS/WIP Structure is an Abstract Class. – in RDS/WIP but no definition.
Framework	"A support which is a rigid supportive and/or protective openwork or structural frame of an object".
StructuralSection	A linear or curved structural element of a defined profile – Not yet in the RDS/WIP.
StructuralBeam	"A support that is a large and straight piece of normally timber or iron forming one of the main structural members of a building or supporting structure". In the RDS/WIP it is Beam in the context of a "Structural Functional component class".
StructuralBrace	"A device that is the staying or supporting rods or ties which are used in the stiffening of a structure or construction. The brace is normally positioned with an angle to the main axis of the structure". In the RDS/WIP there is BRACE and FRAMEWORK BRACE.
StructuralColumn	"A support which is a cylindrical or slightly tapering body of conciderably greater length than diameter, erected vertically as a support". In the RDS/WIP this is a COLUMN.

Plate	"A physical object that is a piece of material with constant thickness that may be bent or formed".
Ceiling	"A lining which is the overhead inside lining of a room".
Floor	"A device which is the lower inside surface of a hollow structure".
Wall	"A divider for space that may also insulate, protect, secure and support loads".

Process Plant 3D model overview

The model contains three independent views

- Spatial layout
- Topology
- Materials definitions

The 3D model for Process Plant is a spatial layout of the plant organised in accordance with the topology of the plant. The model may be physically split into manageable sections of the plant and also by discipline. The topology of the plant is that of the engineering organisation of the plant.

The topology of the plant for Equipment and Piping is the same as that for the P&ID as the 3D model is the 3D spatial layout of the processes defined on the P&ID.

The spatial layout of the model reflects the physical plant that it is the design of. The coordinate system used may be varied but there needs to be a reference to the plant datum such that every plant item can be spatially located with respect to the Plant Datum.

Spatial layout

This is the graphical representation of the model depicting the layout of Equipment, PipingNetworkSystems, Structural Steel, Cable Trays, HVAC, Mechanical, Civil and Electrical and their structure in a 3D model. This contains Catalogue components to represent the piping and instrumentation components and their connectivity is spatially defined. The model will normally be split by discipline and also by area or other organisation. Nozzles are explicit, however gaskets and bolts may be implied.

The origin of the model may be the plant datum or a local origin with information either within the model or in other metadata to locate and orient the model with respect to the Plant Datum.

Equipment, being one off (for the most part) is geometrically defined with the instance data and will contain the Nozzles that are part of it. Equipment will usually be defined as a collection of CSG primitives but also may be defined using a BREP model.

PipingComponents and ProcessInstruments are, in most cases, standard catalogue items and the definition of the geometry for these can be held once in a ShapeCatalogue and identified by its ComponentName. The ShapeCatalogue can be stand alone or contained in the XML file for each 3D model in which case only the Components for that model will be present.

Each instance of a Component will define the Position of it, the ComponentName and the attributes specific to that instance (eg Tag). The ConnectionPoint Nodes will be located relative to the origin of the model.

In the model a PipingNetworkSegment is a set of PipingComponents connected to each other by Pipes. Where there is a junction in the PipingNetworkSystem (PipingNetworkBranch) there will be the junction of three PipingNetworkSegments each of which will have a Pipe that connects to a specific Node of the PipingNetworkBranch. In many cases only the ProcessInstruments will be modelled in 3D and those not connected to the process will not.

For Structural Steel, most of the sections will be in accordance with a particular standard (eg. AISC) and the SectionName will identify the specific shape. These shapes are defined in the relevant standard and need not be held in the ShapeCatalogue. The resulting geometry for each specific section may be contained as a SolidOfExtrusion for it.

For CableTrays and HVAC, there may be standard off the shelf components but much will be specifically designed. The geometry of each element of a CableTraySection or HVACSection should carry explicit definition as a collection of extrusions, revolutions and possibly BREP model

Mechanical and Civil may be fairly dumb and in these models their inner details are not required – it is mainly spatial layout of the objects as a whole.

Electrical may contain components but is unlikely to contain a great deal of detail.

Topology

This view is concerned with the connectivity of the model. Connectivity involves PipingNetworkSegments, Structural Steel plates and sections and CableTraySegments. Connectivity is mainly confined to each discipline.

For Equipment the main aspect here is that the Equipment has as Tag and contains the Nozzles, that are part of the Equipment, which also have a Tag.

The key top level topology for piping is a PipingNetworkSystem which is a connected set of PipingNetworkSegments. The second level of topology is that of the PipingNetworkSegment itself which is an ordered sequence of PipingComponents and ProcessInstruments. Each PipingNetworkSegment will contain a Connection object that will identify the plant items and the Node of the item that it is connected to both upstream and downstream.

Each PipingComponent and ProcessInstrument contains a set of Nodes which are the points at which the Pipes or other Components connect to the Component. A CenterLine also defines the path of the PipingNetworkSegment and this will have a coordinate that coincides with each Node. These Nodes are contained in a ConnectionPoints object. Equipment will contain Nozzles which in turn will contain a ConnectionPoints object for the Nodes.

A PipingSystem is a collection of PipingNetworkSystems, Equipment, ProcessInstruments and possibly InstrumentLoops. A PipingSystem can be formed for many purposes. It can be as a parent for a given fluid system or for a purpose such as a commissioning package. The same plant items can belong to many PipingSystems at any one time.

Materials definition

For the Piping this is mostly specification driven and most PipingComponents will be selected by a lookup process in the Specification. Depending on the Specification and the NominalDiameter the shape of the PipingComponent can be different. A given SpecificationEntry will have a reference to a specific ShapeCatalogue PipingComponent identified by its ComponentName.

The details for Equipment etc will be on the Equipment Datasheet which are not within the scope of this document.

Model Overview

Whilst there are many differences between the total model for a P&ID and a 3D model, the core topology model for Equipment, process and instrumentation is the same. The P&ID is a functional specification and the 3D is the spatial implementation of this model.

The coordinate system used for 3D is the right handed Cartesian system with the origin typically either at PlantDatum or at the lower left corner of the model or some key object.

There are three overlapping areas of interest which relate to ADI Matrix projects 1, 2 and 3 - P&ID exchange, 3D model exchange and P&ID – 3D model exchange (and comparison). Indeed model comparison is valuable for P&ID to P&ID and 3D to 3D as well as between them. The common model facilitates the interaction between them.

All objects contain a Presentation object that allows for layers, colour, line types and text fonts.

The main hierarchy of the model is as follows:

PlantModel
- PlantInformation
- Specification
- ShapeCatalogue

- Site
  - PlantArea
- CableTray
  - CableTraySection
    - CableTrayBend
    - CableTrayComponent
    - CableTrayStraight
- Component
- Equipment
  - Nozzle
- HVAC
  - HVACSegment
    - HVACBend
    - HVACComponent
    - HVACDuct

- InstrumentComponent
- PipingSystem
- PipingNetworkSystem
  - PipingNetworkSegment
    - PipeBend
    - Pipe
    - Weld
    - PipingComponent
    - ProcessInstrument
- ProcessInstrument
- PipeSupport
- Plate
- Structure
  - Structure
  - Framework
- Framework
  - StructuralSection
- StructuralSection

- Ceiling
- Floor
- Wall

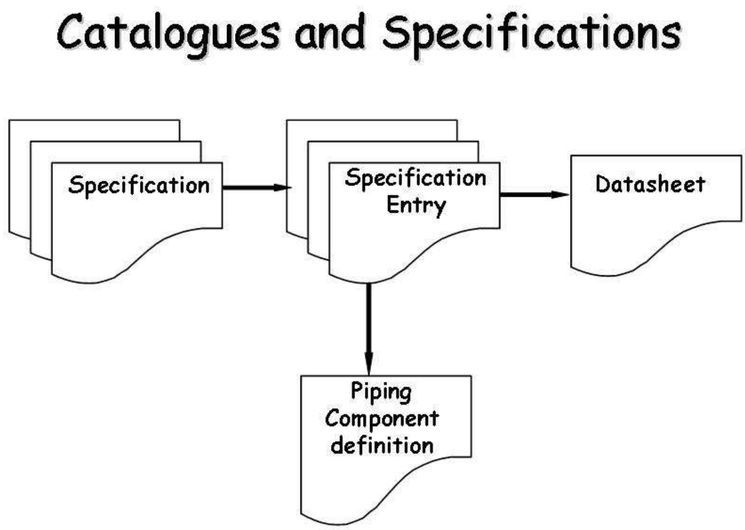
PlantInformation

This object is a container for the metadata for the model and contains amongst other aspects the Schema version (it conforms to), originating system, time and date, default units of measure and a flag to identify this is a 3D model and the discipline.

Specifications

The general principal is that the neutral Catalogues and Specifications contain a full definition of the engineering information such that system specific Catalogues and Specifications can be created from it.

This diagram shows the structure for these.



The Specification Element contains Attributes that define the common parameters for this Specification and any number of SpecificationEntry Elements that relate the ComponentClass and NominalDiameter to a ComponentName.

This reflects the engineering usage, a specification is basically a table of allowable engineering attribute combinations. The SpecificationEntry is concerned with largely non geometric attributes (Rating etc) for a specific selection.

<Specification Name=”” Units=”” Version=”” Date=””>
<SpecificationEntry/>
.. the other attributes as Element references ..
</Specification>

There are approximately 18 attributes of the Specification which define Rating, Minimum and Maximum temperatures and pressures etc that apply to this Specification.

Each SpecificationEntry identifies a specific ComponentClass and the PipingComponent that represents the geometry which is referenced by ComponentName. The definition of the PipingComponent will be in a ShapeCatalogue either in the same XML file, another or may not exist at all. The Specification and SpecificationEntry is about Meta Data and is just as valid if the definition it references does not yet exist. The ComponentType identifies whether the component shape definition is explicit or parametric.

For parametric components the SpecificationEntry will contain a Parameters Element which will contain the values of the parameters to be used when the shape definition is evaluated.

```
<SpecificationEntry ComponentClass=”” ComponentType=”” ComponentName=””
StockNumber=”” ISOSymbol=””>
        <Parameters/>
        .. the other attributes as Element references ..
</Specification>
```

ShapeCatalogue

The ShapeCatalogue is the container for the definition of Components used in the 3D model for PipingComponents and ProcessInstruments and can include Equipment.

The geometric definition of the Components can be CSG primitives or BREP models. The geometry is defined with respect to the origin of the component which in most cases is its centre. The main flow direction is along the X axis of the component.

This can also be a standalone ShapeCatalogue containing a full library of Components for a given standard or company.

The Catalogue is only concerned with shape – it could be explicit or parametric. The shape of components that have different engineering parameters can be the same hence the separation – it is the ComponentName attribute that is the link from the SpecificationEntry to the definition in the ShapeCatalogue.

For parametric components parametric geometry objects are used which have the same name as their explicit counterparts, but their names are preceded with the letter “P”. eg a parametric Cone is a PCone. The value of any of the attributes of a parametric object can be a formula that references parameters or can simply be a number. There are also basic objects for Position, Location, Axis and Reference as well as for ConnectionPoints.

The ShapeCatalogue Structure is:

```
<ShapeCatalogue Name=”” Units=”” Version=”” Date=””>
```

```
        <Equipment/>
        <PipingComponent/>
        <ProcessInstrument/>
        <InstrumentComponent/>
        <Component/>
</ShapeCatalogue>
```

Site

This is the top level organisation object and may contain several PlantAreas and is optional.

PlantArea

This is an organisational area within a Site and is optional.

CableTray

A CableTray is a fully connected collection of CableTraySections which in turn contain CableTrayComponents, CableTrayBends and CableTrayStraights.

CableTraySection

A CableTraySection is a section of a CableTray that has one start and one end. There is no need to end a CableTraySection at a Reducer Component but it will end at a CableTrayTee. Note that a CableTrayTee is a CableTrayComponent with a ComponentClass of CableTrayTee.

CableTrayBend

A CableTrayBend is a piece of CableTray that may be a straight section of CableTray bent on site to change the direction of a Cabletray in the plane of the CableTraySection. The angle of the bend is as required and so is unlikely to be a Catalogue item.

CableTrayComponent

A CableTrayComponent is mainly a Catalogue item (eg Tee) however it may also be a custom designed component.

CableTrayStraight

A CableTrayStraight is a straight section of CableTray. This is a bulk item that will be cut to size when installed.

Equipment

Equipment is usually one off but can contain nested Equipment and Components (Mechanical objects etc). This contains the geometry of the Equipment, some key attributes and nested symbols for the Nozzles it contains. The geometry can be a collection of CSG primitives or BREP models.

The Key attributes are:
    Tag
    ComponentClass – type of equipment

Vessel Trim

Vessel trim piping is piping that is fitted to the Equipment by the Equipment manufacturer before delivery and it may or may not have line labels (Tag).

Vessel trim is made up of PipingNetworkSystems and PipingNetworkSegments which can contain PipingComponents and ProcessInstruments.

In 3D it is unlikely that these can be separately identified as they may not have Tags.

Nozzle

A Nozzle contains the key Attribute Tag and the geometry as well as a ConnectionPoints object that will contain two or three Nodes. The mandatory ones are at the origin and the point where the CenterLine connects. This is for consistency with PipingComponents.

HVAC

HVAC is a fully connected set of HVACSegments which in turn contain HVACBends, HVACComponents and HVACDucts.

HVACSegment

An HVACSegment is a section of HVAC that has one start and one end. There is no need to end the segment at an HVACComponent that is a reducer but it will end with an HVACComponent that is an HVACTee.

HVACBend

An HVACBend is a piece of an HVACDuct that may be a straight section, bent on site to change the direction of a duct. The angle and direction of the bend is as required and so is unlikely to be a Catalogue item.

HVACComponent

An HVACComponent is mainly a Catalogue item (eg Tee) however it may also be a custom designed component.

HVACDuct

An HVACDuct is a straight section of duct that is usually a bulk item cut to size.

PipingSystem

A PipingSystem is a stand alone object that has no direct children but may have many Associations. Whilst it is possible, it is not envisaged that the Plant items involved will have Associations to the PipingSystems to which they belong.

Note that these are lifecyle objects in that they may be required for a specific part of the lifecycle but have little meaning elsewhere (eg. For a Commissioning Package).

PipingNetworkSystems

PipingNetworkSystems are concerned with the arrangements of interconnected PipingNetworkSegments which in turn contain PipingComponents, ProcessInstruments that take part in the flow of the fluid that they contain. The 3D model is a spatial layout and represents a one to one relationship with the P&ID model at the PipingNetworkSegment level, with the exception of the special symbols on the P&ID. Also there are PipingComponents in the 3D model that are not on the P&ID (pipe supports, elbows etc).

A key here is that the topology for both is the same as this is what can drive from or validate the 3D model against the P&ID. The definition of the PipingNetworkSegment is a vital part of this that enables the PipingNetworkSystem to be decomposed into single flow sequences that directly relate. Where a key engineering parameter changes or the flow splits then the segment ends.

PipingNetworkSegments

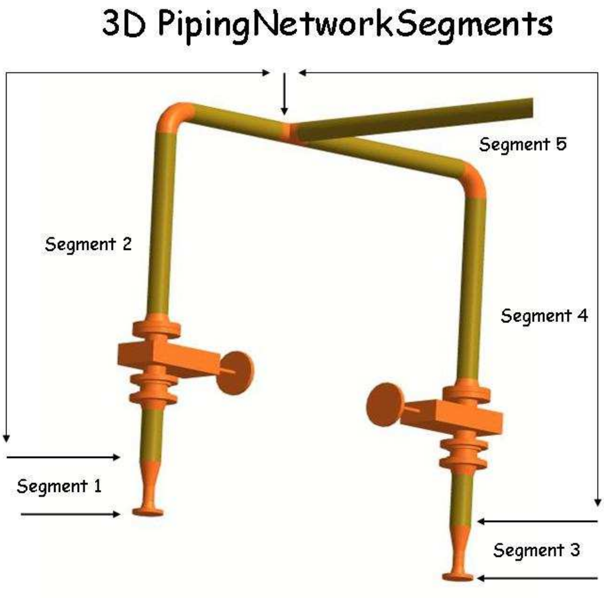
The key to a PipingNetworkSegment is that it has a single start and end. Components in it are ordered from head to tail. The connectivity of PipingNetworkSegments themselves is through the Connection object, which identifies the plant item and Node of the item that the segment is connected from (upstream) and to (downstream). The plant item can be identified by its Tag or ID if it has no Tag.

Each PipingComponent and ProcessInstrument will have Attributes Flowin and Flowout which identify the Node of the Component where the upstream and downstream connections are made respectively.

The topology for PipingComponents and ProcessInstruments (inline instruments) in the PipingNetworkSegment is by their order in the PipingNetworkSegment whereas the connections to ProcessInstruments that are not part of the flow (eg. Pressure indicator), are not. Olets may be present in the PipingNetworkSegment where small bore pipe connects between a Pipe and a ProcessInstrument. Offline instruments may or may not be modelled in 3D.

The PipingNetworkSegment has a CenterLine that defines the route of the PipingNetworkSegment. This should have a coordinate at the origin of each Component as well as the start and end of the segment. The topology has a clear order. Where the flow of a PipingNetworkSegment is both ways then there will be a DualFlow attribute set to True and the topology defines the order from one end to the other. In this case either end can be the head.

This diagram shows an example of part of a PipingNetworkSystem and it constituent PipingNetworkSegments.



XML Fragment

Segment 1

This starts at the Nozzle of a Pump (not shown) and ends at the top end of the Reducer (Large radius). The segment ends because of the diameter change.

```
<PipingNetworkSystem  ID=”XMP12” Tag=”100-B-1”>
    <PipingNetworkSegment  ID=”XMP14” Tag=”100-B-1-B1-1”>
        <Connection FromID=”P1502A-N2” ToID=”XMP_25” ToNode=”1”/>
        <Pipe …/>
        <PipingComponent ID=”XMP_15” ComponentClass=”Gasket” …/>
        <PipingComponent ID=”XMP_23” ComponentClass=”Flange”  …/>
        <PipingComponent ID=”XMP_25” ComponentClass=”Reducer” …/>
    </ PipingNetworkSegment>
```

Segment 2

This starts at the top end of the Reducer and ends at the branch Node of the Tee.

```
 <PipingNetworkSegment ID=”XMP_32” Tag=”100-B-1-B1-2”>
        <Connection FromID=”XMP_25 FromNode=”1” ToID=”100-B-1-B1-TEE” ToNode=”3”
        <Pipe ID=”XMP_28” …/>
        <PipingComponent ID=”XMP_23” ComponentClass=”Flange”  …/>
        <PipingComponent ID=”XMP_15” ComponentClass=”Gasket” …/>
        <PipingComponent ID=”XMP_25” Tag=”V102” ComponentClass=”WaferCheckValve” …/>
        <PipingComponent ID=”XMP_26” ComponentClass=”Gasket” …/>
        <PipingComponent ID=”XMP_27” Tag=”V103” ComponentClass=”GateValve” …/>
        <PipingComponent ID=”XMP_28” ComponentClass=”Gasket” …/>
        <PipingComponent ID=”XMP_29” ComponentClass=”Flange”  …/>
        <Pipe ID=”XMP_30” …/>
        <PipingComponent ID=”XMP_31” ComponentClass=”Elbow”  …/>
        <PipingComponent ID=”XMP_32” Tag=”100-B-1-B1-TEE”
                ComponentClass=”PipingNetworkBranch”>
                <ConnectionPoints NumPoints=”4” Flowin=”1” Flowout=”3” ../>
</ PipingNetworkSegment>
```

Segment 3 and 4

These are the same as Segment 1 and 2 respectively except that the Connection for Segment 4 is to Node 1 of the PipingNetworkBranch. The PipingNetworkBranch is the last PipingComponent in the first Segment that references is as the downstream end of that segment.

PipeBend

A PipeBend is not a PipingComponent but the result of an action when installing a Pipe where the Pipe is physically bent to the desired angle. As such it will not appear in the BOM.

Pipe

A Pipe is a tube that conveys the fluid between each of the PipingComponents that make up the PipingNetworkSegments. The material will be defined by the Specification of the PipingNetworkSegment.

ProcessInstrument

A ProcessInstrument is a component that is part of the PipingNetworkSegment or is connected to it small bore Pipe. Eg. Control Valve or Pressure indicator.

InstrumentComponent

An Instrument is a component that is not connected directly to the process. These are unlikely to be modelled in 3D and if so would be stand alone components.

PipeSupport

A PipeSupport is not part of the flow and is therefore not Part of the PipingNetworkSegment. They are however associated with a Pipe or PipingComponent. The PipeSupport will have an Association "is associated with" to the Pipe or PipingComponent and the Pipe or PipingComponent will have an Association "refers to" to the PipeSupport.

Weld

A Weld may or may not be modelled. It is not part of a segment and is associated with two 3D objects (eg. Two Pipes). This will have an Association of "is associated with" between itself and the objects.

Plate

A plate can be either planar or can be defined by a free form surface (Nurb Surface). If no surface is defined, it is assumed to be planar. It can have a trimming outer boundary.

Cutouts for plates are supported. A Cutout is a closed boundary inside the outer boundary of the plate.

A Plate can also contain Fittings

Structure

A Structure is a collection structural, civil and mechanical items. It is a logical collection for a purpose.

A Structure may contain Structures as well as civil and mechanical elements and Frameworks.

Framework

A Framework is a collection of structural elements which may or may not have a topology.

A Framework may contain frameworks as well as StructuralSections.

StructuralSection

A StructuralSection is a general, linear or curved structural element of a defined profile. The profile can be constant or tapered. These can be custom, but for the most part they conform to a particular standard and are catalogue items.

These are in most systems intelligent with the design system able to perform automatic cutbacks and in many cases generate the joint details.

The key Attributes for a StructuralSection are:-

- SectionName
- SectionProfileType
- Standard
- CardinalPoint
- RotationAboutPlacement
- Grade

It may also contain the Elements for:-

- PlacementCurve
- ProfileCurve
- ObstructionProfileCurve
- FireProofProfileCurve
- StartProfileCurve
- EndProfileCurve
- ConnectionPoints
- Solid of Extrusion or SolidOfRevolution
- Cutout
- Component
- Fitting

The geometry is optional and there are more attributes identified in the Schema that will be expanded upon in a later version.

StructuralBeam

A StructuralBeam is a specialisation of a StructuralSection - it is usually horizontal.

StructuralBrace

A StructuralBrace is an angled StructuralSection that provides additional strength to around joint.

StructuralColumn

A StructuralColumn is a vertical StructuralSection that is designed for support.