

Tomas Sandberg

PDMS – COMOS Data transfer development

Metropolia University of Applied Sciences

Bachelor of Engineering

Mechanical Engineering

Bachelor's Thesis

14 December 2017

Author Title	Tomas Sandberg PDMS – COMOS Data transfer development
Number of Pages Date	53 pages + 7 appendixes 14 December 2017
Degree	Bachelor of Science (Technology)
Degree Programme	Mechanical Engineering
Specialisation option	Design Engineering
Instructors	Nisula Sami, Plant Engineering Manager Pekka Salonen, Senior Lecturer
<p>This Bachelor's Thesis was conducted for Andritz Oy, a Finnish subsidiary of Andritz AG, which supplies systems, equipment and services to the pulp and paper industry. The purpose of the thesis was to develop a pipe data transfer tool between the AVEVA's CAD software solutions (PDMS/E3D) and the Siemens' process design software COMOS.</p> <p>The objective of this thesis was to analyse and research the required and necessary data to be transferred from the 3D modelling software (AVEVA PDMS and E3D) to the process design software (COMOS). At the beginning of this study, a one-way data transfer was already existing and used for testing purposes (COMOS to AVEVA PDMS and E3D).</p> <p>For the completion of the thesis, research on the work mechanisms of the existing transfer tool was conducted. Additionally, further research was conducted to define the necessary improvements on the existing COMOS – PDMS tool and the requirements of the developed PDMS – COMOS tool, with the key process engineers, design engineers and the software administrators.</p> <p>As a result, data transfer tools are being integrated in the starting projects providing tangible benefits for users of both software solutions. In the future, more tools are planned to be integrated in the data transfer interface, such as model visualization and document sharing tools.</p>	
Keywords	PDMS, COMOS, PLM, Pipework design, Data transfer

Tekijä Otsikko	Tomas Sandberg PDMS – COMOS -tiedonsiirron kehitystyö
Sivumäärä Aika	53 sivua + 7 liitettä 14.12.2017
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Kone- ja Tuotantotekniikka
Suuntautumisvaihtoehto	Suunnittelutekniikka
Ohjaajat	Laitossuunnittelupäällikkö Sami Nisula Lehtori Pekka Salonen
<p>Tämä insinöörityö on tehty Andritz Oy:lle, Andritz AG:n Suomessa olevalle tytäryhtiölle, joka toimittaa järjestelmiä, laitteita ja palveluita paperi- ja selluteollisuuteen. Työn tarkoitus oli kehittää putkiston tiedonsiirtotyökalu AVEVAn CAD-ohjelmistojen ja Siemensin CO-MOS- prosessinsuunnitteluohjelmistojen välille.</p> <p>Työssä selvitettiin tarvittavat siirrettävät tiedot 3D-mallinnusohjelmasta (AVEVA PDMS ja E3D), prosessinsuunnitteluohjelmaan (COMOS). Työn alussa yhdensuuntainen tietojensiirtojärjestelmä oli jo kehitetty ja käytössä testivaiheessa (COMOS-ohjelmasta AVEVA PDMS- ja E3D -ohjelmiin).</p> <p>Tämän työn aikana tarkastettiin ja rajattiin tarvittavat parantamisen kohteet olemassa olevassa COMOS – PDMS -tiedonsiirtotyökalussa sekä kehitettävän PDMS – COMOS -työkalun vaatimukset, yhdessä eri avainasemassa olevien prosessinsuunnittelijoiden, suunnitteluinsinöörin sekä ohjelmistojen ylläpitäjien kanssa.</p> <p>Tiedonsiirtotyökalun lisääminen alkaviin projekteihin aloitettiin, ja se tarjosi merkittäviä etuja ohjelmistonkäyttäjille. Suunnitelmissa on lisätä muita ohjelmistoratkaisuja datan siirtoon käyttöliittymään, kuten mallin katselu- sekä tiedoston julkaisu- ja jako-ohjelmistoja.</p>	
Avainsanat	PDMS, COMOS, PLM, Putkisuunnittelu, Tiedonsiirto

Foreword

I express my gratitude to my parents, for all the support offered during my studies, my friends and relatives who never let me give up on my studies and helped look up also on the times I was down. I want also to thank my manager, Sami Nisula, for offering the opportunity to work in this challenging project, and the colleagues in the company who provided me guidance and help in the areas where I needed additional knowledge when writing this Thesis.

Estibaliz Ponce Miñaur

Thomas Mattick

Apolinaria Gainulenko

Pekka Rosenblad

Andritz Oy

Antti Tiihala

Otso Tiili

Ismo Latti

Jukka Kivistö

Ilari Ruohomaa

Susanna Vuorinen

Janne Tiihonen

Tomas Sandberg, B.Sc. (Tech.)

Contents

Abbreviations

1	Introduction	1
1.1	Background	1
1.2	Target and limits	2
1.3	Completion and Methods	2
2	Andritz	3
2.1	History	3
2.2	Pulp & Paper	4
3	Pulp Crafting	5
3.1	Cooking	5
3.2	Washing	7
3.3	Oxygen Delignification	9
3.4	Knott separation and Screening	10
3.5	Bleaching	10
4	Industrial pipes	12
4.1	Pipe Spacing	12
4.2	Valves and Instruments	12
4.3	Media	13
4.4	Thermal expansion	13
4.5	Supports	14
4.6	Fiberline Pipes	16
4.7	Directives and Standards	16
4.7.1	Pressure Equipment Directive	16
4.7.2	Standards	17

4.8	Design timing	17
5	COMOS	18
5.1	Structure	18
5.2	COMOS Objects	19
5.2.1	Base objects	19
5.2.2	Engineering Objects	19
5.2.3	Attributes	19
5.2.4	Equipment	20
5.3	Pipes	20
5.4	Process Flow Diagram	21
6	AVEVA software solutions	23
6.1	Hierarchy	24
6.2	Attributes	24
6.3	Design Module	26
6.3.1	Equipment	26
6.3.2	Pipework	28
6.4	Schematics	31
6.4.1	Diagrams	32
6.4.2	Schematic 3D Integration	32
6.5	AVEVA PDMS	33
6.6	AVEVA Everything3D (AVEVA E3D)	34
6.7	AVEVA PML	36
7	COMOS – PDMS data transfer	37
7.1	PDMS – COMOS Link	37
7.2	COMOS Export	38
7.3	AVEVA PDMS/E3D Import	39
7.4	Transferred data	39
8	PDMS – COMOS data transfer	41
8.1	Data storage in COMOS	41

8.2	PDMS export	42
8.3	COMOS import	44
8.4	Transferred PDMS data	44
8.4.1	Pipe centreline length	45
8.4.2	Pipe Maximum Elevation	45
8.4.3	Pipe Elbow counter	45
8.4.4	Isometric View	46
8.4.5	Instrument Position	47
8.4.6	Pipe 3D Status	47
9	Results and conclusions	49
9.1	Results	49
9.2	Future development	50
9.2.1	COMOS to PDMS Data Transfer	51
9.2.2	PDMS/E3D to COMOS Data transfer	51
9.3	Conclusions	52

Appendixes

- Appendix 1. COMOS – PDMS/E3D data import tool MXC file code
- Appendix 2. COMOS Exported XML file extract
- Appendix 3. PDMS/E3D – Comos data export tool MXC file Code
- Appendix 4. PDMS/E3D exported XML file extract
- Appendix 5. PED Classification charts, for pipes
- Appendix 6. Flow resistance monogram for clean, 20 °C water
- Appendix 7. Local flow resistance monogram for MC pulp media

Abbreviations

AEI	Automation Electrical Instrumentation
ANSI	American National Standard Institute.
ASME	The American Society of Mechanical Engineers.
COMOS	Process, automation and electrification design software by Siemens.
DIN	Deutsches Institut für Normung (German Institute for Standardization)
DN	Nominal Dimension
E3D	Everything 3D. Software solution from AVEVA to replace PDMS.
EN	Euronorm. International technical standard equivalent of a national standard in all member countries, and replaces any prior conflicting national standard
Equipment	Refers to element types for the AVEVA PDMS/E3D and COMOS Software solutions from chapter 5 onwards.
GUI	Graphical User Interface
ISO	International Standards Organization.
JIS	Japanese Industrial Standard
MC	Medium Consistency. Pulp dilution in water at 7 – 12 % by mass concentration.
MXC	COMOS Enterprise Server command file
PDMS	Plant Design Management System, 3D Plant design software by AVEVA Ltd.

PED	Pressure Equipment Directive.
PFD	Process Flow Diagram
PML	Programmable Macro Language
PN	Nominal Pressure (indicated in bars)
PSK	Prosessiteollisuuden Standardoimiskeskus ry (Finnish Process Industry Standardization Centre).
SFS	Suomi-Finland Standard.
UDA	User Defined Attributes
UI	User Interface
XML	Extensible Mark-up Language, mark-up language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.
XMR	COMOS Enterprise Server Reply File

1 Introduction

During the work of this Thesis made for Andritz Oy, a data transfer tool between COMOS and PDMS/E3D software's was developed. The aim of the work is to develop the recently released data transfer tool between COMOS and PDMS/E3D and research the requirements for tool to transfer data from PDMS/E3D to COMOS. The data transfer tool development between COMOS and AVEVA PDMS/E3D is part of larger software tool integration, involving the different tools used for the design and the company – client/subcontractor communication.

This Thesis includes a detailed description of the preliminary information required to understand it. Preliminary information includes the description of the pulp crafting process, industrial piping design as well as description of the software solutions involved (COMOS and AVEVA PDMS/E3D). Preliminary information is compiled from chapter 3 till chapter 6.

1.1 Background

The current PDMS version in use includes an automatic pipe routing tool, assisting the designers to route pipelines, but it required the manual input of pipeline data by the user from the data received from the process design software COMOS. The required time to manually input data is not cost effective as the estimation of the amount of pipe materials and data for each pipe is time consuming and may lead to human errors.

Possible data export formats from COMOS are in some extent readable for PDMS/E3D, allowing the use of a data transfer tool in PDMS, which would update the pipe data in the 3D model using as source the data from COMOS. In the same way, data export formats from PDMS/E3D are also readable for COMOS. Bidirectional data transfer can be configured between the before mentioned software solutions.

1.2 Target and limits

The target of this Thesis is to develop a tool to assist in the pipe data transfer from the process design 3D design software PDMS to the COMOS software, to provide data to the process engineers to calculate different variables of the pipe which are dependent of the defined model attributes. The tool is to be created having in mind the requirements and needs of process engineers, the Fiberline process in pulp factories and the compatibility with the possible data export formats from COMOS and the readable formats for PDMS.

The built in PML language in PDMS, allows the user to create tools to decrease the work of the designer. The Software customization through PML induces the risk that modified macros are not compatible with later software versions. PML is an object-oriented language, not providing inheritance.

1.3 Completion and Methods

Without previous knowledge on the COMOS software, ANDRITZ Oy offered a basic training at the beginning of the project work, as well as continuous guidance in learning the use of the software.

For the data transfer tool development, several meetings were held with the key personnel of the relevant disciplines of the company (Design, Process and AEI – Automation Electrical Instruments) to discuss the PDMS – COMOS Data Transfer tool requirements, as well as discussion with the software administrators of PDMS and COMOS inside the company to define the limitations of the software, if any.

2 Andritz

Andritz AG is a worldwide company supplying equipment and solutions for different industries. The headquarters of the Andritz Group is located in Graz, Austria. The company employs approximately 25 200 people and operates in 250 locations worldwide.



Photo 1: Andritz GMBH Headquarters in Graz (Austria)

Andritz Oy is a subsidiary of the group which operates in Finland and focuses on the Pulp and Paper industry systems. Its products are wood processing, fiber processes, chemical recovery and pulp processing, and offers biomass boilers and gasification facilities for energy production. Andritz Oy employs 1 200 people in Finland. Andritz Oy has product “centres of excellence” located in Lahti,

Kotka, Savonlinna, Tampere, and Varkaus. The headquarters of the company is located in Helsinki.

2.1 History

The company was founded in 1852 by the Hungarian-born Josef Körösi (Image 1) as a small iron foundry in the, by the time, independent municipality of Andritz. The latter had first been trained as an apprentice in a Hungarian iron trade in Hungary, before he came to Graz in 1847. Although Körösi initially produced only smaller castings, the company expanded rapidly and produced water turbines, cranes and pumps. As early as in 1860, Andritz AG employed more than 500 people, and in 1870 the number of employees was already 1300. Steam boilers, steam engines, mining machines and even bridges were also produced.



Image 1: Andritz Founder, Josef Körösi

Since 1990, Andritz has acquired and integrated almost 70 companies, pursuing its policy of complementary acquisitions. The largest transactions in terms of sales were

the acquisitions of Ahlstrom Machinery Group in 2000/2001, VA TECH HYDRO in 2006, and more than 95% of Schuler in 2013.

2.2 Pulp & Paper

Pulp & Paper is one of five product branches of Andritz AG (Hydro, Pulp & Paper, Metals, Separation). Pulp & Paper is further divided in four fields of operation, Pulp, Paper, Power and Service.

Pulp & Paper supplies equipment, systems, and services for the production and processing of all types of pulp, paper, tissue, and cardboard. The technologies cover the processing of logs, fibers, and waste paper; the production of chemical pulp, mechanical pulp, and recycled fibers; the recovery and reuse of chemicals; the preparation of paper machine furnish; the production of paper, tissue, and cardboard; the calendaring and coating of paper; as well as treatment of reject materials and sludge.

The service range includes modernization, rebuilds, spare and wear parts, service and maintenance, as well as machine transfer and second-hand equipment, Biomass, steam, and recovery boilers, as well as gasification plants for power generation, flue gas cleaning plants, plants for the production of nonwovens, dissolving pulp, and panel boards (MDF). Recycling plants are also allocated to the business area.

3 Pulp Crafting

Pulp factories consist of different plants, Pulp manufacturing takes place in the Fiberline, and it can be a mechanical or a chemical process. The chemical pulp crafting, also known as Kraft Pulping Process, is presented below, as only its requirements were considered during the creation of the data transfer tool.

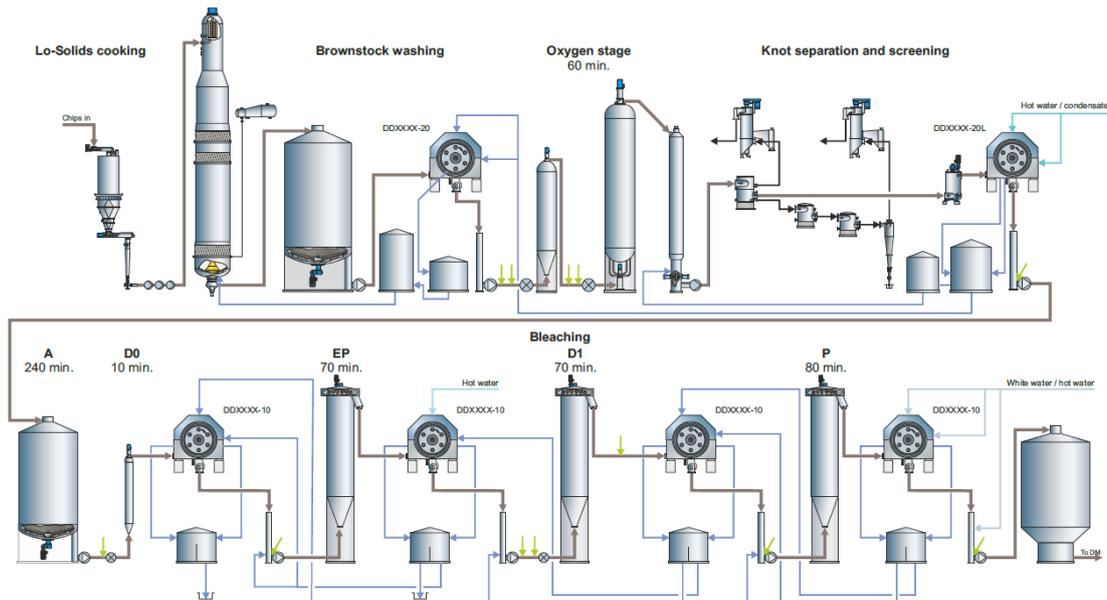


Figure 1: Fiberline Chemical Process

3.1 Cooking

Pulp crafting process starts with the preparation of the wood to be processed in the Fiberline, in the Woodyard. Raw Wood is first peeled and chopped to small pieces of roughly 20x20x5 mm in size and transported by the Chip Conveyor Belt to the cooking plant.

Chips are fed in to the Diamondback Chip Bin with the Airlock Screw Feeder. In the Diamondback Chip Bin chips are moistured with steam and impregnated with a Weak White Liquor and Strong White Liquor solution ($\text{NaOH} + \text{Na}_2\text{S}$), from where they pass through the Chip Metering Screw to the Chip Tube. At this point chips are pumped to the Digester where these are cooked with high pressure and high temperature sulphate

liquor - H_2SO_3 - to separate the cellulose fibers from the wood lignin. After the cooking, the chips are fed into an atmospheric blow tank to separate the fibers from the wood lignin.

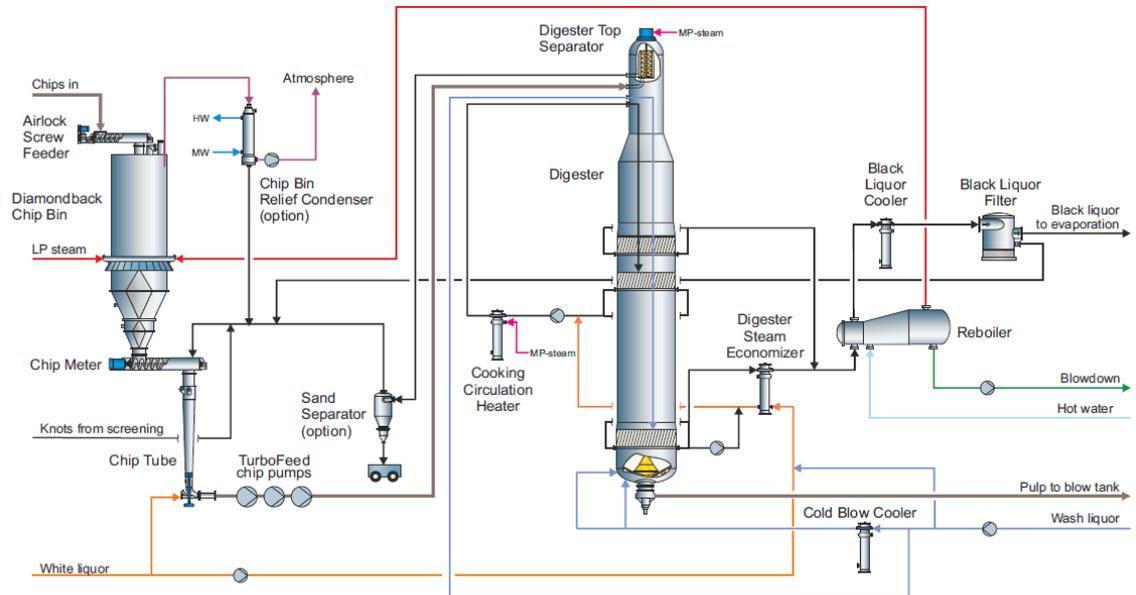


Figure 2: Cooking process

Optionally an Impregnation Vessel may be included in the process before the Digester. Then the liquor is impregnated in the chips in the Impregnation Vessel instead of the Digester. (Figure 3)

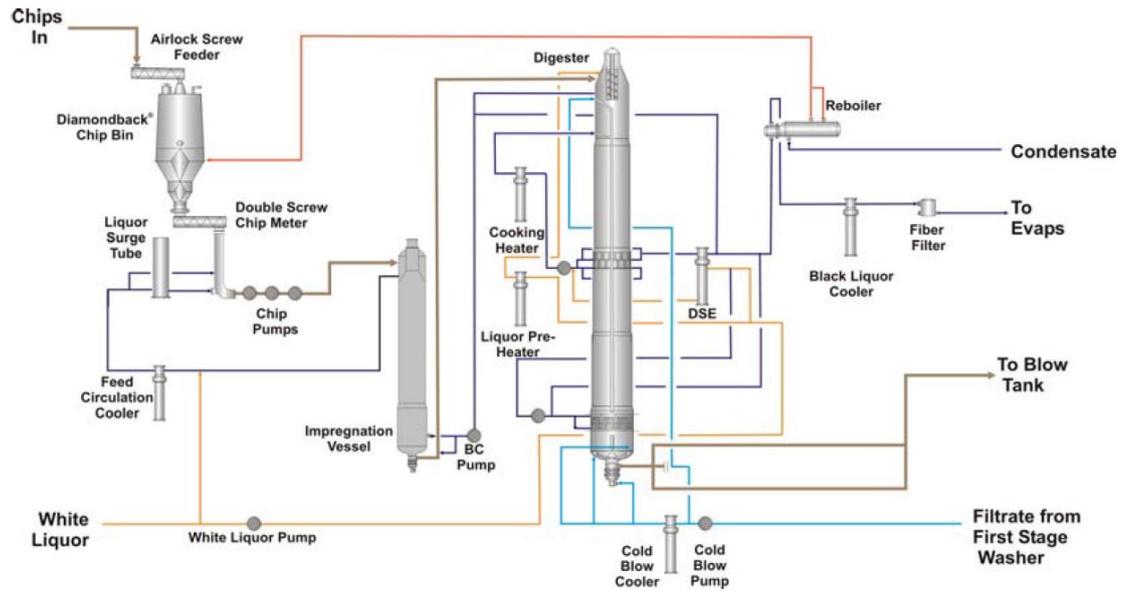


Figure 3: Cooking process with Impregnation vessel

3.2 Washing

Lignin (liquor and lignin) is separated from the pulp in the Washers (figure 4). Water circulates in counter flow; already used water is reused to clean dirtier pulp in the washer. The washing process is set in between the different delignification and bleaching stages of the process. Pulp washers use water in a counter flow way.

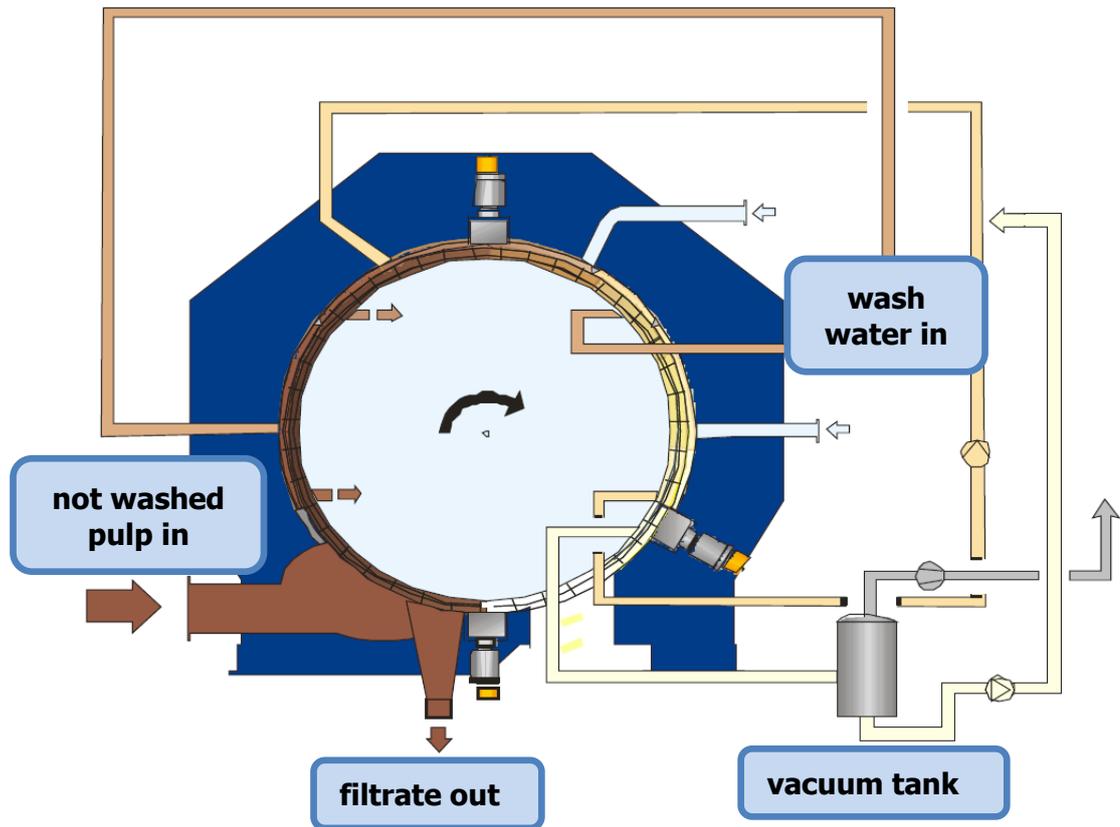


Figure 4: DD Washer wash water cycle principle

Pulp wash is performed in the Brownstock stage as well as all delignification stages. The pulp solution is pumped to the Washer from the Storage Tank or Reactor, depending on the stage in the process. The cleaned pulp passes through the Standpipe from where it is pumped to the next reactor or tank for the next stage bleaching or storage. The filtrate is conducted to the Filtrate Storage Tank, from where a part goes back to the Digester or to a previous washing stage.

The side product lignin from the Fiberline is recovered for the reuse the liquor. The lignin is first dried in the evaporation plant. After it is feed to the Recovery boiler, where it burns turning into green liquor, and producing electrical power for the plant and the local power grid. Lime is added to the Green Liquor ($\text{CaO} + \text{Na}_2\text{CO}_3 = \text{NaOH} + \text{CaCO}_3$) in the Causticizing Plant to convert it to White Liquor (NaOH), which will be used in the Cooking plant Digester to cook the wood chips. (Figure 5)

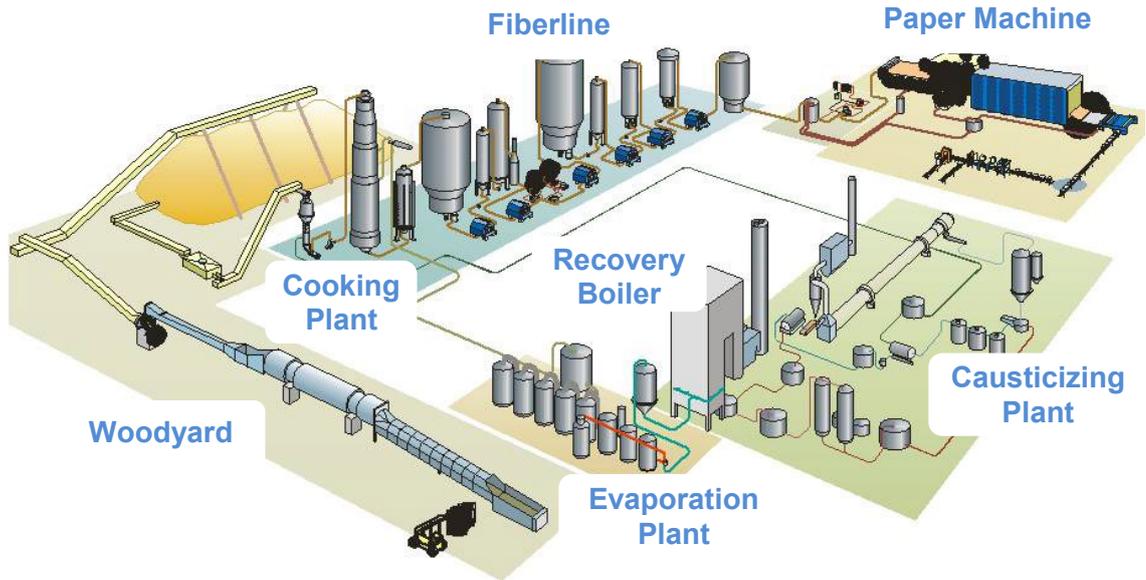


Figure 5: Pulp and Paper plant process diagram

3.3 Oxygen Delignification

In the delignification process, lignin is removed from the pulp by oxygen delignification in one or two reactors, depending on the process requirements. After this, the pulp is washed before bleaching. This is often known as Brown Stage.

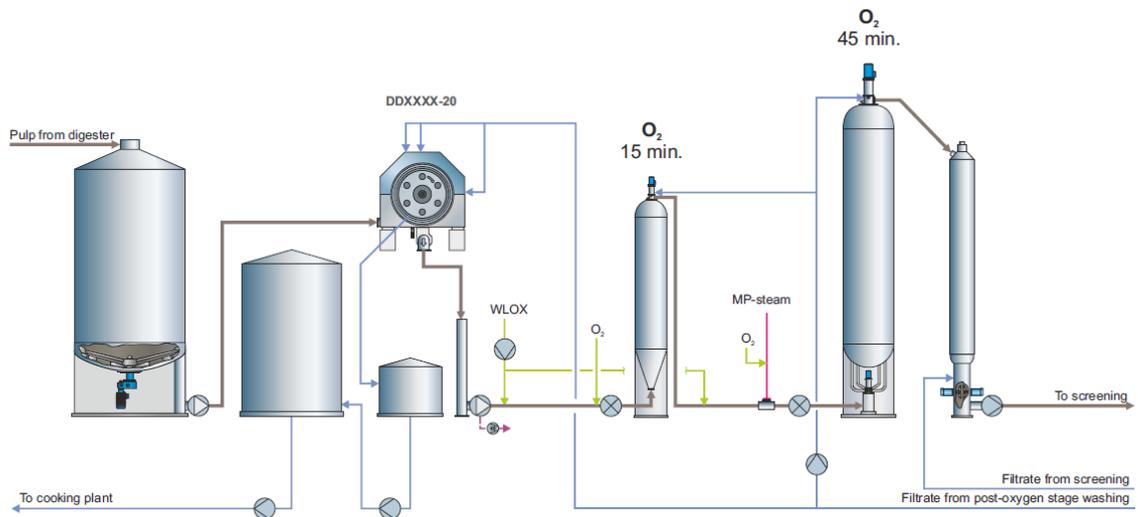


Figure 6: Brown Stock washing and Oxygen Delignification

3.4 Knott separation and Screening

The screening process is used to separate the possible knots, dirt and other debris (reject) from the pulp. Screening is carried out in a cascade process to avoid excessive amounts of good pulp to be lost with the rejects. The rejects are then either refined or sent back to the Digester. In modern processes, the screening is done after the delignification, as part of the knots become good fiber in the delignification process, avoiding unnecessary rejects.

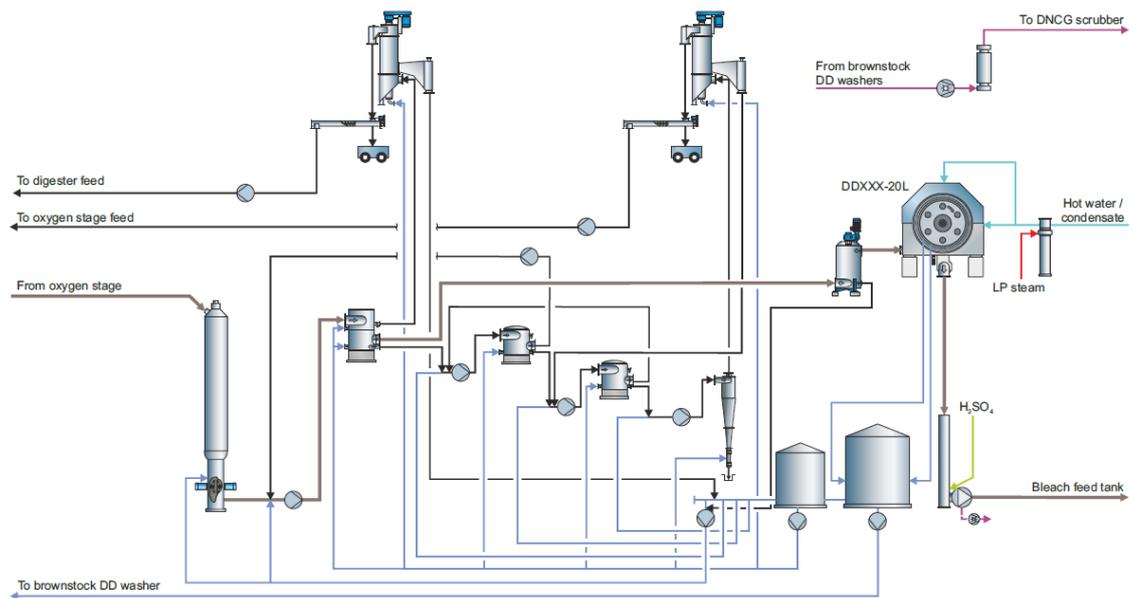


Figure 7: Screening & Post-oxygen Washing

3.5 Bleaching

Pulp Bleaching is achieved in several stages located between the washers. In modern factories the desired pulp brightness is reached with 3 – 5 bleaching stages.

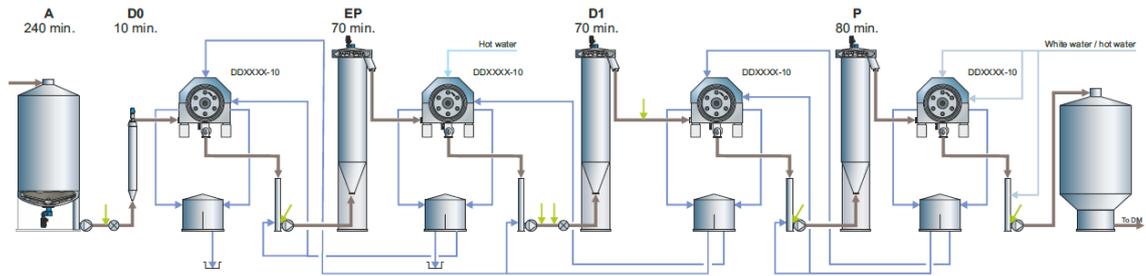


Figure 8: Bleaching stages and washing in a Fiberline

In the Bleaching stages, pulp is pumped from a Storage Tank or Reactor to the DD Washer, cleaner pulp is fed to the Standpipe from where it is pumped to the next Reactor or Storage Tank. In some Bleaching stages, chemicals are added to the pulp between the Standpipe and the Reactor through the Chemical Feeder, proper mixing is enforced with the Chemical Mixer. The chemicals added to the pulp in the bleaching stages are acidic (Oxygen and/or Chlorine).

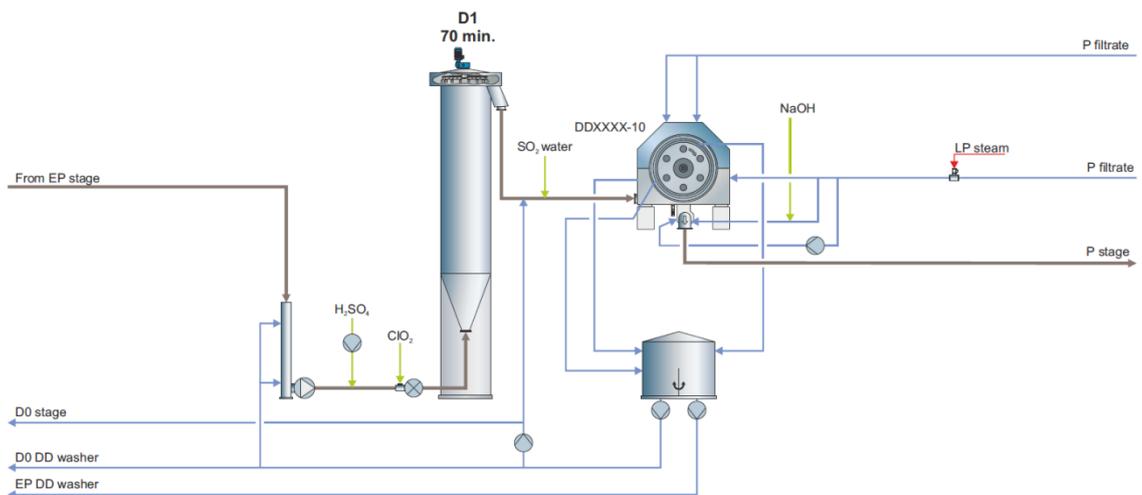


Figure 9: D1 Stage Wash

ratio to the pipe's nominal diameter, making instrumentation information a key factor when routing the pipes. Below are listed the most commonly used valves and instruments used in the industry.

Valves

- Ball Valve
- Butterfly Valve
- Check Valve
- Gate Valve
- Globe Valve
- Segment Valve
- 3-Way Valve
- 4-Way Valve.

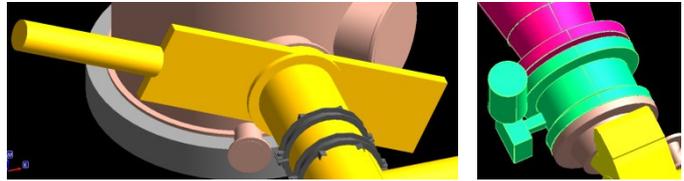


Image 3: Gate Valve and Butterfly Valve in PDMS model

Other Instruments

- Control Valve
- Meters (level, pressure, consistency, conductivity, temperature, flow)
- Samplers.

4.3 Media

Media defines different requirements of the pipe line routing. A clear example of this are the filtrate lines, which require a slope upwards (between 1° and 1.5°), and as few as possible turns with the lowest possible turning degree (max 45°).

4.4 Thermal expansion

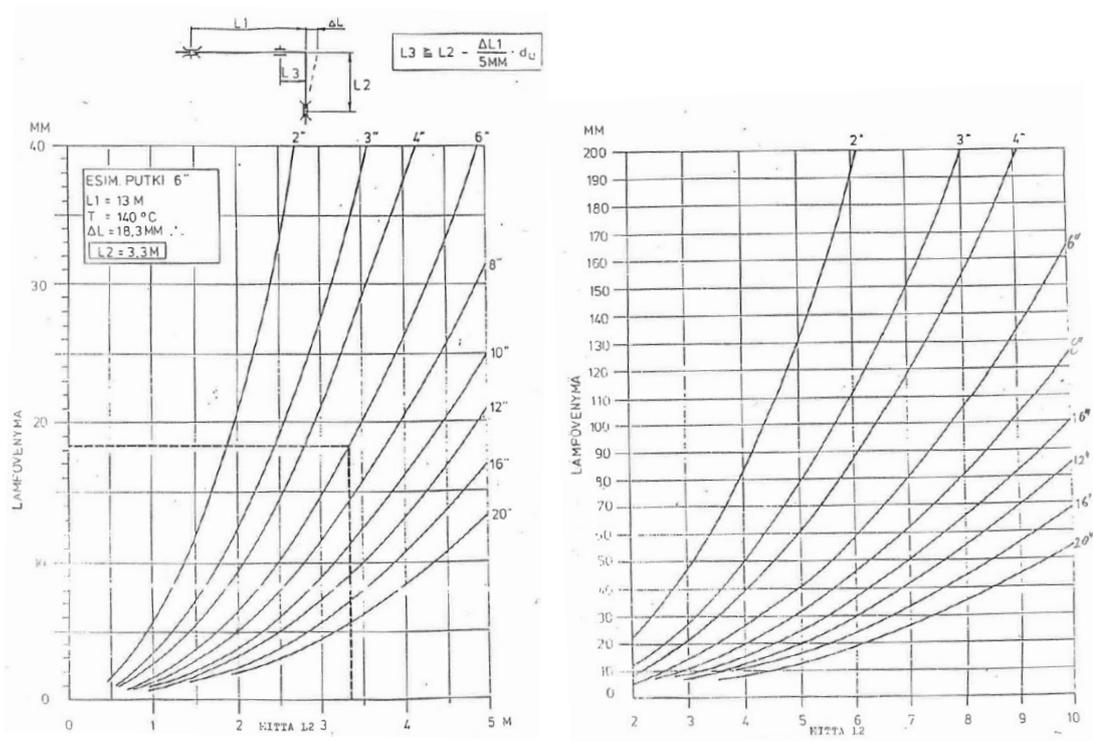
Thermal expansion compensation is required in certain pipes. This is evaluated considering different factors such as pipe length, media temperature and pipe DN size. Sometimes thermal expansion compensation is considered in the pipe routing, even in the case it is not necessary for the pipe in question. For the compensation, elbows are placed in strategical locations to limit the internal tension inside the pipe to a minimum value.

Thermal expansion is calculated using the static thermal expansion equation (1):

$$\Delta L = \alpha_L \cdot \Delta T \cdot L \tag{1}$$

Where: ΔL : Length variation
 α_L : Thermal expansion coefficient
 ΔT : Temperature variation
 L : Initial length

Thermal expansion affects the location and the type of the supports along the pipe. The following graphics show the recommended location of supports nearby a thermal expansion compensation elbow:



Graphic 1: Support distances to nearby thermal expansion elbow for carbon steel pipes

4.5 Supports

Pipe support is a relevant factor to consider when choosing a route for a pipe. Most Fiberline pipes require a minimum support distance between supports from 5 to 15 m

(liquid media), depending on the state of the media flowing, the pipe diameter, bends and the support type (Sliding, Guided or Anchor). As seen in table 1.

Table 1. Pipe support recommended distances and insulation correction factors for carbon steel pipes (PSK 7304).

Pipe DN	Gas				Liquid			
	Support type ¹		Insulation thickness ²		Support type ¹		Insulation thickness ²	
	Free	Rigid	60 mm	120 mm	Free	Rigid	60 mm	120 mm
15	1,6	2,8	0,7	0,5	1,5	2,6	0,7	0,5
20	2,0	3,4	0,7	0,5	1,8	3,1	0,7	0,5
25	2,3	4,0	0,7	0,5	2,0	3,5	0,8	0,6
32	2,7	4,6	0,8	0,6	2,4	4,1	0,8	0,6
40	3,0	5,1	0,8	0,6	2,5	4,4	0,8	0,6
50	3,5	6,0	0,8	0,6	2,9	4,9	0,8	0,7
65	4,1	7,0	0,8	0,6	3,3	5,4	0,9	0,7
80	4,5	7,8	0,8	0,7	3,6	5,8	0,9	0,8
100	5,4	9,1	0,8	0,7	4,2	6,3	0,9	0,8
125	6,1	10,0	0,9	0,7	4,8	6,8	0,9	0,8
150	7,0	11,0	0,9	0,8	5,3	7,4	0,9	0,9
200	8,3	12,6	0,9	0,8	6,2	8,1	1,0	0,9
250	9,6	14,2	0,9	0,8	7,0	8,7	1,0	0,9
300	10,9	15,4	0,9	0,8	7,6	9,3	1,0	0,9
350	11,6	16,2	0,9	0,8	7,7	9,4	1,0	0,9
400	12,6	17,3	0,9	0,9	8,2	10,1	1,0	0,9
450	13,7	18,4	0,9	0,9	8,4	10,2	1,0	0,9
500	14,7	19,4	0,9	0,9	8,5	10,4	1,0	0,9
600	16,7	21,3	0,9	0,9	8,7	10,7	1,0	1,0
700	18,5	23,1	0,9	0,9	9,3	11,4	1,0	1,0
800	20,1	24,7	0,9	0,9	9,9	12,1	1,0	1,0
900	21,3	26,1	0,9	0,9	10,9	13,4	1,0	1,0
1000	22,5	27,6	0,9	0,9	11,0	13,5	1,0	1,0
1200	24,7	30,2	1,0	0,9	12,3	15,1	1,0	1,0

¹: Support spacing (m) for uninsulated carbon steel pipes

²: support spacing correction factors for carbon steel insulated pipes

Support types (slide, guided or anchor) on the piping do not have a direct impact on whether the span is deemed rigidly or freely supported.

Pipe routes may require additional length (of up to several meters per straight section) when considering the possible support locations, as supports in larger pipes (DN 300

or above). Secondary support structures are to be anchored to a stiff enough structure. The anchor structure can be either concrete or steel (building's beams and columns, equipment shells or platform structures).

Support locations depend not only on the building's structural elements (columns and beams) location, but they also depend on the limitations set by the thermal expansion compensation turns mentioned before.

4.6 Fiberline Pipes

In the Fiberline process, the main process lines contain pulp diluted on water at 7 – 12 % by mass concentration. These lines are the most critical when calculating pipe dimensions, as pipe length, elbows, valves and other components “induce” a flow loss on the scale of meters per meter, when with medias comparable to water, the flow loss is on the scale of meters per 100 meters. In Annexes 6 and 7, are the monograms for the flow loss in different components for water and pulp media.

4.7 Directives and Standards

4.7.1 Pressure Equipment Directive

PED (Pressure Equipment Directive) regulates the inspection parameters of the pressure equipment, piping, safety accessories, pressure equipment and assemblies intended to be used for fluid media, manufactured and used in the EU member states.

The inspection parameters requirements are set taking into consideration certain technical values such as media (hazard level, state – liquid or gaseous –, temperature and pressure), equipment and piping dimensions, etc. PED Class is required in certain European standards for piping. PED Classification charts for pipes and pipe components are available in annex 5.

4.7.2 Standards

Standards set certain guidelines and limitations for the design of piping components and the routing of the industrial pipes, and in general the design for industrial purposes, around the world. The most commonly used standards in the pulp and paper industry are SFS/PSK (Finland), ISO/EN (Europe), ASME/ANSI (Americas) and JIS (Asia).

4.8 Design timing

The process and the 3-D model design work starts before the project itself is started. At this stage, the new equipment and new building elements, when applicable, are located on the model, equipment design is done at this stage as well. Once the contract has been signed, piping is designed and minor changes to the smaller location modifications are done to equipment, to allow better routes for the pipes.

Pipe design takes several months overall and during this time, changes to the process are done as well, when required. The manual updating of these changes requires many work hours that aren't available, making the available data outdated in a very short time period. Material lists are provided to the subcontractors several times during the design stage to allow the purchase of the needed materials, especially for materials with higher resistance qualities such as austenitic steel (DIN 1.4550, DIN 1.4845), high additive content steel or titanium pipe components.

5 COMOS

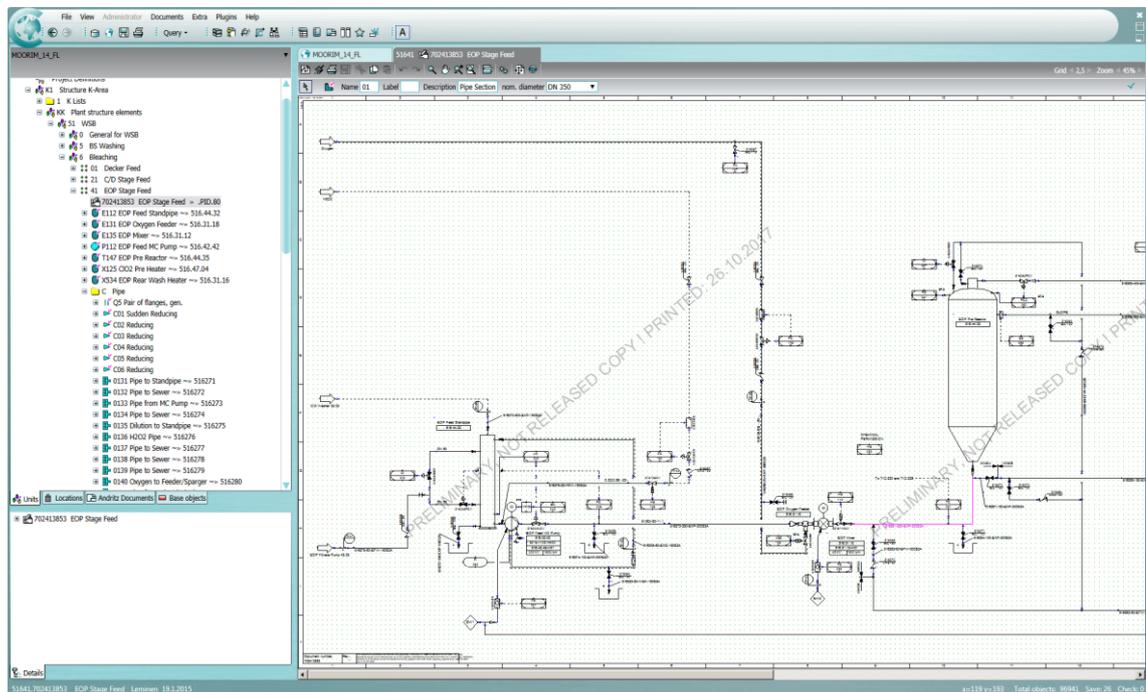


Image 4: P&ID view in COMOS

Siemens offers a unified data platform, COMOS which provides plant design engineers, plant operating personnel, company management and solution partners with a continuous flow of data that meets their specific needs across different project stages.

COMOS objects are stored on a central database. This makes the COMOS objects available for various engineering projects. Thus interdisciplinary engineering always accesses the same object.

5.1 Structure

The structure is divided into a hierarchy, Plant Level – Division – Main Process – Position:

- Plant level includes all media catalogues.

- Divisions are used to divide the plant in different global processes or departments, i.e. in a Pulp factory those may be Recovery Boiler, White Liquor Plant, Evaporation Plant, Power Boiler, Wood Yard, Cooking, Fiberline and/or Causticizing.
- Main Process is a more detailed level of the division
- Position includes all detailed processes (P&ID drawings) and contains all equipment.
- Equipment and process object level contains the detailed process engineering and E&I Engineering.

In the last level we find the different objects visible in the flowsheet; equipment (and the process relevant components of the equipment under each equipment, Nozzles and Motors), and grouped in folders the Pipes (segments and sections), Loops and Hand valves.

5.2 COMOS Objects

COMOS objects have a unique name (position name), to avoid double objects on the same project. There are various types of Objects.

5.2.1 Base objects

Base objects are general templates. Information is prepared in the base objects e.g. labels, fields, connectors, symbols and attributes.

5.2.2 Engineering Objects

Engineering objects are the result of the designer's work. Engineering objects are derived from base objects and inherit all attributes definitions of the base object. Each engineering object is based on one base object.

5.2.3 Attributes

Attributes in *Base Objects* and *Engineering Objects* are objects itself. The interesting attributes from COMOS in Pipe elements are:

- Description

- Nominal Diameter
- Design pressure
- Material
- Section Connections (Inlet/Outlet)

5.2.4 Equipment

Equipment is an object integrated in the plant under Position (e.g. 01 Feed water tank). These objects are tanks, silos, pumps, conveyors etc. Equipment can be connected to pipes through Nozzles. By default, the equipment may not have any nozzle when imported from the library, in which case the user creates the nozzle when connecting a pipe to the equipment.

5.3 Pipes

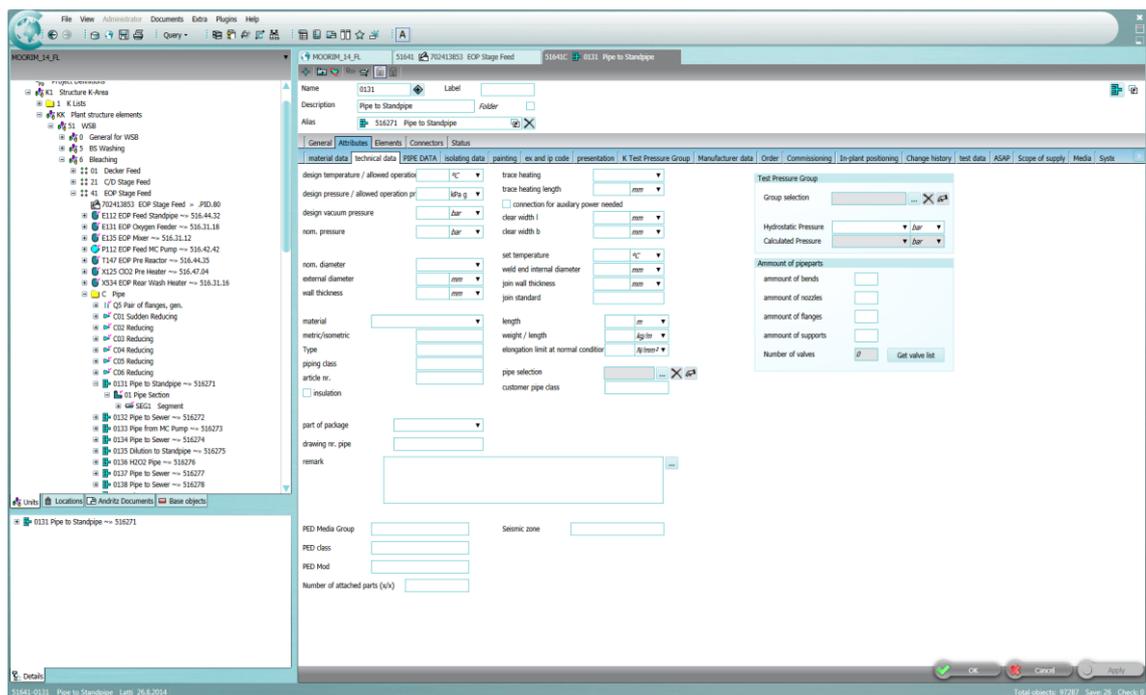


Image 5: COMOS Pipe element attributes view

Pipes are divided in *Sections*, which contain the pipe *Segments*. Pipe elements contain the technical data of the pipe (material, temperature, pressure, etc.). Section elements contain the Pipe's Nominal dimension in addition to the technical data of the pipe.

Segment elements are the graphical representation of the pipe in the P&ID and do not contain any technical data.

5.4 Process Flow Diagram

Process Flow Diagrams (PFD), commonly known as flowsheets, are used in various engineering disciplines (Process Engineering, AEI, Lay-out and Piping Engineering). PFDs contain all required plant process, which includes equipment, pipe data, instrumentation, valves and the control relation between the different meters, the automatic valves and the possible tie-in pipe connections. The pipe data used by the lay-out design engineers includes:

- Pipe Specification: material and Nominal Pressure
- Nominal Diameter: targeted maximum diameter of the pipe along the route.
- Pipe Head/Tail connection(s)
- Media: content in the pipe
- Pipe components required by the process (Tees, Reducers, Step pipes, etc.)
- Meters
- Valves
- Equipment connections

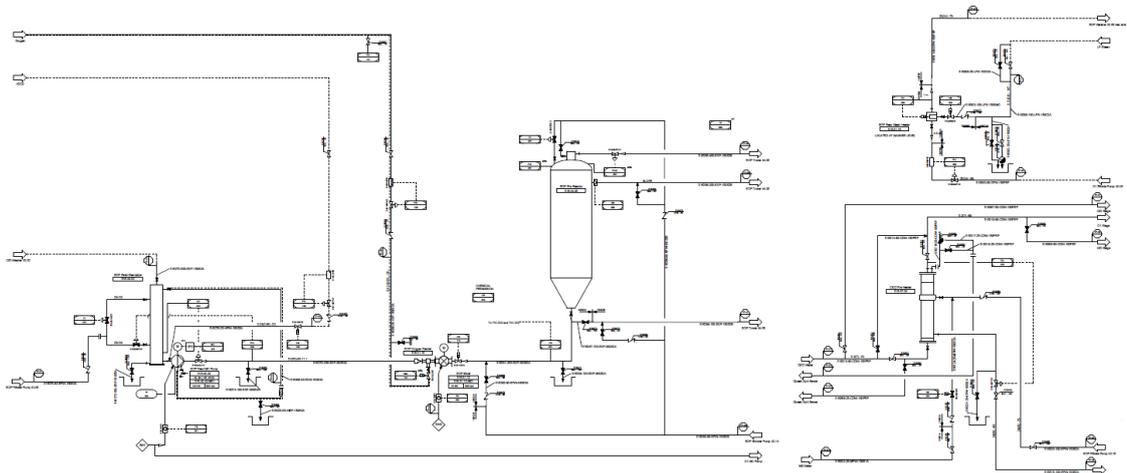


Image 6: EOP Stage PFD

Symbols in PFDs are standardized (ISO 10628). However, companies usually use their own variants for a more detailed representation of the process. Each equipment, instrument, pipe media (solid, fluid, gas, flammable, inert, etc.), has its own symbol. Pipe components can also have their own symbols in PFDs (flange, tee, reducer, step pipe, hose, etc.). PFD can also distinguish the pipes status (existing, new, to be demolished, etc.). PFDs contain detailed data related to the flow balance of the process. The flow balance related data values are the media flow, density and temperature.

6.1 Hierarchy

Hierarchy is used to limit the element types the user can design, as well as assign common attributes to all sub elements (i.e. purpose, colour, position to owner, etc.). The hierarchy also allows the management of the drawing lists in the drafting module, adding or removing not only the element in the drawing list, but all its sub elements as well. The first levels in the PDMS design hierarchy are World → Zone → Site.

- Structure → Framework → Sub-framework → Section / Panel / Plate / Wall / Floor / Primitive → Penetration
- Equipment → Sub-equipment → Primitives/Nozzles
- Pipe → Branch → Pipe Components & Instruments
- Supports → Restrain → Hanger → Clamp/Lug/Stopper.

Each structure element has an array of attributes, depending on the element type.

6.2 Attributes

All attributes of an element are parametric, allowing the user modify or create model elements through commands, introduced through a command window.

Each element in the Plant Database has a finite number of Attributes. Attributes are common for a group of elements, whereas the data type (string, number, etc.) differs according to the element type. When an element is created in PDMS/E3D, a group of attributes is fed to the database. Common Attributes to all elements are:

- **Reference Number**, PDMS/E3D assigns to each element, its unique reference number. The same reference number cannot be assigned to two elements, even if one of the elements has been erased from the database, and the reference number cannot be modified at any moment.
- **Name**: allows the user to define Element Names to ease navigation in the hierarchy tree. The Name cannot contain spaces and it is case sensitive. Element names always start with a forward slash (/), which PDMS/E3D adds automatically. The same name can be assigned to two different elements, with the condition that they do not exist in the same database at the same time.
- **Type**: defines what element is defined, defining what other attributes specify the element. This attribute cannot be modified at any point.
- **Owner**: defines the position in the hierarchy, and what attributes are inherited from the owner.

- **Position:** defines the position of the element in reference to the owner.
- **Orientation:** each element has its own X, Y & Z axis, according to which the element visualization is created, the model is defined by the N/S, E/W & U/D axis. The orientation attribute defines the orientation of the element axis in relation to the model axis.
- **Obstruction:** defines how the clash analysis tool analyses the element in case of other elements overlap it. By default, PDMS assigns to each element obstruction 2 (hard obstruction, no overlapping allowed). The user, if needed, can change the value to 1 – soft obstruction, for path ways and service openings – or to 0 – no obstruction, used when one element is inside another element and the clash analysis is done in reference to the other element.
- Catalogue reference: cross-reference pointing to a specific entry in the Catalogue database.
- **Specification reference:** catalogue reference specific for structural elements (beams, columns, profiles, fittings, etc.).

Database elements may also contain so-called pseudo-attributes. Pseudo-attribute information is derived from the attributes and some tools allow the modification of attributes through the modification of the related pseudo-attributes.

Attribute	Value
RefNo	=201343431/19060
Name	VL-122218
Type	EQUI
Lock	<input type="checkbox"/>
Owner	026200_PUMPS
Description	DD1 suodoksen palautuspumppu
Function	APP32-125
Purpose	EQUI
Number	0
Area	0
Usrweight	unset
Usrweight	unset
Usrcogravity	unset
Usrwcogravity	unset
Position WRT Owner	E 122500mm N 61735mm U 755mm
Orientation WRT Owner	Y is N and Z is U
Desparam	unset
Built	<input type="checkbox"/>
Ispec	Nulref
Uwmbxt	unset
Dscode	unset
Ptspec	unset
Inschedule	unset
Skey	unset
StlRef	Nulref
Sprel	Nulref
Fstatus	unset
Mdsysf	unset
Inprtref	Nulref
Ouprtref	Nulref
:EXCLUDED	<input type="checkbox"/>
:ComosUID	0000000000
:AND_STATUS	
:MPOINT	unset
:Colour	1
:EQTYPE	unset
:ALTPOS	unset
:BUILT-BY	unset
:DRYWEI	unset
:DYNFAC	1.3000
:WINDF	unset
:WINDM	unset
:ERTQFH	unset
:ERTQFV	unset
:ERTQM	unset

Image 8: PDMS Attribute Viewer

AVEVA solutions allow the user to define its own attributes, known as UDAs (User Defined Attributes). These attributes can be identified easily by the “:” in front of the attribute name. The user can freely define all parameters of the UDAs, such as in which element types it is shown, content validation (text string, number, list, etc.), size, default value (if applicable). UDAs are created and edited with PML code.

6.3 Design Module

PDMS contains several modules, the only relevant ones for the designer, are the Design Module and Draft Module. All elements in the different modules are structured in a hierarchy tree.

In the Design Module, the user models equipment (tanks, pumps, etc.), pipes, structures, and construction buildings. The design can be made faster by taking advantage of the libraries inside the database, and thanks to libraries and databases, different information can be exported for reports.

The Design Module comprises the different tools to create a representation of different elements of the 3D model. Tools are grouped depending on the design area:

- Equipment
- Pipework
- Cable trays
- HVAC (Ducts)
- Structures (Buildings, Pipe Bridges, Platforms, Pipe Supports)
- Beams & Columns (Sections)
- Panels & plates
- Walls & Floors
- Access, Stairs & ladders.

6.3.1 Equipment

Equipment (EQUI) items are built up in PDMS using elements known as Primitives. Each Equipment can comprise any number of primitive shapes positioned to form the item.

Sub-equipment (SUBE) is an optional element to further subdivide an EQUI. The SUBE can also own primitive elements. When used, the equipment is divided in parts according to the function (motor, body, pump, tubes, supports, foundation, grouting, etc.).

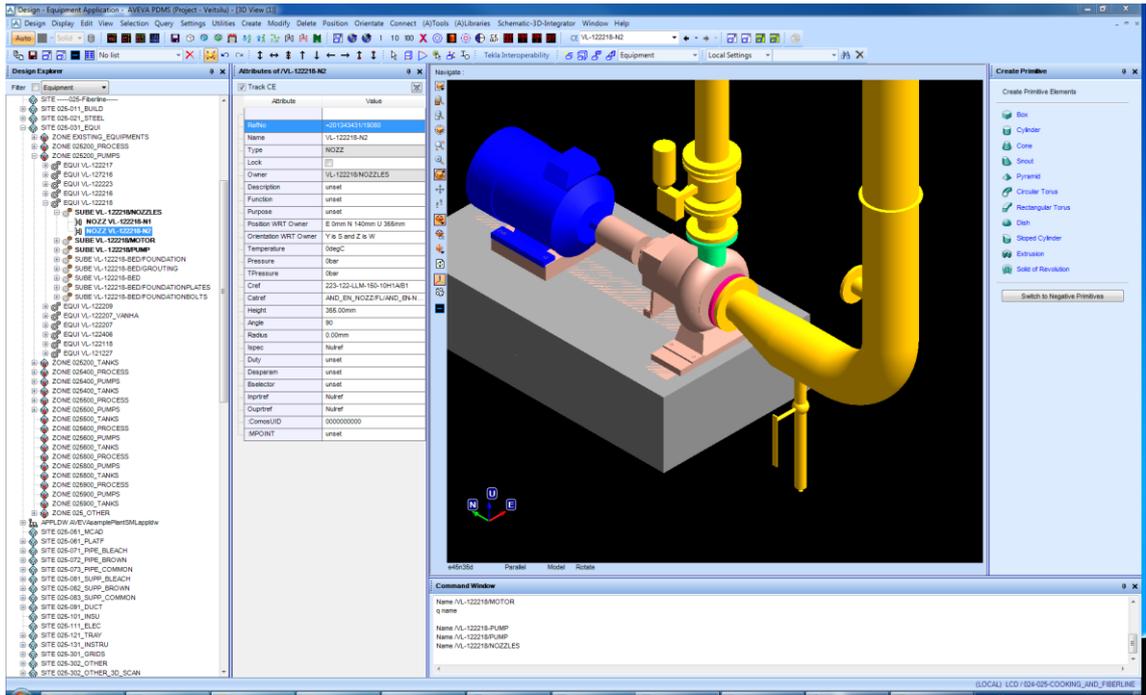


Image 9: PDMS Equipment design Module view (Equipment Nozzle highlighted)

Primitives may be owned directly by the Equipment element or by a Sub-Equipment element. Primitives are used in other disciplines as well to create catalogue components (Pipe components, Structural profiles, cable trays, stairs and ladders, etc.).

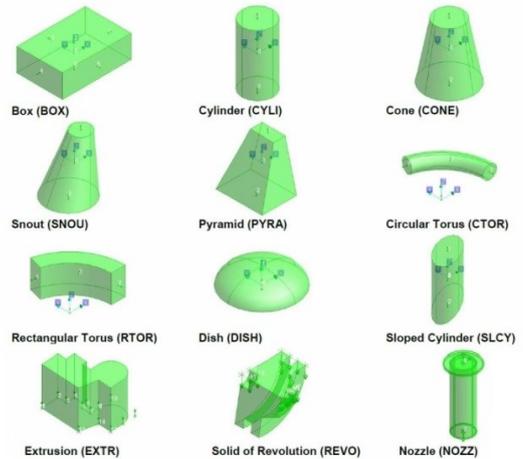


Figure 10: Graphic representation of available primitives in PDMS/E3D

The 3-dimensional shapes in Figure 10 are available as Primitives (Nozzles are not primitives themselves, but catalogue elements). Primitives can also be used to create voids, with the use of Negative Primitives.

Primitives are the core parts of any other element in the 3D model. In the design module primitives are the blocks used to create equipment elements and catalogue elements such as pipe components, air duct components, cable trays, structural components (concrete columns and beams, steel profiles, panels, walls, floors, penetrations and openings, stairs, ladders and handrails) and pipe supports.

The range of possible primitives allows the designer to model with good enough detail the volume required by the equipment. Primitives are based on three basic geometric shapes (square, triangle and circle), the use area of the software (chemical industry, nautical and off-shore pipe design) does not require more detail than the provided by the before mentioned shapes. Detailed design on non-piping elements is not required and a tolerance of ± 10 mm is commonly accepted. Nozzles are required to be positioned with ± 0.1 mm accuracy.

Primitives can also be used to define volumes reserved in the model for personnel and the mobile machinery access to the different parts of the plant. This is exploited with the Obstruction attribute and is to be set to value 1.

Nozzles (NOZZ) are significant to the Design because they provide the link between Equipment and a Pipe (Branch) which is connected to each one. Nozzles have a specification attribute, which defines the possible pipe components that can be attached to it, i.e. a flanged Nozzle of PN6 will not accept a PN10 Gasket. As the equipment is often imported to the model through a library, Nozzles have the same name and description.

6.3.2 Pipework

The Pipework tool set comprises the different tools to create pipes in the 3D model, with which the designer can create elements referencing to the pipe component catalogue.

Pipes run between several end connection points and are grouped by the flowing media. Each Pipe must have at least one Branch; the maximum amount is not limited.

Branch elements are sections of a pipe, which have defined start and finish points (Head and Tail). They may be connected to Nozzles (Equipment), Tees, other Pipe/Branch Head or Tail, or other Pipe components with more than two pipe connection points, depending on the configuration of the pipe, or left open ended.

A Branch can contain an unlimited quantity of **Pipe Components** which define the Branch's shape and geometry, and therefore the Pipe's. Pipe components are stored in a catalogue built in PDMS and are designed in the database according to the standard

for the specific pipe specification. When the designer wants to use a specific component, a copy of the component can be imported to the 3D model from the pipe component catalogue.

Pipe tubes are automatically imported to the 3D model by the PDMS Pipe Tools, and its 3D representation is visible in the 3D model when the inlet and outlet of the consecutive pipe components are aligned within the built-in tolerance. Pipe tubes are not visible automatically in the hierarchy, but Pipe Tube's attributes can be modified through the other pipe components (LSTUBE Attribute, defines the leaving pipe tube representation). Pipe components may have more than two connections. However, only two are used at the most for the connection of the component to the other components of the branch.

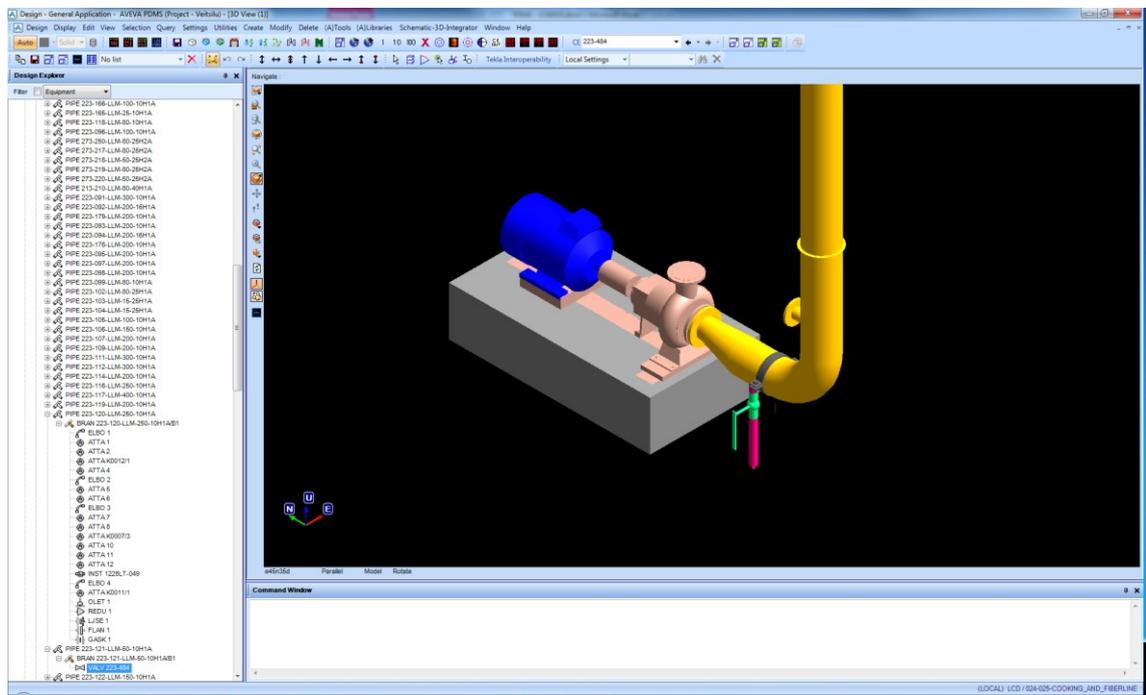


Image 10: AVEVA Pipework view Drain Pipe's Hand Valve selected and drain pipe branch highlighted

Pipe branches require start and end points before the designer can import any pipe component under the branch elements. Below are listed the possible connection points of the head and the tail of a branch:

- Nozzle
- Branch head/tail

- Tee
- O-let
- First/Last member.

Nozzle connections are required when the pipe is connected to Equipment. Head/Tail connection is used when the pipe specification change is required by the process, mostly due to a pipe material change or the media inside the pipe changes. Elbow and O-let connections are used to connect the unconnected end of the before mentioned components to another branch. When the pipe end is open, the pipe start and/or end point is connected to the First/Last Member of the Branch accordingly.

The pipe routing tool in PDMS offers some more options than the ones listed above. Those other connection points are not listed in this Thesis as they are not used to route the pipes in a Fiberline.

To route the pipe branch, the designer first assigns a connection point to the Head and the Tail of the branch and then selects the different pipe components used to route the pipe branch. The designer can design the pipe in the flow direction or in the counter-flow direction. By default, the pipe is routed from Head to Tail, unless the setting is changed in the component selection tool. The tool connects the component directly to the previous component (or the next, in the case of counter-flow routing). In case of inconsistency between two consecutive components (Nominal pressure, bore, etc.), the tool does not connect the components adding a spool space (100 mm) between the components.

The tool does not offer the Tube component, as the pipe tubes are created automatically by the tool when the outlet of the component and the inlet of the next component are aligned. The routing tool assists the designer in the routing task with the Align and Orient buttons, available in the component selection and component modification tools.

Automatic Pipe Routing

The PDMS design module allows the automatic generation of pipes using the orthogonal route when the pipe head and tail are known. The automatic pipe routing creates the centreline of the pipe using elbows and bends, through the simplest route without constraints, resulting in the shortest possible route.

The automatic pipe routing tool avoids all possible obstructions already existing, such as equipment, structure and building elements. The user can define penalty volumes which specify the additional restricted areas (pass ways, equipment maintenance required volumes, etc.) with the weight attribute. The user can configure preferred volumes for the pipe route (a maximum of 1,000 routing volume elements are considered by the automatic routing tool); planes that guide the automatic routing tool to preferred areas (a maximum of 150 routing plane elements are considered by the automatic routing tool) and the axis configuration, which determines the axis system to follow in case it is not the same as the axis system of the World.

Pipes to be automatically routed must have empty branches with only the head and tail connections defined. Nozzle connections have a default spool distance of 3 times the bore size of the associated nozzle between the flange and the first elbow. The user can change the spool distance to another value (e.g. a value 4 creates a spool distance 4 times of the nozzle bore).

When creating a route for multiple pipes with the Automatic Pipe Routing tool, the user must also determine a set of variables to the tool, such as the location and dimensions of the rack, the arrangement configuration of the pipes in the rack, the order in which the pipes will be placed and the minimum clearance between adjacent pipes. Additional commands can be used for the detail design of the routing with the automatic pipe-routing tool.

6.4 Schematics

The AVEVA Engineering contains the piping data database (Paragon). P&ID Data can be imported to AVEVA Databases from other software and it is stored in the Schematic Database. The Schematic Database was introduced for the first time within the software solution package E3D. Within the Schematic Database, the user is able to do the process design. It also allows the option to import the data from other software such as COMOS.

Once the data is imported to the schematic database, the data can be viewed and managed via a hierarchical data explorer (the Schematic Explorer). This explorer includes facilities for releasing selected schematic elements to the AVEVA Schematic 3D

Integrator, for integration with the 3D Design Model database. Facilities are also provided to view P&IDs, both before and after import. A set of standard PDMS report templates for the schematic database are also included. The Architecture of the Schematics database is available in Diagram 1.

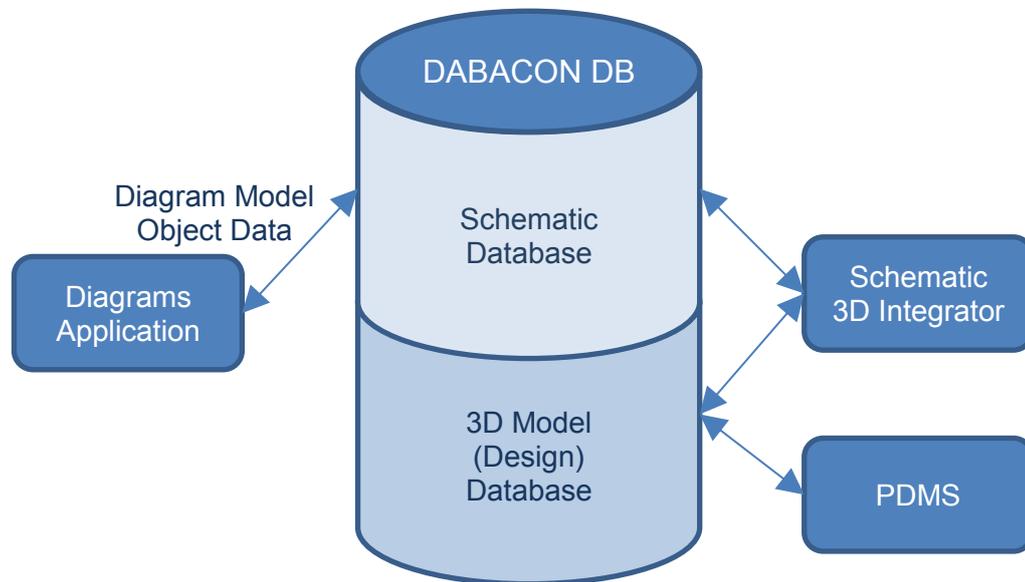


Diagram 1: Schematics Database Architecture

6.4.1 Diagrams

The module additionally allows the import of schematic elements from other software, such as COMOS, through the Schematic Model Manager tool. The functions available in the application include standard MS Visio functions and specific Diagrams functions, in addition to the standard PDMS functions.

The object data created through the Diagrams application is stored in the Schematic Database. This data is possible to re-use for 3D modelling through the use of Schematic 3D Integrator.

6.4.2 Schematic 3D Integration

The Schematic 3D Integrator allows the user to create elements in the 3D model using the data from the Schematic database. It also allows the users link the elements in the

schematic database to the corresponding 3D model elements and compare them for inconsistencies in connectivity or attributes.

6.5 AVEVA PDMS

With AVEVA PDMS (Plant Design Management System) software the user can create process plants' 3D models. All projects have their own database, which is available for all users involved in the specific project. Users can read information, e.g. text reports, technical drawings and the 3D visualization from all different modules. The user can navigate to any element of the different databases and read its content from any module, using the command window or the attribute viewer. To be able to edit the different attributes, the user needs to have the rights to write on the specific database where the data is stored.

AVEVA PDMS Environment is the group of settings for the GUI (Graphical User Interface). It defines the position and size of the different tool windows and bars, as well as the colouring rules of the different elements in the 3D model.

All model elements can be exported to a macro text file which contains all commands to create the different sub-elements under the exported element. The macro file is editable with a basic text editor (i.e. notepad) and can be imported to a model either with the Import tool, or simply dragging the file to the Command Window, and PDMS will attempt to create the elements on the macro file under the selected element in the database if the user has the writing rights to the database.

6.6 AVEVA Everything3D (AVEVA E3D)

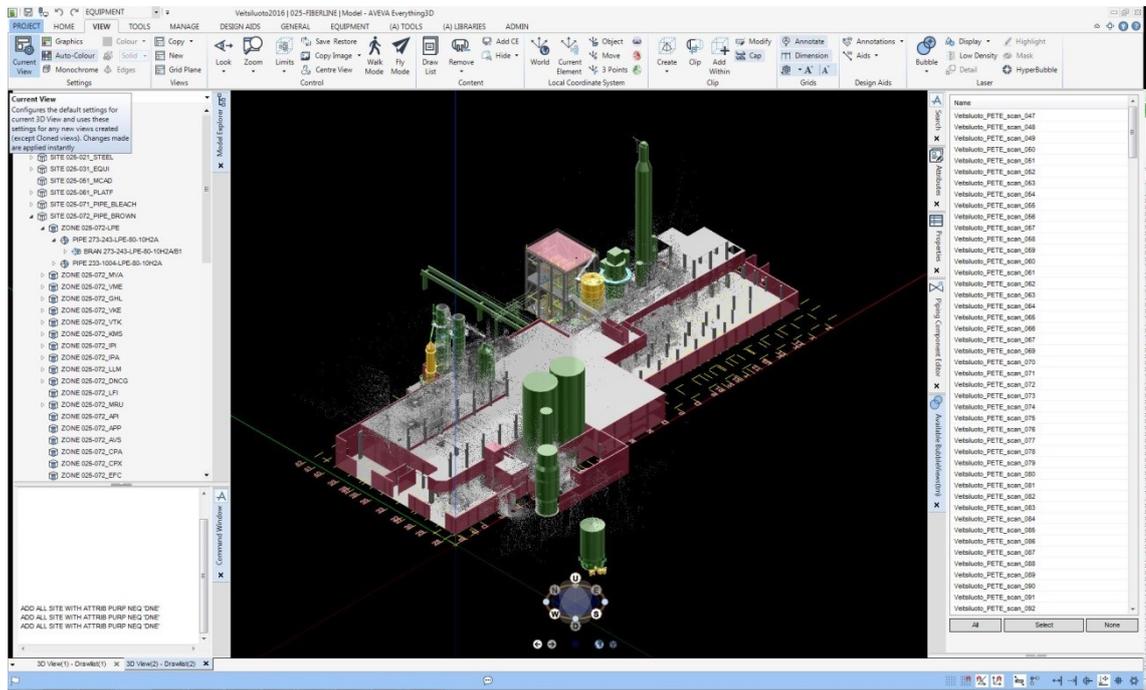


Image 11: E3D Model with integrated Point Cloud visible

AVEVA E3D is a 3D design solution, based on the AVEVA PDMS, for the process plant and power industries. Both software solutions share the same structure in the database level and the User Interface level.

E3D offers the possibility to integrate a scanned point cloud to the database allowing the user to make measurements between elements of the point cloud and the 3D model. The user can navigate to each scan point bubble. Navigation to the bubbles can be done through the bubble list (Image 11), or by selecting a bubble in the 3D Model Viewer.

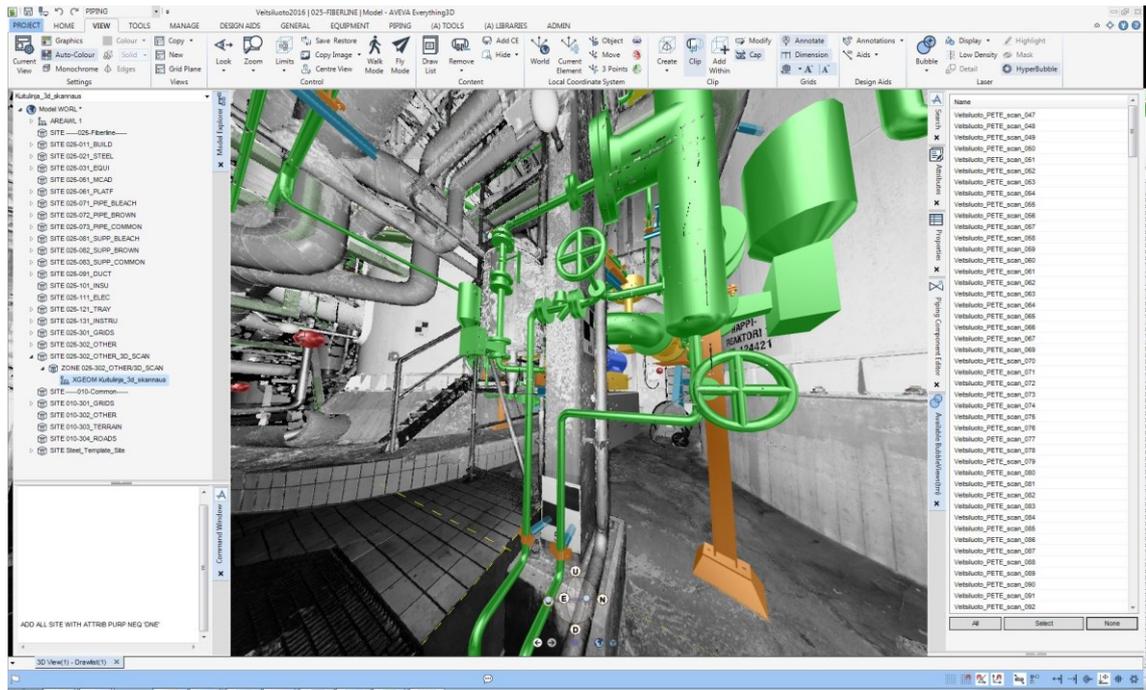


Image 12: Bubble view of the E3D model with Point Cloud

The point cloud can be visualized as surfaces also outside the bubble visualization using the Hyperbubble tool. The Hyperbubble visualization integrates in the 3D Model Viewer the 3D colour image acquired by the 3D scanning device.



Image 13: E3D Hyperbubble view, with scanned model and 3D model design elements

The pipework tools have been centralized in a single tool and new functions have been added. The new features include the reconnection of a branch head/tail back to the set head tail element when the connection is not to the branch's first/last member. The new AutoRoute function routes the branch between the existing components using the default selections from the branch specification. When components are placed on the branch, there is a required minimum snap distance (set in ratio to the OD). If the tube between components is shorter than the predefined snap distance, the components will be connected rather than placed.

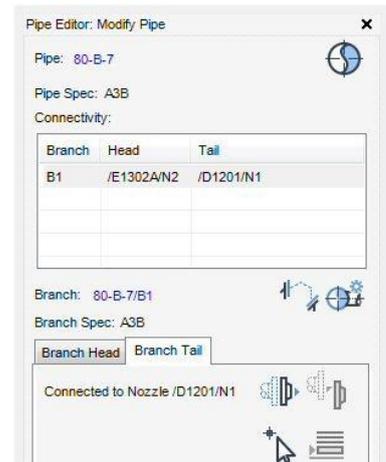


Image 14: Pipe editor form in E3D

6.7 AVEVA PML

PML (Programmable Macro Language) is AVEVA's programming language for the customisation of the PDMS software. The current PML (referred as PML2) is mainly an extension of the original PML facilities. PML Internal format for string data is Unicode UTF8 format. PML numbering starts from 1, instead of the commonly used by other computer language codes, 0.

PML is used to design additional and better suitable GUIs for the user's needs. These can be additional tools, elements, attributes, dialogues, forms, menu objects, etc. To build the code for different tools, the user can use either PML1, PML2 or combine them. The PML code is based on objects to which the user can apply methods, which may return a different type of object for the output variable. The methods can be chained. In the form and menu object customization code, there are also the gadgets, members and methods.

7 COMOS – PDMS data transfer

For the data transfer from COMOS to PDMS to be successful, all relevant elements (Pipes and the pipe connection points) require to have a defined “:COMOSUID” attribute set. This requirement provides the administrators the necessary freedom to use enough variety of tools and methods for the data transfer.

The transferred data is stored in the E3D’s Schematic and Engineering databases of the project. The Schematic and Engineering database allows the storage of PFDs, and make them available to the PDMS/E3D user.

7.1 PDMS – COMOS Link

The COMOS – PDMS link creates a link between the same components in PDMS and COMOS. To create the link, the “:COMOSUID” in PDMS/E3D is required and it is selected from the pipe list. To be able to create the pipes with the “:COMOSUID” attribute, the designer is required to login to the project with the Engineering and Schematics integration license.

7.2 COMOS Export

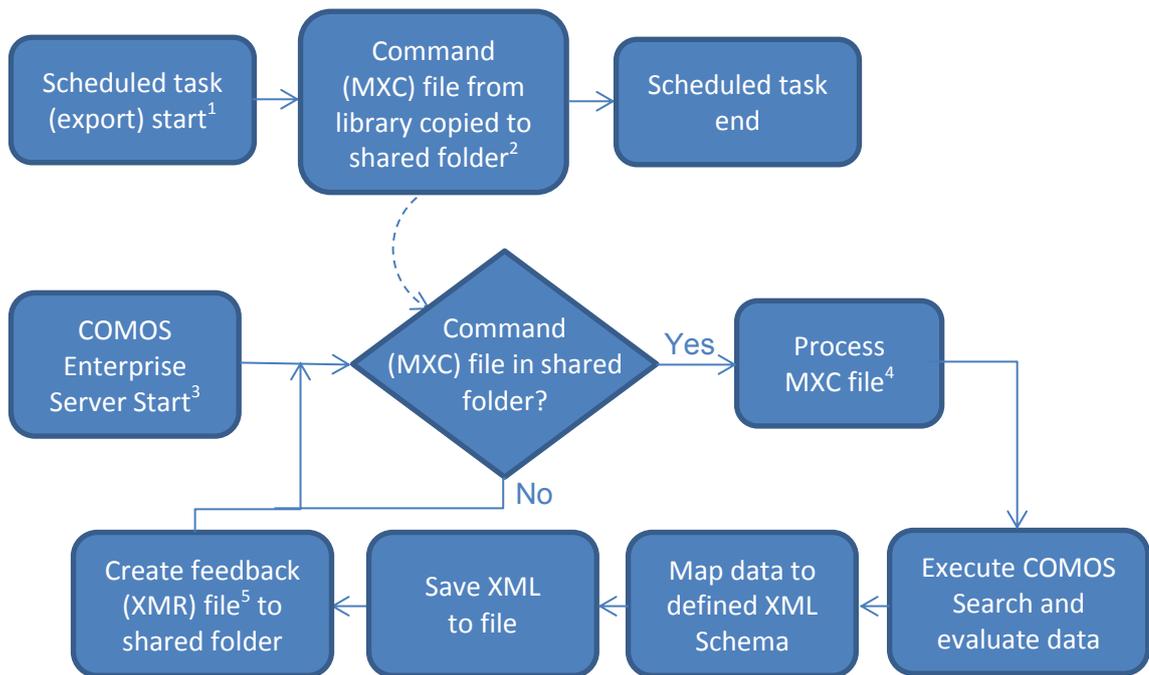


Diagram 2: COMOS Data Export

1. Windows scheduler task
2. Predefined system folder that is polled by COMOS enterprise server for command files
3. Windows service starting automatically on server start up
4. Command file contains parameters of executed action such as COMOS project, action type, target file path etc.
5. Feedback file created automatically by system at the end of each task. It provides status and other details of the task execution

The data transfer is initiated by the Windows scheduler in a daily basis at a predefined time, copying an MXC file to a shared folder starting the file command and ending it with the creation of an XMR feedback file to the predefined shared folder.

COMOS data exporter creates two XML data files, Level 1 and Level 2 files. The level 1 file contains the pipe specification data (UID, name, description, flow sheet number, etc.), the pipe technical data (specification, PED class, media, Nominal Diameter, operating values – pressure, temperature, etc.) and connections (head and tail). The level 2 data file contains the instrumentation and component (reducers, tees, valves, meters, etc.) data.

7.3 AVEVA PDMS/E3D Import

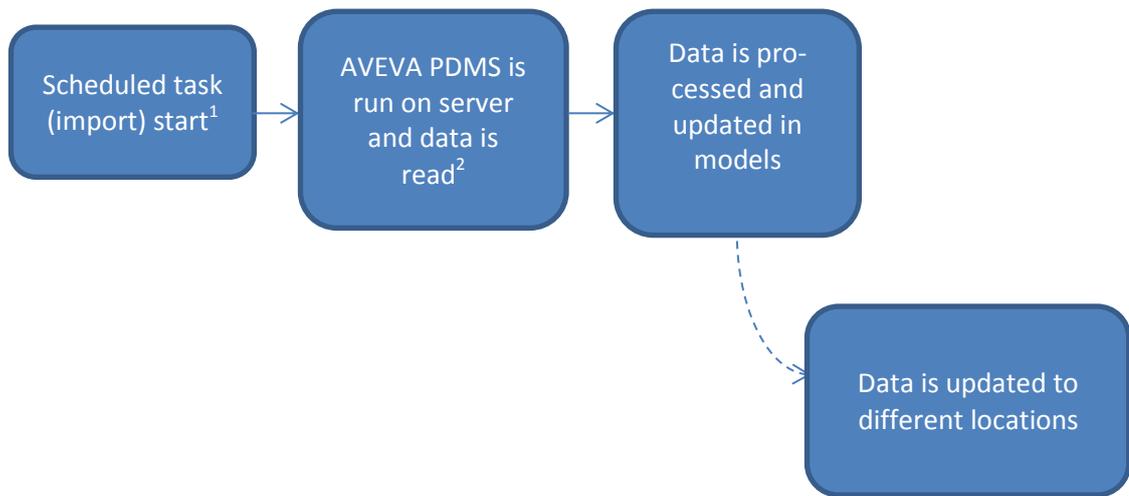


Diagram 3: Data Import tool to AVEVA PDMS

1. Windows scheduler task
2. Predefined system folder that is read by AVEVA PDMS

The PDMS Import tool code is available in Appendix 1.

Similarly as in the COMOS export tool, the PDMS import tool is initiated by a Windows Scheduler command and then the AVEVA PDMS import tool reads the XML files created by the COMOS export tool and updates the data on the different PDMS databases.

The PDMS/E3D Import tool is a set of different tools. One tool collects the required data from the different attributes of the model elements. A group of tools are required to translate the *String* data format received from the XML files to the data format accepted by the attribute receiving the data before the data is introduced to the attribute. This is the case for data included in e.g. operating pressure or temperature data (*Real*) of the Pipe/Branch elements.

7.4 Transferred data

The attributes data transferred from the COMOS databases to the PDMS/E3D databases grouped by the level are visible in table 2. Some attribute values are transferred independently of the element type, while other attribute values depend on the element type of the owner.

Table 2. Data transferred from COMOS to PDMS/E3D

Attribute name in PDMS/E3D	Level 1 object types						Level 2 object types				
	Pipe	Pump	Fan	Tank	Conveyor	GenEQ	Valve	Inline Instr.	I Nozzle	P Nozzle	Motor
@ObjectLevel	1	1	1	1	1	1	2	2	2	2	2
@ObjectType	PIPE	EQUI	EQUI	EQUI	EQUI	EQUI	VALV	INST	NOZZLE	NOZZLE	MOTOR
@Name											
ComosUID	X	X	X	X	X	X	X	X	X	X	X
Description	X	X	X	X	X	X	X	X	X	X	X
ComosBaseOb	X	X	X	X	X	X	X	X	X	X	X
ComosFullLabel	X	X	X	X	X	X	X	X	X	X	X
ComosCustomerTag	X	X	X	X	X	X	X	X	X	X	X
ProcessArea	X	X	X	X	X	X	X	X	X	X	X
UploadDate	X	X	X	X	X	X	X	X	X	X	X
Date	X	X	X	X	X	X	X	X	X	X	X
FlowSheetNumber	X	X	X	X	X	X	X	X	X	X	X
ComosDescription	X	X	X	X	X	X	X	X	X	X	X
Owner	X	X	X	X	X	X	X	X	X	X	X
ComosActive	X	X	X	X	X	X	X	X	X	X	X
DetailType	X	X	X	X	X	X	X	X	X	X	X
Manufacturer		X	X	X	X	X	X	X	X	X	X
ManufacturerType		X	X	X	X	X	X	X	X	X	X
Supplier		X	X	X	X	X	X	X	X	X	X
Medium	X	X	X	X	X	X	X	X	X	X	X
ScopeOfSupply	X	X	X	X	X	X	X	X	X	X	X
PipeSpecification	X										
PEClass	X										
DN	X										
OperatingPressure	X										
PS	X										
OperatingTemperature	X										
TS	X										
FromDescription	X										
FromComosUID	X										
GroupNumber	X										
TracingSpecification	X										
InsulationSpecification	X										
IsometricDrawingNumber	X										
MechanicalErection	X										
Team	X										
PumpType		X									
PumpCapacity		X									
PumpHead		X									
FanCapacity			X								
FanPressureDifference			X								
GrosVolume				X							
ConveyorType					X						
ValveDN							X				
ValveType							X				

The data transfer tools only replace attribute data content in the case the attribute of the element in COMOS contains information. New elements are created with the attribute data content. When the new element in COMOS did not exist in the PDMS/E3D database before the data transfer. If an element or the attribute content is deleted from the database, the data transfer does not delete the element or the content of the attribute in the PDMS/E3D databases.

8 PDMS – COMOS data transfer

Data transfer from PDMS/E3D to COMOS can have a positive impact on different project stages. During the design stage, it provides pipe data for the process engineering team, providing relevant data to calculate the different process variables. After the delivery of the project to the client, drawings and the 3D model are updated with the possible modifications made on site; and in consequence the P&IDs are also updated.

Process Engineers take advantage of the data transfer from PDMS to COMOS to be able to determine the pipes DN size and add it to the base templates, after which the designer is able to update the modified pipes according to the feedback from the Process Engineer.

Process Engineers require an accurate pipe isometric to be able to calculate with precision the pipe design parameters, which cannot be provided before design data and process data is shared and updated from each side several times. However, process engineering is able to provide pipe parameters with sufficient accuracy with a more simple set of data, which designers are able to provide already in the earlier stages of the project.

8.1 Data storage in COMOS

COMOS Pipe elements have attributes for the data received from PDMS. Two new tools are required in PDMS/E3D before the automatic data transfer can be developed efficiently. The first tool counts the number of elbows in the main branch of the pipe element. The second tool calculates the height difference between the component on the highest and the component on the lowest position. Position related attributes, and pipe maximum height should be given in relation to an agreed point on the model, for example a Grid Datum element of the mill area is a good option.

In-plant positioning	In-plant coordinates
Global object coordinates <input type="text"/>	
x position <input type="text"/>	x coordinates <input type="text"/>
y position <input type="text"/>	y coordinates <input type="text"/>
z position <input type="text"/>	Level <input type="text"/>
Ground level <input type="text"/> m <input type="button" value="v"/>	

Image 15: COMOS Instrument Attributes for PDMS - COMOS data transfer

8.2 PDMS export

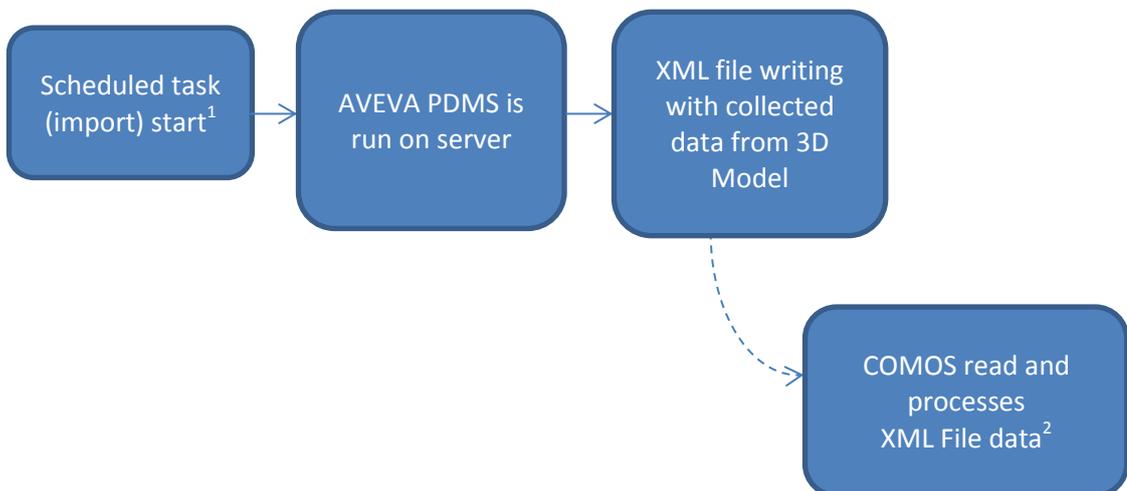


Diagram 4: Data Import from AVEVA PDMS

1. Windows scheduler task
2. Predefined system folder that is read by COMOS

The PDMS Export tool XML file content is available in Appendix 3.

The PDMS/E3D data export PML code, collects the data to be transferred from the 3D model in a XML file which is stored in a shared folder on the company web drives. Data collection and file writing is scheduled to be run once a day in the silent office hours of the day. The PML code creates also a PNG image collection of the existing equipment and pipe branches. The images are screen captures of the isometric view of the

equipment or branch in question. Isometric images are stored in the predefined folder with the COMOSUID value of the branch or equipment as name.

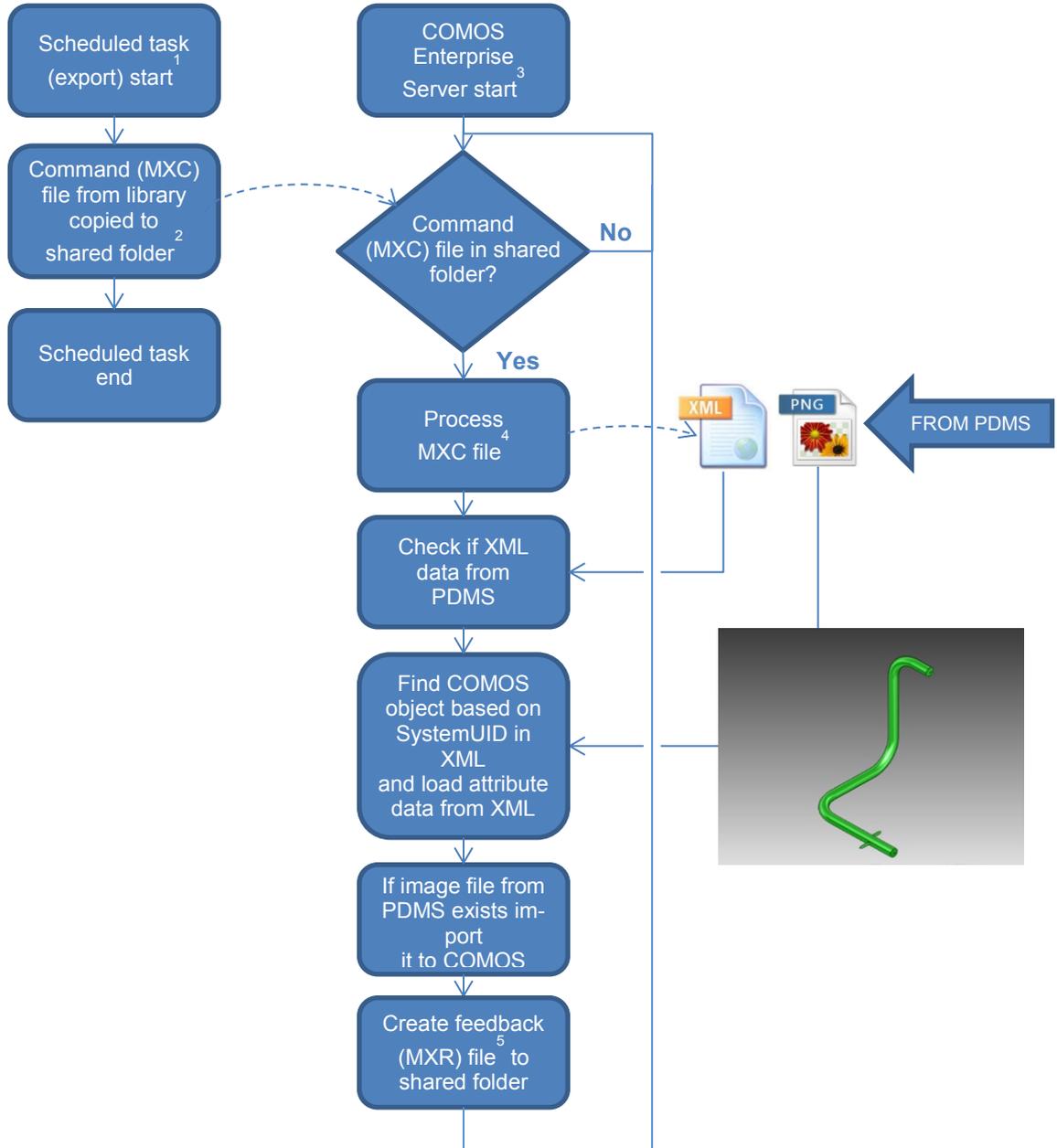


Diagram 5: COMOS Import Tool diagram

1. Windows scheduler task
2. Predefined system folder that is polled by COMOS enterprise server for command files

3. Windows service starting automatically on server start up
4. Command file contains parameters of executed action such as COMOS project, action type, target file path etc.
5. Feedback file created automatically by system at the end of each task. It provides status and other details of the task execution

The PDMS/E3D export tool is a set of different tools. Some tools collect the data or calculate it from different attributes and/or pseudo-attributes of the different elements. Other tools translate the data attributes to a COMOS readable format. The main tool creates the XML file and exports it to a predefined shared folder.

8.3 COMOS import

The COMOS import tool tracks in a loop if a new XML file has been created in the predefined shared folder. Once the file is located, the importer tool updates the data to the COMOS elements based on the COMOSUID attribute value. The import tool also locates the PNG file based on the COMOSUID value, if the PNG file is located. Once it is located, the COMOS importer links the image to the COMOS element.

8.4 Transferred PDMS data

The transferred data from PDMS to COMOS should include the pipe branch's centre-line length, elbow quantity and the maximum elevation and instrument's position. Some data is already stored in the PDMS database as attribute of the design element, this being the case for the pipe centreline length (stored in the branch element) and the instrument position; for the other two variables it is necessary to build additional tools to collect the data.

Pipe data required to be exported from PDMS is only available in the Branch level, and it is imported to the equivalent element in COMOS to the Pipe element. This requires the definition of the main branch, either in the export tool or in the model.

To have the data transferred under the elements in COMOS which they belong to, COMOS UID attribute is to be used as element identifiers, in the same way as in the data transfer from COMOS to PDMS.

length	13.129	m	▼
weight / length		kg/m	▼
elongation limit at normal conditior		N/mm ²	▼
pipe selection		...	✕ ↻
customer pipe class			

Image 16: Technical data attributes imported from PDMS to COMOS

8.4.1 Pipe centreline length

Pipe centreline length is not possible to collect directly from the pipe elements as the specific information is stored on the branch elements. Most pipes have several minor branches for drains and sampling, which are not relevant for the process calculations.

Most pipes are designed in PDMS using only one branch under each pipe element. In the case of a pipe containing more than one branch, the main branch is in most of the cases designed first, thus, allowing a direct identification through the hierarchy tree (*Pipe_ID-MEDIA-DN-SPEC/B1*). This may not be always the case, as this hierarchy structure is not instructed or checked at any point.

8.4.2 Pipe Maximum Elevation

Pipe maximum elevation can be calculated with different methods. One possible method, and simple to implement in a tool, would be through the evaluation of the pipe volume. The provided value represents the highest point of the outer plane of the pipe, when the required data value would be the highest point of the centreline of the pipe to be evaluated.

8.4.3 Pipe Elbow counter

Elbows have a relevant influence in the overall flow resistance in a pipe, when considering MC pulp as media. With liquid media (i.e. water) the flow loss is approximately in the rate of few meters per hundred meters of pipe length, when considering MC pulp as media, the flow loss can be as high as meters per meter of pipe length (Appendices 6 and 7).

8.4.4 Isometric View

Pipe isometric drawings offer the most accurate data for the process engineers to calculate the required pipe data. A pipe isometric export tool is already in use in the PDMS tool set. A simple command in the PDMS export tool to trigger the isometric export tool would provide process engineers the pipe isometric through the data transfer tool.

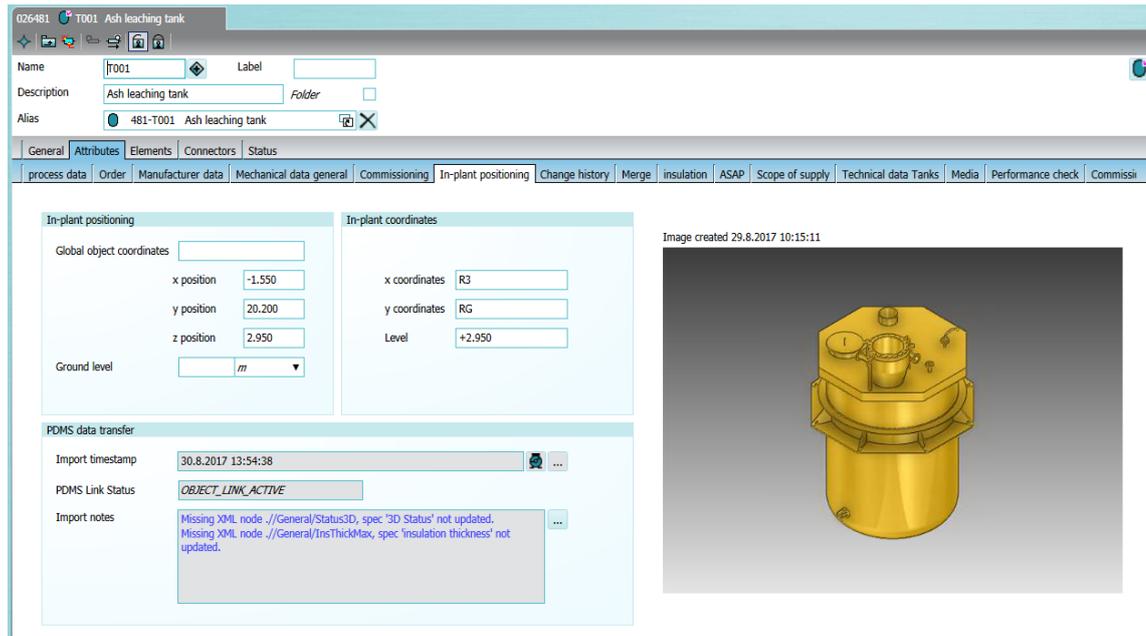


Image 17: In Plant positioning attributes in COMOS with ISO-metric view of equipment element in PDMS Database

In the current version of the tool, a screenshot is taken of the pipe or the equipment and exported to PDMS (images 17 and 18). Screenshots of Equipment 3D models in PDMS are also exported to COMOS. Below you can see images of the screenshots taken by the PDMS – COMOS Data transfer tool for Equipment and Pipes.

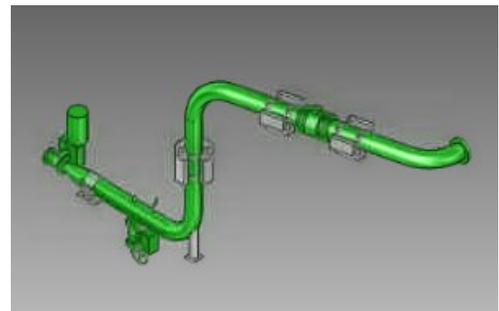


Image 18: Isometric view of a pipe exported from PDMS to COMOS

8.4.5 Instrument Position

Instrument position is already an attribute of the instrument element, however, the exported position is in reference to the World origin, which is very rarely the same as the agreed element for the origin. In the same model there may be several reference elements, each one for the instruments in different areas of the mode; i.e. Cooking instruments can be positioned in reference of the Cooking Grid Datum element, and the Fiberline instruments positioned in reference of the Fiberline Grid Datum.

8.4.6 Pipe 3D Status

The 3D Status attribute defines the design status:

- Created
- Preliminary
- For construction
- Final
- As-Built

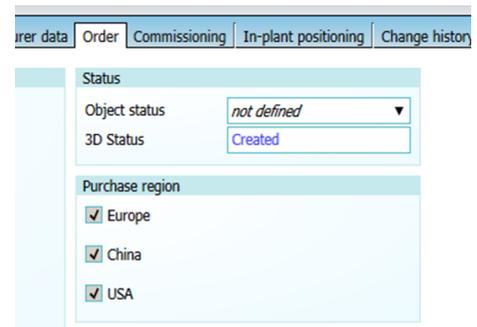


Image 19: Object status in COMOS

“Preliminary” status is used on the early stages of a project, when the design may be changed remarkably. “For Construction” status is used after the detailed design will not cause any remarkable changes in the global aspect. “Final” status is used after the design has been approved by the client, any unclear details are to be marked as “on Hold”; after the final status drawings had been delivered to the client. “As-Built” is used after the possible changes to the design made by the constructors on site, are updated to the design.

The most relevant 3D Status values, from the PDMS/E3D – COMOS data transfer tool aspect, are “*Preliminary*”, “*For construction*” and “*Final*”, as they define the quality and quantity of the future possible changes in the design.

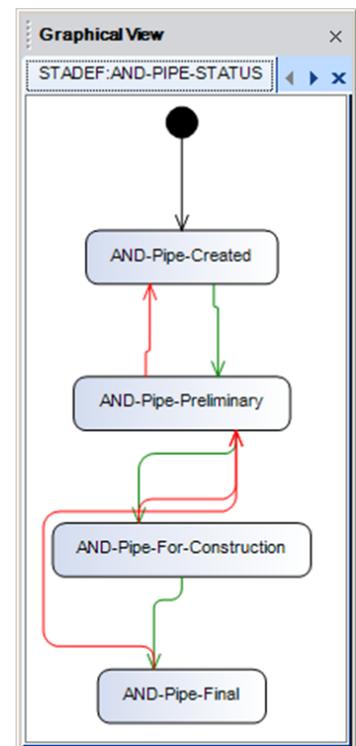


Image 20: Pipe 3D status values in PDMS/E3D

The 3D status value aims to inform the user on the stage on the pipe design:

- Created: the pipe has been created in the model and components are being placed/removed in the model to define the pipe routing
- Preliminary: the pipe may be changed in a regular basis and the specification may also be changed
- For Construction: pipe design is ready, but is to be expected to see modifications on the pipe
- Final: the design of the pipe is ready and the pipe is already in production, in case of unclear details, these are to be marked as on "Hold", till further notice.

9 Results and conclusions

9.1 Results

During the writing process of this Thesis, the PDMS/E3D – COMOS data transfer tool was developed and implemented in the starting projects. The transferred data results were collected from the different projects where the tool was implemented in different regions worldwide (Europe, America and Asia), several of them involving the Fiberline process.

Data transfer tools between COMOS and AVEVA PDMS/E3D software were developed to offer a bidirectional data transfer between the software's. First data is transferred from COMOS to PDMS/E3D where data is used to route the pipes. When equipment and instrumentations are positioned in the model and the pipe routes are ready, the 3D model pipe, instrument and equipment data is transferred back to COMOS, where the process engineer evaluates if the pipe data from COMOS requires modifications.

Data transfer between the software solutions is carried out on a daily basis during the silent office hours to avoid data transfer of work in progress. Data is transferred by an administrator bot user in the company server that triggers the data transfer tools to write XML files with the data collected from the corresponding databases. The admin bot user also triggers the tools used to read the XML files and write the data in the corresponding databases of both software solutions (COMOS and PDMS/E3D). Data transfer removes the need to allocate work hours for manual data transfer, which requires excessive amount of the available work hours when allocated and removes almost completely possible mistakes in data transfer.

Data transfer tools have a window for the development in the near future, to enable data transfer of more variables and take advantage of the data transfer for more tools, especially on the design field. The development of certain tool features, require the integration of other tools used by the company to the data transfer network. Some of these tools are planned to be synchronized in the next steps of the tool development. This is the case for the data transfer from PDMS to Navisworks, a 3D model view and analysis software used to share the model with the client.

9.2 Future development

The tool is in continuous development and there are some aspects being considered. All future changes are to be implemented in the right pace. The implementation and publication of too many features and tools refrains users from implementing them and developers are not able to analyse the benefits and possible improvements of each tool.

The XML file based data transfer is a remnant of the first stages of the development to allow administrators review and analyse the content of the transferred data, as data file based transfer is not required anymore. On the short term, data transfer could be reduced to a single XML file containing both component data levels. On the Long term development, both software solutions allow the data transfer without data files, allowing the possibility to set the data transfer on a continuous cycle. This continuous cycle data transfer option is to be studied further, as it may not be relevant considering the time designers need to complete the pipe routing during the early stages of the project.

A dialog window, to notify the user of all issues during the latest data transfer, also in the case when the data transfer was completed without any issues, would be of benefit for the user, as well as for the future development and maintenance of the data transfer tools. Dialog windows are to be considered in both software solutions, for the export and import process.

Data transfer timing may change to provide more than one data transfer during the working day. A continuous data transfer has been considered, as in many cases the design office and the project site are in different time zones. When providing multiple data transfers during the day, it may become crucial to differentiate the system, which data is reliable and accurate enough to be transferred, to avoid the transfer of wrong data caused by the human error.

The development of the tools is also to be revised to respond to the needs and requirements of all different divisions (Woodyard, Evaporation Plant, White Liquor Plant, Recovery and Power Boiler). In the case of Power, Recovery Boiler and Evaporation Plant require ducting elements in the design. Ducting data requires the cross section profile definition. Ducting elements profiles can vary between circular to rectangular including the transition elements.

The equipment databases in PDMS/E3D are to be revised and nozzle naming updated to correspond with the naming structure in COMOS. Equipment accesses and sight glasses are modelled using nozzle elements under the Nozzle Sub-Equipment elements in PDMS/E3D. To ensure accurate data transfer, nozzles used in the design to other purposes than pipe connections may be moved to another Sub-Equipment element or designed using other primitives. This option requires the revision and update when needed, of the whole equipment library, and the rules of the different tools to create equipment.

9.2.1 COMOS to PDMS Data Transfer

There are certain features that require special attention in the COMOS – PDMS data transfer tool. The most relevant ones to be mentioned are attributes with empty value, removed elements and XML data files in the servers.

Attributes with empty value require attention to the differentiation considering the reason for the value to be empty; the most usual one being the case where a value has not been set at any time during the project, and the most relevant case where the value has been removed. Attribute content may be removed for several reasons, for example to indicate that the element is on hold awaiting a decision. It may also indicate that the previously existing data was not reliable or relevant for the specific project.

Different elements may also be removed during the design stage; in these cases, elements should be saved on both databases to allow the recognition of such elements and have them reviewed manually. This procedure requires a new attribute/hierarchy structure in the databases.

9.2.2 PDMS/E3D to COMOS Data transfer

Pipe data from PDMS/E3D exported to COMOS is collected mainly from an element under the Pipe element in the 3D model database, Branch. To ensure the correct data is transferred to COMOS, it is required to define the rules used by the tool to identify the branch used to collect the data to be transferred to COMOS. The current design order used by the model designers provides already a reliable rule to identify the

branch from where data is collected (branch 1). This requires the instruction of the designers to reserve the branch name *PIPE_ID/B1* to the main branch.

Pipe Isometric is a key document for process designers to evaluate if the pipe specifications require to be modified. At the moment, pipe isometric drawing export to COMOS require a tool in PDMS/E3D to allow the latter to communicate with another software solution used in the company for documentation and file sharing (internally and externally) ADMS, a database used to share different files between the different Andritz departments internally and the client/subcontractors. The Pipe isometric export tool would create the pipe isometric drawing, export it to ADMS and share with COMOS the file address in ADMS.

9.3 Conclusions

Data transfer tool from PDMS/E3D to COMOS was implemented successfully in new projects.. The developed data transfer tool transfers the relevant data for the process engineers, including:

- Pipe centre line length
- Pipe maximum elevation
- Pipe elbow counter
- Pipe isometric view
- Equipment isometric view
- Instrument 3D position
- Pipe 3D status.

The data transfer tool offers a seamless bidirectional data transfer between the process design software COMOS and the plant design software from AVEVA (PDMS and E3D). Data transfer offers the following benefits in various aspects:

- seamless change management, real-time access to data during design work
- limiting human errors by removing part of the manual data input work
- work process support, e.g. “Unlinked pipes”
- reduction of waiting time to access preliminary process data
- reduction of the required time for the design process in the project

From the AVEVA PDMS/E3D specific point of view, the data transfer tools offer benefits in aspects like automated generation of PDMS elements as part of the early project stage. While from the COMOS point of view, the benefits include:

- COMOS data attribute visualization in 3D models
- list creation directly from data available in COMOS
- COMOS data revision directly from the model

Data transfer tools (COMOS – PDMS/E3D and PDMS/E3D – COMOS) require constant follow-up and development. The data transfer tools are planned to be extended to other tools used in the Andritz – Client communication tools like ADMS and Navisworks, offering the possibility to transfer more data values between COMOS and AVEVA solutions.

Sources

1. <https://www.andritz.com/> → About Andritz → History. Andritz AG website (23.01.2017).
2. <https://www.andritz.com/> → Pulp & Paper → locations → Andritz Oy. Andritz AG website (6.9.2017).
3. Puumassa Valmistus II Osa 1, VIRKOLA Nils-Erik, 2 painos (1983) ISBN 95199117-3-1
4. A Modern ECF Fiberline for the Production of Bleached Eucalyptus Pulp, REYES Jorge, VALDEBENITO Fernando, POULIN Ted M. Ted M. http://www.eucalyptus.com.br/icep02/reyes_valdebenito_poulin.pdf
5. PSK 7306. PSK Standard database
6. COMOS PT 10.1 Basic. Andritz AG Internal documentation
7. Interview with MARKKANEN Jani, Team Leader, Engineering Tools Support, Andritz Oy (Helsinki, 2.3.2017)
8. Interview with LATTI Ismo, COMOS Administrator, Fiberline and White Liquor Plant Industries, Andritz Oy (Helsinki, 2.3.2017)
9. AVEVA Everything3D™ User bulletin published by AVEVA Ltd. 2015.
10. AVEVA Everything 3D Product brochure 2015
11. Interview with ROSENBLAD Pekka, Senior pipe design specialist, Pöyry Finland Oy (Kotka, 10.08.2017)
12. TM-1401 AVEVA Plant (12 Series) Programmable Macro Language (Basic) Rev 3.0. Andritz AG Internal documentation
13. Interview with VUORINEN Susanna, Process Engineering Manager, Andritz Oy (Kotka, 31.07.2017)
14. Interview with TIIHALA Antti, Engineering Tools Support, Andritz Oy (Kotka, 16.05.2017)
15. Interview with TIIILI Otso, Senior Development Engineer, COMOS, Recovery & Power Boiler Industries Andritz Oy (Helsinki, 31.08.2017)

16. EUR-Lex - 32014L0068. Directive 97/23/EC of the European Parliament and of the Council on the approximation of the laws of the Member States concerning pressure equipment. <http://eur-lex.europa.eu/eli/dir/1997/23/2013-01-01>

Appendix 1. COMOS – PDMS/E3D data import tool MXC file code

```

define function !!AndTagsImportXml(!fileName is STRING) is BOOLEAN

    if !!MODULE().appkey.Neq('tags') then
        return FALSE
    endif

    if !!AndObj(object FILE(!filename)).Exists().Not() then
        return FALSE
    endif

    !xml = ''
    !xmlFile = object AndXmlFile()
    !xmlFile.Open(!fileName)
    handle ANY
        return FALSE
    endhandle

do
    if !xml.Neq('E<>1<>Object<>') then
        !xml = !xmlFile.Read()
        break if !xml.Unset()
    endif

    if !xml.Eq('E<>1<>Object<>') then
        !name = ''
        !type = ''
        do
            !xml = !xmlFile.Read()
            break if !xml.Unset()
            break if !xml.Substring(1, 6).Neq('A<>2<>')
            if !xml.Substring(1, 12).Eq('A<>2<>Name<>') then
                !name = !xml.Split('<>').Last().Substring(2)
            endif
            if !xml.Substring(1, 15).Eq('A<>2<>ObjType<>') then
                !type = !xml.Split('<>').Last().Substring(2)
            endif
        enddo

        skip if !name.Empty() OR !type.Empty()

        !engite = object DBREF('/') + !name
        handle ANY
            !engite = object DBREF()
        endhandle

        if !engite.Badref() then
            !owner = object DBREF('/AD_TAGS/Import')
            handle ANY
                !xmlFile.Close()
                return FALSE
            endhandle
            !ce = !!CE
            !!CE = !owner

            if !type.Uppcase().Eq('NOZZLE') then
                !type = 'NOZZ'
            elseif !type.Uppcase().Eq('MOTOR') then
                !type = 'SUBE'
            endif
        endif
    endif

```

```

!command = 'NEW :AN' + !type + 'TAG'
$!command
handle NONE
    !engite = !!CE
elsehandle ANY
    !!CE = !ce
    skip
endhandle
!!CE = !ce
endif

if !engite.Badref() then
    !xmlFile.Close()
    return FALSE
endif

!engite.Name = '/' + !name
handle ANY
    !xmlFile.Close()
    return FALSE
endhandle

!doclinks = object ARRAY()
do
    !xml = !xmlFile.Read()
    break if !xml.Unset()

    do
        break if !xml.Neq('E<>2<>DocLink<>')
        !doclink = object ARRAY()
        do
            !xml = !xmlFile.Read()
            break if !xml.Unset()
            break if !xml.Substring(1, 6).Neq('A<>3<>')
            !doclink.Append(!xml.Trim())
        enddo
        if !doclink.Empty().Not() then
            !doclinks.Append(!doclink)
        endif
    enddo

    skip if !xml.Substring(1, 6).Eq('E<>2<>')
    break if !xml.Substring(1, 6).Neq('E<>3<>')

    !att = !xml.Split('<>')[3].Substring(2).Trim()
    !val = !xml.Split('<>')[4].Substring(2).Trim()

    if !att.Ucase().Eq('OWNER') then
        !engite.Attribute(':TagsOwner') = !val
        handle ANY
        endhandle
    endif

    !engite.Attribute(': ' + !att) = !val
    handle ANY
        !engite.Attribute(!att) = !val
        handle ANY
            !arr = object ARRAY()
            !arr.Append(object REAL(!val.Replace('@',
'').Replace('|', '|').Replace('|', '')))
            handle NONE
                !engite.Attribute(': ' + !att) = !arr
                handle ANY
                endhandle
                !engite.Attribute(!att) = !arr
                handle ANY
                endhandle
        endhandle
    endhandle
enddo

```

```
                elsehandle ANY
                endhandle
            endhandle
        endhandle
    enddo
    !!AndHandleDocLinks(!engite, !doclinks)
endif
enddo

!xmlFile.Close()
return TRUE

endfunction
```

Appendix 2. COMOS Exported XML file extract

Level 1 XML file

```

<ComosDataToPDMS_LVL1>
  <Object Level="1" ObjType="PIPE" Name="A3TNHMOW7B">
    <General>
      <ComosUID>A3TNHMOW7B</ComosUID>
      <Description>Pipe 3</Description>
      <ComosBaseOb>@1RI|@PP</ComosBaseOb>
      <ComosFullLabel>K1KR0881.11.0003</ComosFullLabel>
      <ComosCustomerTag>0003-WCW</ComosCustomerTag>
      <ProcessArea>088</ProcessArea>
      <UploadDate>23.12.2016</UploadDate>
      <FlowSheetNumber>A-123456</FlowSheetNumber>
      <ComosDescription>Pipe 3</ComosDescription>
      <ComosActive>Active</ComosActive>
    </General>
    <Detail>
      <DetailType>PIPE</DetailType>
      <PipeSpecification/>
      <PEDClass/>
      <Media>Cold water</Media>
      <DN/>
      <OperatingPressure/>
      <PS/>
      <OperatingTemperature/>
      <TS/>
      <FromDescription>Pump</FromDescription>
      <ToDescription>Reactor</ToDescription>
      <GroupNumber>171.002</GroupNumber>
      <TracingSpecification/>
      <InsulationSpecification/>
      <IsometricDrawingNumber/>
      <MechanicalErection/>
      <Team/>
    </Detail>
  </Object>
</ComosDataToPDMS_LVL1>

```

Level 2 XML file

```

<ComosDataToPDMS_LVL2>
  <Object Name="A3R8N66UG9" ObjType="MOTOR" Level="2">
    <General>
      <ComosUID></A3R8N66UG9</ComosUID>
      <Description>3-phase motor</Description>
      <ComosBaseOb>@1EA|M|M|D|B</ComosBaseOb>
      <ComosFullLabel>041E917-M01</ComosFullLabel>
      <ComosCustomerTag>041E917-M01</ComosCustomerTag>
      <ProcessArea>088</ProcessArea>
      <UploadDate>NA</UploadDate>
      <Date>NA</Date>
      <FlowSheetNumber>0003017_A0041001</FlowSheetNumber>
      <ComosDescription>3-phase motor</ComosDescription>
      <Owner>A3R8N9PMG9</Owner>
      <ComosActive>Active</ComosActive>
    </General>
    <Detail>
      <DetailType>MOTOR</DetailType>
      <Manufacturer></Manufacturer>
      <ManufacturerType></ManufacturerType>
      <Supplier></Supplier>
      <Medium></Medium>
      <ValveDN></ValveDN>
      <ValveType></ValveType>
      <ScopeOfSupply>Remove from motor list</ScopeOfSupply>
    </Detail>
    <Custom>
    </Object>
  </ComosDataToPDMS_LVL2>

```

Appendix 3. PDMS/E3D – Comos data export tool MXC file Code

```

define function !!AndTagsExportXml() is BOOLEAN

    if !!MODULE().appkey.Neq('tags') then
        return FALSE
    endif

    !import = object DBREF('/AD_TAGS/Import')
    handle ANY
        return FALSE
    endhandle

    if !!AndObj(object FILE(!import.Desc)).Exists().Not() then
        return FALSE
    endif

    !exportPath = !import.Desc.Replace('TagsExport',
'TagsImport').Replace('_LVL1.xml', '.xml')
    if !exportPath.Eq(!import.Desc) then
        return FALSE
    endif

    !rawTypes = !!AndVar(|COLL ALL UDA WITH UPCASE ( UDNAME ) EQ
'COMOSUID'|).First().Elelist
    handle ANY
        return FALSE
    endhandle
    !types = object ARRAY()
    do !rawType values !rawTypes
        !dbType = !!AndObj(object ELE
ENTTYPE(!rawType)).DbType().First().UpCase()
        handle ANY
            !dbType = ''
        endhandle
        if !dbType.Eq('DESI') then
            !types.Append(!rawType)
        endif
    enddo
    if !types.Empty() then
        return TRUE
    endif

    !export = object FILE(!exportPath)
    !export.Open('OVERWRITE')
    !export.WriteRecord('<PDMSDataToComos>')

    !collection = object COLLECTION()
    !collection.Types(!types)
    !collection.Filter(object EXPRESSION(|TYPE NEQ 'ENGITE' AND NOT ( UNSET (
ENGLNK ) ))))
    !elements = !collection.Results()
    !collection.Filter(object EXPRESSION(|UNSET ( ENGLNK ) AND NOT ( UNSET (
:ComosUID ) ))))
    !elementsExtra = !collection.Results()
    do !e values !elementsExtra
        skip if !!AndExist('/' + !e.:ComosUID).Not()
        !testDuplicates = !!AndVar(|COLL ALL | + !e.Type + | WITH :ComosUID
EQ '|' + !e.:ComosUID + '|')
        skip if !testDuplicates.Size().Gt(1)
        !elements.Append(!e)
        $P AndTagsExportXml: exported without englkn $!<e.Name>
    enddo

```

```

enddo

if !elements.Empty() then
    !export.WriteRecord('      <!-- Linked objects are not found -->')
endif

!currentTeam = ''
!exportedUids = object ARRAY()
do !element values !elements
    !e = object DBREF(!element.Englnk.Refno.String())
    handle ANY
        !e = !element
        skip if !!AndExist(!e.:ComosUID).Not()
    endhandle

    if !exportedUids.FindFirst(!e.:ComosUID).Set() then
        !export.WriteRecord('      <!-- Please check ' +
!element.Refno.String() + ' (' + !e.:ComosUID + ') -->')
        skip
    else
        !exportedUids.Append(!e.:ComosUID)
    endif

    !export.WriteRecord('      <Object Name="' + !e.:ComosUID + '"
ObjType="' + !element.Type + '>')
    if !element.Englnk.Badref() then
        !export.WriteRecord('      <!-- Exported without active
englnk -->')
    endif
    !export.WriteRecord('      <General>')

    !export.WriteRecord('      <ComosCustomerTag>' +
!e.:ComosCustomerTag.Replace("|", |&quot;|).Replace(|'|,
|&apos;|).Replace(|<|, |&lt;|).Replace(|>|, |&gt;|).Replace(|&|, |&amp;|) +
'</ComosCustomerTag>')
    handle ANY
        endhandle
    !export.WriteRecord('      <ComosDescription>' +
!e.:ComosDescription.Replace("|", |&quot;|).Replace(|'|,
|&apos;|).Replace(|<|, |&lt;|).Replace(|>|, |&gt;|).Replace(|&|, |&amp;|) +
'</ComosDescription>')
    handle ANY
        endhandle

    if !currentTeam.Neq(!element.Dbname.Split('/').First()) then
        !newTeam = !element.Dbname.Split('/').First()
        !collection = object COLLECTION()
        !collection.Type('SCTN')
        !collection.Filter(object EXPRESSION('GTYPE EQ |XGRD| AND PART
( DBNAME, 1, | / | ) EQ |' + !newTeam + '|'))
        !xgrds = !collection.Results()
        !collection.Filter(object EXPRESSION('GTYPE EQ |YGRD| AND PART
( DBNAME, 1, | / | ) EQ |' + !newTeam + '|'))
        !ygrds = !collection.Results()
        !currentTeam = !newTeam
    endif

    !positionTest = !element.Position
    handle NONE
        !export.WriteRecord('      <PositionX>' +
!element.Position.Wrt(!element.Evaluate(object
BLOCK('WORL'))).East.ConvertUnits('M').Value().String('D3') + '</PositionX>')
        !export.WriteRecord('      <PositionY>' +
!element.Position.Wrt(!element.Evaluate(object
BLOCK('WORL'))).North.ConvertUnits('M').Value().String('D3') + '</PositionY>')
    endhandle
enddo

```

```

!export.WriteRecord('                <PositionZ>' +
!element.Position.Wrt(!element.Evaluate(object
BLOCK('WORL'))).Up.ConvertUnits('M').Value().String('D3') + '</PositionZ>')

!coords = !!AndGetGridLocs(!xgrds, !ygrds, !element,
!element.Position)
if !coords.Empty() then
!coords = !!AndGetGridLocs(!!AndVar('COLL ALL SCTN WITH
GTYPE EQ |XGRD|'), !!AndVar('COLL ALL SCTN WITH GTYPE EQ |YGRD|'), !element,
!element.Position)
!export.WriteRecord('                <!-- Grids are not
in the same team! -->')
endif

if !coords.Empty().Not() then
!export.WriteRecord('                <CoordX>' +
!coords[1] + '</CoordX>')
!export.WriteRecord('                <CoordY>' +
!coords[2] + '</CoordY>')
!export.WriteRecord('                <Level>' +
!coords[3] + '</Level>')
endif
elsehandle ANY
endhandle

if !element.Type.Eq('PIPE') then
!collection = object COLLECTION()
!collection.Type('BRAN')
!collection.Scope(!element)
!brans = !collection.Results()
!cllength = 0
do !b values !brans
!cllength = !cllength +
!b.Clength.ConvertUnits('M').Value()
handle ANY
skip
endhandle
enddo

!headtail = !!AndPipeHeadTail(!element, false)
if !headtail.Empty().Not() then
!headPos = !headtail[1]
!tailPos = !headtail[2]
!export.WriteRecord('                <PositionX>' +
!headPos.East.ConvertUnits('M').Value().String('D3') + '</PositionX>')
!export.WriteRecord('                <PositionY>' +
!headPos.North.ConvertUnits('M').Value().String('D3') + '</PositionY>')
!export.WriteRecord('                <PositionZ>' +
!headPos.Up.ConvertUnits('M').Value().String('D3') + '</PositionZ>')
!export.WriteRecord('                <PositionTailX>' +
!tailPos.East.ConvertUnits('M').Value().String('D3') + '</PositionTailX>')
!export.WriteRecord('                <PositionTailY>' +
!tailPos.North.ConvertUnits('M').Value().String('D3') + '</PositionTailY>')
!export.WriteRecord('                <PositionTailZ>' +
!tailPos.Up.ConvertUnits('M').Value().String('D3') + '</PositionTailZ>')

!coords = !!AndGetGridLocs(!xgrds, !ygrds, !element,
!headPos)

if !coords.Empty() then
!coords = !!AndGetGridLocs(!!AndVar('COLL ALL SCTN
WITH GTYPE EQ |XGRD|'), !!AndVar('COLL ALL SCTN WITH GTYPE EQ |YGRD|'),
!element, !headPos)
!export.WriteRecord('                <!--
Grids are not in the same team! -->')
endif
if !coords.Empty().Not() then

```

```

                                !export.WriteRecord('
<CoordX>' + !coords[1] + '</CoordX>')
                                !export.WriteRecord('
<CoordY>' + !coords[2] + '</CoordY>')
                                !export.WriteRecord('
<Level>' + !coords[3] + '</Level>')
                                endif
                                !coords = !!AndGetGridLocs(!xgrds, !ygrds, !element,
!tailPos)
                                if !coords.Empty() then
                                    !coords = !!AndGetGridLocs(!!AndVar('COLL ALL SCTN
WITH GTYPE EQ |XGRD|'), !!AndVar('COLL ALL SCTN WITH GTYPE EQ |YGRD|'),
!element, !tailPos)
                                endif
                                if !coords.Empty().Not() then
                                    !export.WriteRecord('
<CoordTailX>' + !coords[1] + '</CoordTailX>')
                                    !export.WriteRecord('
<CoordTailY>' + !coords[2] + '</CoordTailY>')
                                    !export.WriteRecord('
<LevelTail>' + !coords[3] + '</LevelTail>')
                                endif
                                else
                                    !export.WriteRecord('
<PositionX>NA</PositionX>')
                                    !export.WriteRecord('
<PositionY>NA</PositionY>')
                                    !export.WriteRecord('
<PositionZ>NA</PositionZ>')
                                    !export.WriteRecord('
<PositionTailX>NA</PositionTailX>')
                                    !export.WriteRecord('
<PositionTailY>NA</PositionTailY>')
                                    !export.WriteRecord('
<PositionTailZ>NA</PositionTailZ>')
                                    !export.WriteRecord('
<CoordX>NA</CoordX>')
                                    !export.WriteRecord('
<CoordY>NA</CoordY>')
                                    !export.WriteRecord('
<Level>NA</Level>')
                                    !export.WriteRecord('
<CoordTailX>NA</CoordTailX>')
                                    !export.WriteRecord('
<CoordTailY>NA</CoordTailY>')
                                    !export.WriteRecord('
<LevelTail>NA</LevelTail>')
                                endif
                                if !clength.Gt(0) then
                                    !export.WriteRecord('
!clength.String('D3') + '</Length>')
                                else
                                    !export.WriteRecord('
<Length>0.000</Length>')
                                endif
                                !collection = object COLLECTION()
                                !collection.Types(SPLIT('ELBO BEND'))
                                !collection.Scope(!element)
                                !bends = !collection.Results()
                                if !bends.Empty().Not() then
                                    !export.WriteRecord('
!bends.Size().String() + '</BendCount>')
                                else
                                    !export.WriteRecord('
<BendCount>0</BendCount>')
                                endif
                                !export.WriteRecord('
<Length>' +
<BendCount>' +

```

```

endif

if !!AndGetInsuThick(!element).Set() then
    !export.WriteRecord('                <InsThickMax>' +
!!AndGetInsuThick(!element).ConvertUnits('MM').Value().String('D0') +
'</InsThickMax>')
endif
endif

if !element.ISNAMED then
    !export.WriteRecord('                <Name3D>' + !element.NAMN
+ '</Name3D>')
endif

!status = object ARRAY()
!status.Append(object DBREF('/AND-' + !element.Type + '-STATUS'))
handle NONE
    !export.WriteRecord('                <Status3D>' +
!element.Attribute('STVDSC', !status).String() + '</Status3D>')
    handle ANY
        !export.WriteRecord('
<Status3D></Status3D>')
    endhandle
elsehandle ANY
    !export.WriteRecord('                <Status3D></Status3D>')
endhandle

!doclinks = !!AndVar('COLL ALL :ANDOCTAG WITH OWNER EQ ' +
!e.Refno.String() + ' FOR ' + !e.Refno.String())
if !doclinks.Empty().Not() then
    !export.WriteRecord('                <DocLinkCount>' +
!doclinks.Size().String() + '</DocLinkCount>')
else
    !export.WriteRecord('
<DocLinkCount>0</DocLinkCount>')
endif

!export.WriteRecord('                </General>')
!export.WriteRecord('                </Object>')
enddo
!export.WriteRecord('</PDMSDataToComos>')
!export.Close()

return TRUE

endfunction

```

Appendix 4. PDMS/E3D exported XML file extract

PML Extract for Pipe Element

```
<PDMSDataToComos>
  <Object ObjType="PIPE" Name="A3RY73TMK5">
    <General>
      <ComosCustomerTag>10HAD23BR302-WBO-150 SEC I</ComosCustomerTag>
      <ComosDescription>2nd pass RW supply pipe</ComosDescription>
      <Length>1.188</Length>
      <BendCount>1</BendCount>
      <InsThickMax>140</InsThickMax>
      <Status3D> Preliminary </Status3D>
    </General>
  </Object>
</PDMSDataToComos>
```

PML Code extract for Equipment element

```
<PDMSDataToComos>
  <Object Name="A3PQ0EQ5YU" ObjType="EQUI">
    <General>
      <ComosCustomerTag>691PU0904</ComosCustomerTag>
      <ComosDescription>Feedwater Pump 4</ComosDescription>
      <PositionX>-14.700</PositionX>
      <PositionY>22.390</PositionY>
      <PositionZ>38.730</PositionZ>
      <CoordX>43</CoordX>
      <CoordY>PB</CoordY>
      <Level>+0.730</Level>
      <Name3D>691PU0904</Name3D>
    </General>
  </Object>
</PDMSDataToComos>
```

Appendix 5. PED Classification charts, for pipes

For gases, liquefied gases, gases dissolved under pressure, vapours and those liquids whose vapour pressure at the maximum allowable temperature is greater than 0,5 bar above normal atmospheric pressure (1 013 mbar); Chart 1 and 2:

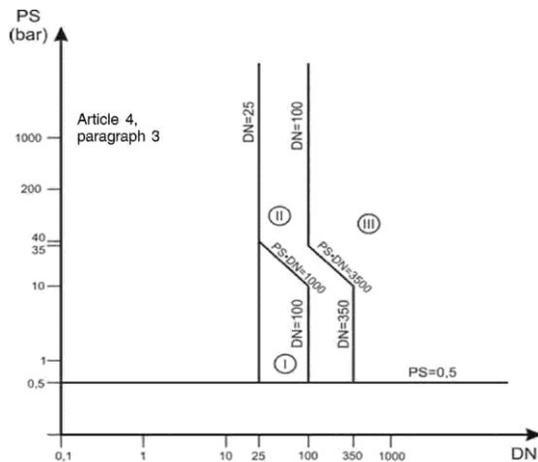


Chart 2: Piping for unstable gaseous fluids with a DN greater than 25

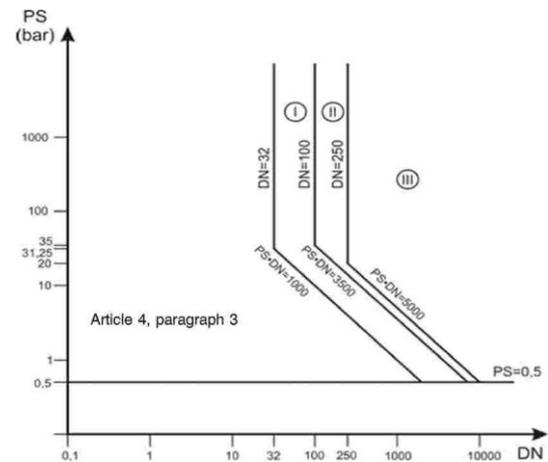


Chart 2: Piping for stable gaseous fluids with a DN greater than 32 and a product of PS and DN greater than 1 000 bar

For liquids having a vapour pressure at the maximum allowable temperature of not more than 0,5 bar above normal atmospheric pressure (1 013 mbar): Chart 3 and 4:

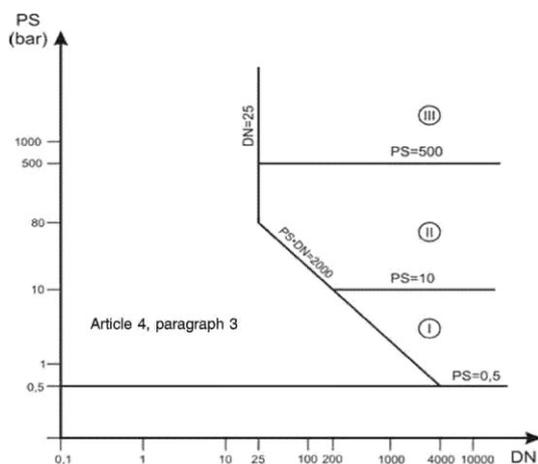


Chart 4: Piping for unstable liquid fluids with a DN greater than 25 and a product of PS and DN greater than 2 000 bar

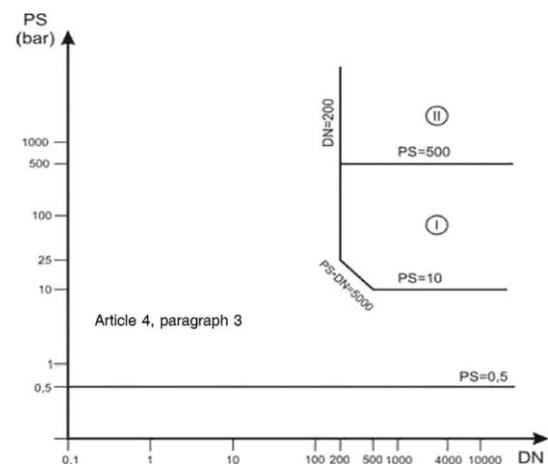
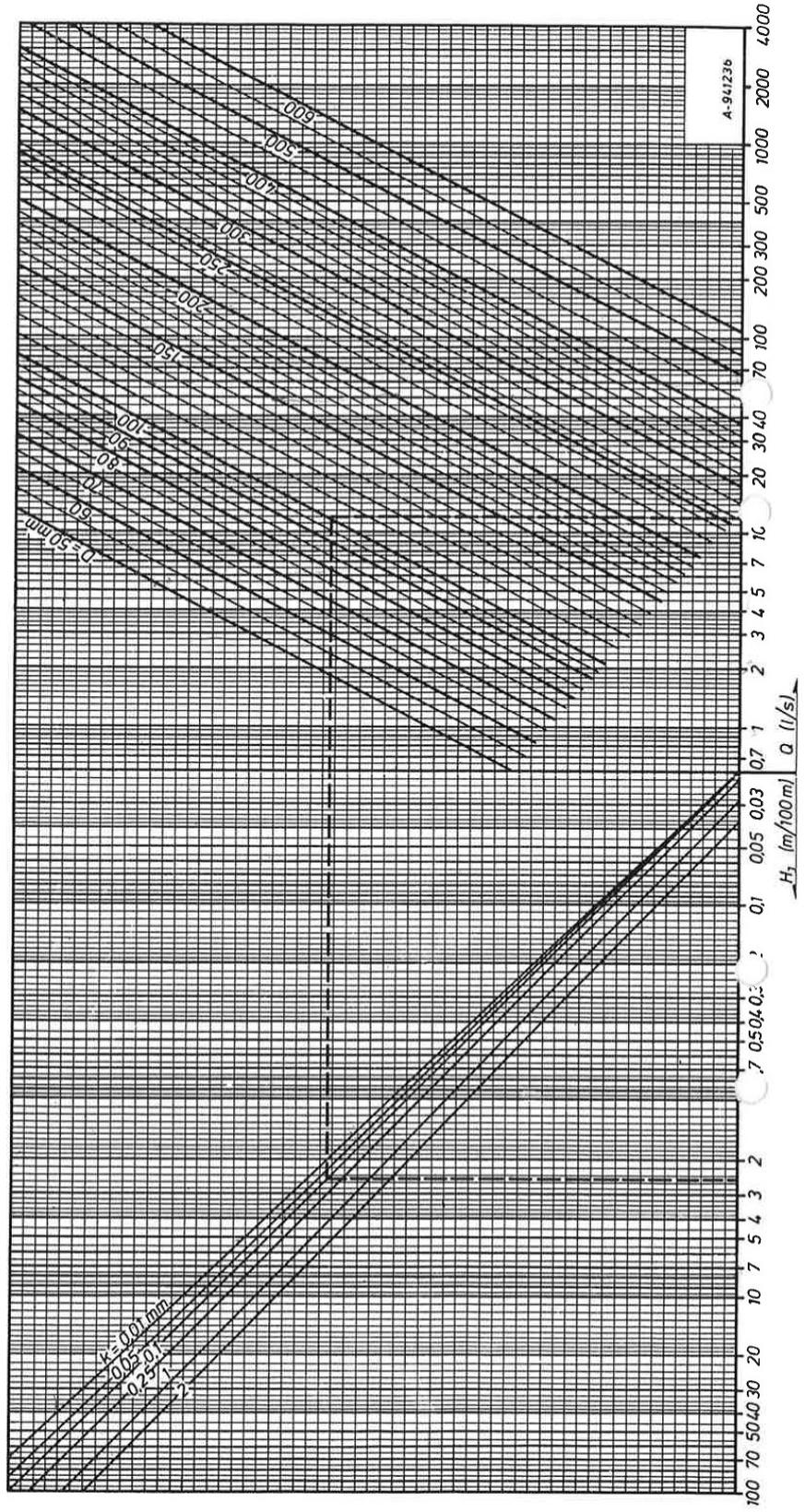


Chart 4: Piping for stable liquid fluids with a PS greater than 10 bar, a DN greater than 200 and a product of PS and DN greater than 5 000 bar

Appendix 6. Flow resistance monogram for clean, 20 °C water

- Q = Flow l/s
- D = Pipe inner diameter
- k = Surface roughness
- H_f = Flow resistance m/100 m



Appendix 7. Local flow resistance monogram for MC pulp media

Q = Flow l/s
 D = Pipe inner diameter
 v = Flow speed inside the pipe
 ζ = Local flow resistance multiplier
 k = Surface roughness
 HJ = Flow resistance m

Local flow resistance guide values

- | | |
|-------------------------------|-----------------------|
| | ζ |
| - Long Elbow 90°, R/D = 1,5 | 0,3 |
| - Flow out at the end of tube | 1,0 (without reducer) |
| - non-return valve | 1,0 - 2,5 |
| - Slide valve | 0,2 |
| - Slanted connection valve | 1 - 2 |
| - Ball valve | 5 - 6 |

