

KYMENLAAKSON AMMATTIKORKEAKOULU

Energiatekniikan koulutusohjelma/Automaation suunnittelu

Ville Kerttula

PARAMETRISTEN MAKROJEN KÄYTTÖ LAITOSTEN PUTKISTOSUUNNITTELUSSA

Insinööritö 2014



## SISÄLLYSLUETTELO

TIIVISTELMÄ	I
ABSTRACT	II
ALKUSANAT	III
KESKEISET KÄSITTEET	IV
1 JOHDANTO	9
2 LAITOSSUUNNITTELUN PERUSTEITA	14
2.1 Laitossuunnittelussa käytettäviä työkaluohjelmia	16
2.2 Laitossuunnittelun dokumentaatio	17
2.3 Putkistosuunnitteluun perehdytys	21
3 SMARTPLANT 3D -LAITOSSUUNNITTELUOHJELMAN ESITTELY	23
4 PARAMETRISEN OHJELMOINNIN PERUSTEET	32
4.1 XML-kieli ohjelmoinnissa	33
4.2 Olio-ohjelmointikielen alkeet	35
4.3 Ohjelmoitaessa huomioitavat asiat putkistosuunnittelun yhteydessä	37
4.4 Valmiiden kokoonpanomakrojen käyttö	38
5 PARAMETRINEN OHJELMOINTI PLANT DESIGN SYSTEM -OHJELMASSA	41
5.1 PDS-laitossuunnitteluohjelma	41
5.2 Parametrinen ohjelmointi	42
5.3 PDS-kokoonpanon sijoittaminen malliin	48
6 PARAMETRISEN OHJELMOINNIN PERUSTEET SMARTPLANT 3D:SSÄ	50
6.1 Kokoonpanon nimeäminen	51
6.2 Kokoonpanon sijoittamismahdollisuuksien rajaaminen	52
6.3 Kokoonpanon mallikuvan luominen	53
6.4 Parametrien määrittely	54
6.5 Parametrien validointi	56
6.6 Putkilinjan ja -osuuden määrittäminen	60
6.7 Erilaisia tapoja putkiosuuksien ohjelmointiin	62
6.8 Sovitteen tekeminen	66
6.9 Komponenttien ja instrumenttien rakentaminen	67

6.10 If-Then-määrittely	71
6.10 Komponenttien ominaisuuksien spesifiointi	72
6.11 Geometriatyypimäärittelyn hyödyntäminen ja käyttö	75
6.12 Referenssipisteiden tallennus ja käyttö	77
6.13 Ohjelman kirjoitussäännöt	79
<b>7 OHJELMOINTI YKSITYISKOHTAISIN ESIMERKEIN</b>	<b>82</b>
7.1 Kokoonpanon luominen - Place Parametric Assembly	87
7.2 Ohjelman rakentaminen	89
<b>8 PUTKISTON OLIOIDEN TIETOKANTAOMINAISUUKSIIN VIITTAAMINEN</b>	<b>92</b>
8.1 Kokoonpanokansion luominen	92
8.2 Tietokantaan viittaaminen työkaluohjelmalla	94
8.3 Putkiston osaominaisuuksien määrittely ohjelmoinnissa	98
8.4 Tietokannasta löytyvät komponentit ja niiden tärkeimmät määritteet	102
<b>9 TOTEUTETUT OHJELMOIDUT MALLIT</b>	<b>104</b>
9.1 Esimerkkiohjelma	107
9.2 Havaittuja ongelmakohtia	113
<b>10 POHDINTA JA YHTEENVETO</b>	<b>114</b>
<b>11 LÄHDELUETTELO</b>	<b>116</b>
<b>12 KUVALUETTELO</b>	<b>119</b>
<b>LIITTEET</b>	<b>121</b>
Liite 1. Olio-ohjelmointi sanasto	121
Liite 2. PDS-assembly, ilmanpoisto	123
Liite 3. SP3D-assembly, ilmanpoisto	124
Liite 4. Vetyverkon virtauskaavio	126
Liite 5. Laitossuunnittelun kierto	127
Liite 6. Putkistolinja, isometri ja osaluettelo	128
Liite 7. Luotuja kokoonpanoja	130

# TIIVISTELMÄ

KYMENLAAKSON AMMATTIKORKEAKOULU

Energiatekniikka

KERTTULA, VILLE

PARAMETRISTEN MAKROJEN KÄYTTÖ  
LAITOSTEN PUTKISTOSUUNNITELUSSA

Insinööriä

120 sivua + 11 liitesivua

Työn ohjaajat

TkL Lassi Salminen

System Manager Jouko Halme

Toimeksiantaja

Neste Jacobs Oy

Maaliskuu 2014

Avainsanat

Parametri, PAL, SP3D, Putkistosuunnittelu, Makro

Tämä työ käsittelee laitossuunnittelun 3D-mallintamista ja erityisesti sen osittaista automatisoimista kokoonpanomakrojen avulla. Työn tavoitteena oli tuottaa riittävä määrä parametrisiä kokoonpanomalleja putkistosuunnittelun nopeuttamiseksi ja laatutason nostamiseksi.

Työn parametrisen ohjelmoinnin teoriapohja tuli pääosin Intergraphilta, tietokoneavusteisen laitossuunnittelun kehittämiseen erikoistuneelta yritykseltä. Yleisen ohjelmoinnin lähteinä työssä käytettiin muun muassa teoksia C++ ja olio-ohjelmointi sekä Designing XML Databases. Työssä käytettävät spesifikaatiot tulivat työn tilanneelta yritykseltä Neste Jacobs Oy:ltä.

Työlle asetetut tavoitteet saavutettiin suunnittelussa laajuudessa. Lopputuotteena oli joukko PAL-kielisiä ohjelmia (Parametric Assembly Language), jotka rakentavat automaattisesti putkiluokkia ja tyyppidokumentteja vastaavia putkiston osakokonaisuuksia. Parametristen kokoonpanojen käyttönotolla saavutetaan merkittäviä hyötyjä niin kuluvan ajan sekä saavutettavan laatutason suhteen.

Työn suurimpia haasteita olivat NJ:n spesifikaatioiden mukaisten kokoonpanojen tekeminen sekä SmartPlant 3D-mallinnussessiossa näennäisesti toimivien mallien muokkaaminen generointiohjelman (Isogen) tuottamiin isometripiirustuksiin oikein piirtyviksi.

Työn laajuudeksi muodostui kolmekymmentä kokoonpanoa. Määrä kattaa kuitenkin vain käytetyimmät putkistosuunnittelun osakokonaisuudet, ja kaikin puolin olisi luontevaa jatkaa kokoonpanokansion täydennystä, jopa useaan sataan kokoonpanoon. Tällöin parametristen makrojen käytön kokonaisyöty tulisi parhaiten esille.

## ABSTRACT

KYMENLAAKSON AMMATTIKORKEAKOULU

University of Applied Sciences

Energy Engineering

KERTTULA, VILLE	Smart 3D, Parametric Assembly Development
Bachelor's Thesis	120 pages + 11 pages of appendices
Supervisors	Lassi Salminen, Lic.Sc.(Tech) Jouko Halme, System Manager
Commissioned by	Neste Jacobs Oy
March 2014	
Keywords	Parameter, PAL, SP3D, Plant Design, Assembly

This work is related to the plant design 3D-modeling, with a particular focus on its partial automation with assembly macros. The purpose of the project was to produce a sufficient number of parametric assemblies for speeding up the piping designing and for improving its quality.

The theoretical framework of this projects concerning parametric programming came mainly from Intergraph, the corporation specialized in Computer-aided plant designing. The general programming sources used in this project were C++ and Object-Oriented Programming, as well as Designing XML -Databases. The specifications used in the thesis came from the client company Neste Jacobs Ltd.

The objectives of this thesis were achieved to the extent planned. The end product was a set of PAL-language programs (Parametric Assembly Language), which automatically build pipe class and type document congruent pipeline entities. The use of parametric assemblies gives the corporation significant savings in piping designing. Furthermore their usage improves the quality of the models.

The biggest challenges of the thesis were to create assemblies loyal to NJ's specifications and to get the models that appeared to be successful correctly finalized in the piping isometric (Isogen) drawings. The extent of the work covered approximately -thirty-assemblies.

The focus of the thesis, however, is limited to the most commonly used piping design sub-assemblies. Therefore it would be advisable to continue building the assembly folder, even up to several hundred assemblies. As this would enable expressing the parametric macros total benefits.

## ALKUSANAT

Tämä insinööri työ *Smart 3D, Parametric Assembly Development* on tehty Neste Jacobs Oy:n Kotkan toimipisteessä. Työnohjaajana toimi Jouko Halme, Neste Jacobsin Kotkan toimipisteen System Manager: CAD Applications. Työnohjauksesta taas vastasi Kymenlaakson ammattikorkeakoulun lehtori, tekniikan lisensiaatti Lassi Salminen. Ohjeistusta sain myös NJ:n kokeneilta putkistosuunnittelijoilta Esa Gustafssonilta ja Markku Hietavuorelta.

Isot kiitokset eritoten Joukolle ja Lassille, joilta löytyi tietoa, taitoa ja halua opastaa allekirjoittanutta aina tarpeen tullen.

Kotkassa

16.7.2014

*Ville Kerttala*

## KESKEISET KÄSITTEET

NJ	Neste Jacobs Oy, vaativien teknologia-, suunnittelu- ja projektinjohtopalveluiden ja ratkaisujen toimittaja.
NO	Neste Oil Oyj, puhtaan liikenteen polttoaineisiin keskittyvä jalostus- ja markkinointiyhtiö.
SP3D	SmartPlant 3D, pitkälle automatisoitu sääntöihin ja riippuvuuksiin perustuva laitossuunnitteluohjelma.
PDS	Plant Design System, laitossuunnitteluohjelma, joka tarvitsee Microstationia mallipohjaiseen, toimittaja Intergarph.
PDMS	Laitossuunnitteluohjelma, valmistaja Aveva.
Makro	Automatisoitu kokoonpano, jonka avulla sovellus saadaan tekemään tiettyjä tehtäviä käyttäjän puolesta.
Clippaus	Näkyväksi halutun alueen rajaaminen aputyökalulla.
Isometri	Putkiston isometriseen kuvantoon laadittu valmistuspiirustus.
NPD	Nominal Pipe Diameter, nimellinen putken halkaisija.
Laserkeilain	Laite, joka mittaa todellisilla mitoilla "pistepilvikuvan" keilattavasta näkymästä XYZ-koordinaatistoon. Tämän avulla voidaan luoda 3-ulotteinen malli keilatusta kappaleesta.
Assembly-malli	Putkikomponentteja sisältävä kokoonpano.
WSE	Workspace Explorer, SmartPlant 3D-ohjelman mallityökalu, joka näyttää

valitussa mallissa olevat elementit hierarkkiseen rakenteeseen järjestettynä.

PAL

Parametric Assembly Language, SmartPlant 3D:n käyttämä ohjelmointikieli.

ParametricAssembly

Muuttuviin ominaisuuksiin perustuva putkiston osien kokoonpano.

CAD

Computer-aided design, Tietokoneavusteinen suunnittelu.

As Built

Laitossuunnitteluprojektin vaihe, jolloin työn suunnitteluasiakirjat on korjattu, tarkastettu ja hyväksytty vastaamaan rakennettua laitosta.

Sketsaus

Sketch, on PDS:n toiminto, jolla ideoidaan putkilinjan sijainti vetämällä keskilinjanreitti, jonka ympärille putkielementit rakennetaan.



## 1 JOHDANTO

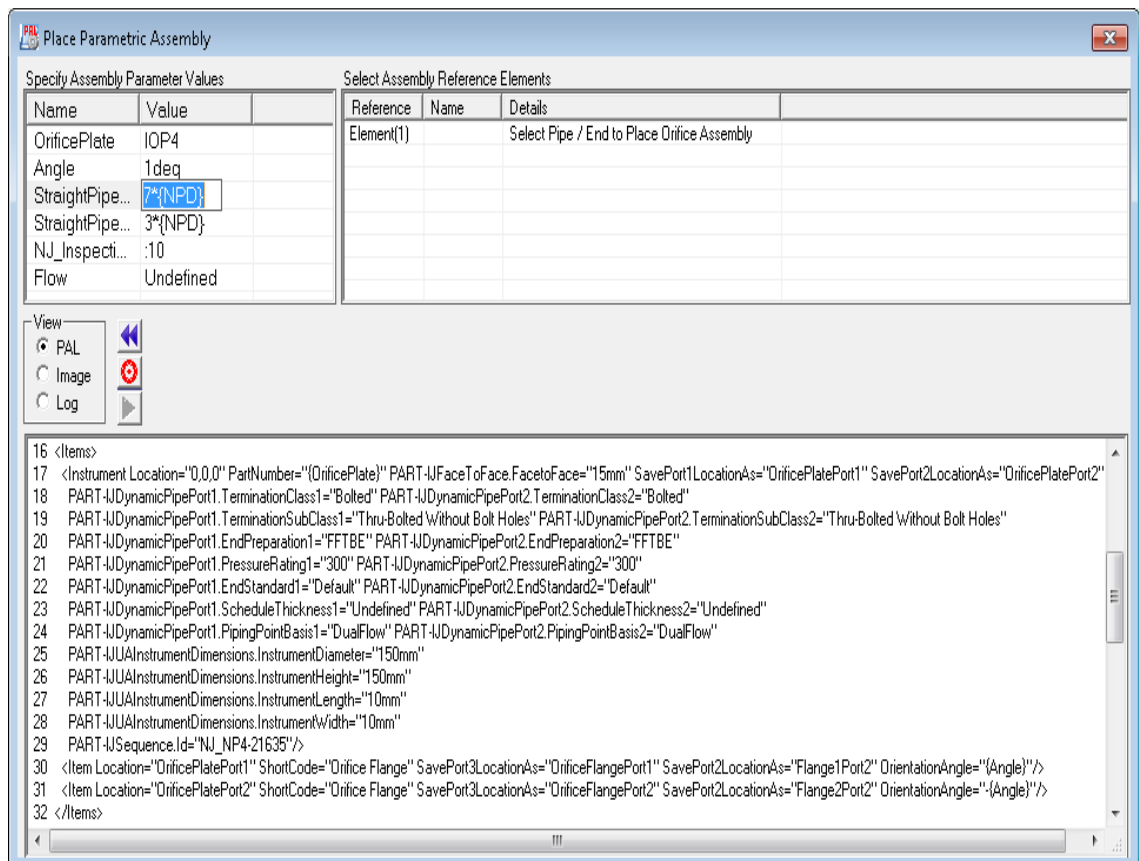
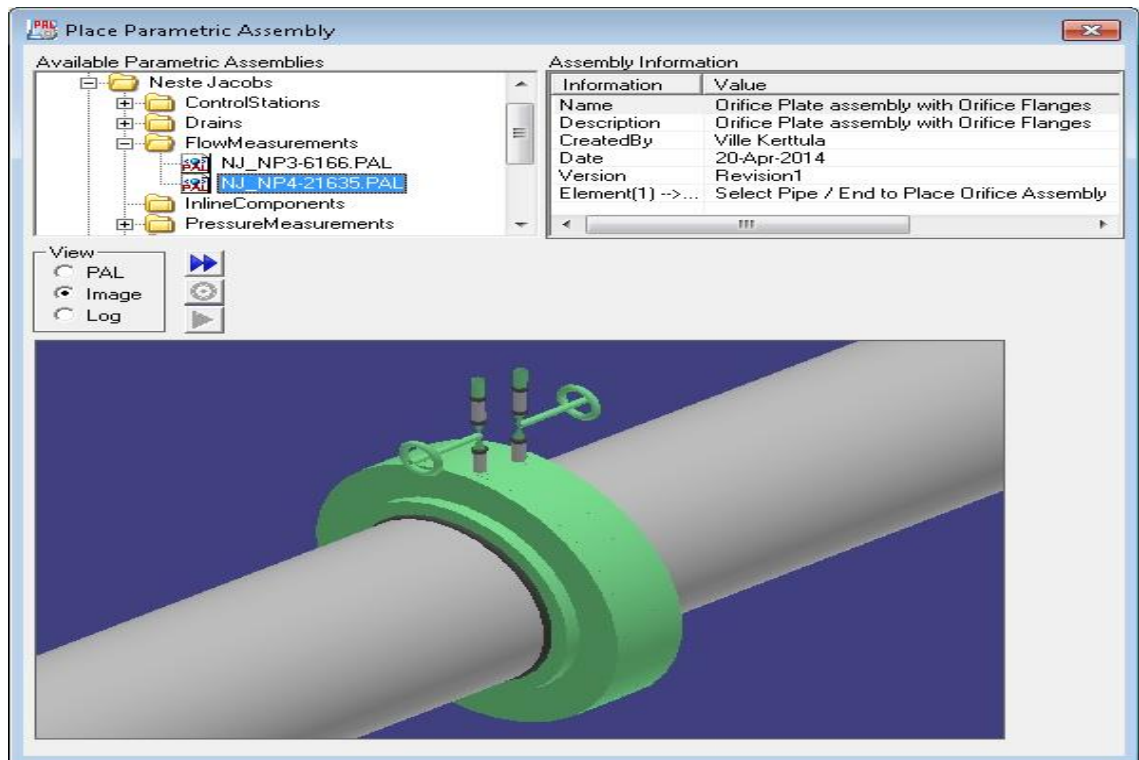
Tämä työ käsittelee laitossuunnittelun 3D-mallintamista ja erityisesti sen osittaista automatisoimista kokoonpanomakrojen avulla. Näitä ohjelmoituja kokoonpanoja putkistosuunnittelijat voivat sijoittaa 3D-pohjaiseen laitossuunnitteluohjelmaan nopeuttaakseen laitossuunnittelua ja ehkäistäkseen inhimillisen virheen syntymistä malliin.

Työn toimeksiantaja on Neste Jacobs Oy, joka on vaativien korkealaatuisten teknologia-, suunnittelu- ja projektipalvelujen tarjoaja. NJ toimii ensisijaisesti öljy-, kaasu-, petrokemian- ja kemianteollisuuden sekä biotekniikan alalla. NJ:llä on yli 50 vuoden kokemus teknologian kehittämisen ja investointihankkeiden toteutuksesta sekä ylläpito- ja suorituskyvyn parantamisesta Euroopassa, Pohjois- ja Etelä-Amerikassa, Aasiassa ja Lähi-idässä. Pohjoismaiden lisäksi NJ pyrkii laajentamaan toimintojaan kasvaville markkinoille Lähi-idässä ja Venäjällä. NJ:n palveluksessa on noin 1 000 ammattilaista maailmanlaajuisesti. (Neste Jacobs 2014)

Tämän työn kohdeyrityksen käyttämän SmartPlant 3D -laitossuunnittelujärjestelmän (SP3D) implementointi on aloitettu NJ:llä vuonna 2007, tavoitteena korvata 1980-luvulla käyttöön otettu Intergraphin PDS-järjestelmä. SP3D-järjestelmällä on tehty useampia esisuunnitteluprojekteja ja muutamia pienempiä neitseelliseen ympäristöön sijoitettuja projekteja. NJ:llä on nyt parhaillaan menossa isomman luokan SmartPlant 3D-putkistosuunnitteluprojekti kevytbensiinin isomerointi yksiköstä, Porvoon jalostamon (Neste Oilin) tuotantolinjalle nro 1.

Tämän työn tavoitteena on tuottaa riittävä määrä parametrisiä kokoonpanomalleja putkistosuunnittelun nopeuttamiseksi ja laatutason nostamiseksi. Lopputuotteena on PAL-kielisiä ohjelmia (Parametric Assembly Language), jotka rakentavat automaattisesti putkiluokkia ja tyyppidokumentteja vastaavia putkiston osakokonaisuuksia. Parametrinen kokoonpanojen käyttöönotolla saavutetaan merkittäviä säästöjä suunnittelutöiden ajassa sekä laadussa.

Seuraavassa on kuva kokoonpanojen sijoitusikkunoista.



Kuva 1. Kokoonpanon käyttö- ja määrittelyikkunat.

Yllä esitetyn kokoonpanon sijoitusikkunoissa näkyvät Items-osion koodirivit, jotka sanelevat muun muassa putkilinjan mukaan määräytyvän paineluokan, laippojen liitäntätavat ja paljon muita ominaisuuksia, joiden määrittelyyn suunnittelijalla menisi kallista aikaa. Tällaisia kokoonpanomakroja on tehty laitossuunnittelupuolella mm. laitossuunnitteluohjelmalle Plant Design System. Sen XML-pohjainen PAL-kieli on kirjoitusasultaan hyvin erilaista verrattuna SP3D:n PAL-kieleen. SP3D:n ohjelmointitoiminnot ovat myös pidemmälle automatisoituja, mikä vastaavasti tarkoittaa sitä, että ohjelmakoodia luotaessa on keskityttävä asioiden yhteisvaikutukseen.

Tässä työssä käytettävän laitossuunnitteluohjelma SP3D:n parhaita puolia on se, että siinä samaa mallia pystytään työstämään eri toimipisteistä käsin samaan aikaan, jopa usean henkilön voimin. SP3D:ssä laitoksen tietokannasta voidaan ottaa kerrallaan käsiteltäväksi sopiva osuus niin sanottujen filttäreiden avulla. SmartPlant 3D:n keskeiset osat ovat laite-, putkisto-, rakenne-, sähkö-, instrumentointi-, ilmastointi-, kannakointi- ja perustussuunnittelu. SP3D-ohjelmassa koko mallinnus pohjautuu tietokantaan mahdollistaen attribuuttien ”käyttämisen” oliopohjaisille elementeille kuten pumppu tai venttiili. SP3D:n vahvuuksia on sen pitkälle viety suunnitteluautomaatio. Elementtien ja elementtikokonaisuuksien suunnittelua on helpotettu niin, että ohjelmoimalla valmiita oliomallikokonaisuuksia, niitä voidaan myöhemmin lisätä kokoonpanoina suoraan mallikuvaan. Valmiisiin kokoonpanoihin voidaan määrittellä tietoja muun muassa materiaalista, tyypistä, lämpötilojen kestosta, putkistoissa virtaavasta aineesta sekä virtaussuunnasta. Näin ollen myös osakokonaisuuksien (putkisto, laitesuunnittelu) valmistuminen nopeutuu huomattavasti, kun jokaista säätöventtiiliryhmää ei tarvitse suunnitella alusta asti, vaan sen voi sijoittaa malliin koodaajan määrittelemien parametrein.

Työn kohdeyrityksen Neste Jacobsin projektien aikana on selkeästi todettu tarve kehittää automaattisia rutiineja, jotka kattaisivat ainakin käytössä olevat putkiston ja instrumentoinnin tyyppidokumentit muun muassa tyhjennykset, ilmaukset, näytteenotot, paine- ja lämpötilamittaukset, virtausmittaukset ja säätöventtiiliryhmät.

Tässä työssä keskitytäänkin automaattisten rutiinien eli kokoonpanojen tekoon C++-kielen tapaisella olio-ohjelmointikielellä. Työn tarkoituksena oli tuottaa riittävä määrä "kokoonpanomakroja" SP3D:hen, jotta laitossuunnitteluohjelman projektien aikainen putkistosuunnittelu täsmentyisi ja nopeutuisi.

Tämä työ "Smart 3D, Parametric Assembly Development" on urakaluonteinen toiminnallinen kehittämistehtävä, jota tullaan tarkastelemaan exploratorisen projektityön tapaan. Teknis-ammattillisesti tarkasteltuna työn pääpaino on projektin prosessissa ja akateemisesti taas parametriseen ohjelmoinnin käsittelyssä. Insinööriyön alussa tullaan perehtymään parametriseen ohjelmointiin ja työssä käytettävään ohjelmaan SP3D:hen. Alkeiden sisäistämisen jälkeen siirrytään parametriseen ohjelmoinnin koodaamiseen PAL-kielellä. Työn lopussa esitellään saatuja tuloksia.

Työ etenee lineaarisesti työn teoriapohjan esittelystä kokoonpanojen ohjelmointiin ja tuloksiin. Luvussa 2 käsitellään laitossuunnittelun perusteita. Perusteista syvennytään eritoten laitossuunnittelussa käytettäviin ohjelmiin, dokumentointiin sekä putkistosuunnittelun osa-alueisiin.

Luvussa 3 esitellään NJ:llä parametrisiä makroja hyödyntävä laitossuunnitteluohjelma SP3D.

Luvussa 4 perehdytään parametriseen ohjelmoinnin perusteisiin. Mistä ohjelmointi on saanut alkunsa? Minkä tyyppinen kieli olio-ohjelmointikieli on? Minkälaisia asioita tulee huomioida putkistosuunnittelun ohjelmoinnin yhteydessä. Lisäksi esitellään kokoonpanojen sijoitusikkuna.

Luvussa 5 esitellään laitossuunnitteluohjelma Plant Design System, jolle SP3D:stä poiketen on jo kehitetty laaja määrä kokoonpanomakroja.

Luvussa 6 keskitytään PAL-ohjelmoinnin perusteisiin sekä kirjoitussääntöihin.

Luvussa 7 käydään yhden toteutetun kokoonpanon tekemiseen liittyvät vaiheet läpi yksityiskohtaisesti alkaen tyyppikuvasta aina valmiiseen isometriin.

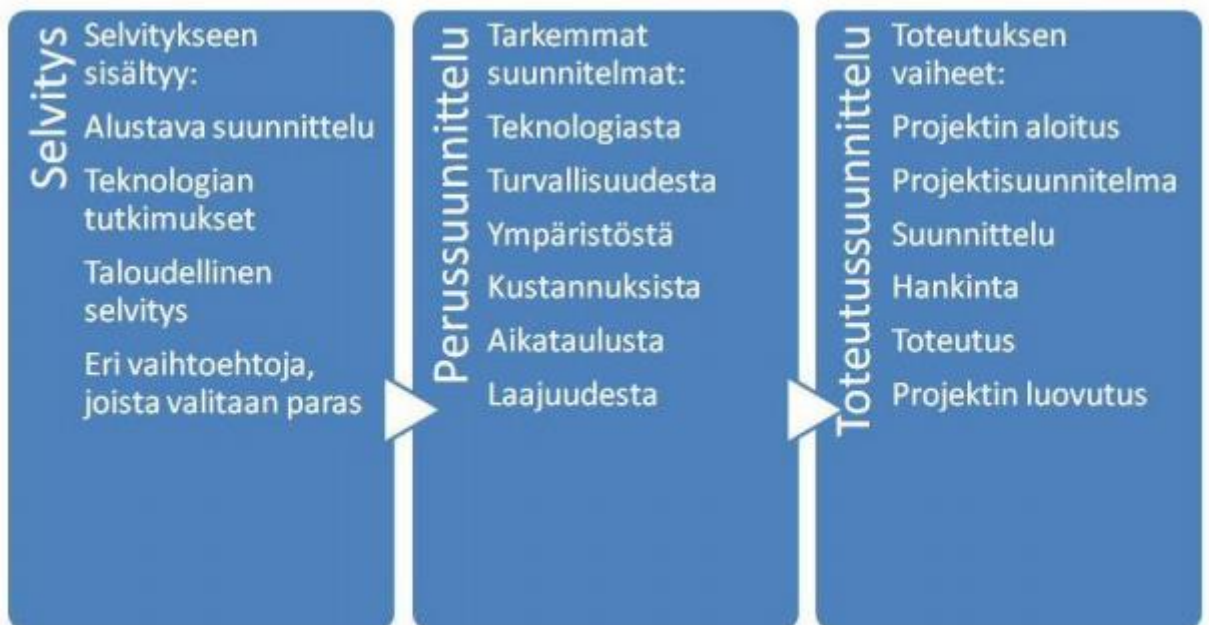
Luvussa 8 käsitellään SP3D:n standardikirjaston oliotietueisiin viittaamista siihen tarkoitettulla työkalulla, Schema Browserilla.

Luvuissa 9-12 keskitytään työn tuloksiin ja kehityskohteisiin sekä esitellään toteutettuja ratkaisuja.

## 2 LAITOSSUUNNITTELUN PERUSTEITA

Tässä luvussa perehdytään laitossuunnittelun perusteisiin, käydään lävitse laitosinvestointiprojektin vaiheet sekä osoitetaan laitossuunnittelun dokumenttien ja informaationvirtojen tärkeys laitossuunnitteluprojekteissa.

Laitossuunnittelulla tarkoitetaan tehdaskokonaisuuksien prosessilaitteiden suunnittelua ja kyseiselle työlle kertyvän hinnan selvittämistä. Laitossuunnittelu etenee yleisesti ottaen seuraavin vaihein.



Kuva 2. Laitosinvestointiprojektin vaiheet. (Pajuniemi 2009, 6-9)

Liitteessä 5 esitellään laitossuunnitteluprosessin toteutuskaavio sijoitussuunnittelun aloituksesta niin sanottuun As Built -vaiheeseen.

Laitossuunnittelu aloitetaan selvityksestä. Selvityksessä lasketaan taloudelliset vaikutukset, määritetään reitti- ja sähkösuunnittelun alustavat suunnittelut. Esisuunnittelujen jälkeen tehdään katselmus niihin laadituista dokumenteista ja asiakas päättää tarjouksen hyväksymisestä. Tämän jälkeen alkavat omat laitossuunnitteluryhmät tehdä heille

rajattuja suunnittelualueita. NJ:n tapauksessa tällaisia suunnitteluryhmiä ovat: Design Systems, Equipment Engineering, Civil Engineering, Layout ja Piping 1, Layout ja Piping 2 sekä Mechanical Engineering. Jokaisella ryhmällä on oma vastuhenkilönsä, jolle on annettu tietty aika saada suunnittelut (2D/3D) lopullisen mallikatselmuksen vaatimaan ulkoasuun. Henkilökohtaisten malliosuukien valmistuttua lähettää osuuden suunnittelija mallista (esimerkiksi PDS:n Isogen-ohjelmalla) generoimansa isometrin Kronodock-tietokantaan. PDS:ssä Isogen-ohjelma lukee Isorevision-tietokannasta osan tiedoista ja laitossuunnittelumallista muun muassa geometrian ja materiaalit. Kronodoc-sovelluksessa työstä vastaava tarkastaja varmistaa isometrin vastaavan kaaviopiirustuksia ja lähettää tarkastetun isometrin aluevastaavalle. Vastaava käy vielä kerran isometrin läpi ja merkitsee sen statukseksi hyväksytyyn, jolloin työ lähetetään eteenpäin ns. valmiina. Kaikkien isometriä tultua hyväksytyiksi tehdään loppukatselmus. Loppukatselmus tehdään juuri ennen laitoksen aiottua käynnistämisaikajankohdalla. Katselmuksessa laitosta käydään läpi 3D-mallissa ”kulkien”, jonka jälkeen laitossuunnitelma viimeistellään. (Neste Jacobs 2014, keskustelut)

Kuvassa 3 on havainnollistettu dokumenttien ja suunnittelun limittäisyyksiä laitossuunnittelussa.



Kuva 3. Laitossuunnittelun tiedonkulku. (Kesti 1992, 13)

Kuten kuvasta voi päätellä, erinäiset dokumentit ja niiden tulkinta ovat erittäin tärkeässä osassa laitosten suunnittelussa. Laitossuunnittelun tärkeimmäksi yksittäiseksi osaksi muodostuu kuitenkin putkistosuunnittelu. Putkistosuunnittelussa käytetään 2D/3D-pohjaisia laitossuunnittelusovelluksia, kuten PDS, PDMS, SmartPlant 3D tai MicroStation. Suunnittelu NJ:llä noudattaa kansainvälisiä ja yrityskohtaisia standardeja sekä ohjeita. Putkistosuunnittelu on jaettu kahteen ryhmään, Layout ja Piping 1 ja 2, joita johtavat ryhmänjohtajat. Ryhmien sisällä jaetaan putkisto-osuuksien suunnittelutehtävät ja aikataulut.

## 2.1 Laitossuunnittelussa käytettäviä työkaluohjelmia

Laitossuunnittelua tehdään pääosin erilaisten siihen tarkoitettujen työkaluohjelmien avulla. Suunnittelulle, dokumenttien hallinnalle ja isometrien generoinnille on omat työkaluohjelmansa. Tässä luvussa esitellään NJ:n käyttämiä työkaluohjelmia sekä selvitetään niiden käyttötarkoituksia.

Laitossuunnittelun aikana joudutaan käyttämään erilaisia ohjelmia dokumenttien hallintaan, suunnitteluun, katselmuksiin ja dokumenttien päivitykseen. Tällaisia ohjelmia ovat esimerkiksi Neste Jacobsin käyttämät Isogen, Isorevisio, Kronodoc, ProjectWise, Foundation, Smart Review, Navisworks ja Microsoft Officen-sovellukset.

Isogen on ohjelma, jolla 3D-mallista tehdään isometri. Isogen tekee 3D-malliin tehdystä kokoonpanosta graafisen isometripiirustuksen, jossa näkyvät mm. käytetyt putket, niiden koot, tyypit, tarkastusluokat ja käytetyt komponentit. Isometripiirustukseen tulee myös osaluettelo, joka sisältää kaikki kyseisen isometrin rakentamiseen tarvittavat tarvikkeet täydellisillä materiaalitilausnimikkeillään. PDS:ssä käytetään Isogen-ohjelman tukena sovellusta Isorevisio. PDS:ssä Isogen lukee Isorevisiotietokannasta isometriin muun muassa revisiomerkinlöjä ja projektitietoja.

Kronodoc on dokumenttien hallintaohjelma, joka toimii projektin aikaisena jakelutietokantana. Kronodociin tallennetaan kaikki isometrit. Siellä tehdään isometrien tarkastukset sekä hyväksynät. Sitten, kun työt on saatu ns.



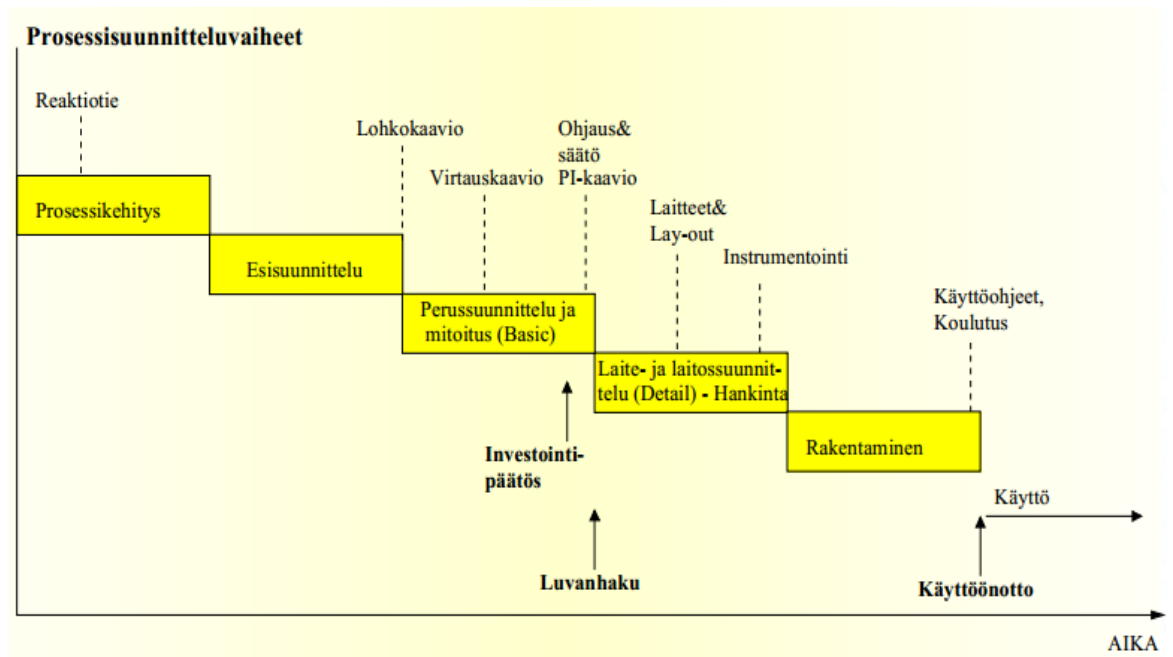
As Built -vaiheeseen, arkistoinnista vastaavat ihmiset siirtävät projektin tiedot lopulliseen dokumenttien hallintajärjestelmään, *Foundationiin*.

Smart Review ja NavisWorks -tyyppiset mallin katselmukseen tarkoitetut ohjelmat ovat myös hyvin tärkeitä laitossuunnittelussa. Malleista tehdään projektien aikana säännöllisiä katselmuksia. Smart Review ja Navisworks ovat mallin tietojen tarkasteluun suunniteltuja ohjelmia. Näillä ohjelmilla havaitaan esimerkiksi tilan puutteet, laitteiden ja venttiilien oikeanlaiset operointisuunnat, operointikorkeudet, huollettavuus sekä prosessille ja kuormitukselle kriittisten putkistojen oikeanlainen sijoitus.

## 2.2 Laitossuunnittelun dokumentaatio

Laitossuunnittelussa vaaditaan paljon erilaisia dokumentteja niin suunnittelua kuin kunnossapitoakin varten. Tässä luvussa selvitetään putkistosuunnittelun dokumenttien ja laitossuunnittelun aikataulujen yhteyttä esittelemällä NJ:n projektidokumenttityyppejä.

Putkistosuunnittelu kulkee tarkan prosessisuunnittelumallin mukaan, jossa esiintyvät osa-alueet esitellään seuraavaksi niiden esiintymisvaiheen mukaisessa järjestyksessä dokumenttipiirustusnimikkeiden avulla. Esisuunnittelua on käyty läpi jo aiemmin, joten seuraavassa keskitytään perussuunnitteluun liittyviin asioihin.



Kuva 4. Dokumenttien esiintyminen prosessisuunnittelussa vaiheittain.

(Heikkilä-Malmen 2014)

Virtauskaavion laatiminen on prosessisuunnittelijoiden tehtävä ja se tehdään heti esisuunnittelun jälkeen. Virtauskaaviosta nähdään prosessin pääelementit ja materiaalivirrat. Virtauskaavio on osa perussuunnittelua ja siinä esitellään prosessilaitteita standardin mukaisin symbolein. Virtauskaaviota käytetään prosessin toiminnan hahmottamiseen.

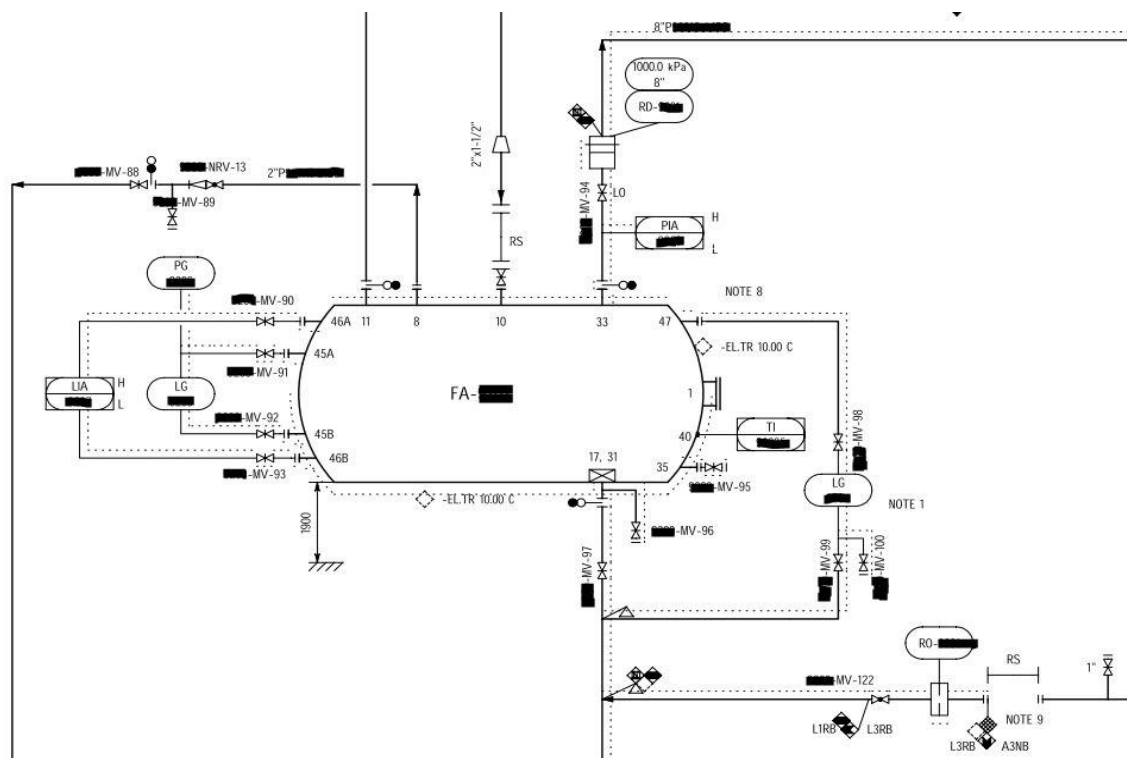
Liitteessä 4 on esitetty yhden yrityksen spesifikaatioiden mukainen näkemys vetyverkon virtauskaaviosta ja sen merkinnöistä.

Tarkemmin määriteltynä virtauskaavio sisältää prosessin kuvauksen toiminnasta, aine- ja energiataseet sekä virtausmäärät, paineet ja lämpötilat, kuten myös päälaitteet, niiden nimet, tunnuksat ja kokoa tai kapasiteettia ilmaisevat suureet.

PI-kaavio (P&I drawing) on joko prosessi- ja instrumentointikaavio tai putki- ja instrumentointikaavio. PI-kaavion laatii prosessisuunnittelu. PI-kaavioon suunnitellaan prosessivirrat sekä putkien tyhjennys- tai ilmastusyhteisiin liittyvät oleelliset asiat. Yleensä PI-kaavio tehdään CAD-pohjaisella ohjelmistotyökalulla ja sitä

käytetään suunnittelun, käytön, käyttöönoton ja kunnossapidon aputyökaluna. PI-kaavio sisältää prosessivirtojen yksityiskohtaisen kulun merkinnät. Näitä ovat: kaikki laitteet, putket, venttiilit, varusteet sekä instrumentointi, putkitunnukset, saatot ja muut putkistosuunnittelun tarvitsemat lähtötiedot.

Kuvassa 5 on esitetty prosessisäiliön yhteyteen tulevien laitteiden, putkien sekä niiden tunnuksien merkintätapoja.



Kuva 5. PI-kaavio prosessisäiliöstä.

Putkiston tasopiirustus (Arrangement drawing-GA) on putkien sijoittelupiirustus ja sen tekee yleissuunnittelu. Tasopiirustus on nimensä mukaisesti 2D-tasokuva kohteesta, johon putket tulee saada sijoitettua. Putkiston tasopiirustuksessa näkyvät: Kaikki putkistot lukuun ottamatta höyrysaatto- ja impulssiputkia; mittakaava 1:25 tai 1:33 ISBL ja 1:50 tai 1:100 OSBL; putkiston tasokorkeudet, kannakkeet, instrumenttien, instrumenttikaappien ja kaapelihyllyjen sijoitus; tarvittavat leikkauskuvannot putkiston asennusta varten. Mikäli työhön liittyy useampia

tasopiirustuksia, laaditaan Plot Plan Key Plan-piirustus, jossa näkyvät kaikkien tasopiirustusten rajat.

Putkiyhteyksikaavio (Pipeline Diagram) on prosessi- tai putkistosuunnittelun laatima dokumentti, jota pidetään ajan tasalla vain suunnittelun ajan. Putkiyhteyksikaavio sisältää tiedot eri yksikköjen, säiliöalueiden ja lähettämöiden välisistä putkiyhteyksistä. Yksiköt ovat kaaviomaisesti maantieteellisessä sijaintijärjestyksessä. Yksiköistä näkyviä tietoja ovat mm. linjanumerot ja pääsulkuventtiilit, pumppaamot jne.

Laitesijoitus (Plot Plan, ISBL-alue): Layout-suunnittelu laatii laitesijoituspiirustuksen, joka on osa esisuunnittelua ja josta ilmenee sijoitettavien laitteiden paikat ja tilantarve. Laitesijoituspiirustuksessa (1:200 tai 1:250) esitetään laitteet, rakennukset ja rakenteet ääriivoin sekä putkisiltojen sijoitukset mitoitettuina.

Isometrinen putki- ja laitesuunnittelu on putkiston valmistuspiirustus. Se tehdään 3D-mallista Isogen-nimisellä ohjelmalla ja siinä näkyy kaikki putkistoon liittyvät tiedot laitoksesta. Isometrinen putki- ja laitesuunnittelu on tärkeimpiä putkistosuunnittelun dokumentteja, koska siinä näkyvät: putkikohtaiset valmistus- ja asennustiedot; putkiston osien mitat, rakenneaineet, aineistodistusvaatimukset sekä kannatus-, ohjaus- ja kiintopisteet. Erikoistapauksessa siinä esitellään myös lämpökäsittely – esimerkiksi mikäli lämpökäsittely aiheutuu virtaavasta aineesta tai muista erikoistarpeista (PWHT).

Edellä mainittujen lisäksi on olemassa vielä muitakin suunnittelijan dokumentteja, kuten Aluekartat, Perustus- ja maanalaisten putkien sijoituspiirustus, Maanalaisten rakenteiden sijoituspiirustus, Pituusleikkauspiirustus, Kannakkeiden taso-, korkeus- ja kuormitustaulukko, Perustusten mitta- ja kuormituspiirustus, Kannakkeiden ja putkisiltojen teräsrakennepiirustus, Lämpösaattopiirustukset, Putki- ja eristysluettelo, Laiteluettelo, Putkistomateriaalin ennakkovaraus (Työnumerokohtainen), Putkiston materiaaliluettelo, Teräsmateriaalin osaluettelo, Ritilöiden hankinta määrittely, työmäärittely, Dokumenttiluettelo sekä Putki- ja

laitenumerointi ohje.

(Toivonen 2011)

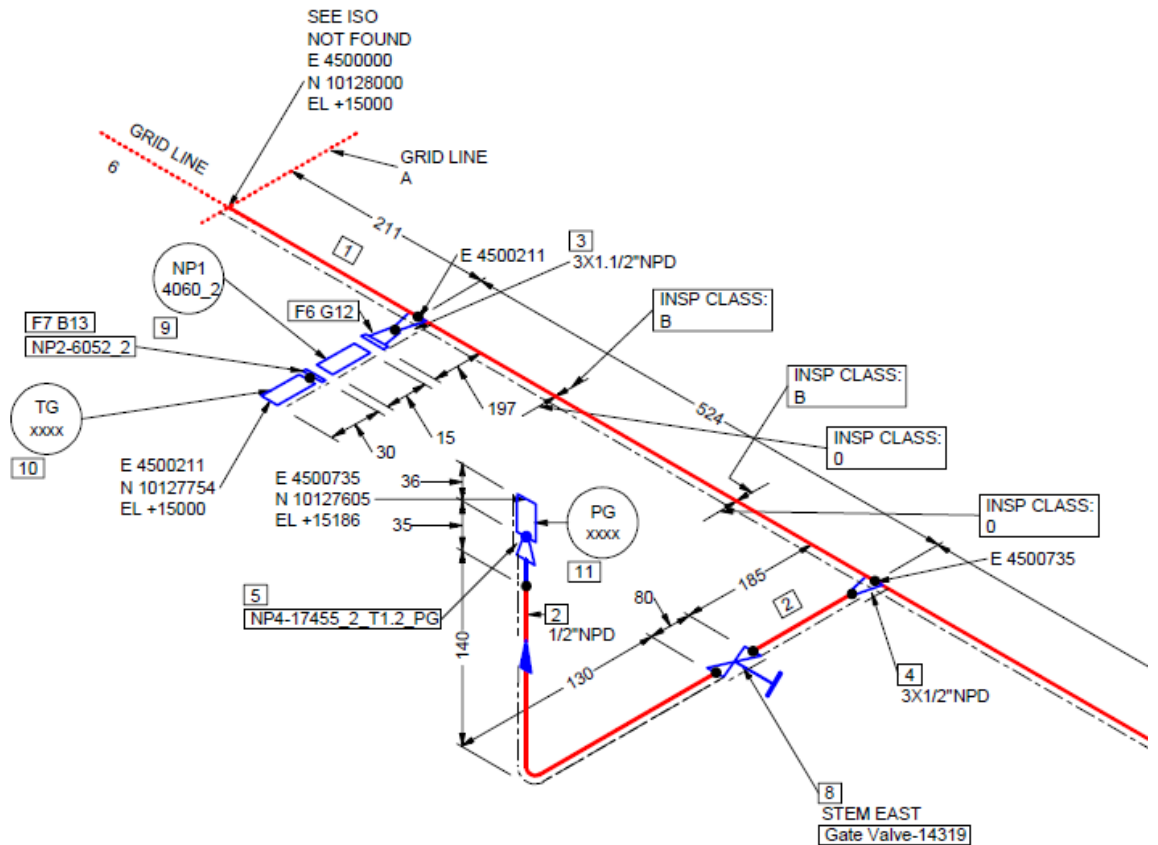
### 2.3 Putkistosuunnitteluun perehdytys

Tässä luvussa käydään lävitse putkistosuunnittelun kulkua ja esitetään valmiita kokoonpanoja sisältävä isometripiirustus.

Suunnittelu aloitetaan putkien sijoittelusta. Putkistoista tehdään reittikaavio, jonka perusteella putkistosuunnittelijat suunnittelevat 3D-mallin esimerkiksi SP3D:llä. Mallin valmistuttua siitä ajetaan putkistoisometri. *Putkistoisometri on aksometrisen projektiotavan mukainen yhdensuuntaisprojektio.* (Pere 2009, 14-48)

Isometrit voidaan jaotella neljään ryhmään, joita ovat: alue-, ryhmä-, linja-, ja linjaosaisometrit. Alueisometrissa esitellään alueellisesti rajatun putkiston putket ja laitteet. Ryhmäisometrissa esitellään tietyn prosessivaiheen putkistoja. Linja- ja linjaosaisometreissa taas esitellään yksittäisiä putki-, apu-, tai ohituslinjoja. Isogen-ohjelma toimii niin, että se hakee tiedot suunnittelijan tekemästä mallista sekä Isorevisio-nimisestä ohjelmasta. Jos mallissa on vikaa, se näkyy isometrin generointiprosessin aikana.

Kuvassa 6 on esitetty Isogen ohjelmalla generoitu putkistoisometri, josta selviää kaikki putkiston toteutuksen kannalta olennainen tieto. Tällaista tietoa on muun muassa putkilinjan halkaisija, eristysluokka, tarkistusluokka, linjalle asennettavat laitteet yms.



Kuva 6. Putkistoisometri.

Suunnittelun aikana laadittavia piirustuksia ovat muun muassa putkiston tasopiirustus, putkiyhteykskaavio sekä *putkistoisometri*. Kyseiset piirustukset laatii yleissuunnittelu. Edellä mainittujen lisäksi laaditaan vielä materiaaliluettelot tarvittaville elementeille.

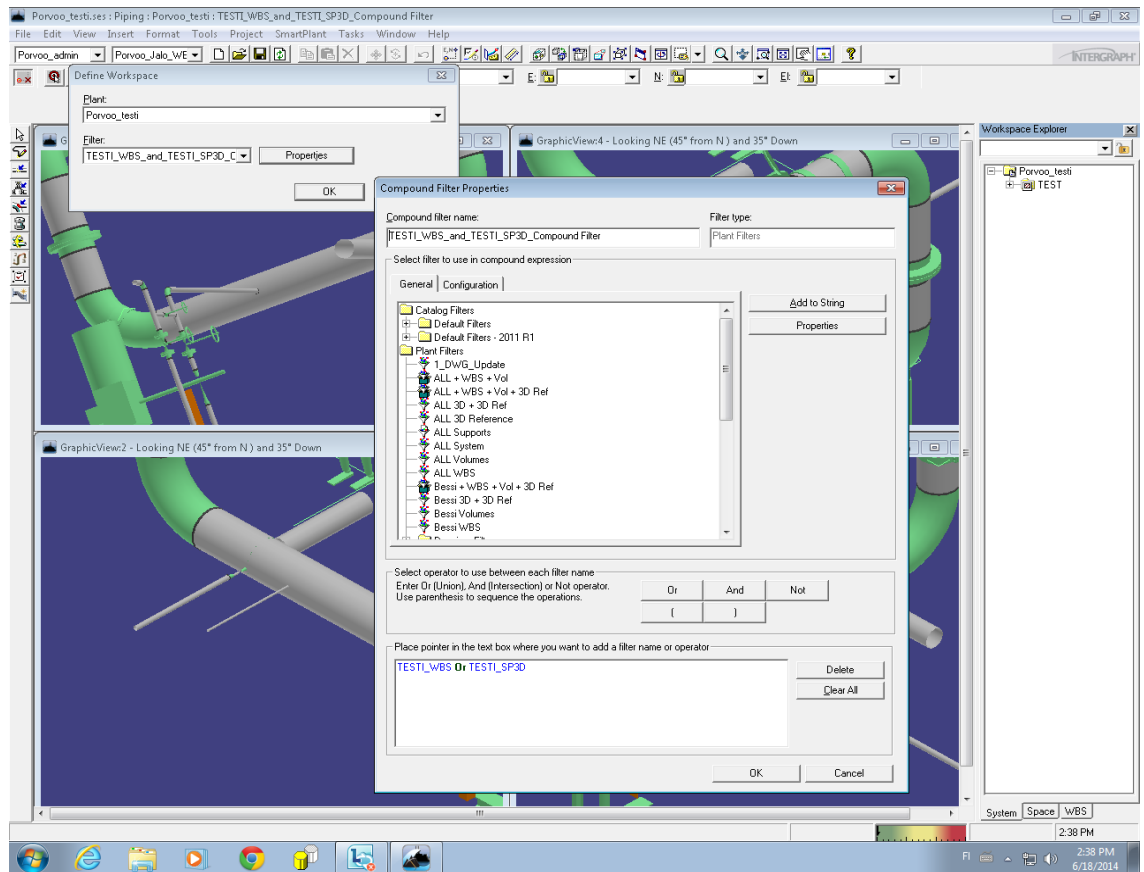
Putkistojen määrittely tehdään käyttötarpeiden mukaan. Käyttötarpeiden määrittelyjen jälkeen tehdään risteävät putkistotasot, huomioidaan putkistojen lämpöliikkeet, kannatukset ja eristykset, ja lopulta ryhdytään elementtien, kuten säätöventtiiliryhmien ja virtausmittausten sijoitukseen. Suunnittelussa toteutetaan ensiksi isoimmat tilaa vievimmät putket, sen jälkeen pienemmät putkistot sekä komponentit. (Neste Jacobs 2014, sisäinen materiaali)

### 3 SMARTPLANT 3D -LAIROSSUUNNITTELUOHJELMAN ESITTELY

SmartPlant 3D on Intergraphin kehittämä 3D-mallinnukseen soveltuva laitossuunnitteluohjelma. SP3D on ollut vasta muutaman vuoden markkinoilla, ja se kehitettiin korvaamaan Intergraphin edellinen laitossuunnitteluun tarkoitettu ohjelma, Plant Design System. SP3D:n toiminta perustuu tietokantapohjaisiin elementteihin, joita voidaan sijoitella laitosmalliin. Ohjelma on jaettu osiin eli taskeihin siten, että kaikille suunnitteluryhmille kuten laite-, putkisto- ja rakennesuunnittelu on omat taskinsa. Ohjelmassa on määritelty laitekirjasto, josta yleisimmät laitossuunnittelussa käytettävät elementit löytyvät. Ohjelman keskeisimmät osat ovat: Grid, Structure, Common, Piping, Systems and specifications, Equipment and Drawings. Näitä taskeja käsitellään tarkemmin jäljempänä.

SP3D:n toiminnassa filteröinnillä eli suodatuksella on merkittävä rooli. Suodatuksella tarkoitetaan mahdollisuutta valita käsiteltäväksi esimerkiksi tietty laitosmallin osa, vaikkapa jokin rakennus. Koko laitos mallinnetaan hierarkkiseen hakemistojärjestelmään, jonka rakenne on laitoksen suunnittelijoiden määriteltävissä. Tämä järjestelmä on nimeltään WSE, Workspace Explorer, joka näkyy muun muassa kuvissa 11 ja 12. Mitkä tahansa hakemistojärjestelmän osat voidaan valita aktiivisiksi työskentelynäkymään. Toisaalta suodatuksella voidaan myös rajata käsiteltäväksi esimerkiksi tietyn tyyppiset elementit, kuten teräsrakenteet, halutut laitteet tai putkistot.

Kuvassa 7 on esitetty suodattimen eli filtlerin valinta. Kuvassa on valittu laitos Porvoo\_testi ja sen alta näkymää tarkemmin rajaavaksi suodattimeksi TESTI\_WBS. Filtterin valinta tehdään joko valitsemalla se pudotusvalikosta tai rakentamalla oma suodatin menuvalinnalla File - Define Workspace.



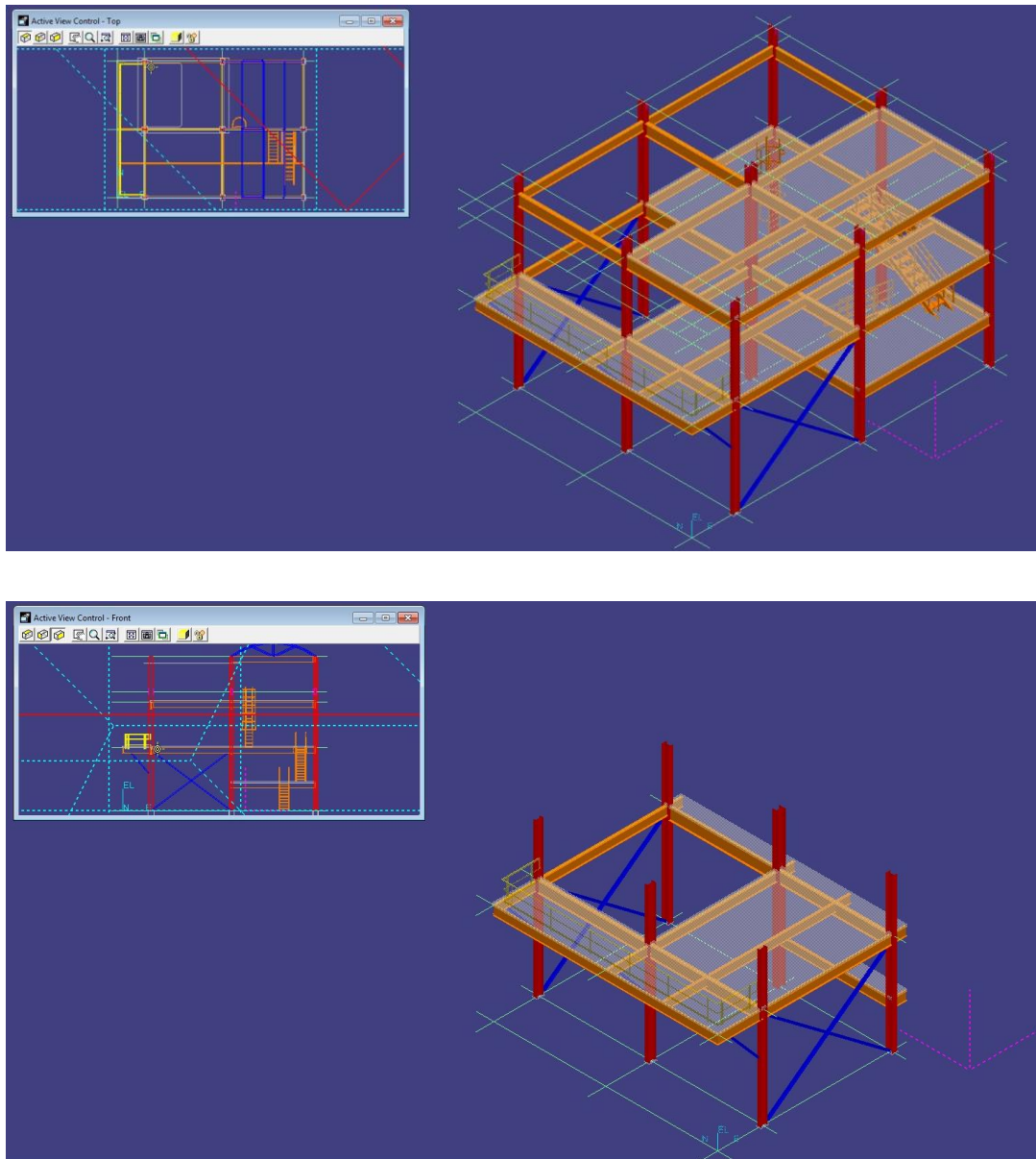
Kuva 7. Suodatuksen eli Filterin valinta.

Seuraavassa esitellään hiukan tarkemmin SmartPlant 3D-ohjelman keskeisimmät osat.

Common on ohjelman päätäski, jonka keskeisimpiä toimintoja ovat: *Clippaus*, katselukulman vaihto, zoomaus, filterin valinta, sovitus, kontrollipisteiden valinta ja referenssielementtien määrittäminen. Clippauksella tarkoitetaan mahdollisuutta rajata tarkasteltavaksi jokin tietty geometrinen osa mallista.

Kuvassa 8 on esimerkki, jossa on otettu erään rakennuksen 1. kerroksen osa lähempään tarkasteluun. Rajaamalla ylimääräiset objektit kuvasta pois, pääsee suunnittelija huomattavasti helpommin käsiksi haluamiinsa kohtiin.





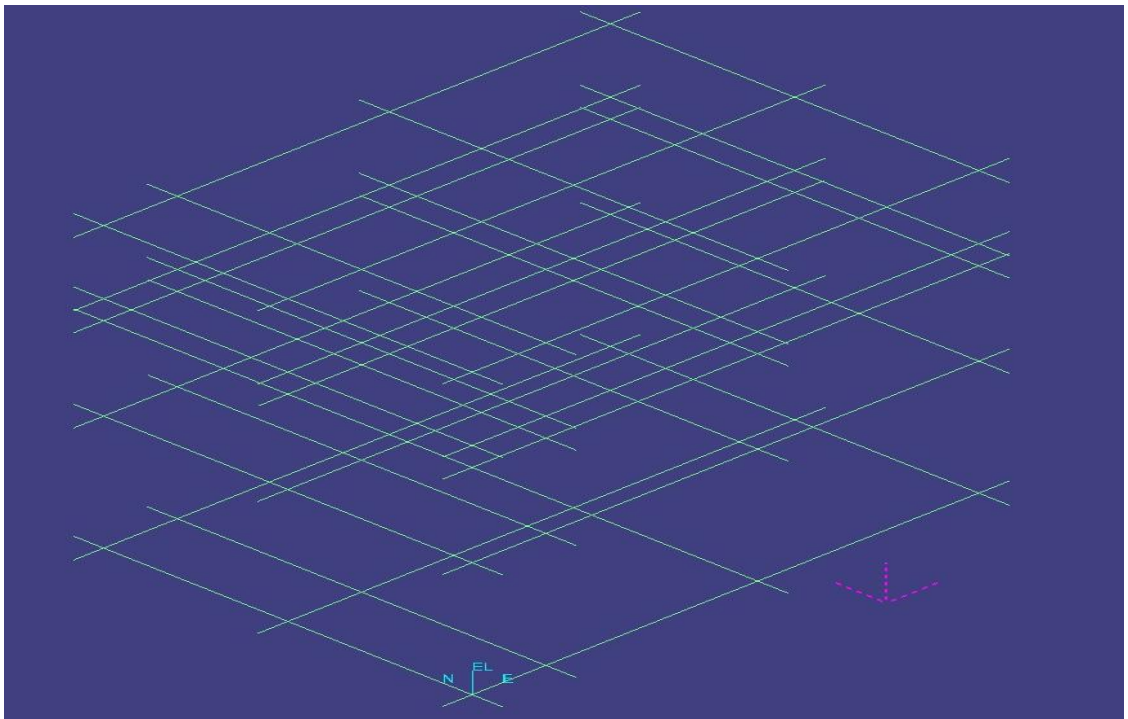
*Kuva 8.* Clippaus rajatun alueen mukaan. Ylemmässä kuvassa näkyy koko rakennus ja alempaan kuvaan on rajattu siitä ensimmäisen kerroksen etuosa.

Kuten kuvassa 8 näkyy, suodattamalla voidaan rajata mallissa näkyviä elementtikonaisuuksia. Tätä varten SP3D-ohjelma sisältää perussuodatinvalikoiman. Jos perussuodatinvalikoima ei ole riittävän tarkka, on ohjelman käyttäjän myös mahdollista tehdä ja tallentaa ohjelman suodatinkirjastoon myös omia filttareitä

Ohjelmassa on monia tapoja tarkastella mallissa esiintyviä elementtejä, rajata näkyviä alueita ja leikata osia halutulla tavalla. Yksi merkittävimpiä näkymän

leikkaustoimintoja on *clippaukseen* sisältyvä aputyökalu Active View Control, jolla voi rajata näkymää eri kuvakulmien mukaisesti erillisessä näkymän rajausikkunassa. Kuvassa 8 näkyy mallin näkymä sekä Active View Control rajaus paneeli, jossa mallin näkyvää osaa voidaan rajata erisuuntaisilla näkymää rajaavilla katkoviivoilla.

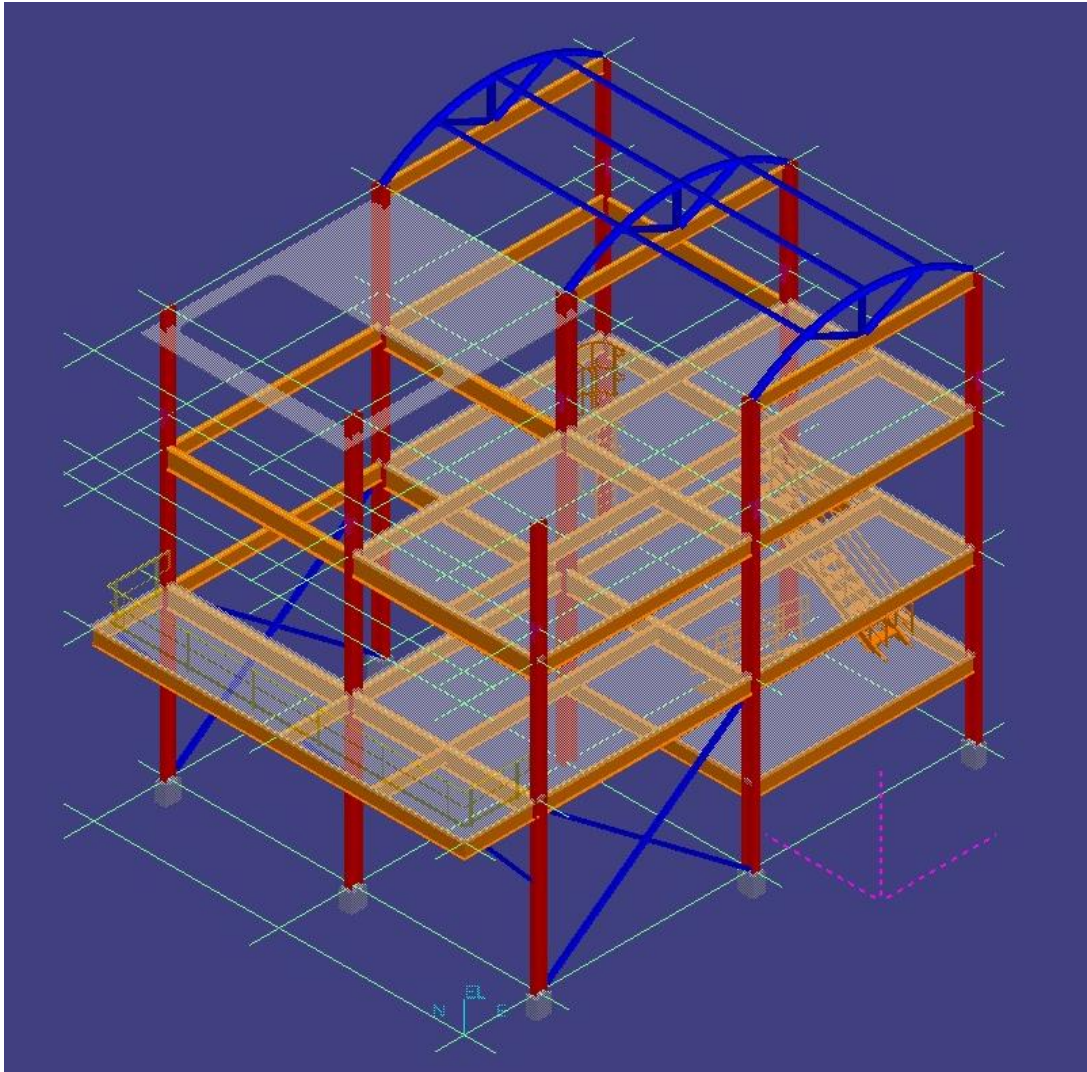
Grids-taskissa määritellään malleissa käytettävät koordinaatistot ja niiden aputasoviivastot. Käytettävissä on suorakaiteen muotoisia ja pyöreitä apuviivastoja. Grid-taskissa määriteltyjen koordinaatistojen E-, N- ja EL-tasot (X, Y, Z) näkyvät Workspace Explorerissa, joka on käyttöliittymässä esiintyvien osien elementtikirjasto. Yksinkertaisin tapa tehdä gridit on käyttämällä toimintoa Grid Wizard. Tällä toiminnolla määritellään luotavan rakennuksen perustasot kaikissa koordinaattisuunnissa. Esimerkiksi kuvan 9 viivastossa on 5 korkeustasoa sekä 3 N-akselin ja 4 E-akselin suhteen määräytyvää tasoa.



Kuva 9. Grid-viivastot, Kymenlaakson ammattikorkeakoulun koulutusmateriaali.

Kuvassa 9 on esitetty kuvan 8 rakennuksen pohjana oleva Grid-viivasto. Tämän avulla mm. rakennuksen pilarit ja palkit voidaan helposti sijoittaa oikeille paikoilleen.

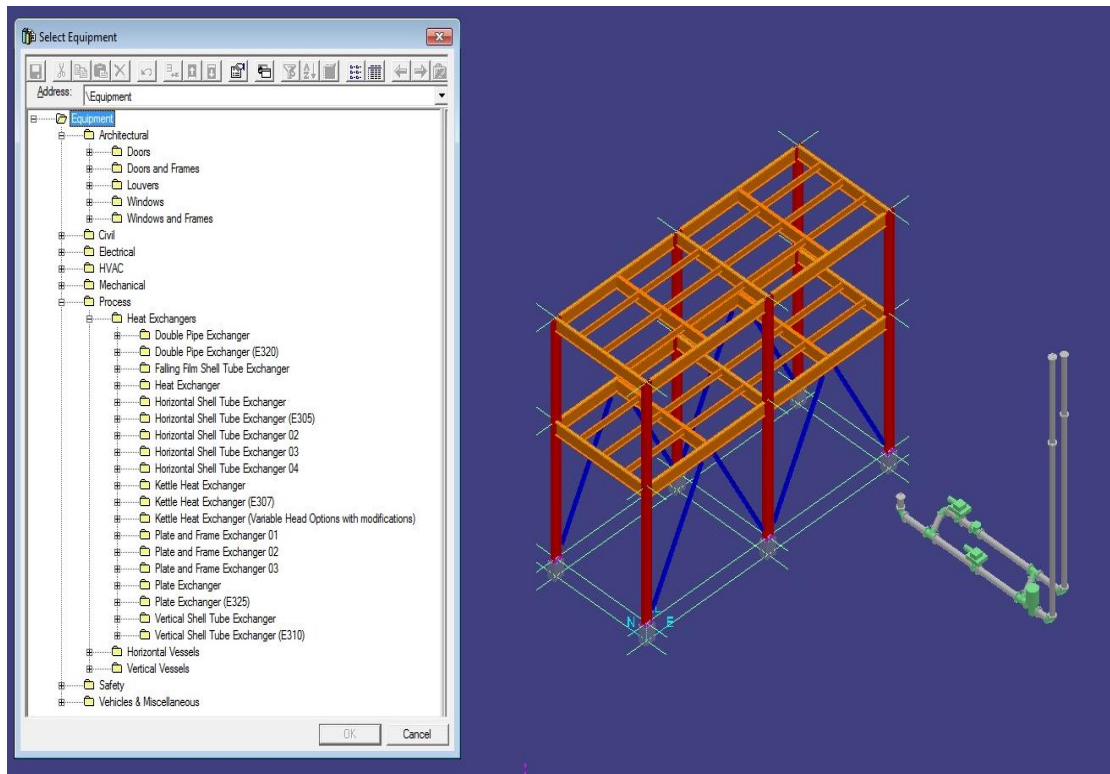
Structure-taskissa määritellään mallin rakenne ja siinä voidaan tehdä muun muassa: Teräsrakenteet, tuet, perustukset, lattiat, portaat, tikapuut, käsituet ja muut rakenteisiin liittyvät elementit. Rakenteet mallinnetaan yleensä Grid-viivoja apuna käyttäen.



*Kuva 10.* Structure-tason toiminnoilla rakennettu yksikkö.

Equipment-taskissa määritellään mallissa käytettävät laitteet, kuten esimerkiksi: Säiliöt, pumput ja lämmönvaihtimet. Laitteita voi luoda itse tai valitsemalla ja määrittelemällä käyttöasetukset laitekirjastosta löytyville ”parametrimalleille”.

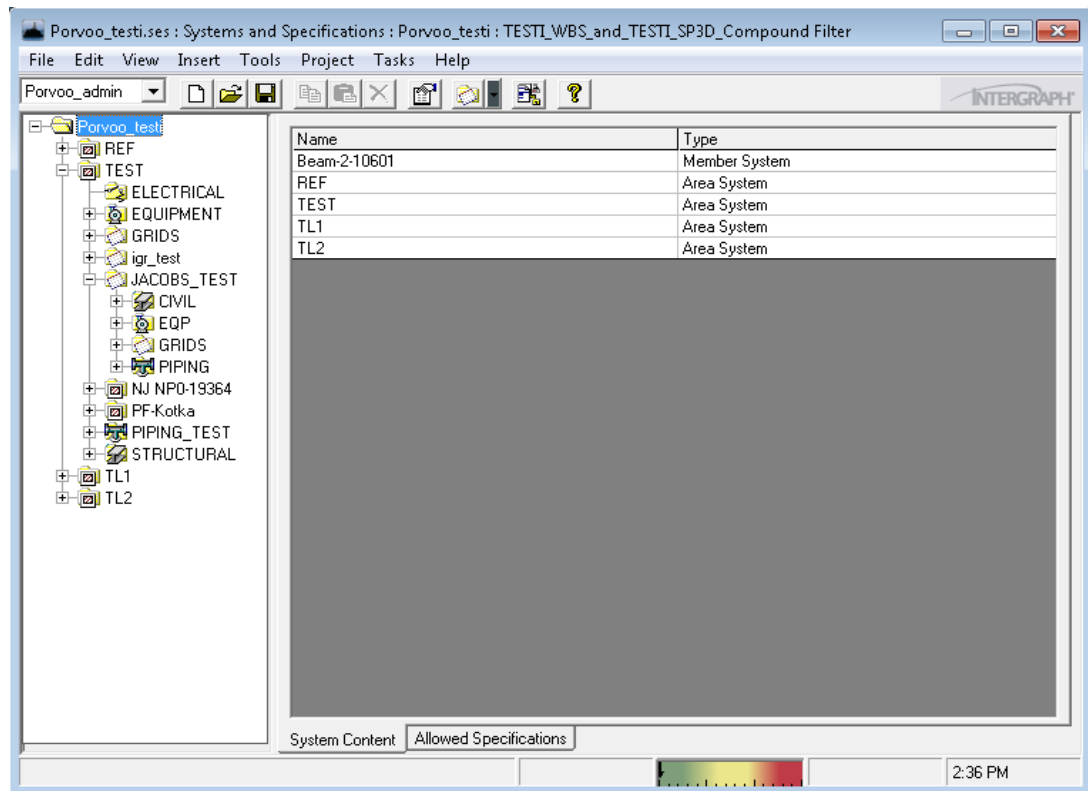
Kuvassa 11 on havainnollistettu elementin sijoittamista (Place Equipment) toiminnolla. Kuvassa näkyy SP3D standardikirjaston lämmönvaihdinvalikoima.



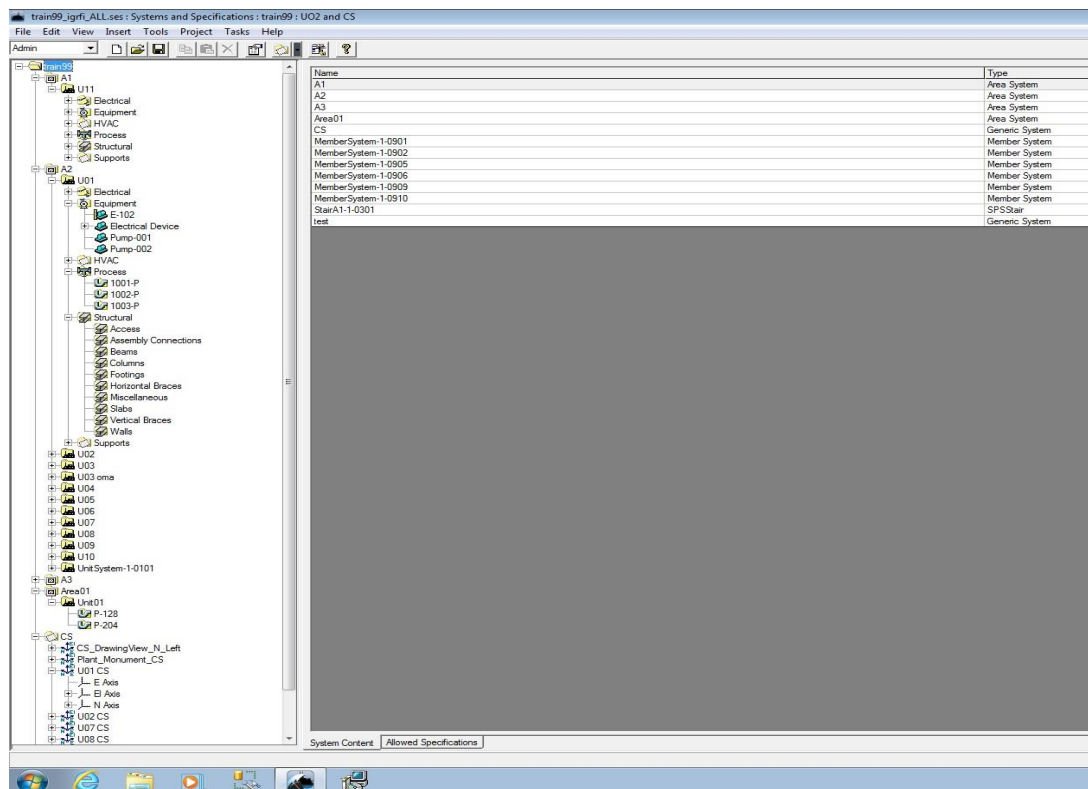
*Kuva 11.* Lämmönvaihtimen valinta malliin, Place Equipment -toimintolaatikolla.

Systems and Specifications -taskissa tehdään ohjelman järjestelmämäärittelyt. Tässä taskissa voidaan määrittellä muun muassa Workspace Explorerin rakenne.





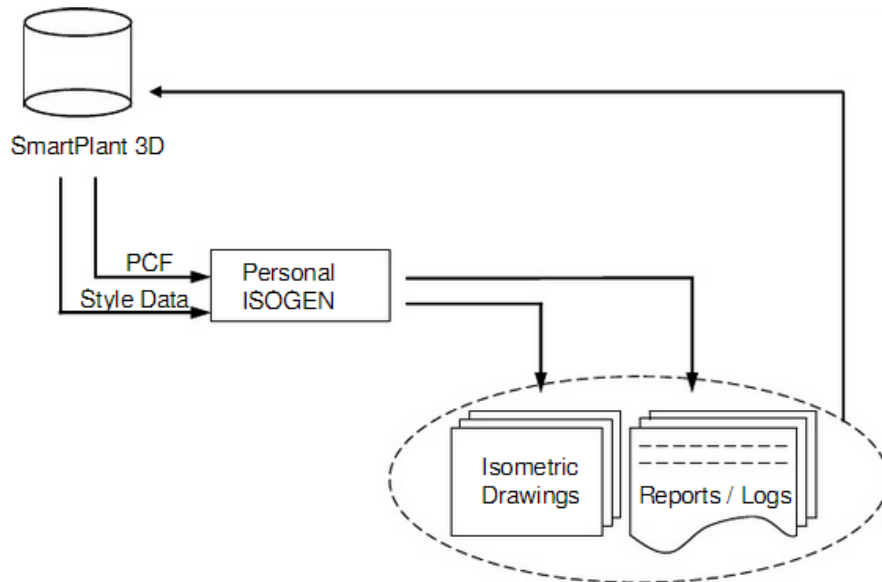
Kuva 12. Systems and Specifications-osion näkymä 1.



Kuva 13. Systems and Specifications-osion näkymä 2.

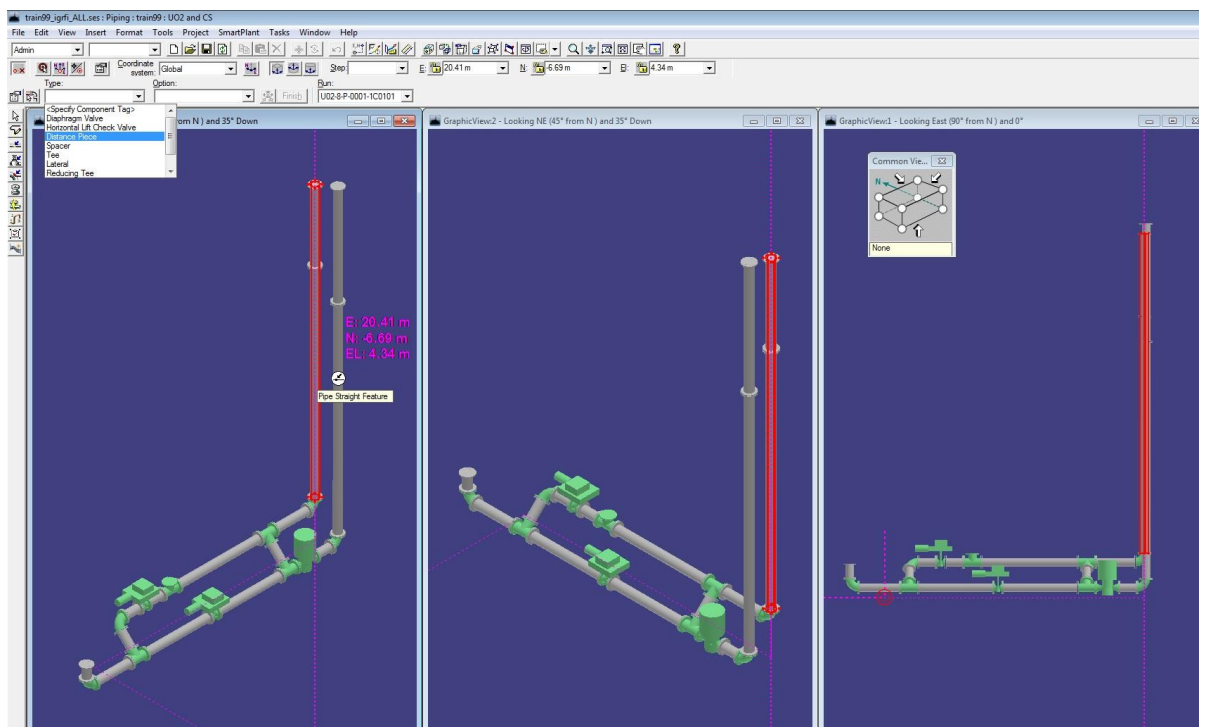
Kuvassa 12 on määritelty WSE näkymä. Kyseessä on testisessio, joten WSE:hen on valittu vain yleisesti käytetyimmät rakennekansiot kuten Electrical, Equipment, Grids ja Piping. Kuvassa 13 on esimerkki laitoksen hierarkkinen WSE rakennenäkö. Kuvan 13 WSE näkyy sisältävän mm. laitoksen alueita koskevat tiedot kansioissa A1 (Area1) sekä A2 (Area2). Kuvassa 13 näkyy hyvin tarkasti mm. Rakenne (Structural) kansion sisältämät alikansiot kuten pilarit ja palkit (Columns ja Beams) sekä CS kansion sisältämät koordinaattijärjestelmät. Jokainen kansio sisältää tarkat tiedot laitoksessa käytettävistä laitteista, putkista yms. Mitä yksityiskohtaisempaa tietoa laitoksesta halutaan tarkastella, sitä tarkemmat tiedot laitoksen hierarkkisesta WSE-rakenteesta tarvitaan. Haluttua elementtiä voi tosin myös hakea sen nimikoodin avulla, mikäli tämä on tiedossa.

Drawings and Reports-osiossa tehdään automatisoidusti ennalta määritellyt taso- ja leikkauspiirustukset, isometriset putkistopiirustukset, piirustus- ja materiaaliraportit, laite- ja instrumentointiraportit sekä diagnostiikkaraportit.



Kuva 14. Smartplant 3D-ohjelman Drawings & Reports-osion yhteys isometrin generointiin.

Piping-taskissa tehdään putkistosuunnitteluun liittyviä asioita. Se on ohjelman osa josta löytyvät kaikki putkistosuunnitteluun käytettävät elementit. Piping-taskin keskeisimpiä toimintoja ovat putkilinjojen vetäminen sekä putkistokomponenttien ja -instrumenttien sijoittaminen malliin. Tässä taskissa kaikki putkistot jaetaan osakokonaisuuksiin, kuten Pipeline, PipeRun, Feature ja Part. Pipeline tarkoittaa putkilinjakokonaisuutta, PipeRun kokonaisuuteen kuuluvaa putkiosuutta, jonka ominaisuudet pysyvät samana. Esimerkiksi haaroituksesta ja toisaalta nimellishalkaisijan muutoskohdasta alkaa uusi PipeRun. Feature on yhden putkiosan malli (teorettinen putkilinjan osa) ja Part on putkiosaan liittyvä todellinen putkikomponentti, esimerkiksi 90-asteen mutka.



Kuva 15. Komponentin lisääminen putkiosuuteen Place Component-toiminnolla. Kuvassa näkyy sama putkilinja kolmesta eri kuvakulmasta.

Kuvassa 15 on valittu näkyviin putkiosuus, (pystysuora putki) johon halutaan lisätä komponentti. Vaihtoehdot osuuteen sopiville komponenteille tulevat näkyviin Type-pudotusvalikkoon.

#### 4 PARAMETRISEN OHJELMOINNIN PERUSTEET

Tämä luku tarjoaa katsauksen parametriseen ohjelmoinnin kehitykseen. Luvussa käsitellään ohjelmoinnin perusteita sekä esitellään yleisimpiä ohjelmointikieliä. Luku sisältää myös kappaleen kokoonpanomakrojen käytöstä SmartPlant 3D -laitossuunnitteluohjelmassa. Luvun tarkoituksena on antaa kokonaiskuva parametrisestä ohjelmoinnista, jonka yksityiskohtia käsitellään myöhemmin luvuissa 5, 6, 7 ja 8.

Parametrinen ohjelmointi juontaa juurensa 1960-luvulta. Ensin yritykset siirtyivät paperipohjaisten kone-, laite- ja laitossuunnitelmien teosta tietokonepohjaiseen suunnitteluun (CAD) ja seuraavassa vaiheessa alettiin korvata rautalankamalleja ensin 2D- ja sitten 3D-malleilla. PDS oli ensimmäisiä tällaisia 2D/3D-malleja tekeviä laitossuunnitteluohjelmia. Markkinoille tuli erilaisia ohjelmia, joista jokainen käytti hiukan erilaista ohjelmointikieltä. Kielistä C++ ja XML tuli suosittuja ohjelmointikieliä 2D/3D laitos- ja konesuunnittelussa. Sellaisista ominaisuuksista, joille voidaan ohjelmallisesti antaa muuttuvia arvoja, käytetään nimitystä parametrit. Parametreja voivat olla esimerkiksi tyyppi, materiaali, lämmönkestävyys ja virtaussuunta.

Parametriseella ohjelmoinnilla tarkoitetaan mahdollisuutta saada sovellus tekemään asioita annettujen tietojen, niin sanottujen parametrien avulla. Parametrinen ohjelmointi voi hyväksi käyttää tietokantaa esimerkiksi Visual Basic-kielisen ohjelman kautta. Tietokannan tietuerakenteet vastaavat tällöin luokkamäärittelyjä erinäisille objekteille kuten pumput, venttiilit ja muut vastaavat rakenteet, joista jokaiselle rakenteelle on määritelty tietyt ominaisuudet. Nämä ovat tietokannan näkökulmasta tietueen kenttiä. Näin ollen esimerkiksi palloventtiilillä on tiettyjä osia yleisen venttiilin tietueesta. Yhteinen ominaisuus on esimerkiksi virtaussuunta. Kaikille venttiileille voi määrittää virtaussuunnan, joka tallentuu tietokantaan tätä venttiiliä vastaavaan tietueeseen. Virtaussuunta on siis yksi esimerkki tietueen kentästä, joka jakautuu pienempiin tietueisiin, ja toisinpäin. Tällöin parametrinen ohjelmoija voi rakentaa koodatuista tietueista itselleen sopivan kokonaisuuden



viittaamalla tietueiden tietokannassa sijaitseviin attribuutteihin. Asiaa on käsitelty hieman tarkemmin luvussa 8 (Schema Browser).

SP3D käyttää parametrisessä ohjelmoinnissa sille ominaisesti varattuja olio-ohjelmointisanoja sekä määrittelyjä, joiden käyttöä esitellään myöhemmin tärkeimpien määritelmien osalta SP3D:n kokoonpanoja käsittelevässä luvussa 6.

SP3D käyttää PAL-nimistä tiedostoformaattia parametrinen kokoonpanojen tallennusmuotona. PAL tulee sanoista "Parametric Assembly Language", joka viittaa C-kielen tapaiseen ohjelmointikielen. PAL-tiedostomuoto on XML dokumentaatio-ohjelmointia lisättynä PAL-määrittelyillä erikoissanoilla. SP3D:n käyttämiä erikoissanoja sisältäviä osioita ovat PAL-formaatissa mm: Information, GIFFile, ReferenceElements, Parameters, ValidateParameters, Systems, Equipments, Network, Items, Overrides, SetAttributesOnRunItems, ConnectFreeEnds, Supports ja SetAttributes. Näistä edellä mainituista tärkeimpiä tullaan käsittelemään luvussa 6 "Parametrisen ohjelmoinnin perusteet SmartPlant 3D:ssä". (Intergraph PAL 2009)

#### 4.1 XML-kieli ohjelmoinnissa

Tässä luvussa esitellään ohjelmoinnissa paljon käytetty kieli; XML. Ymmärtämällä yhden ohjelmointikielen perusteet, kasvavat valmiudet ymmärtää muitakin ohjelmointikieliä. Siksi seuraavassa esitellään XML-kieltä, valotetaan kielen taustaa esimerkein, ja osoitetaan yhteys SP3D:n oliopohjaiseen ohjelmointikielen.

XML-lyhenne tulee sanoista: "Extensible Markup Language".

SP3D:ssäkin ohjelmointi saa perussääntönsä XML-kielestä. Jos olio-ohjelmointi olisi puu, olisi XML-kieli sen juuret. XML-kieli on kehitetty, jotta dataa eli tietoa voitaisiin ilmaista tekstimuodossa. XML-kielessä tekstile on myös annettu tiettyjä erikoisominaisuuksia. Yhtenä ominaisuuksista on tapa hyödyntää ihmiselle selkeitä sanoja. Jos halutaan ilmaista johonkin tiettyyn alaan liittyvää dataa, käytetään sille ominaisia termejä. Jokaiselle termille voidaan sitten määritellä tietty toimintatyyppi. Voi olla, että yksi termi pitää sisällään vain kyvyn olla luettavissa ja toinen taas kyvyn

lukea edellistä. Tällöin, kun viitataan termistä toiseen, itse asiassa vain kommunikoidaan kahden sanan välillä. XML-kielen toinen erikoinen ominaisuus on sen sisältämät erikoismerkit. Näillä merkeillä voidaan "aidata" tietoa sen termin sisälle, johon sen halutaan kuuluvan. (Graves 2002)

Esimerkki 1. "Aitaus", SP3D

```
<Network>
    <PipeLine Use="Element(1)" />
</Network>
```

Tällöin ilmaistaan, että tiedon Pipeline Use halutaan sisältyvän termin Network alle. Eli toisin sanoen kyseistä tietoa käsitellään vain Network-osiossa, sille määritellyin ehdoin.

XML-kieli on keinotekoinen, ihmisten kehittämä symbolipohjainen kieli. XML-kielessä tietoa sisältäviä termejä sanotaan tageiksi. Jokainen tagi erotellaan kielipohjassa, on suurempi- ja pienempi kuin merkeillä <tag>. Tagin loppupiste taas ilmaistaan lisäämällä kauttaviiva termille </tag>. Tagit tunnetaan myös toisella nimellä elementit.

Jos puhutaan elementeistä, viitataan jotakin kokonaisuutta viittaavaan termiin esimerkiksi Network. Tällöin annettaessa tietoa elementtinimelle Network, voidaan tietoa vielä eritellä alielementeille, kuten Item ShortCode ja Point Distance. XML-kielessä tieto annetaan elementeille pääosin jatkuvana tekstinä, kun taas olio-ohjelmoinnissa pyritään tekniseen, lyhyeen ilmaisumuotoon.

Vertaus 1. XML

```
<Para>
Flange is a technical object, which default version we are going to use at this point
with options...
</Para>
```

Vertaus 2. SP3D

```
<Parameter>
```

```
<Parameter Name="Flange" Default="Flange" Options="Flange" />
```

```
</Parameter>
```

Kuten esimerkeistä voi päätellä erot ovat merkkien käytössä ja ilmaisun tiiveydessä.

Riippuen kielelle asetelluista määrittelyistä, on jokaiselle ohjelmointikielelle omat speksinsä. PDS laitosuunnitteluohjelmalla esimerkiksi on eri tagit, kuin SP3D:llä, mistä lisää vertailua luvussa 4.4 ”Valmiiden kokoonpano makrojen käyttö”.

## 4.2 Olio-ohjelmointikielen alkeet

Tässä luvussa käydään lävitse olio-ohjelmoinnin termejä ja niiden merkityksiä ohjelmointikielessä. Luvussa selvitetään myös mistä kieli on saanut alkunsa ja miten se yksilöityy omaksi kielekseen.

Olio-ohjelmointi saa juurensa vuodesta 1967. Tällöin ensimmäinen olio-ohjelmoinnin piirteitä sisältävä ohjelmointikieli SIMULA otettiin käyttöön. Vuodesta -67 on tultu pitkälle ja nykyään menestynein olio-ohjelmointikieli tunnetaan nimellä C++. Olio-ohjelmointi voidaan jakaa kahteen eri osaan, A valmiiden luokkien hyödyntämiseen ja B, uudelleenkäytettävien luokkien tuottamiseen. Puhuttaessa valmiiden luokkien hyödyntämisestä tarkoitetaan luokkakirjaston käyttöä. Tällöin käytetään ohjelmoijan luomia, omiin luokkiinsa jaoteltuja olioita. Kun taas puhutaan uudelleenkäytettävien luokkien tuottamisesta, tarkoitetaan sillä luokkakirjaston rakentamista ja täydentämistä.

Luvussa olio-ohjelmoinnista tullaan osoittamaan, kuinka olio-ohjelmoinnin peruskaava näkyy muun muassa SP3D:ssä ja mitä olio-ohjelmoinnin peruskäsitteillä tarkoitetaan.

Mikä sitten todella on *olio*? Olio sinänsä on hyvin vaikea määrittellä käsitteenä, koska se voi olla lähes mitä tahansa. Se voi olla vaikkapa: Asia, esine, tai symboli. Toimintona se on kuitenkin vain: *'Ohjelman toteutuksen kannalta ajonaikainen keskusmuistista tehty tilanvaraus'*. Olio on käytettävissä heti, kun tiedetään sen käyttötarkoitus ja ulkoinen rajapinta; eli palvelut. (Hietanen 2001)

Oliot on määriteltävä erilaisiin luokkiin. Luokkatyyppi määrittelee olioiden attribuutti-, palvelu- ja metodisisällön. Metodilla tarkoitetaan tässä yhteydessä aliohjelmia. Attribuutilla taas tarkoitetaan tietoa esimerkkinä nimi, ja palvelulla jotain tekoa, josta esimerkkinä voisi olla vaikkapa laskutoiminto. (Hietanen 2001)

*Valmiita luokkia* voidaan hyödyntää kahdella tavalla. Luomalla ja alustamalla olion, tai lähettämällä sille viestejä, joiden perusteella se osaa toteuttaa omat toimintonsa.

Esimerkki 1. SP3D

```
<Instrument PartNumber="3OP14" Secondary="Secondary"
```

```
PART-IJFaceToCenter.Face1toCenter="{F1C}"
```

```
PART-IJFaceToCenter.Face2toCenter="{F2C}" GeometryType="INLINE" />
```

Esimerkissä ohjelmoijan luomalle luokkaolion (3OP14) viestitetään, että sen tulee vaihtaa mittojaan, parametrien FIC ja F2C <Parameter Name>-osion määrittelyjen mukaan. *Parameters* osioon viittaus tiedetään aaltosulkeiden käytöstä kohdissa "{F1C}" ja "{F2C}".

*Kapselointi*, on tapa sisällyttää luokan tietoa hierarkkisessa mallissa. Tällöin on olemassa päätyyppi, jonka alta löytyy sille ominaisia ominaisuuksia. Jos päätyyppimäärittelyllään tarkastusluokaksi, on kapseloinnissa tällöin kyse tarkastusluokkatietojen piilottamisesta tarkastusluokkatietueen alle. Se on kuin kansio, jonka alta löytyy tiedostoja.

Esimerkki 2. SP3D, tarkastusluokka

```
<IJUAIInspectionClass.NJ_Inspection_Class="{NJ_Inspection_Class}"/>
```

Esimerkissä NJ\_Inspection\_Class tietueen parametrit löytyvät viittauksen Default kohdasta ohjelmassa, kun ohjelmoija on ne sinne laatinut.

*Periytymisellä* tarkoitetaan yliluokan perustietojen siirtymistä aliluokkaan. Jos yliluokka on laippa, on aliluokka tällöin esimerkiksi umpilaippa. Tällöin voidaan päätellä, että umpilaippa sisältää ainakin jossain määrin "laipan" tietoja. Umpilaippa siis perii laipan perusominaisuudet.

#### 4.3 Ohjelmoitaessa huomioitavat asiat putkistosuunnittelun yhteydessä

Tässä luvussa käydään lävitse putkistosuunnittelua, johon koko työn ohjelmointiosa keskittyy. Putkistosuunnittelu etenee tietyllä kaavalla, ja tässä osiossa keskitytäänkin eritoten putkistosuunnittelun etenemiseen projekteissa, ja lähinnä sivutaan sen ohjelmointipuolta.

Putkistosuunnittelu on suurin yksittäinen osa-alue laitossuunnitteluprojekteissa. Yritykset uhraavat jopa vuosia suurten teollisuuslaitosten putkistosuunnitteluun. Putkistosuunnittelua tehtäessä tulee aina pitää mielessä seuraavat asiat: Suunniteltavan tilan 3-ulotteiset mitat, käytettävissä olevien tilojen reitit, suunnittelussa käytettävä putkistomateriaali, käytettävän tilan suojausluokitus, putkistoissa virtaava-aine sekä virtaussuunnat.

Putkistosuunnittelussa tehdään aluksi reitityspiirustukset, joiden jälkeen putkistosta tehdään 3D-malli. 3-ulotteisen mallin etuna on aina, että nähdään mihin kohtiin putket voidaan sijoittaa, mistä käsin operaattorit pystyvät käyttämään säätölaitteita, sekä miltä kokonaisuus näyttää lähes oikeannäköisenä mallina.

3D-suunnittelu toteutetaan laitossuunnitteluohjelmilla, joilla tehdyille malleille ajetaan lujuuslaskennat. Esimerkiksi Smart 3D -paketissa käytetään FemDatan

lujuuslaskentaohjelmaa nimeltä FinnSap. FinnSap ohjelman laskelmien perusteella tehdään korjaukset laitossuunnitteluohjelmassa, kuten SP3D, jonka jälkeen tehdään isometrit malleista Isogen-nimisellä ohjelmalla. Tämän jälkeen putkistosuunnittelu etenee laitossuunnittelun tapaan. Eli Isometrit lähetetään Kronodockiin, jossa ne tarkistetaan ja hyväksytään. Tämän jälkeen ne saavat statuksen ”hyväksytty” ja arkistointi lähettää ne eteenpäin, joko tietokantaohjelmaan ProjectWise, tai uudempaan ohjelmaan nimeltä Foundation, joka on niiden lopullinen sijoituspaikka.

Lopuksi kokonaisuus mallista katsotaan ja tarkistetaan mallin näytäntäohjelmalla (NavisWorks) ja siinä havaitut virheet korjataan, ja malli laitetaan toteutukseen.

#### 4.4 Valmiiden kokoonpanomakrojen käyttö

Tässä luvussa käsitellään kokoonpanomakrojen tarkoitusta ja mahdollisuuksia laitossuunnittelussa.

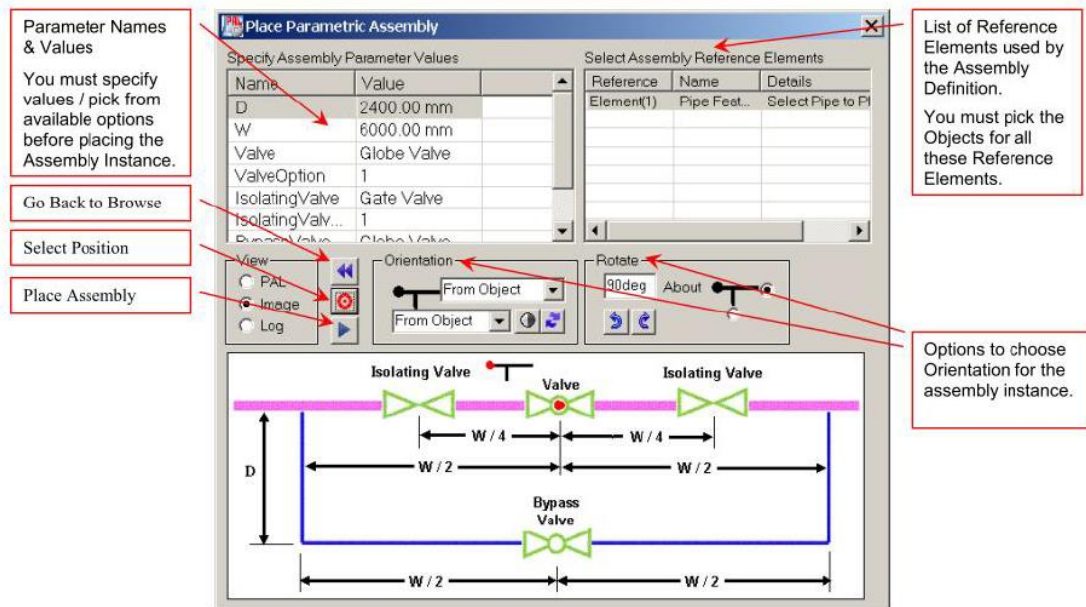
Kokoonpanomakroja tehdään, jotta voidaan nopeuttaa putkistosuunnittelua, ehkäistä suunnittelijoiden tekemiä virheitä sekä estää vääränlaisista liitoksista johtuvat rikkoutuvat linjat. Nopeammalla suunnittelulla säästetään rahaa ja pysytään paremmin projektien aikatauluissa.

NJ teetättää kokoonpanomakroja, koska valmiit esimerkkimakrot, joita on vain muutama, ovat ”raakileita” (eivät toimi NJ:n ympäristössä), ja eivät siis nopeuta työskentelyä millään tavalla.

SP3D:n makrot ovat pitkälle automatisoituja. Jokainen komponentti ja sen vaihtoehdot on ilmoitettu sanallisella nimellä. SP3D:n makrot estävät vääränlaisen kokonaisuuden sijoittamisen linjaan. Olio-pohjaisena kielenä, sen perusteet on suhteellisen helppo omaksua lyhyessäkin ajassa.

Myös SP3D:n makrot ovat huomattavasti parempia kuin esimerkiksi PDS:ssä. Siellä kieli on koodipohjaista sekä komponentteihin viitataan koodinimillä. PDS:ssä on mahdollista tehdä pahimmassa tapauksessa koko laitoksen seisauttavia

putkilinjakokonaisuuksia, ilman suurempia ohjelmallisia varoituksia, toisin kuin SP3D:ssä. PDS:ssäkin on kuitenkin työkalu Verify Data Integrity of a Model, jolla mallin virheitä saadaan huomattavasti karsittua pois. Loput virheet mallista poistetaan Isogenin tekemän loki tiedoston viittauksien mukaan.



Kuva 16. Kokoonpanon sijoitusikkuna. (Intergraph, PAL. 2009)

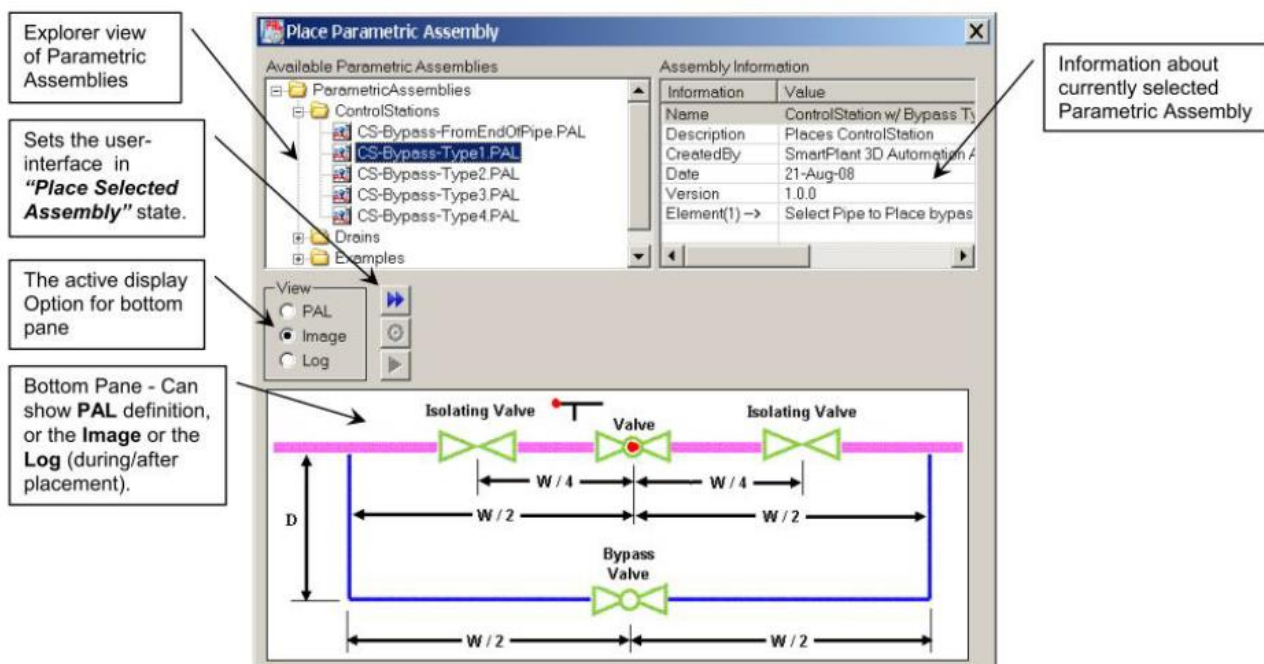
Kuvassa 16 on esitetty kokoonpanon sijoitusikkuna, josta valitaan parametreille, kuten virtaussuunta halutut arvot. Kokoonpanon luoja asettamat oletusarvot näkyvät Value-otsikon alapuolella olevissa kentissä. Kokoonpanon luoja on kuitenkin voinut määrittellä esimerkiksi virtaussuunnalle vaihtoehdoisen pudotusvalikon. Vaihtoehdoisen valikon virtaussuunnalle saa auki klikkaamalla Value-otsikon alta halutun parametrin Flow vastinetta Down. Ohjelmoija on myös voinut tehdä mahdolliseksi Value-arvon kirjoittamisen taulukkoon.

Kuvassa 16 on myös esitetty Select Assembly Reference Elements-taulukko. Kohdassa Reference on määriteltä, mitä mallin olemassa olevaa kokoonpano lukee/mihin se kiinnittyy painettaessa nuolikuviolla varustettua toimintolaatikkoa. Kohdassa Name on määriteltä onko Element(1) putkiosuus, ominaisuus tms. Kohta Details selventää kokoonpanon sisältöä ja rakentumistapaa.

Place Parametric Assembly-sijoitusikkunasta löytyvät vielä osiot View, Orientation sekä Rotate.

View osiosta voi valita ikkunan alaosassa olevan (kuvassa tyhjän) laatikon näkymän. View PAL-kohdan valitsemalla tulee alalaatikkoon esille kokoonpanon ohjelmakoodi. View Image puolestaan näyttää kokoonpanon kuvan jos sellainen on kokoonpanolle tehty. View Log-ikkuna pysyy tyhjänä kunnes kokoonpanon sijoitusikkunasta valitaan nuolella varustettu kokoonpanon rakennustoiminnon aloittava kohta. Toiminnon alkaessa rakentaa kokoonpanoa malliin alkaa ohjelma kirjata rakennustietoja Log-laatikkoon. Kokoonpanoa sijoituksen virheet ja onnistuneet tiedot tulevat näkyville Logiin. Jos koodissa on pahoja virheitä, keskeyttää ohjelma kokoonpanon teon ja Log-ikkunasta löytyy yleensä jonkinlainen viittaus epäkohtaan. Valitettavasti usein epäkohtien viittaukset ovat kuitenkin erittäin vajavaisia.

*Orientation ja Rotate* osiot ovat kokoonpanon sijoitussuunnan ja kulman asettamiseen. Hyvä ohjelmoija koodaa kuitenkin ohjelman niin, ettei kyseisiin toimintoihin tarvitse puuttua ollenkaan.



Kuva 17. Kokoonpanon sijoittamisen ohjeistusta. (Intergraph, PAL. 2009)



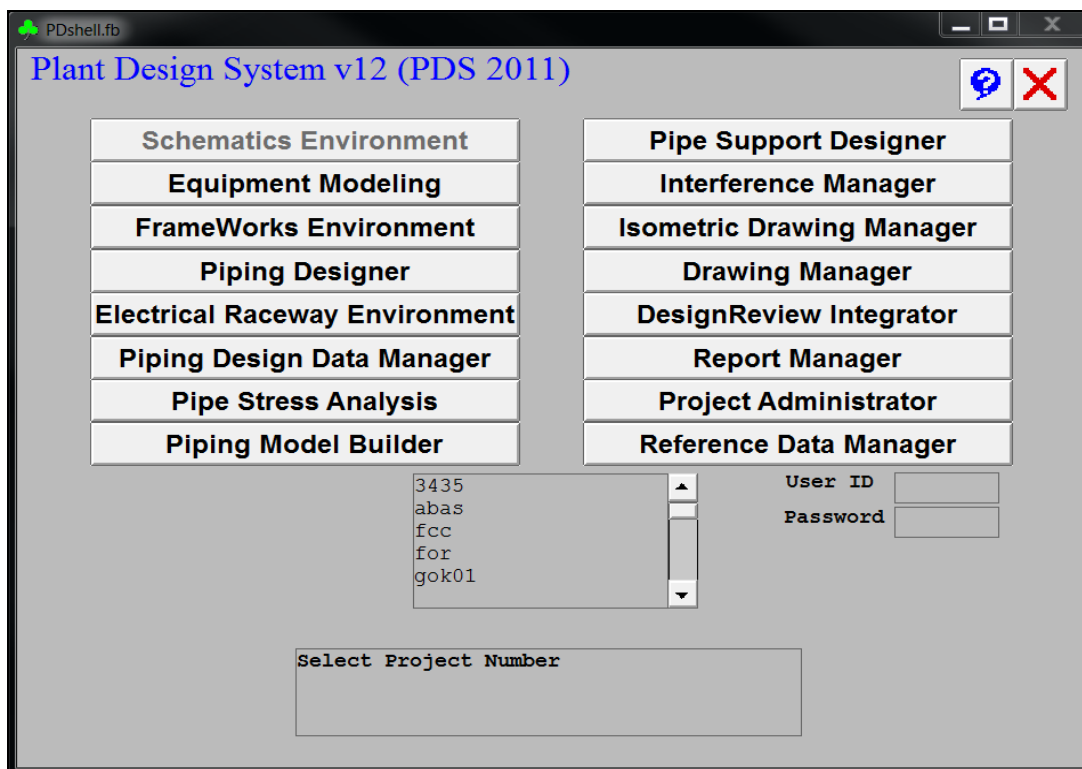
## 5 PARAMETRINEN OHJELMOINTI PLANT DESIGN SYSTEM -OHJELMASSA

Tässä luvussa käydään lävitse PDS:n ilmanpoistolle tehtyä ohjelmaa. Esimerkillä ilmanpoistosta annetaan mahdollisuus perehtyä PDS:n SP3D:stä poikkeavaan kieleen sekä kielten eroavaisuuksiin ohjelmoinnissa. Luvun tarkoitus on laajentaa ymmärrystä ohjelmointikielten käytöstä ja automatisoivien makrojen tekotavoista.

Plant Design System on laitossuunnitteluohjelma, joka hakee tietokannasta valmiiksi määritellyt komponentit ja spesifikaatiot. PDS-ohjelmassa ohjelmointi PAL-kielellä on kuitenkin alkeellisempaa ja matemaattisempaa, kuin nykyään ilmestyvissä uusissa laitossuunnitteluohjelmissa. Seuraavassa on kerrottu tarkemmin tavoista, joilla PAL- ohjelmointia tehdään PDS:ssä. PDS:ssä on samankaltainen ohjelmointipuu, kuin SP3D:ssä. Ohjelmakoodin luonnin jokaiselle osiolle on siis oma otsikko, joiden avulla ohjelmakoodin tekoa seuraavassa esitellään.

### 5.1 PDS-laitossuunnitteluohjelma

PDS on yksi vanhimpia ja sitä kautta käytetyimpiä laitossuunnitteluohjelmia. PDS:n kehittäjä on Intergraph. PDS käyttää grafiikkamoottorinaan ohjelmaa Microstation, joten PDS-ohjelman käyttö edellyttää myös MicroStation-lisenssiä. PDS:n valikot ja toimintatavat ovat hyvin alkeellisen tuntuisia, mutta loppujen lopuksi toiminnoiltaan erittäin käytännöllisiä. PDS:n automatisoinnin ehdotonta kärkeä ovat kokoonpanomakrot, joiden avulla koodatut putkilinjakokonaisuudet voidaan sijoittaa malliin hyvinkin pienin toimenpitein. PDS jakautuu pääosin kuuteentoista moduuliin, jotka näkyvät kuvassa 18.

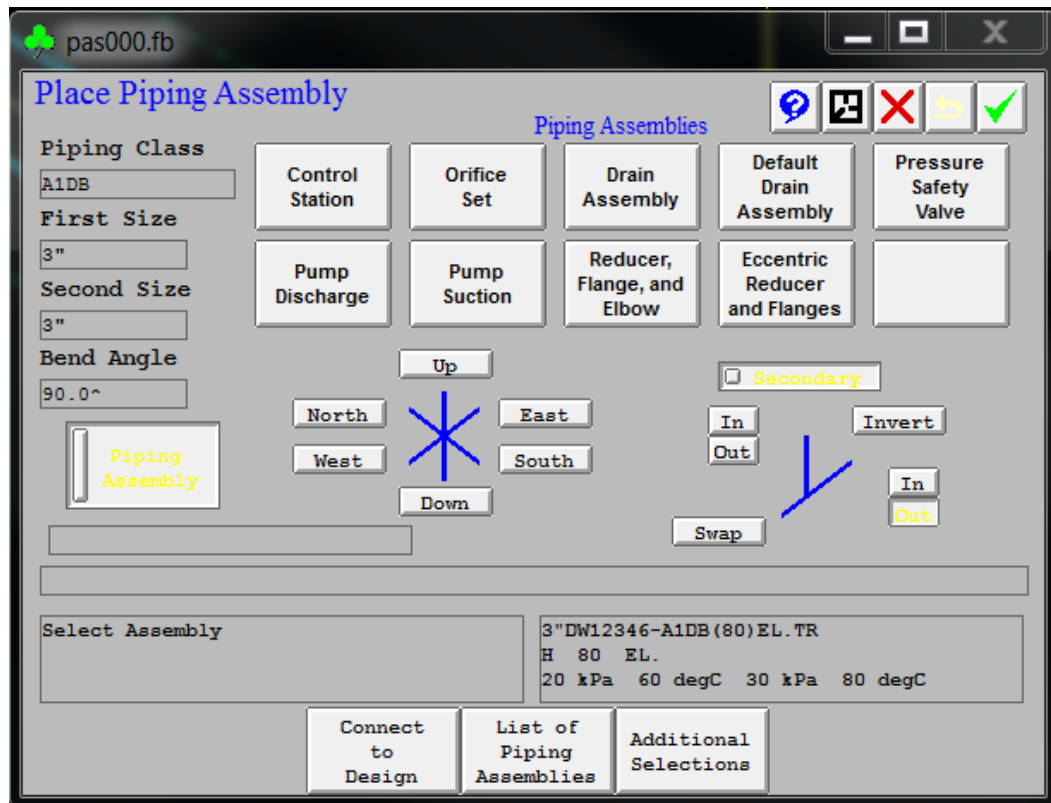


Kuva 18. PDS-ohjelman moduulit.

Moduulit käsittelevät niille rajattuja suunnittelualueita ja alueille valittuja työkaluja. Putkistosuunnittelu on PDS:n käytetyin toimintamoduuli. PDS:ssä putkistosuunnittelu aloitetaan sketsaamalla segmenttireititys, johon putkistokomponentit ja putket voidaan lisätä. Kaikki segmentin kiinnityspisteet pohjautuvat laitoksille määritettyihin grid-tasoviivastoihin. Suunnittelutyön tekeminen tällä ohjelmalla on hyvinkin tarkkaavaisuutta vaativaa ja sääntöjä seuraavaa. Ohjelman putkistosuunnitteluosiossa ei ole undo-toimintoa. Osittain tämän undo-toiminnon puuttumisen takia ohjelman kokoonpano-osio on laajennettu yli sataan assemblyyn, joiden käyttö vähentää huomattavasti aloittavien suunnittelijoiden tekemiä virheitä sekä nopeuttaa putkistosuunnittelutyötä yleensäkin. Toinen syy on tietenkin ajallinen voitto.

## 5.2 Parametrinen ohjelmointi

PDS:lle luotu kokoonpanokirjasto mahdollistaa komponenttiryhmiä nopean määrittelyn (parametroinnin) ja sijoittamisen malliin.



Kuva 19. Kokoonpanon sijoituslomake eli formi.

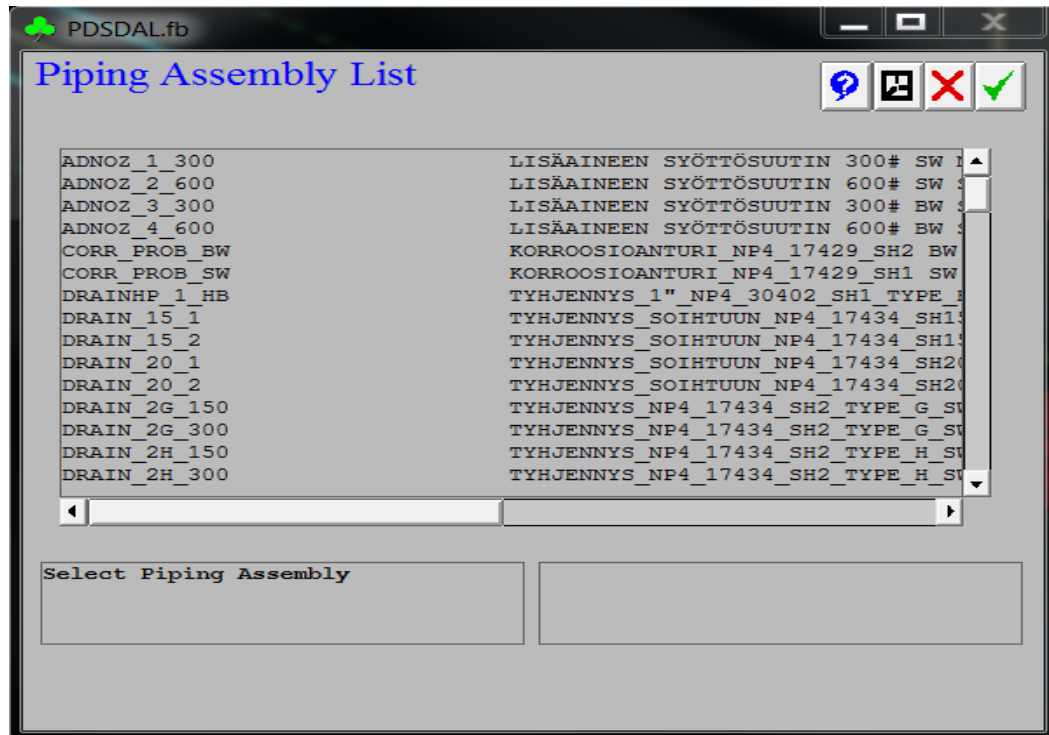
Kokoonpanojen rakennus on PDS:ssä aivan oma maailmansa verrattaessa esimerkiksi SP3D:hen. PDS:n käyttämä vahvasti XML-taustainen ohjelmointikieli on juuri sellaista, kuin 80-luvun ohjelmointi oli yleensä, yksinkertaista ja tikapuumaista. PDS:ssä kokoonpanot ohjelmoidaan tekstitiedostoina niin, että rakentaessa putkilinjaa komponentit ohjelmoidaan ohjelmaan täsmälleen siinä järjestyksessä, kuin ne putkea pitkin kuljettaessa tulisivat vastaan.

PDS:n ohjelmointia on käyty pääpiirteittäin läpi seuraavan ilmanpoisto esimerkin käsittelyn avulla. Ohjelma on esitetty kokonaisuudessaan liitteessä 2.

Ilmanpoisto

```
# BW_ILMANPOISTO_NP4_17433_SH2_TYPE_UVA
```

Tämä kokoonpano asettaa ilmanpoiston, joka on nimetty NJ:n spesifikaation mukaisesti. #-merkillä voidaan viitata kokoonpanokirjastosta löytyvään kokoonpanoon. Tällaisesta kirjastosta on esimerkki kuvassa 20.



Kuva 20. Luotu kokoonpanokirjasto.

*Ohjelmakoodi:*

PAL 'VEN\_TYP\_A\_BW'

DISPLAY\_TUTORIAL = 'PAS000'

Kun kokoonpanosta halutaan tietää sen sijoittamiseen liittyviä tietoja, kehoitetaan ohjelmaa näyttämään (display) tietoja ilmanpoistoon liittyen.

SECOND\_SIZE = 0.75

Avainsanoilla SECOND\_SIZE kerrotaan ohjelmalle, että putkiosuuden halkaisija katsotaan putkimateriaaliluokkataulukosta, jolloin arvoksi tulee tässä tapauksessa 0.75 tuumaa.

```
PLACE BRANCH , 90_DEG , BY CENTER
```

Tämä komento etsii 90-asteen haaroitukselle oikeantyyppisen komponentin haaroitustaulukosta putkiosuuden tulevan halkaisijan ja lähtevän vastineen perusteella. Jos haaroitus on putkisegmentillä tai vapaassa avaruudessa tulisi käyttää komentoa By Center.

```
IF ( NUMBER_CPS .EQ. 2 ) THEN
```

```
REF_PNT_A = CP2
```

```
ELSE
```

```
REF_PNT_A = CP3
```

```
ENDIF
```

Tämän IF-lauseen merkitys on seuraava: Jos liitäntäpisteiden määrä on tasan kaksi tällöin referenssipisteenä A käytetään liitäntäpistettä 2. Muutoin referenssipisteenä A käytetään liitäntäpistettä 3.

```
CONNECT TO REF_PNT_A
```

Tässä määritellään aiemmin tehty kokoonpanon osa liitettäväksi referenssipisteeseen A.

```
PROMPT_MESSAGE = 'ENTER NEW PIPING O-SPEC'
```

Seuraavaksi määritellään ja näytetään käyttäjälle kokoonpanon sijoitukseen liittyvä ohje.

```
DISPLAY = PROMPT_MESSAGE
```

```
MATERIALS_CLASS_B = USER_INPUT
```

Joskus putkiston materiaaliluokkaa joudutaan vaihtamaan, esimerkiksi tässä on annettu materiaaliluokan valinta käyttäjän määritettäväksi (User Input). Määrittelyssä tulee kuitenkin noudattaa PROMPT-ohjeen rajausta.

```
ACTIVE_MATERIALS_CLASS = MATERIALS_CLASS_B
```

Nyt kun materiaaliluokka B on tarkemmin määritelty voidaan se aktivoida käyttöön komennolla Active.

```
PLACE FITTING , 6Q2C01 , BY CP2
```

Komennolla PLACE FITTING käsketään ohjelmaa sijoittamaan 6Q2C01 tyyppinen sovite malliin, liitäntäpisteestään kaksi.

```
T34C46 = 'SEE_NP4_17033_SH_2_TYPE_A'
```

Komennolla T34C# voidaan esiasetella tiettyjä arvoja sijoitettavalle komponentille. Nyt #-merkin jälkeinen numero on 46, ja se viittaa sarakenumeroon putkikomponenttidataaulukossa.

```
PLACE FITTING , 6Q2C08 , BY CP1
```

(kts. edellinen PLACE FITTING-toiminto)

```
END
```

### Kokoonpanon nimeäminen

PDS:ssä yleinen ohjeistus on nimetä kokoonpanot käyttäjäystävällisellä tavalla. Tällainen tapa on antaa kokoonpanolle päänimike esim. tyhjennys, joka on suunnittelijalle/kokoonpanon käyttäjälle tuttu. Kokoonpanon loppuosa nimetään yleensä kokoonpanolle tehdyn tyyppikuvan tunnuksella. Tähän kaikkeen on käytössä yhteensä 100 merkkiä.

### Kokonaisuuksien rakentaminen

PDS:sän ohjelmointi on XML-pohjaista, kuten jo aikaisemmin todettiin. Tällöin, kun ohjelmalla halutaan luoda jotain, tulee se tehdä antamalla yksinkertaisia käskymuotoisia komentoja. Tällaisia komentoja ovat PDS:n "varatut sanat", kuten

CONNECT ja PLACE. Kokonaisuuksien rakentaminen onkin pitkälle avainsanojen ja niihin liittyvien tarkenteiden laittamista niiden esiintymisjärjestykseen.

### Putkiosuuden rakentaminen

Putkiosuuden rakentaminen PDS:ssä tehdään hyvin samalla tavalla kuin SP3D:n Pipe Length -toimintoa käytettäessä. PDS:n avainsanavastineet ovat "MOVE SPECIFIED DISTANCE". Tässä toimintatavassa ei kuitenkaan voida määrittää suuntaa, vaan se määräytyy primääriakselin mukaan.

### Yksityiskohtaisempi ominaisuuksien määrittely

SmartPlant 3D:ssä tarkemmat ominaisuudet määritellään valitsemalla joko osa, putkiosuus tai muu vastaavaa elementti. Plant Design Systemissä tarkemmat määrittelyt tehdään samalla tavalla, eli lataamalla tiedot tietokannasta juuri kyseiselle osaoliolle. PDS:ssä vain määrittely SetAttributesOnRunItems korvataan sanoilla LOAD SPEC DATA. Tämän jälkeen voidaan antaa samanlaiset määrittelyt kuten TERM\_TYPE\_1 TAI PR\_RATING\_1.

### Suuntien ja operointisuuntien määrittely

Laitteiden ja putkistojen suunnat voidaan määrittää mm. primääri- ja sekundääriakselien mukaan, avaruuspisteiden mukaan tai edellisten komponenttien mukaan.

### Muuta

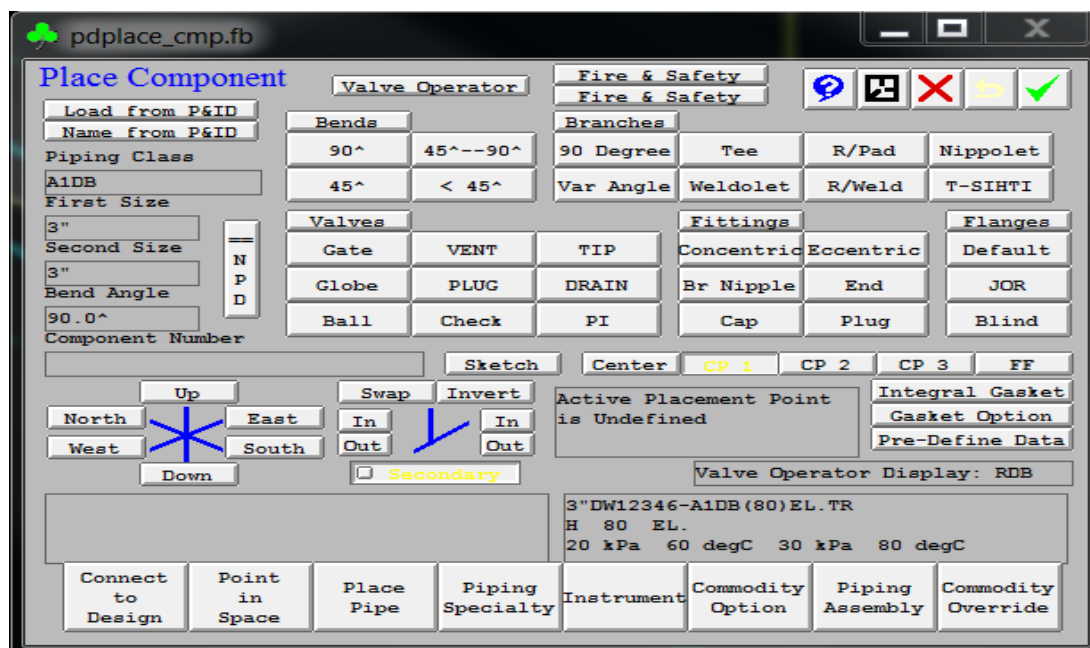
Putkistosuunnittelun kannalta SP3D- ja PDS-ohjelmissa suurin ero liittyy ohjelmien automatisointikykyihin. Periaatteessa PDS pystyy monella tapaa tekemään samoja asioita kuin SP3D. SP3D on kuitenkin käyttöliittymältään ymmärrettävämpi ja suurissa projekteissa huomattavasti nopeampi käyttää. SP3D:n Undo-toiminto, hiukan pidemmälle viedyt parametointi- sekä määrittelymahdollisuudet, helpompi putkistosuunnittelutapa ja graafisesti hienompi käyttöliittymä osoittavat, miksi

SP3D:hen ollaan siirtymässä nyt ja tulevaisuudessa. Suurin syy sille miksi PDS on olemassa, on se, että se on käytössä niin monessa paikassa, integroituna niin moneen järjestelmään ja sen tiedon siirto SP3D:hen (vaikka tämä on saman yhtiön tuote) on erittäin vaikeaa. Jotta näiden ohjelmien toimintoja voidaan paremmin verrata toisiinsa, käydään seuraavassa vielä lävitse PDS:n kokoonpanojen sijoittamista malliin.

### 5.3 PDS-kokoonpanon sijoittaminen malliin

PDS-ohjelmassa kokoonpanon sijoitusikkunaa, kuten muitakin ikkunoita, kutsutaan formeiksi. Jokaiselle formille on määritelty ohjelmassa oletusmallit. Ohjelman hieno piirre on se, että formien valinta ikkunoita voi kuitenkin määrittellä myös itse. Kyseinen toiminto on nimeltään Formbuilder.

Kuvassa 21 on muun muassa valittavina putkiluokka-, koko- ja taivutusparametrejä.

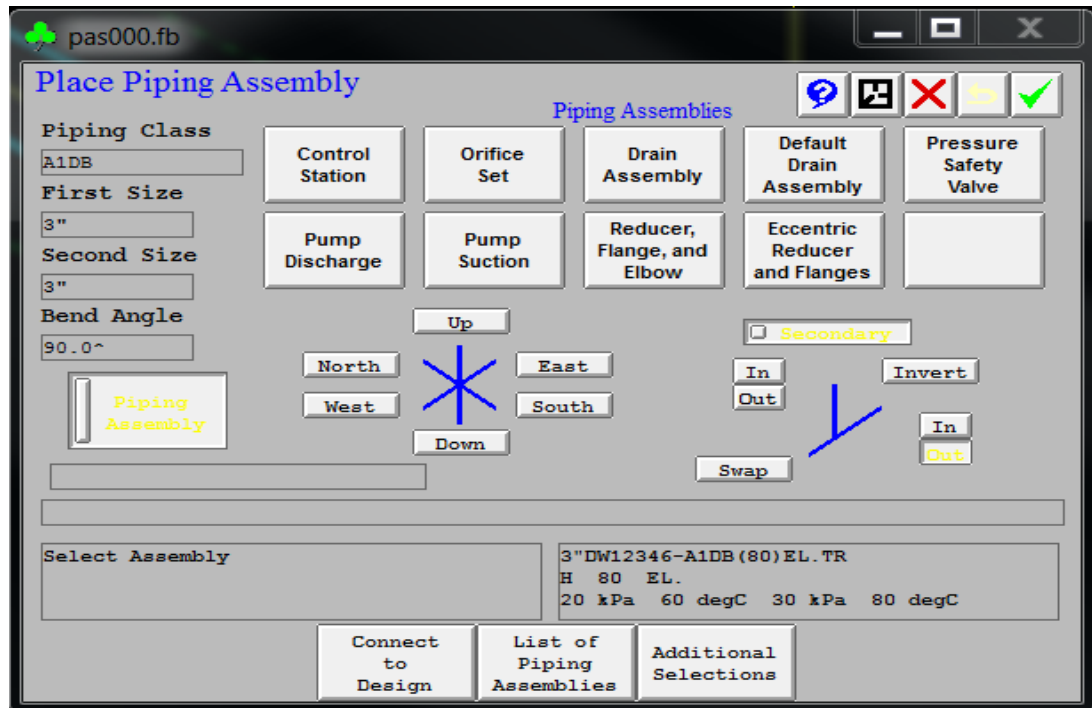


Kuva 21. Parametroitavia asioita.

PDS:sän kokoonpanon sijoittaminen aloitetaan valitsemalla Place Component. Toisin kuin SP3D:ssä, jossa kokoonpanot ovat ihan oma automatisointityökalunsa, PDS:ssä kokoonpanomakrot löytyvät samasta formista yksittäisten komponenttien kanssa.



Place Component -ikkunassa valitaan Piping Assembly. Tällöin aukeaa ikkuna Place Piping Assembly, josta pääsee valitsemaan kuten SP3D:n kokoonpanokirjastosta, millainen kokoonpano on kyseessä (kuva alla).



Kuva 22. Kokoonpanon sijoitusikkuna PDS.

Tässä ikkunassa valittavia parametrejä ovat mm. edellä esitetyt sijoitussuunta, putkiluokka, putkikoko tms. Valittaessa esimerkiksi List of Piping Assemblies-toiminto, päästään kuvan 20 mukaiseen näkymään.

Itse kokoonpanon sijoittaminen tapahtuu valitsemalla Connect to Design.

Kokonaisuutena PDS:n kokoonpanon luontitoiminto on varsin toimiva, ja siksi siihen on joissain yrityksissä luotu useita satoja tällaisia sijoitettavia kokoonpanoja.

PDS on siitä erikoinen ohjelma, että se rakentaa kappaleet täsmälleen siinä järjestyksessä, kuin ne esiintyvät ohjelmakoodissa. Eli kaikki putket, sovitteet ja liitokset on määriteltävä siinä esiintymisjärjestyksessä missä ne tulevat peräkkäin esim. putkilinjaan lisättävän kokoonpanon kohdalla.

## 6 PARAMETRISEN OHJELMOINNIN PERUSTEET SMARTPLANT 3D:SSÄ

Tässä luvussa annetaan valmiudet kirjoittaa PAL-ohjelmakoodia sekä tutustutaan ohjelmoinnin kirjoitussääntöihin.

PAL-ohjelmointi on parametreihin perustuvan kielen sääntöjen noudattamista. PAL-ohjelmoinnissa SP3D:ssä tehdään ensin tietokanta, jonka jälkeen siihen viitataan. Viittaukset mallissa käytettävien objektien tietueisiin tehdään esimerkiksi ohjelmalla Notepad++, kirjoittamalla tietueelle tietokannasta varattu ”koodinimi”, esimerkiksi ”Gate Valve”, oikeassa ulkoasussa, oikeaan kohtaan. SP3D:llä on omat sääntönsä, samoin kuin PDS:llä. Tässä luvussa on esitetty PAL tunnisteen yleisimpiä määritelmiä.

SmartPlant 3D:ssä parametrisiä kokoonpanoja hallinnoiva vastuhenkilö määrittelee ohjelman automatisointityökaluun, minkälaisia kokoonpanoja on saatavilla, millä parametreillä ne on toteutettu ja miten ne on organisoitu standardikansioon. Parametriset kokoonpanot tallennetaan tekstitiedostoina laitoksen systeemihakemistoon (SharedContent, Alias\_SymbolShare).

Parametrinen ohjelmointi on mahdollistettu kaikille käyttäjäoikeudet omaaville, parametriseen ohjelmoinnin lisätyökalun hankkineille käyttäjille. Kaikkia ohjelmoituja kokoonpanoja voi muokata malleissa täysin samalla tavalla, kuin perinteistä 3D-putkistosuunnittelua tehtäessä.

SP3D:n PAL-kielisessä ohjelmoinnissa ”varatuilla sanoilla” muodostetaan hierarkiapuu, jossa kokoonpanoa käsittelevät asiat otetaan esille ohjelmoijan haluamassa järjestyksessä.

Alla on esitetty perinteinen ”varattujen sanojen” mukainen hierarkiapuu.

```
<ParametricAssemblyDefinition>
```

```
</Information Name>
```

```
<ReferenceElements>
```

```
<Element Prompt/>
```

```
</ReferenceElements>
```

```

<GiFFile Name>
<Parameters>
</Parameters>
<Network>
</Network>
<Items>
</Items>
<Overrides>
</Overrides>
</ParametricAssemblyDefinition>

```

Hierarkiapuussa osioita voi järjestellä eri järjestykseen, mutta ohjelman suunnittelijan mielestä loogisin ratkaisu on oletusmallin mukainen kokoonpanon rakentaminen. Tässä perinteisessä mallissa kokoonpanon ohjelman kirjoittaminen lähtee liikkeelle kokoonpanon nimeämisestä selkokiekisellä käyttäjäystävällisellä nimellä.

Seuraavassa on havainnollistettu ohjelman kirjoitusprosessi putkistosuunnittelulle tärkeimpien kontekstien osalta.

## 6.1 Kokoonpanon nimeäminen

Kokoonpanon ohjelmointi aloitetaan yleensä kokoonpanon nimeämisellä tyyppikuvan mukaisella nimellä ja toimintaa kuvaavalla tarkenteella. Tekstieditorissa ohjelman aloitus näyttäisi seuraavanlaiselta.

Esimerkki 1. Kokoonpanon nimeäminen.

```

<ParametricAssemblyDefinition>
<Information Name="Vent - Pipe + Valve + ClosingItem"
Description="Piping Vent Assembly with Closing Item" CreatedBy="Ville Kerttula"
Date="7-Apr-2014" Version="Revision1"/>
</ParametricAssemblyDefinition>

```

```
<Information Name="Orifice Plate assembly with Orifice Flanges"
Description="Orifice Plate assembly with Orifice Flanges" CreatedBy="Ville Kerttula"
Date="20-Apr-2014" Version="Revision1"/>
```

Esimerkissä nimetyt kokoonpanot käsittelevät tyhjennystä ja virtausmittausta.

Esimerkin kenttänimike Information Name tarkoittaa kokoonpanon nimeä, joka tulee näkyville kokoonpanon sijoitusikkunassa. Description taas tarkoittaa edellä mainittua tarkennetta, jonka tulisi kertoa kokoonpanon käyttäjälle, miten kokoonpano toimii ja mitä se sisältää. Loput ohjelmakoodin termit, kuten CreatedBy, Date sekä Version selventävät kokoonpanon käyttäjälle, kuka kokoonpanon on luonut sekä milloin ja montako kertaa kyseisen kokoonpanon ohjelmakoodia on muokattu.

## 6.2 Kokoonpanon sijoittamismahdollisuuksien rajaaminen

Joissain tapauksissa kokoonpanojen kiinnittymispiste halutaan rajata tarkemmin. Näin saadaan karsittua pois inhimillisen vääränlaisen sijoittamisen mahdollisuus. Rajaamalla kokoonpanon sijoitusmahdollisuuksia voidaan jo ohjelman alkuvaiheessa määritellä, miten ohjelmaa lähdetään rakentamaan. Jos kokoonpano on esimerkiksi lauhdutuslinjaan liittyvä, tulee kaikki muut liitosmahdollisuudet rajata pois inhimillisen sijoitusvirheen välttämiseksi. Seuraava esimerkki valaisee asiaa.

Sijoittamismahdollisuuksien rajaamistoiminto esiintyy ohjelmakoodissa nimellä Reference Elements. Tällä nimellä viitataan referenssipisteeseen johon kokoonpano halutaan kiinnittää, tai josta se halutaan laittaa rakentumaan. Alla lisää havainnollistuksia toiminnon käyttämisestä.

...

```
<ReferenceElements>
```

```
<Element Prompt="Select Pipe Feature / Pipe Part to Place Vent "/>
```

```
</ReferenceElements>
```

...

```
<Element Prompt="Select Pipe / End to Place Orifice Assembly"
Type="JRtePipePathFeat"/>
```

Yllä käytetyllä ohjelmointiin varatulla sanalla JRtePipePathFeat estetään kokoonpanon sijoitus muualle kuin putkilinjalle.

### 6.3 Kokoonpanon mallikuvan luominen

Kokoonpanon graafinen malli on sijoittajalle informatiivinen kuvanto kokoonpanon ulkoasusta, sisällöstä sekä mahdollisesti kokeneelle käyttäjälle myös lisäinfo kokoonpanon toimintaa ajatellen. Kokeneelle käyttäjälle mallikuva kertoo mm. mitä laitteita, komponentteja ja putkia se sisältää, minkä tyyppisiä ne ovat ja minkälaisilla etäisyyksillä elementit sijaitsevat toisiinsa nähden. Mallikuvasta näkee myös laitteiden ja venttiilien operointisuunnat ja kiertomahdollisuudet.

Kokoonpanon mallikuva tulee kokoonpanon sijoittajalle esille valitsemalla Place Parametric Assembly sijoitusikkunan kohdan View Image. Mallikuvan tulee olla kuvatiedostotyyppiä (esim. jpg., png. tai gif) ja se tulee tallentaa sitä koskevan kokoonpanon kanssa samaan tiedostokansioon. Jos näin ei, tehdä, malli ei tule näkymään sijoitusikkunassa.

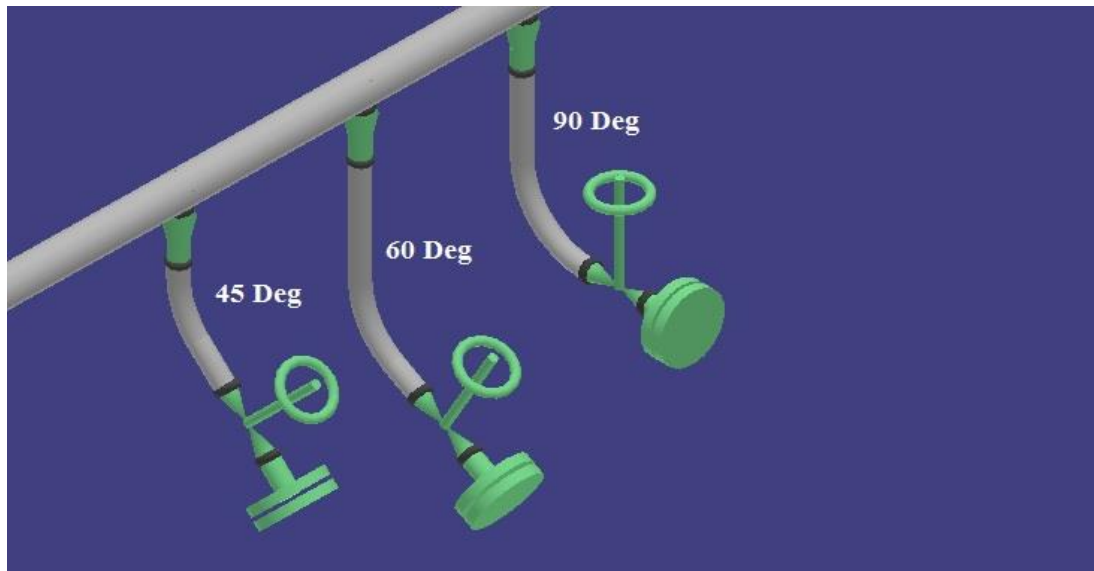
Ohjelmakoodissa kokoonpanon mallikuvan paikan määrittely sijoittuu referenssipisteen jälkeen.

Esimerkki 2. Mallikuva.

```
<GIFFile Name="Vent.gif"/>
<GIFFile Name="NJ_NP4-21635.jpg"/>
```

Esimerkissä 2 GIFFile name kuvastaa kuvatiedostomuodolle "varattua sanaa". Lainauksien sisällä oleva osa kannattaa nimetä kokoonpanon tyyppikuvan mukaan, kuten itse kokoonpanokin. Esimerkin 3 tiedostopäätteet voivat olla mitä tahansa ohjelman tunnistavaa kuvatiedostotyyppiä.

Kokoonpanon mallikuva kannattaa ottaa vasta valmiista kokoonpanosta, tällöin se kuvastaa parhaiten kokoonpanon grafiikkaa. Mallikuvan voi kuitenkin tehdä myös peruspiirto-ohjelmalla. Jotta mallikuva olisi mahdollisimman hyödyllinen, kannattaa siihen lisätä omia merkintöjä kuvankäsittelyohjelmalla, tästä on havainnollistus kuvassa 23.



*Kuva 23.* Mallikuvan täydennys kuvankäsittelyohjelmalla.

Kuvan 23 mallikuva havainnollistaa minkälaisena kokoonpano muodostuu eri taivutuskulmissa.

#### 6.4 Parametrien määrittely

Parametrit ovat kokoonpanon tekstimuotoisesti tai numeerisesti määriteltäviä arvoja, jotka kuvaavat selkokielellä kokoonpanon toimintaa ja sen sisältämiä elementtejä.

Ohjelmakoodin parametrejä käsittelevässä osassa määritellään parametriä sijoittajalleen kuvaava nimi, parametrin oletusarvo, oletusarvonyksikkö oletusarvon vaihtoehdot sekä mahdollisesti myös pakotettu parametrin arvo (value). Parametrit osiossa voidaan määritellä myös parametrin arvon ”lukeminen” mallissa olevasta

osasta. Tällöin määrittely on esimerkin 3 tummennettujen kohtien mukainen. Referenssipisteen luvusta on vielä tarkemmin seuraavassa esimerkissä.

Esimerkki 3. Parametrin arvon lukeminen referenssi elementistä.

```
<ParameterFromReferenceElement Name="NPD$" Value="NominalDiameter$@Element(1)"/>
```

Esimerkissä 3 on luettu putkilinjaosuuden halkaisijan yksikköjärjestelmän mukainen lukuarvo (tuuma) ja käytetty erikoismerkkejä, jotka muuttavat luettavan parametrin arvon yksikön numeeriseksi. Täytäntötyypiseen merkintään on käytetty dollarin merkkiä. Alla on esitetty kokoonpano kokonaisuus, johon parametriarvon lukeminen sisältyy.

Esimerkki 4. Säätoventtiiliryhmän parametrit osion määrittelyjä.

```
<Parameters>
<Parameter Name="D" Default="1300mm" UnitType="DISTANCE"
Options="1.0m|2.0m|3.0m"/>
<Parameter Name="Width" Default="800mm" UnitType="DISTANCE"/>
<Parameter Name="D1" Default="300mm" UnitType="DISTANCE"
Options="250mm|100mm"/>
<Parameter Name="ControlValve" Default="IPDA24"/>
<Parameter Name="L1" Default="100MM" UnitType="DISTANCE"/>
<Parameter Name="NJ_Inspection_Class" Default=":10" />
<Parameter Name="Flow" Default="DOWN"
Options="Up|Down|Both|No|Undefined"/>
<ParameterFromReferenceElement Name="NPD"
Value="NominalDiameter@Element(1)"/>
<Parameter Name="ValveOption" Default="Default" Options="Default|Bellows"/>
<Parameter Name="GateValve" Default="Gate Valve"/>
<Parameter Name="GateValveOption" Default="Default" Options="Default|300"/>
```

```

<Parameter Name="BypassValve" Default="Globe Valve"
Options="Globe Valve|Ball Valve"/>
<Parameter Name="BypassValveOption" Default="Default" Options="Default|300"/>
<Parameter Name="Branch" Default="Nipolet" Options="Default|Weldolet"/>
<Parameter Name="ClosingItem" Default="Blind Flange" Options="Blind Flange"/>
<Parameter Name="ClosingItemOption" Default="Default" Options="300|+300"/>
<Parameter Name="H" Default="152mm" UnitType="DISTANCE"/>
<Parameter Name="ControlValveOperatorDirection" Default="WEST"
Options="EAST|WEST|NORTH|SOUTH"/>
<Parameter Name="DrainValveOperatorDirection" Default="EAST"
Options="EAST|WEST|NORTH|SOUTH"/>
<Parameter Name="F1C" Default="50mm" UnitType="DISTANCE"
Options="120mm|200mm"/>
<Parameter Name="F2C" Default="50mm" UnitType="DISTANCE"
Options="160mm|250mm"/>
<Parameter Name="RotationDirection" Default="Normal"
Options="Secondary|Primary|Normal|Reverse_Secondary|Reverse_Primary|
Reverse_Normal"/>
<ParameterFromReferenceElement Name="NPD$"
Value="NominalDiameter$@Element(1)"/>
<ConstantParameter Name="TWO_INCH" Value="2in" UnitType="NPD"/>
</Parameters>

```

Parametrien määrittely osion jälkeen hierarkiapuussa on vuorossa määriteltyjen parametrien käytön rajaaminen, toisin sanoen ns. validointi.

## 6.5 Parametrien validointi

Validointi-toiminnolla tarkoitetaan mahdollisuutta määritellä missä tilanteissa validointiosion väliin jäävät koodirivit toteutetaan ohjelmassa. Validointi-toiminto toimii myös käänteisesti. Valid If -toiminnolla validointiarvon vastatessa Valid-arvoa



toiminto toteutetaan. Seuraavassa on esimerkki Valid If "varattujen sanojen" käytöstä säätöventtiiliryhmän ohjelmakoodin osana.

Esimerkki 5. Validointi säätöventtiiliryhmän toteutuksen rajaajana.

```
<Parameters>
```

```
...
```

```
<ParameterFromReferenceElement Name="NPD$"
Value="NominalDiameter$@Element(1)"/>
```

```
</Parameters>
```

```
<ValidateParameters>
```

```
<Valid if='StringHasSubString("#2in#3in#4in#6in#", "[NPD$]#")'
```

```
Description="NPD must be 2, 3, 4 or 6 inches"/>
```

```
</ValidateParameters>
```

```
<Network>
```

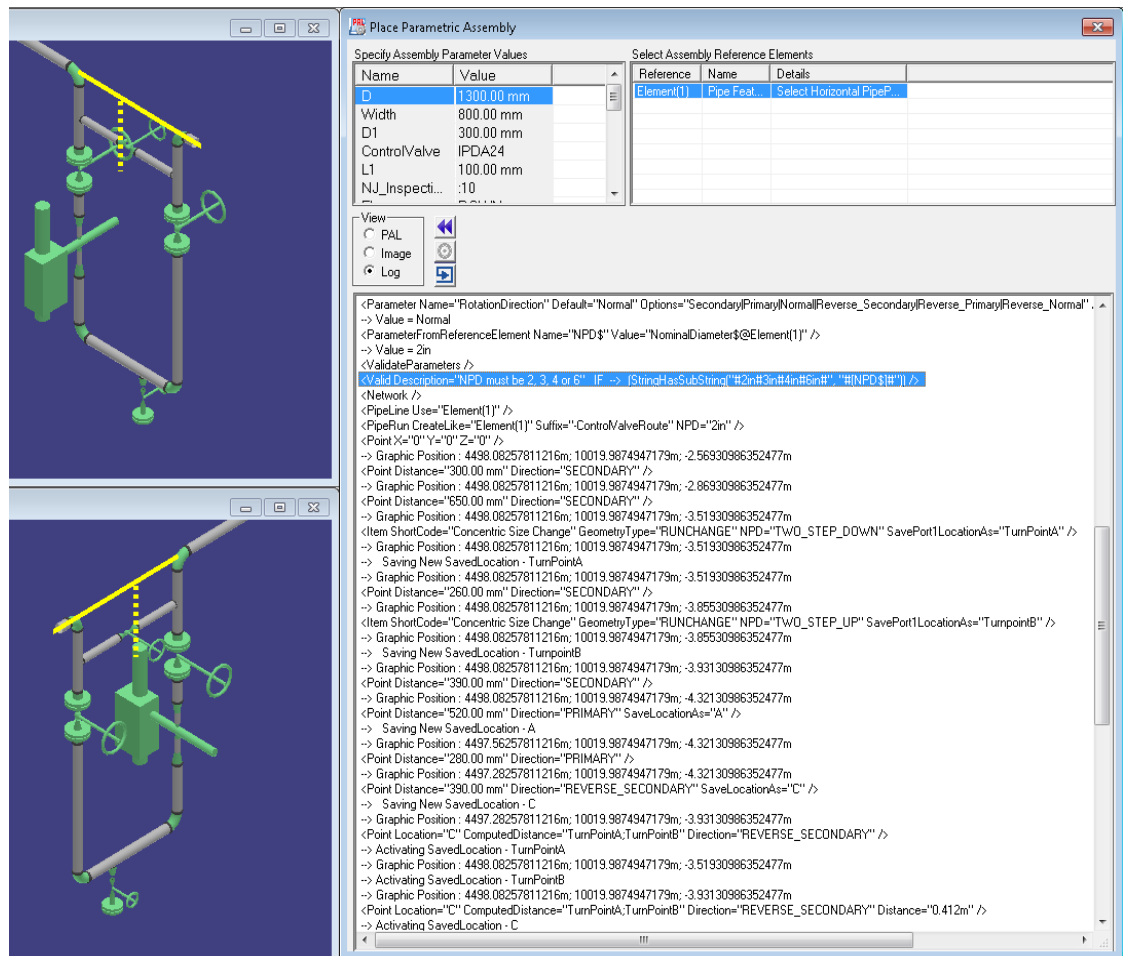
```
<PipeLine Use="Element(1)"/>
```

```
<PipeRun CreateLike="Element(1)" Suffix="-ControlValveRoute" NPD="{NPD$}"/>
```

```
...
```

Esimerkissä 5 StringHasSubString komennolla tarkoitetaan seuraavassa: Jos parametriosiosta löytyy tekstuaalinen ilmaisu 2in, 3in jne. tällöin parametrit-osion oletusarvot voidaan toteuttaa putkihalkaisijan käytön osalta. Sulkeissa käytetty ilmaisu *#[NPD\$]#*, tarkoittaa halkaisijan tekstuaalisen muodon lukua osiosta. Vaihtoehtoiset arvot ilmaistaan ohjelmakoodissa #-merkein rajaamalla.

Konkreettisesti validointi tapahtuu ohjelman luodessa kokoonpanoa. Ohjelman validointia voi seurata sijoitusikkunan Log view näkymässä. Alla on esitetty tilannetta hyvin havainnollistava kuva.



Kuva 24. Säätoventtiiliryhmän validointi sijoitusikkunassa.

Koska validoinnin voi tehdä monella tavalla, on alla esitetty vielä muutama validoinnin muoto.

Esimerkki 6. Venttiili tehdään jos referenssiputkenhalkaisija on 2 tuumaa.

<Network>

...

<Item IF="{NPD}" .EQ. "2in" ShortCode="Gate Valve"/>

...

Esimerkin 6 ohjelmakoodi voidaan tulkita näin: Tee venttiili, jos sen putkiliitososuuden halkaisija on tasan 2 tuumaa. Vertailuoperaattoreita, kuten EQ (equals) on olemassa enemmänkin, mm. LT (Less Than) sekä GT (Greater Than).

Esimerkin 6 If toimintoa voi käyttää myös muodoissa *Feature if*, *Point If* sekä jo edellä mainitut *Valid IF* ja *ConditionallyExecute IF* tapaan.

Validointi ja ehtolause ovat toimintansa kannalta hyvin samanlaisia komentoja ja siksi niitä käsitelläänkin tässä osiossa peräkkäin. Seuraavassa on esitelty ehtolauseen käyttöä käytännössä.

Esimerkki 7. IF-Then määrittely lauhteenpoisto kokoonpanossa.

```
<Network>
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" NPD="{NPD$}" IF="{NPD}" .EQ. "{TWO_INCH}"
or "{NPD}" .LT. "{TWO_INCH}"/>
<PipeRun CreateLike="Element(1)" NPD="3in" IF="{NPD}" .GT. "{TWO_INCH}"/>
<Point X="0" Y="0" Z="0" />
<Point Distance="0.2*{E}" Direction="DOWN" IF="{NPD}" .EQ. "{TWO_INCH}"
or "{NPD}" .LT. "{TWO_INCH}" SaveLocationAs="A"/>
<ConditionallyExecute IF="("{NPD$}".EQ."3in") OR ("{NPD$}".EQ."4in")
OR ("{NPD$}".EQ."5in") OR ("{NPD$}".EQ."6in") OR ("{NPD$}".EQ."7in")
OR "{NPD$}".EQ."8in">
<Point Distance="0.25*{E}" Direction="DOWN" SaveLocationAs="A"/>
</ConditionallyExecute>
```

Yllä esitetystä esimerkistä on esitetty lauhteenpoisto kokoonpanon liitosmahdollisuuksien rajaustoimintoja. Tällaisia toimintoja ovat: Putkiosuus tehdään jos (PipeRun CreateLike - IF), sekä Putkiosuus tehdään tietyissä ennalta määritellyissä olosuhteissa (ConditionallyExecute IF). Esimerkissä 8 kokoonpanon putkiosuudet valikoituvat kokoonpanon liittämisputkiin koon mukaan. Kokoonpanoon liittyvän ohjelmakoodin merkitys voidaan tiivistää seuraavasti: Tehdään putkiosuus, jonka halkaisija on ”esimerkiksi kaksi tuumaa” jos käyttäjän osoittaman, referenssipisteessä olevan elementin halkaisija on ”kaksi tuumaa”.

## 6.6 Putkilinjan ja -osuuden määrittäminen

Putkien koodaaminen voidaan tehdä ohjelmaan joko määrittelemällä putkien yksityiskohtaiset tiedot tai käyttämällä referenssiputkea mallista ja modifioimalla referenssiputken tietoja halutunlaisiksi. Jälkimmäinen tapa on huomattavasti nopeampi, helpompi ja varmempi tapa määritellä eritoten putkiosuus.

Putkiosuudet määritellään yleensä Network-osioon, jonka kautta ohjelma alkaa rakentaa kokoonpanoa, ellei Items-osiota ole laitettu hierarkiaan Network-osion edelle. Seuraavassa on esitetty putkilinjan määrittäminen referenssiputkella mallista.

Esimerkki 8. Putkiosuuden määrittely, tapa 1.

```
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" NPD="{NPD$}" />
<Point X="0" Y="0" Z="0" />
<Point Distance="200mm" Direction="DOWN" SaveLocationAs="A"/>
```

Putkilinjan määrittämiseen riittää yleensä yksi viittaus Network-osion alussa, jolloin ohjelma olettaa linjan pysyvän samana, jollei sitä määritellä uudelleen. Tavassa 1 annetaan referenssipiste 0,0,0, josta liikutaan 200 mm alaspäin. Kyseisessä mallissa ohjelma valitsee itse sille ennalta koodatun tiedon pohjalta sopivan haaroituskomponentin.

Putkiosuuden määrittäminen voidaan tehdä myös seuraavanlaisilla tavoilla.

Putkiosuuden määrittely, tapa 2

```
<Network>
<Item ShortCode="Eccentric Size Change" GeometryType="Runchange"
SavePort2LocationAs="E1" CreateNewRunLike="Element(1)" NPD="0.5in"/>
```

```

<PipeRun CreateLike="Element(1)" NPD="0.75in"/>
<Point X="1m" Y="0" Z="0"/>
<Point X="1m" Y="0.5*{E}" Z="0"/>

<PipeRun CreateLike="Element(1)" NPD="0.75in"/>
<Point X="1m" Y="0.3*{E}" Z="0"/>
<Point Distance="2mm" Direction="Reverse_Primary" Flush="true"
TrimToBranchPort="true" SaveLocationAs="Branch2Port"/>
<Item ShortCode="Eccentric Size Change" GeometryType="Runchange"
SavePort2LocationAs="E2" CreateNewRunLike="Element(1)" NPD="0.5in"/>

<Point Location="E2" ComputedDistance="E2;E1"
Direction="Reverse_PRIMARY" Flush="true"/>

</Network>

```

Tavassa 2 määritellään supistuksien väliin tulevan poikkiputkiosuuden rakentamismalli. Tavassa putkiosuuden tiedot annetaan osittain jo komponentin ohjelmakoodissa ja tällöin putkiosuus tulee määrämittäisenä alkaen kohdasta Point Location.

Putkiosuuksia tulee kokoonpanoon yleensä useampia kuin yksi. Putkiosuus joudutaan määrittelemään uudelleen aina kun joku putken ominaisuus muuttuu, tai kun halutaan määritellä, että putkiosuus rakennetaan vain tietyissä ehdoissa. Edellä mainituista tilanteista on esitetty havainnollistukset seuraavassa.

Esimerkki 9. Putkiosuuden määrittäminen ominaisuuden muuttuessa.

```

<Network>
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" Suffix="-Test1"
NPD="Element(2)|ONE_STEP_DOWN"/>
<Point From="Element(2)"/>

```

Tavassa 3 luotavan putkiosuuden halkaisija luetaan osoittamalla putkilinjan halkaisija referenssielementistä, Element(2). Tällöin kyseinen putkiosuus määritellään tulevaksi yhtä halkaisijakokoa pienemmäksi kuin verrokkinsa.

```
<Point Distance="{D}" SaveLocationAs="A"/>
```

Esimerkki 10. Putkiosuus rakennetaan vain tietyissä ehdoissa.

```
<Network>
```

```
<PipeLine Use="Element(1)"/>
```

```
<PipeRun CreateLike="Element(1)" NPD="{NPD$}" IF="{NPD}" .EQ. "{TWO_INCH}"
or "{NPD}" .LT. "{TWO_INCH}"/>
```

Putkiosuuden rakentamiselle on SP3D:n laitossuunnitteluohjelmassa mahdollistettu useita erilaisia tapoja. Seuraavassa kyseisiin tapoihin perehdytään tarkemmin.

## 6.7 Erilaisia tapoja putkiosuuksien ohjelmointiin

Putkiosuus voidaan rakentaa mm. määrittelemällä: putkiosuuden alkupiste sekä loppupiste tai antamalla osuuden alkupiste ja suunta. Seuraavassa esitellään viisi erilaista tapaa putkiosuuden tekemiseen.

Ensimmäiseksi esitellään yksinkertaisin putkiosuuden rakentamistapa.

Esimerkki 11. Putkiosuuden määritteleminen alkupisteen ja loppupisteen avulla.

```
<PipeRun CreateLike="Element(1)" NPD="0.75in"/>
```

```
<Point X="1m" Y="0" Z="0"/>
```

```
<Point X="1m" Y="0.5*{E}" Z="0"/>
```

Kun putkiosuus määritellään CAD-pohjaisessa laitossuunnitteluohjelmassa, pitäisi olla itsestään selvyyttä, että määrittelyssä käytetään koordinaattipisteitä, kuten edellisessä esimerkissä. Toisena tapana esitellään yleisin putkiosuuden ohjelmointitapa.

Esimerkki 12. Putkiosuuden rakentaminen putkipituuden ja suunnan avulla.

```
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" NPD="{NPD$}" />
<Point X="0" Y="0" Z="0" />
<Point Distance="200mm" Direction="DOWN" SaveLocationAs="A"/>
```

Esimerkissä 12 osoitetaan piste, josta halutaan liikkua määrämittainen matka alaspäin.

Kolmantena esitellään määrittelytapa, jossa putkiosuuden suunta määräytyy edellisten elementtien suuntaiseksi.

Esimerkki 13. Pelkällä putkipituudella määrittely.

```
<Network>
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" Suffix="-1"/>
<Point X="0" Y="0" Z="0" SaveLocationAs="Origin"/>
<Point Distance="2mm" Direction="SECONDARY"/>
<Pipe Length="200mm"/>
```

*Esimerkissä 13 arvo 2 mm kuvastaa sovitetta, ei putkea!*

Esimerkissä 13 käytetään Pipe Length-määrittelyä, jolloin suunta on osoitettava jollakin tavalla. Kyseisessä esimerkissä putkiosuuden suunnanositusmenetelmänä on käytetty sovitetta, jolle on annettu suunta päälinjasta poikkeavaksi (Secondary). Tällöin Pipe Length-toiminto lukee suunnan edelliseltä koodiriviltä.

Neljäntenä putkiosuuden määrittelytapana esitellään hyvin käytännöllinen tapa nimeltä ComputedDistance. Tämä tapa on kehitetty tilanteisiin, joissa vaikkapa putkihalkaisija muuttuu, mutta putkipituus halutaan pitää kokoonpanossa samana.

Sama tilanne toistuu, jos kokoonpanon laitteet lukevat kokotietonsa putkilinjasta ja suurenevat, jolloin myös putkiosuus pituus muuttuu.

Esimerkki 14. Putkiosuuden pituuden rakentaminen ja vakiointi tietyn pituiseksi.

```
<Item ShortCode="Eccentric Size Change" GeometryType="Runchange"
SavePort2LocationAs="E1" CreateNewRunLike="Element(1)" NPD="0.5in"/>
```

```
<PipeRun CreateLike="Element(1)" NPD="0.75in"/>
```

```
<Point X="1m" Y="0" Z="0"/>
```

```
<Point X="1m" Y="0.5*{E}" Z="0"/>
```

```
<PipeRun CreateLike="Element(1)" NPD="0.75in"/>
```

```
<Point X="1m" Y="0.3*{E}" Z="0"/>
```

```
<Point Distance="2mm" Direction="Reverse_Primary" Flush="true"
```

```
TrimToBranchPort="true" SaveLocationAs="Branch2Port"/>
```

```
<Item ShortCode="Eccentric Size Change" GeometryType="Runchange"
SavePort2LocationAs="E2" CreateNewRunLike="Element(1)" NPD="0.5in"/>
```

```
<Point Location="E2" ComputedDistance="E2;E1" Direction="Reverse_PRIMARY"
Flush="true"/>
```

```
</Network>
```

Tavassa 5 putkiosuuden määrittely tehdään ”varatuilla sanoilla” Point From tai Point To. Tämänkaltainen määrittely mahdollistaa käyttäjälle putkiosuuden rakentamisen aloitus- tai lopetuspuolelta.

Esimerkki 15. Putkiosuuden määrittely tavan 5 mukaan.

```
<Items>
```



```

<Instrument Location="0,0,0" PartNumber="{OrificePlate}" Name="FI-xxx"
PART-IJFaceToFace.FaceToFace="15mm" SavePort1LocationAs="OrificePlatePort1"
SavePort2LocationAs="OrificePlatePort2"
<Item Location="OrificePlatePort1" ShortCode="Orifice Flange"
SavePort3LocationAs="OrificeFlangePort1" SavePort2LocationAs="Flange1Port2"
OrientationAngle="{Angle}"/>
</Items>
<Network>
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" Flow="{Flow}"
IJUAIInspectionClass.NJ_Inspection_Class="{NJ_Inspection_Class}"
NPD="OrificeFlangePort1" NameRule="PipeRunNJ"/>
<Point From="OrificeFlangePort1" Direction="UP"/>
<Pipe Length="70mm"/>

```

Tavassa 5 osoitetaan origopisteeksi käyttäjän osoittama piste referenssielementistä (Element(1)). Tähän pisteeseen rakennetaan instrumentti, jonka toisesta päästä (Port1) putkiosuus Point From -komennolla määritellään alkamaan.

Toinen yksinkertaisempi malli putkiosuus määrittelyihin on esitetty alla.

```

<Network>
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" Suffix="-Test1"
NPD="Element(2)|ONE_STEP_DOWN"/>
<Point From="Element(2)"/>
<Point Distance="{D}" SaveLocationAs="A"/>

```

Tavassa 5 Point To- komento eroaa Point From- komennosta vain siten, että To-komennossa viitataan putkiosuuden loppupisteeseen.

## 6.8 Sovitteen tekeminen

Sovite on esimerkiksi hitsaussauma, jolla voidaan yhdistää kaksi komponenttia. Yleensä ohjelma osaa tehdä hitsisaumat itse, toisaalta välillä tulee vastaan tilanne, jossa toiminto ei toteudukaan. Tästä on esimerkkinä ensin ohjelma, joka jättää hitsisauman tekemättä, ja toisena sen korjattu versio.

```

<Network>
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" NPD="1.5in" Suffix="-Drain"/>
<Point X="0" Y="0" Z="0" />
<Point Distance="{D}" Direction="DOWN" />
<Item ShortCode="{Valve}" OptionCode="{ValveOption}" GeometryType="INLINE" />
<Item ShortCode="{ClosingItem}" OptionCode="{ClosingItemOption}"
GeometryType="END" />
</Network>
<Overrides>
<Feature Location="0,0,0" ShortCode="{Branch}" />
</Overrides>
</ParametricAssemblyDefinition>
...
<Item ShortCode="Gate Valve" OptionCode="{ValveOption}"
Secondary="{RotationDirection}"/>
<Point Distance="2mm" Direction="DOWN"/>
<Item ShortCode="Flange" GeometryType="RUNCHANGE" Spec="A3N"/>

```

Alla on esitetty vielä sovitteen käyttöä haaroituslinjan suunnan osoittajana.

```

<PipeRun CreateLike="Element(1)" NPD="0.75in"/>
<Point X="0" Y="0.3*{E}" Z="0"/>
<Point Distance="2mm" Direction="Primary" Flush="true" TrimToBranchPort="true"
SaveLocationAs="BranchPort"/>

```

Seuraavassa esitellään tapoja ohjelmoida laitteita putkiosuuksiin.

## 6.9 Komponenttien ja instrumenttien rakentaminen

Tässä osiossa määritellään kaikkien putkiosuudelle tulevien laitteiden tms. rakentaminen ja annetaan vinkkejä laitteiden määrittelyyn eri osioissa.

Ensimmäisenä osiossa esitellään komponenttien ja laitteiden ohjelmoiminen Networks-osioon, toisena Items-osioon ja kolmantena Overrides-osioon. Elementtien ohjelmoiminen eri osioihin mahdollistaa erilaisten kokoonpanoratkaisujen toteuttamisen.

### Network-osioon määrittely

Komponenttien ja vastaavien määrittely Network-osioon on PAL-ohjelmoinnin yleisin toteustapa pienehköissä kokoonpanoissa. Komponentteja määritellään Network-osioon myös silloin, kun kyseisiä kokoonpanon osia on mahdoton sijoittaa ns. "itemeinä" jo malliin rakennettuihin putkiin. Tästä tapauksesta on havainnollistus seuraavassa.

Esimerkki 16. Ulospuhalluksen komponenttien määrittely Network-osioon

```
<Item ShortCode="{Branch}"/>
<Point Distance="{D}" Direction="UP" Flush="true"/>
<Item ShortCode="Gate Valve" Secondary="{RotationDirection}"
SavePort2LocationAs="A"/>
<Point Location="A" Direction="UP"/>
<Point Distance="2mm" Direction="UP"/>
<Item ShortCode="Flange" GeometryType="RUNCHANGE" Spec="A3N"/>
<Item ShortCode="Blind Flange" GeometryType="END"
PART-IJSequence.Id="NJ_NP4-17434_6_H"/>
```

Esimerkissä 16 Network-osioon määriteltäessä komponentit voidaan antaa peräkkäisessä ohjelmointijärjestyksessä.

Joissain tapauksissa putkilinjat halutaan pitää täsmälleen tietyn mittaisina muista ohjelmointitoiminnoista riippumatta. Tällöin käytetään komponenttien sijoitteluun Items-osioita. Sijoitettaessa Items-osion komponentteja jo rakennettuihin putkiosuuksiin, leikkaa ohjelma komponenteille niiden tarvitsemat tilat putkiosuuksiin säilyttäen kuitenkin annetun putkipituuden.

Esimerkki 17. Säästöventtiiliryhmän komponenttien määrittely Items-osioon.

```
<Item Location="0.5*{Width},{D1},0" ShortCode="{BypassValve}"
OptionCode="{ValveOption}" Secondary="Normal"/>
<Item Location="0,0.5*{D},0" ShortCode="{GateValve}"
OptionCode="{GateValveOption}" Secondary="Normal"/>
<Instrument Location="0,{D1}+0.65*{D},0" PartNumber="{ControlValve}"
Secondary="Reverse_Normal" PART-IJFaceToCenter.Face1toCenter="{F1C}"
```

Esimerkissä 17 on komponentit määritelty Items-osioon, jotta kokoonpanon komponentit pysyisivät samoilla kokoonpanopaikoillaan kokoonpanon ollessa riippuvainen liitoslinjan putkikoosta. Eli jos liitoslinjan putkikoko muuttuu, tällöin myös kokoonpanon osien koot muuttuvat ja "iteminä" annettuina komponentit pysyvät aina tarkalla määrittely paikalla.

Seuraavassa on esim. Overrides-osioista, joka on suunnattu haaroituksille Items-osion edellä mainittuun tapaan.

Ulospuhalluksen haaroituksen määrittely Overrides-osioon

```
<Parameter Name="Branch" Default="WeldOlet" Options="Sokolet|WeldOlet"/>
```

...

```
<Overrides>
```

```
<Feature Location="0,0,0" ShortCode="{Branch}" />
</Overrides>
</ParametricAssemblyDefinition>
```

Lisäksi esitetään esimerkki, jossa määrittely on korvattu Network-osion koodissa.

```
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)"
<Point Distance="{D}" Direction="UP" Flush="true" />
```

Yllä esitetystä koodinosasta ohjelmoija voi luottaa siihen, että ohjelma osaa valita oikean tyyppisen haaroituksen kokoonpanon alkupisteeseen. Pituuden D tulee siis olla niin pitkä, että vaikkapa 90 mm pitkä Nipolet mahtuu putkelle, jolloin ohjelma poistaa putken kyseiseltä kohdalta ja korvaa sen Nipolet-komponentilla.

Network-osioon on olemassa kuitenkin myös toinen määrittelytapa ja sitä kutsutaan nimellä ”referenssipisteen perusteella määrittely”. Tässä määrittelytavassa käytetään hyväksi SP3D:n PAL-kieleen ohjelmoituja kokoonpano-osien referenssipisteitä. Tällaisia pisteitä ovat ohjelmassa esimerkiksi putkenpää, keskikohta tai ohjelmoinnissa erikseen ohjelmoijan tallentama piste. Tämä tapa tulee esille myöhemmin.

### Items-osioon määrittely

Komponenttien tms. määrittely tehdään Items-osioon, kun halutaan säilyttää kokoonpanon absoluuttiset mitat. Kyseisessä tavassa ohjelmoidaan ensin putkilinjat ja vasta sen jälkeen laitteet. Putkilinjat tehdään Items-osioon esimerkiksi Point Distance -komennolla ja laitteet lisätään jo rakennettuun linjaan Items-osiossa, koordinaatti- tai referenssipisteiden avulla. Tällaista ohjelmointitapaa käytetään vaativissa kokoonpanoissa kuten säätöventtiiliryhmät.

Säätöventtiiliryhmän komponenttien määrittely Items-osioon

```
<Item Location="0.5*{Width},{D1},0" ShortCode="{BypassValve}"
OptionCode="{ValveOption}" Secondary="Normal"/>
<Item Location="0,0.5*{D},0" ShortCode="{GateValve}"
OptionCode="{GateValveOption}" Secondary="Normal"/>
<Instrument Location="0,{D1}+0.65*{D},0" PartNumber="{ControlValve}"
Secondary="Reverse_Normal" PART-IJFaceToCenter.Face1toCenter="{F1C}"
```

Items-osioon määrittelyssä käytetään instrumenttien kohdalla yllä esitettyjä varattuja sanoja, kuten PartNumber sekä Instrument Location joista kohdassa PartNumber voidaan antaa Instrumentin tilausnumero, josta ohjelma tunnistaa numerolle koodatut määrittelyt. Instrument Location -komennolla taas määritellään instrumentin sijoituspiste.

Overrides-osioon määrittely

Overrides-osiota käytetään vain kun halutaan tarkentaa haaroitusten määrittelyjä. Normaaleissa tapauksissa haaroituksia ei tarvitse erikseen määrittellä, vaan ohjelma osaa valita oikean tyyppisen haaroituksen (T-haara, Nipolet) putkiosuuden koon, tyyppin, liikuttavan suunnan, geometrian sekä putkipituuden perusteella.

Esimerkki 18. Ulospuhalluksen haaroituksen määrittely Overrides-osioon.

```
<Parameter Name="Branch" Default="WeldOlet" Options="Sockolet|WeldOlet"/>
...
<Overrides>
<Feature Location="0,0,0" ShortCode="{Branch}" />
</Overrides>
</ParametricAssemblyDefinition>
```

Esimerkki 19. Sama kokoonpano Network osioon tehtynä, ilman Overrides-osioon määrittelyä.

```
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)"
<Point Distance="{D}" Direction="UP" Flush="true" />
```

## 6.10 If-Then-määrittely

Kokoonpanon määrittelyissä joudutaan välillä tukeutumaan If-Then ehtolauseeseen. Tällaisia tilanteita tulee eteen muun muassa silloin, kun halutaan määrittellä kokoonpanon koon olevan erilainen liitettäessä sitä halkaisijaltaan erikokoiseen putkilinjaan. Tästä on havainnollistus alla.

Esimerkki 20. If Then-määrittely lauhteenpoisto kokoonpanossa.

```
<Network>
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" NPD="{NPD$}" IF="{NPD}" .EQ. "{TWO_INCH}"
or "{NPD}" .LT. "{TWO_INCH}"/>
<PipeRun CreateLike="Element(1)" NPD="3in" IF="{NPD}" .GT. "{TWO_INCH}"/>
<Point X="0" Y="0" Z="0" />
<Point Distance="0.2*{E}" Direction="DOWN" IF="{NPD}" .EQ. "{TWO_INCH}"
or "{NPD}" .LT. "{TWO_INCH}" SaveLocationAs="A"/>
<ConditionallyExecute IF="{NPD$}.EQ."3in" OR "{NPD$}.EQ."4in"
OR "{NPD$}.EQ."5in" OR "{NPD$}.EQ."6in" OR "{NPD$}.EQ."7in"
OR "{NPD$}.EQ."8in">
<Point Distance="0.25*{E}" Direction="DOWN" SaveLocationAs="A"/>
</ConditionallyExecute>
```

Yllä olevassa esimerkissä on esitetty lauhteenpoistokokoonpanon liitosmahdollisuuksien rajaustoimintoja. Tällaisia toimintoja ovat: Putkiosuus tehdään jos (Piperun CreateLike - IF), sekä putkiosuus tehdään tietyissä ennalta määritellyissä

olosuhteissa (ConditionallyExecute IF). Esimerkissä 21 kokoonpanon putkiosuudet valikoituvat kokoonpanon liittämisputkilinjan koon mukaan.

Esimerkki 21. Venttiili tehdään jos referenssiputkenhalkaisija on 2 tuumaa.

```
<Network>
```

```
...
```

```
<Item IF="{NPD}" .EQ. "2in" ShortCode="Gate Valve"/>
```

Esimerkin 21 If-toimintoa voi käyttää myös muodoissa *Feature If*, *Point If* sekä jo edellä mainittujen *Valid If* ja *ConditionallyExecute if* - toimintojen tapaan.

If-Then-määrittely voidaan tehdä mm. ominaisuudelle, referenssipisteelle sekä validoinnille.

If then määrittelylle on olemassa myös vaihtoehtoinen komento ConditionallyExecute. Kyseisessä määrittelytavassa voidaan valita minkä ehtojen puitteissa jokin kokoonpano rakennetaan. Conditionally-ehdolla voi rajata vaikkapa vain jonkun tietyn koodiriviosuuden. Tällöin kyseinen koodiriviosuus toteutetaan vain jos Conditionally ehto täyttyy.

Esimerkki 23. ConditionallyExecute ehdolla rajattu putkiosuuden teko.

```
<ConditionallyExecute IF="{NPD$}.EQ."3in") OR ("{NPD$}.EQ."4in") OR
("{NPD$}.EQ."5in") OR ("{NPD$}.EQ."6in") OR ("{NPD$}.EQ."7in") OR
"{NPD$}.EQ."8in">
<Point Distance="0.25*{E}" Direction="DOWN" SaveLocationAs="A"/>
</ConditionallyExecute>
```

## 6.10 Komponenttien ominaisuuksien spesifointi

Putkiosuuksille sijoitettavia tai rakennettavia komponentteja voi määritellä pääosin kahdella eri tavalla. Ensimmäinen tapa on määritellä komponentin tms. spesifointi



varatulla sanalla `SetAttributesOnRunItems` nimettyyn osioon. Tällöin ohjelma rakentaa ensin komponentin ja muuttaa sen jälkikäteen ohjelman käyttäjämäärittelyjen mukaiseksi.

Toinen tapa komponenttien ominaisuuksien määrittelyyn on tehdä määrittely komponentin kanssa samalle koodiriville alla olevan mallin mukaan.

Esimerkki 24. Lauhteenpoistokomponentin ominaisuuksien määrittely.

```
<SetAttributesOnRunItems
RUN-IJRteTesting.TestingType=":55"
PART-IJConstructionInfo.ConstructionType="New"
WELD-IJRteWeldTesting.TestingType="Radiographic"
GASKET-IJRteGasket.GasketReportingClassification=":10"
BOLT-IJRteBolt.BoltReportingClassification=":10"
FEATURE-IJCoatingInfo.CoatingType=":4"
FEATURE-IJCoatingInfo.CoatingColor="Blue"/>
```

Esimerkin 24 osio sijoitetaan yleensä `Network`-osion loppuun omana kokonaisuutenaan.

Tässä esimerkissä 24 on esitetty myös kuinka ominaisuuksia luetaan tietokannan koodikentästä (*Codelist values*), jota käsitellään tarkemmin myöhemmissä luvuissa.

Koska ominaisuuksien määrittely tehdään hiukan eri tavalla jos kyseessä on instrumentti tai vaihtoehtoinen komponentti, on alla esitetty mahdolliset tavat niiden ohjelmointiin.

Esimerkki 25. Säätöventtiiliryhmän toimilaitesinstrumentin ominaisuuksien määrittely.

```
<Instrument Location="0,{D1}+0.65*{D},0" PartNumber="{ControlValve}"
Secondary="Reverse_Normal" PART-IJFaceToCenter.Face1toCenter="{F1C}"
PART-IJFaceToCenter.Face2toCenter="{F2C}"
PART-IJFaceToFace.FaceToFace="100mm"
```

```

PART-IJUAInstrumentActuatorCylinder1.ActuatorCyl1Length="100mm"
PART-IJUAInstrumentActuatorCylinder1.ActuatorCyl1Diameter="90mm"
PART-IJUAInstrumentActuator.ActuatorHeight="100mm"
PART-IJUAInstrumentActuatorCylinder.ActuatorCylLength="300mm"
PART-IJUAInstrumentActuatorCylinder.ActuatorCylDiameter="90mm"
PART-IJUAInstrumentActuatorHeight.ActuatorHeight3="300mm"
PART-IJUAInstrumentActuatorHeight.ActuatorHeight1="100mm"
PART-IJUAInstrumentActuatorHeight.ActuatorHeight2="1mm"
PART-IJUAInstrumentActuatorLength.ActuatorLength="100mm"
PART-IJUAInstrumentActuatorWidth.ActuatorWidth="50mm"
PART-IJUAInstrumentActuatorWidth.ActuatorWidth1="100mm"
PART-IJUAInstrumentActuatorWidth.ActuatorWidth2="1mm"
PART-IJUAInstrumentActuatorWidth.ActuatorWidth3="1mm"
PART-IJUAInstrumentActuatorWidth.ActuatorWidth4="100mm"
PART-IJUAInstrumentActuator.ActuatorDiameter="1mm"/>

```

Esimerkissä 25 PART-sanalla viitataan komponentin osa määrittelyyn. Jos kyseessä olisi toisaalta sana RUN, viitattaisiin putkiosuuden määrittelyvaihtoehtoihin.

Toinen tapa muuttaa komponenttien tms. objektien ominaisuuksia on linkittää ominaisuusmuutos samalle koodiriville muutettavan kappaleen kanssa. Kyseinen tapa toimii kaikissa ohjelman luontiosioissa ja siitä on havainnollistus alla.

...

```

<Instrument PartNumber="IND4" GeometryType="END"
PART-IJUAInstrumentDimensions.InstrumentDiameter="1mm"
PART-IJDynamicPipePort1.TerminationClass1="Male"
PART-IJDynamicPipePort1.EndPreparation1=":391"/>

```

...

Joissakin tapauksissa taas joudutaan käyttämään niin sanottua vaihtoehdon rajaus -komentoa, jossa pakollinen tarkennus komponentin mallista annetaan. Tästä on alla esitetty esimerkki liitospään Swagelok määrittelystä paineen mittauksessa.

Esimerkki 26. OptionCode määrittelyn käyttö.

```
<Item ShortCode="Tube Connection"
OptionCode="Swagelok compression tubing end"/>
```

Esimerkissä 26 vaihtoehtoisuudella viitataan tapaan, jolla tietyt komponentit tulee tarkentaa ohjelmakoodiin. Näistä esimerkkinä on juuri esitelty Tube Connection komponentti, jolle ei ohjelmakoodissa ole oletusarvoa (esim. Swagelok) vaan se pitää selvittää tarkemmin käyttämällä OptionCode-sanaa. Esimerkissä 27 OptionCode sana vastaa Parameters-osion oletusarvokenttää.

### 6.11 Geometriatyypimäärittelyn hyödyntäminen ja käyttö

Putkistosuunnittelussa joudutaan miettimään usein ratkaisuja putkilinjojen mutkien, haaroitusten sekä putkikoon muutosten toteutustapoihin. SP3D:n PAL-kielisessä ohjelmoinnissa edellä mainitut asiat tulee mainita koodirivillä termillä GeometryType. Geometriatyypin määrittelyllä voidaan selventää ohjelmalle mitä sen halutaan tekevän. Alla on esitetty yleisimmät geometriatyypimäärittelyvaihtoehdot.

Esimerkki 27. Taivutuksen määrittely.

```
<Point X="0" Y="{Pipe Length}" Z="0" IF="{Angle} .GT. 56" Direction="DOWN"/>
<Point X="0" Y="{Upper Pipe Length}" Z="0" IF="{Angle} .LT. 56"
Direction="DOWN"/>
<Point Distance="{Lower Pipe Length}" Direction="Secondary"/>
<Point Distance="{C}" Direction="Secondary {Angle} {Lower_Assembly_Orientation}"
Prompt="To Change the Direction of the Lower Assembly - Change Reverse_Normal
to Normal from the Parameters Section"/>
<Pipe Length="{D}"/>
```

Esimerkissä 27 on annettu tietyt putkipituudet erikokoisille kyseisen mutkan mutkille. Suurempi kulma tarvitsee pidemmät putkiosuudet sekä ennen että jälkeen,

siksi Point-komennolla on kaksi erilaista putkiosuus määrittelyä ohjelmakoodissa. Seuraavassa on esitetty mutkan määrittelemisen ohjelmakoodikomponenttina.

Esimerkki. 28 Mutkan määrittely komponenttitunnuksella.

```
PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" Suffix="-Test1"/>
<Point X="0" Y="0" Z="0"/>
<Point Distance="{D}" Direction="Primary"/>
<Item ShortCode="90 Degree Direction Change" Secondary="Normal"
GeometryType="TURN"/>
<Point Distance="{D}"/>
```

Esimerkin 28 komponenttimäärittely on kuitenkin riippuvainen putkiosuuden nimellishalkaisijasta. Eli jos tämä halkaisija on alle 2 tuumaa, määrittelyä ei voi käyttää, koska näin pieniä 90 asteen mutkia ei ole olemassa.

Esimerkki 29. Komponenttisarjan päättymisen ilmaiseminen.

```
<Item ShortCode="Blind Flange" GeometryType="END"/>
</Network>
</ParametricAssemblyDefinition>
```

Esimerkki 30. Putki-osuuden koon muutoksen ilmaiseminen.

```
<Point Distance="{D3}" Direction="REVERSE_NORMAL" Geome-
tryType="RUNCHANGE" Spec="A3N"/>
```

Seuraavassa on edellistä vastaava keskeiselle supistukselle tehty ilmaisumuoto.

```
<Point Distance="0.5*{D}" Direction="SECONDARY"/>
<Item ShortCode="Concentric Size Change" GeometryType="RUNCHANGE"
NPD="TWO_STEP_DOWN" SavePort1LocationAs="TurnPointA"/>
```

```
<Point Distance="0.2*{D}" Direction="SECONDARY"/>
```

## 6.12 Referenssipisteiden tallennus ja käyttö

Esimerkki 31. Referenssipisteiden tallentaminen ja käyttö säätöventtiiliryhmässä.

```
<Network>
```

```
<PipeLine Use="Element(1)"/>
```

```
<PipeRun CreateLike="Element(1)" Suffix="-ControlValveRoute" NPD="{NPD$}"/>
```

```
<Point X="0" Y="0" Z="0"/>
```

```
<Point Distance="{D1}" Direction="SECONDARY"/>
```

```
<Point Distance="0.5*{D}" Direction="SECONDARY"/>
```

```
<Item ShortCode="Concentric Size Change" GeometryType="RUNCHANGE"
```

```
NPD="TWO_STEP_DOWN" SavePort1LocationAs="TurnPointA"/>
```

```
<Point Distance="0.2*{D}" Direction="SECONDARY"/>
```

```
<Item ShortCode="Concentric Size Change" GeometryType="RUNCHANGE"
```

```
NPD="TWO_STEP_UP" SavePort1LocationAs="TurnpointB"/>
```

```
<Point Distance="0.3*{D}" Direction="SECONDARY"/>
```

```
<Point Distance="0.65*{Width}" Direction="PRIMARY" SaveLocationAs="A"/>
```

```
<Point Distance="0.35*{Width}" Direction="PRIMARY"/>
```

```
<Point Distance="0.3*{D}" Direction="REVERSE_SECONDARY" SaveLocationAs="C"/>
```

```
<Point Location="C" ComputedDistance="TurnPointA;TurnPointB"
```

```
Direction="REVERSE_SECONDARY"/>
```

```
<Point Distance="{D1}" Direction="REVERSE_SECONDARY"/>
```

```
<Point Distance="0.5*{D}" Direction="REVERSE_SECONDARY"/>
```

```
<Point Distance="{H}" Direction="PRIMARY"/>
```

Muita tallennettavia kohtia ovat mm. komponenttien keskipisteet, -päät, putkien keskipisteet, päät, putkiosuudet, putkilinjat sekä origo.

Esimerkki 32. T-haaran sijoittaminen referenssipään perusteella.

```
<Item ShortCode="Tee" GeometryType="BRANCH" ReferencePosition="{RefPos}"
PlaceByPort="{PlaceByPort}"/>
```

Referenssipisteiden tallennusta joudutaan käyttämään eritoten edellä mainitussa ComputedDistance komennossa.

Esimerkki 33. Osan nimeäminen isometriin tyyppitunnuksella.

```
<Item ShortCode="Blind Flange" GeometryType="END"
PART-IJSequence.Id="NJ_NP4-17434_6_H"/>
```

Esimerkki 34. Supistuksellisen T-haaran eri suuntien putkihalkaisijan määrittely.

```
<Parameters>
<Parameter Name="HeaderNPD" Default="6in" Options="6in|4in"/>
<Parameter Name="BranchNPD" Default="4in" Options="4in|3in"/>
<Parameter Name="ContinuationNPD" Default="4in" Options="4in|3in|2in"/>
</Parameters>
<Network>
<PipeLine CreateLike="Element(1)"
Name="[HeaderNPD]-[BranchNPD]-[ContinuationNPD]"/>
<PipeRun CreateLike="Element(1)" Name="Header" NPD="{HeaderNPD}"/>
<Point X="-1m" Y="0" Z="0"/>
<Point X="1m" Y="0" Z="0"/>

<PipeRun CreateLike="Element(1)" Name="Branch" NPD="{BranchNPD}"/>
<Point X="0" Y="0" Z="0" />
<Point Distance="2mm" Direction="DOWN" Flush="true" TrimToBranchPort="true"
SaveLocationAs="BranchPort"/>
```

```
<Item IF="{BranchNPD}" .NE. "{ContinuationNPD}"
ShortCode="Concentric Size Change" GeometryType="RUNCHANGE"
CreateNewRunLike="Element(1)" NPD="{ContinuationNPD}"
```

Esimerkki 35. Komponentti sarjan päättymisen ilmaiseminen.

```
<Item ShortCode="Blind Flange" GeometryType="END"/>
</Network>
</ParametricAssemblyDefinition>
```

Esimerkki 37. Putki-osuuden koon muutoksen ilmaiseminen.

```
<Point Distance="{D3}" Direction="REVERSE_NORMAL"
GeometryType="RUNCHANGE" Spec="A3N"/>
```

(Intergraph pdf, Version 2009)

### 6.13 Ohjelman kirjoitussäännöt

Tämä luku käsittelee Parametric Assembly Language-kielen kirjoitusasua, merkistöä ja kielelle varattuja operaattoreita. Tavoitteena on antaa hyvä yleiskuva kielen kirjoitusasun säännöistä ja antaa valmiudet perehtyä seuraavassa luvussa olevaan yksityiskohtaiseen esimerkkiin.

#### Kirjoitussäännöt

SP3D:n ohjelmointikieli muistuttaa ulkoasultaan hierarkiapuuta. Eli ohjelmoinnissa on runko, joka jakautuu osiin, joista lähtee haaroja. Voisi siis sanoa, että tässä ohjelmointikielen tyyppissä on nimettyjä ryhmiä, jotka tekevät oman organisaationsa käskyjä ensin ryhmäkohtaisesti sitten ryhmän sisällä henkilökohtaisesti. Tällöin organisaatiota koskee omat sääntönsä, samoin kuin ryhmää, ja yksilöä.

Koska suuri osa säännöistä on merkkiin liittyviä, tullaan seuraavassa käsittelemään ohjelmointia niiden rajaamin säännöin.

Parametrien käyttöön perustuvan ohjelman kirjoittaminen aloitetaan vasemmasta reunasta SP3D:ssä. Aloitusta kuvaava ja myös sille varattu sana, on "ParametricAssemblyDefinition". Ohjelmointi koodina se näyttää seuraavalta:

```
<ParametricAssemblyDefinition>
```

Vastaava lopetukselle on seuraavanlainen:

```
</ParametricAssemblyDefinition>
```

Ulkoasuun on siis lisätty kenoviiva, joka läpi koko ohjelmoinnin kuvastaa yhden tietyn osion toteuttamisen loppua. Miksi näin? Koska jokaiselle osiolle on omat sääntönsä, joilla ne toimivat.

Ohjelmassa edetään yleensä seuraavanlaisella rakenteella:

```
<ParametricAssemblyDefinition>
  </Information Name>
  <ReferenceElements>
    <Element Prompt/>
  </ReferenceElements>
  <GiFFile Name>
  <Parameters>
  </Parameters>
  <Network>
  </Network>
  <Items>
  </Items>
  <Overrides>
  </Overrides>
</ParametricAssemblyDefinition>
```



Rakenne kuvaelmassa on tummennettuna referenssielementteihin sisältyvä alitietue. Tämä havainnollistaa sitä että, aina tarkempaan määrittelyyn mentäessä siirrytään hierarkiapuussa sisennyksissä oikealle. (Lähes kaikkiin kohtiin voidaan liittää useita alitietueita, jotka löytyvät Intergraphin parametrissa ohjelmointia käsittelevästä oppaasta).

Kaikki ominaisuudet, kuten pituus ja etäisyys määritellään ohjelmassa seuraavan mallin mukaan: `NPD="0,75in"`. Tässä määrittelyssä "varatulle sanalle" esimerkiksi NPD arvo on 0,75in. (Putkihalkaisija)

Kun taas halutaan viitata ohjelmassa tietueeseen "keksityllä" kuvaavalla nimellä, laitetaan viittaus `Parameter Name` -osion kohtaan `Default` ja valitaan oletusarvot/tyypit. Tämän jälkeen `Options`-kohtiin voidaan valita määrittelyt SP3D:ssä olevista valmistyypeistä tai antaa ohjelman hoitaa valinta kirjoittamalla `"Default"` tai vielä pakottamalla arvo kirjoittamalla `Parameters`-osioon `Value="-"/>`.

## 7 OHJELMOINTI YKSITYISKOHTAISIN ESIMERKEIN

Tässä luvussa esitellään koko ohjelmantekoprosessi tyyppiinirustuksesta lähtien, aina valmiiseen sijoitettuun kokoonpanoon. Esittelyssä käytetään ilmanpoistoon tarkoitettua kokoonpanoa. Luvussa käydään myös lävitse tapaa, jolla kokoonpanoja sijoitetaan malliin.

Parametristä ohjelmointia esitellään esimerkillä kokoonpanon luonnista ilmanpoistolle putkistosta. Esimerkkitapauksessa kokoonpanon luonti aloitetaan putkiston putkiluokkapiirustuksesta. Seuraavaksi valitaan suunnittelutyyppi putkiston tyyppiluokan vaihtoehdoista. Putkistosuunnittelulle on tällöin kyseessä suunnittelutyyppi: ”Putkisto”. Seuraavaksi putkiston liitepiirustusluettelosta (H104) etsitään oikeat kokoonpanojen tyyppi nimet. Ilmanpoiston tapauksessa valitaan alla oleva vaihtoehto.

NP4-17433 s.2 Ilmanpoisto  $\frac{3}{4}$ ”(150# - 600#), BW

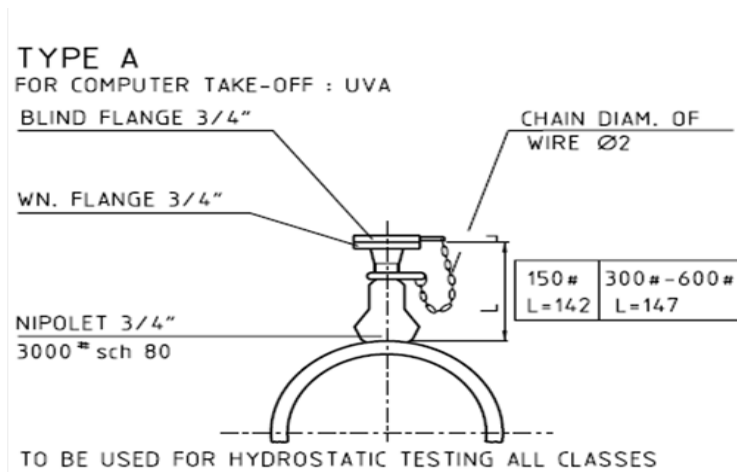
NP4-17433A s.2 Vent  $\frac{3}{4}$ ”(150# - 600#), BW

Tyyppinimien löydyttyä voidaan varmistaa putkiluokkastandardista kokoonpanolle tulevat perusominaisuudet, kuten halkaisija, tarkastusluokka sekä käytettävä materiaali. Tämän jälkeen valitaan putkien putkiluokka pääosin virtaavan aineen mukaan.

PIPE CLASS	MAIN USE AREA	PIPES			FLANGES			FITTINGS		
		Wall thickness	Material	Colour marking	Type	Material	Colour marking	Type	Material	Colour marking
Carbon steel										
A1N0 A1NA A1NB / A1P0 A1PA A1PB	Look the list of fluids from the specification H102	$\frac{1}{2}$ "-1 $\frac{1}{2}$ " Sch.80, 2"-10" Sch.40, 12"-24" WT 3/8" (SMLS), 26"-28" WT 3/8" (WLD), 30"-42" WT 1/2", 44"-52" WT 5/8", 56"-60" WT 3/4"	SA333-6, SA671 CC60/CC65 CL32 S2 (26"-28", 30"-42", 44"-52", 56"-60")	No colour marking	WN 150# $\frac{1}{2}$ "-1 $\frac{1}{2}$ " sch.80, 2"-10" sch.40, 12"-24" WT 3/8", 26"-28" WT 3/8", 30"-42" WT 1/2", 44"-52" WT 5/8", 56"-60" WT 3/4" Blind $\frac{1}{2}$ "-24", 26"-60"	SA 350-LF2 CL1	No colour marking	Elbow 90°, Elbow 45°: $\frac{1}{2}$ "-1 $\frac{1}{2}$ " LR sch.80 , 2"-10" LR sch.40 , 12"-24" LR WT 3/8" (SMLS), 26"-28" LR WT 3/8" (WLD), 30"-42" LR WT 1/2", 44"-52" LR WT 5/8", 56"-60" LR WT 3/4". Reducer end, Cap, Tee: $\frac{1}{2}$ "-1 $\frac{1}{2}$ " sch.80, 2"-10" sch.40 , 12"-24" WT 3/8" (SMLS), 26"-28" WT 3/8" (WLD), 30"-42" WT 1/2", 44"-52" WT 5/8", 56"-60" WT 3/4"	SA420-WPL6	No colour marking
					Spect.blind, Spade Spacer 6"-60" 150#	P355NL1				

Kuva 25. Leikkaus H102 putkiluokkastandardista – Ilmanpoistolle (Niemi, R. 2014).

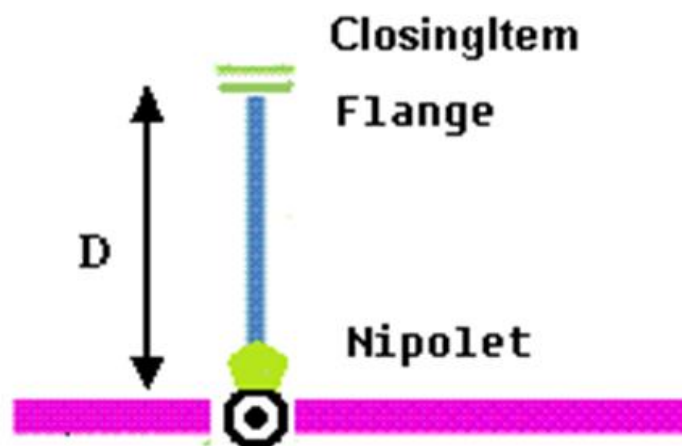
Putkiluokan valinnan jälkeen katsotaan tyyppiierrosarkistosta kyseisen tyyppin tyyppikuva (CAD-piirros). Kuvasta nähdään putkien mitat, käytettävät komponentit sekä tarkemmat määrittelyt. Ilmanpoistoon liittyvä tyyppikuva on esitetty kuvassa 26.



Kuva 26. Leikkaus - Standardin mukaisista tyyppiierroksista - H104 (Yksinkertainen ulospuhallus kokoonpano).

Kuvassa on määritelty käytettävät komponentit:

1. Umpilaippa - Blind Flange
2. Laippa (Vastakappale) - Flange
3. Haaroituksenliitännäkomponentti - Nipolet



Kuva 27. Piirto-ohjelma hahmotelma tyyppiierroksen H104 TYPE A kokoonpanosta.

Alla on esitetty ilmanpoistoa käsittelevälle esimerkkikokoonpanolle suunniteltu ohjelma.

```
<ParametricAssemblyDefinition>
```

```
<Information Name="Vent - Pipe + Flange + ClosingItem"
```

```
Description="Piping Vent Assembly
```

```
with Closing Item" CreatedBy="Ville Kerttula" Date="12-Apr-2014"
```

```
Version="Revision1"/>
```

Ohjelmassa informaatiokenttään on määritely, mitä kokoonpanon tulee sisältää, kuka sen on tehnyt ja milloin. Kyseiset tiedot ovat vain tekstimuotoisia ohjeita, eivät siis varsinaisesti tee mitään.

```
<ReferenceElements>
```

```
<Element Prompt="Select Pipe Feature / Pipe Part to Place Vent"/>
```

```
</ReferenceElements>
```

Referenssi kohdassa on ohjeistettu, mistä kohdasta kokoonpanon voi laittaa rakentumaan mallissa.

```
<GIFFile Name="NJ_Vent.jpg"/>
```

Giffile kohdassa on viitattu kuvaan nimeltä NJ\_Vent.jpg, joka samaan kokoonpanokansioon tehtynä näkyy kokoonpanon sijoitusikkunassa valitsemalla toiminto View -> Image.

```
<Parameters>
```

```
<Parameter Name="D" Default="100MM" UnitType="DISTANCE"/>
```

```
<Parameter Name="NJ_Inspection_Class" Default=":10" />
```

```

<Parameter Name="Flow" Default="Down"
Options="Up|Down|Both|No|Undefined"/>
<ParameterFromReferenceElement Name="Spec" Value="Spec@Element(1)"/>
<Parameter Name="Branch" Default="Nipolet" Options="Nipolet"/>
<Parameter Name="ClosingItem" Default="Blind Flange" Options="Blind Flange"/>
<Parameter Name="ClosingItemOption" Default="Default"/>
</Parameters>

```

Parameters-kohtaan on määritelty tietokannasta luettavat komponentti nimet, putkipituus, virtaussuunta ja mm. tarkastusluokka. Koodiarvolla :10 viitataan Schema Browserista löytyvään tietueeseen (**Code list** value) liittyen tarkastusluokkaan. Ohjelma siis lukee tarkastusluokan tarkan tyyppin tietokannasta, johon se on koodattu tässä tapauksessa ohjelmalla Visual Basic.

```

<Network>
<PipeLine Use="Element(1)"/>
<ConditionallyExecute IF="({Spec}.EQ."A1N") OR ({Spec}.EQ."A3N")
OR ({Spec}.EQ."A6N") OR ({Spec}.EQ."K1N") OR ({Spec}.EQ."K3N")
OR ({Spec}.EQ."K6N") OR ({Spec}.EQ."L1N") OR ({Spec}.EQ."L3N")
OR {Spec}.EQ."L6N""
Prompt="Assembly Valid only for Specs - A1...6N OR K1...6N OR L1...6N">
<PipeRun CreateLike="Element(1)" Flow="{Flow}"
IJUAInspectionClass.NJ_Inspection_Class="{NJ_Inspection_Class}" NPD="0.75in"
NameRule="PipeRunNJ" />
<Point X="0" Y="0" Z="0" />
</ConditionallyExecute>
<Point Distance="{D}" Direction="UP" Flush="true" />
<Point Distance="2mm" Direction="UP" SaveLocationAs="A"/>

```

```

<PipeRun CreateLike="Element(1)" Type="RUNCHANGE" Spec="A3N" NPD="1in"
Flow="{Flow}" IJUAInspectionClass.NJ_Inspection_Class="{NJ_Inspection_Class}"
NameRule="PipeRunNJ"/>
<Point Location="A"/>
<Point Distance="2mm" Direction="UP"/>
<Item ShortCode="Flange"/>
<Item ShortCode="Blind Flange" GeometryType="END"
PART-IJSequence.Id="NJ_NP4-17433_2_A"/>
</Network>

```

Network-kohdassa on osoitettu valittava putkilinja (Pipeline), minkä tyyppinen putkiosuus (PipeRun) on ominaisuuksiltaan, kuten virtaussuunta (Flow), halkaisija (NPD), tarkastusluokka (InspectionClass) sekä nimeämistapa (NameRule). Network- osiossa on myös määritelty (ConditionallyExecute) minkälaiseen putkityyppiin ulospuhalluksen voi liittää. Tässä tapauksessa sallittavat putkiosuudet kuuluvat paineen- ja lämpötilan kestoltaan luokkiin: 150 lb=X1Y, 300 lb=X3Y tai 600 lb. Kyseisiä luokkia vastaavat määritteet tyyppiluokissa ovat A1N, A3N sekä A6N. Käyttämällä ConditionallyExecute määrittelyä, estetään kokoonpanojen sijoittaminen vääräntyyppisiin putkiosuuksiin.

```

<Overrides>
<Feature Location="0,0,0" ShortCode="{Branch}"/>
</Overrides>
<ParametricAssemblyDefinition>

```

Overrides-kohdassa voi esimerkiksi tehdä haaroituskomponentin kokoonpanolle. Sen voi tosin tehdä usealla muullakin tavalla. esimerkiksi sen voi tehdä osoittamalla pisteen ja kertomalla että tehdään putkilinja pisteestä A pisteeseen B, jolloin ohjelma tekee automaattisesti haaroituksen, jos linja on eriävän suuntainen päälinjasta Element(1). Toisella tavalla sen voi tehdä tekemällä haaroituskomponentin

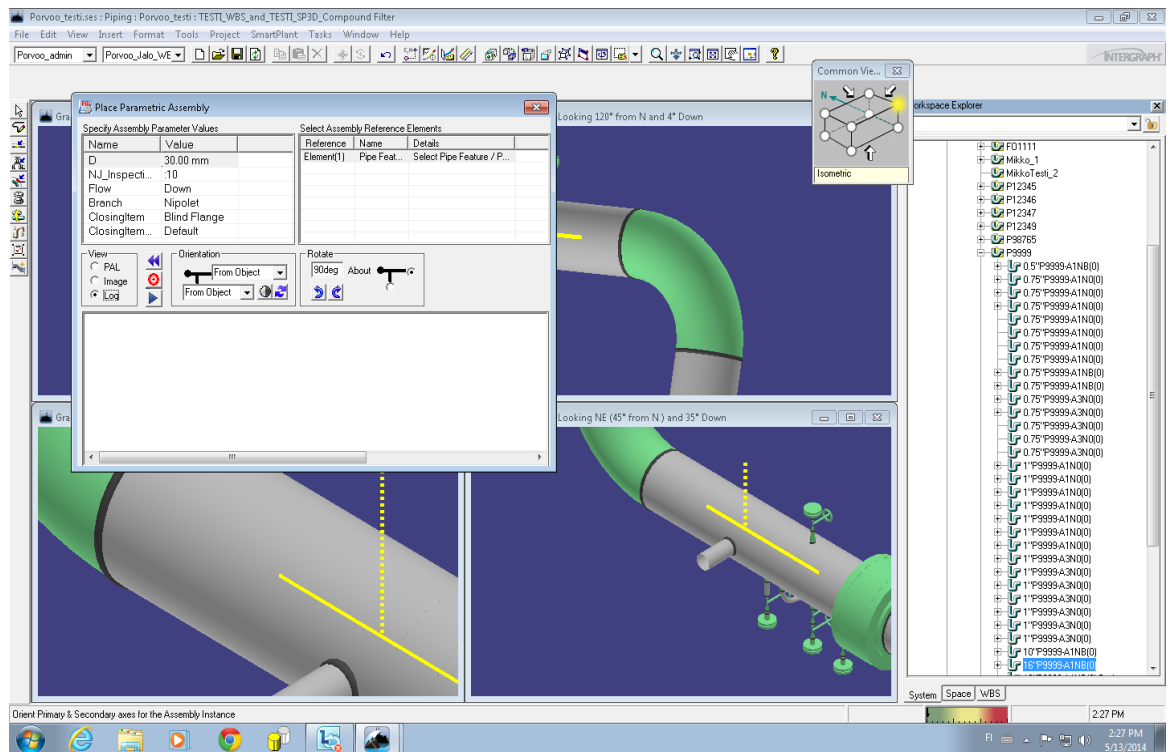
Item ShortCode määrittelyllä Network-osioon. Käytännössä Overrides-määrittely haaroituksille on valinnainen.

Ohjelman koodaamisen jälkeen se tallennetaan esim. SP3D:n testi sessioon, jossa sitä voidaan testata avaamalla välilehdet järjestyksessä: "Tools" -> "Automation" -> "Place Parametric Assembly" ja valitsemalla sijoituspaikaksi keskilinjan.

## 7.1 Kokoonpanon luominen - Place Parametric Assembly

Tässä alaluvussa selvitetään, kuinka luotu kokoonpano sijoitetaan malliin. Luvussa perehdytään kokoonpanon sijoitusikkunaan ja siitä löytyviin määrittelymahdollisuuksiin. Kokoonpanon luonti-ikkunassa valitaan mm. kokoonpanon sijoituspaikka ja kokoonpanon linjakohtaiset parametrit.

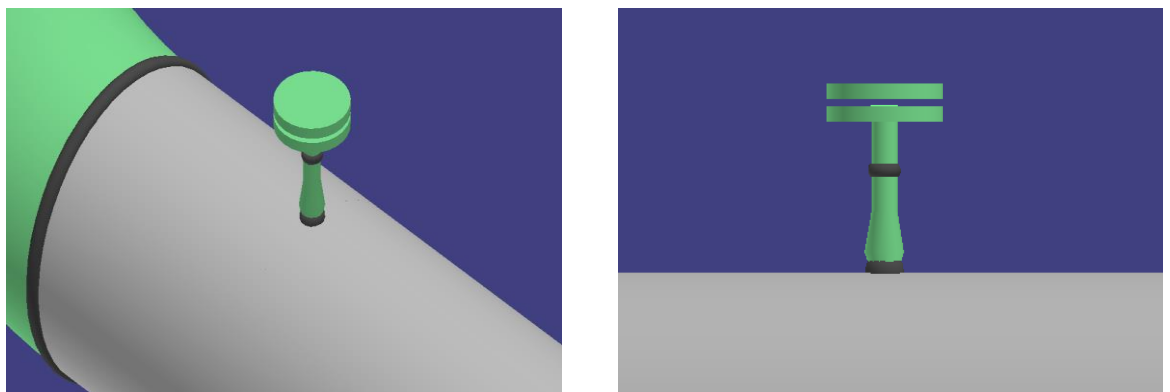
Kokoonpanoikkunassa voidaan muuttaa jo annettuja arvoja sekä määrittää kokoonpanon sijoituspaikka, -kulma, sekä -suunta. Ikkunasta voi myös tarkastella tiettyä tiedostotyyppiä olevaa (.gif /.jpg) kuvaa kokoonpanosta. Jos ohjelmassa on haitallinen virhe, ei ohjelmaa "näy" kokoonpano kansiossa. Näin ohjelma estää vääränlaisen kokoonpanon sijoittamisen malliin. Yleisimpiä virhetyyppejä ovat koodaajan unohtama merkki (</>), vääränlainen viittaus putkiosuuteen (Piperun cREAtELike – tässä on esimerkiksi väärät kirjainkoot) tai ohjelman väärä ulkoasu. Jos kokoonpanon ohjelmassa on pienempi virhe, ohjelma löytyy sessiosta, mutta ohjelman luonnissa tulee Log-ikkunaan ilmoitus, ettei kokoonpanoa voitu luoda joko ollenkaan tai osittain. Tällöin tulee perehtyä ohjelman sekä putkistosuunnittelun sääntöihin syvällisemmin.



Kuva 28. Kokoonpanon luonti-ikkuna.

Kuvassa 28 näkyy valittu putken keskiliinja (keltainen yhtenäinen viiva) sekä kokoonpanon sijoitussuunta, joka on putken keskiliinjasta ylöspäin (katkoviiva).

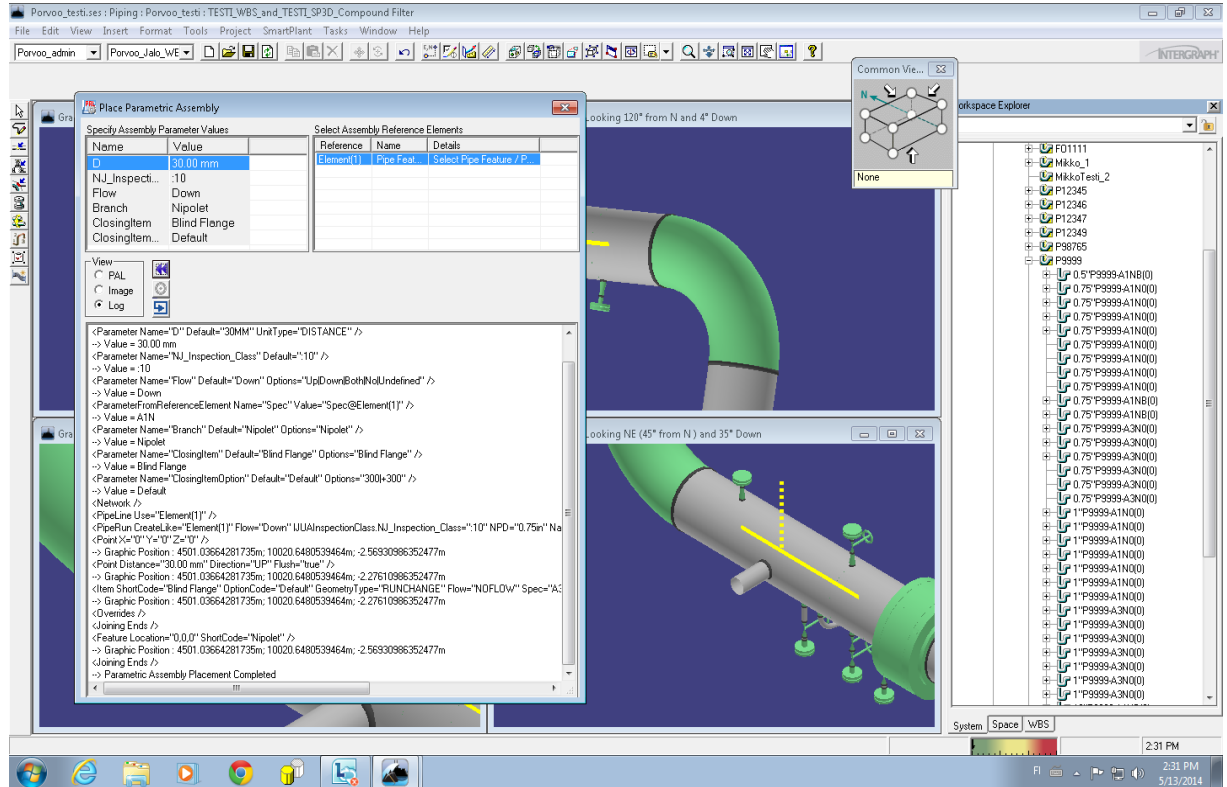
Kokoonpanoikkunasta voi tarkastella ohjelmakoodia valitsemalla View-ikkunasta kohdan PAL. Saman ikkunan kohdasta Image on nähtävissä kuvan 20 kokoonpanolle tehty hahmotelma. Parametrien arvoja voi vielä viimeistellä kokoonpanolle ikkunassa *Specify Assembly Parameter Values*.



Kuva 29. Valmis kokoonpano kahdesta eri kuvakulmasta.



SP3D suunnittelutilassa käytetään yleensä neljään ikkunaan jaettua suunnittelu mallia, jotta suunnittelijat voivat todeta kappaleen olevan ulkoasultaan ja ominaisuuksiltaan oikean näköinen.



Kuva 30. Lokitiedot ovat nähtävissä sijoituksen jälkeen Log-ikkunassa (View-Log).

Onnistuneesta komponentin sijoituksesta tulee ilmoitus "Parametric Assembly Placement Completed".

## 7.2 Ohjelman rakentaminen

Tässä aluvuussa esitellään Network- ja Parameter Name nimisten osioiden viiteyhteys ohjelmassa, ja osoitetaan se havainnollistavalla esimerkillä. Ohjelmassa viitataan usein Parameter Name nimiseen ohjelman määrittelykenttään. Parameter Namekenttä on luotu siksi, että parametreja voidaan nimetä suunnittelijaa helpottavilla nimillä. Näin ollen suunnittelija tunnistaa kokoonpanon parametrien muutosmahdollisuudet sijoittaessaan kokoonpanoa. Ohjelman rakentamisesta

kertovassa osiossa on otettu kahden ohjelmakoodiosan viiteyhteys käsittelyyn erikseen.

Varsinaisesti ohjelma rakentaa kokoonpanon <Network>-osion koodin perusteella, mutta <Parameters>-osiosta voidaan "lukea" arvoja viittaamalla <Network>-osiossa olevaan <Parameter Name>-nimiseen arvoon { } sulkeilla.

Esimerkki 1. Havainnollistus parametrin arvon lukemisesta "Parameter Name" nimisestä kentästä.

```
<Parameters>
<Parameter Name="D" Default="30MM" UnitType="DISTANCE"/>
<Parameter Name="Branch" Default="Nipolet" Options="Nipolet"/>
<Parameter Name="ClosingItem" Default="Blind Flange" Options="Blind Flange"/>
<Parameter Name="ClosingItemOption" Default="Default" Options=""/>
</Parameters>
<Network>
<PipeLine Use="Element(1)"/>
<PipeRun CreateLike="Element(1)" NPD="0.75in" />
<Point X="0" Y="0" Z="0" />
<Point Distance="{D}" Direction="UP" />
<Item ShortCode="{ClosingItem}"
  OptionCode="{ClosingItemOption}"GeometryType="INLINE" />
```

Yllä havainnollistetussa ohjelmassa on käytetty parametrinimeä ClosingItem. Parametrinimi on kuitenkin vapaasti valittavissa, ja se voisi olla yhtäläillä EndPart tms. Tärkeintä on ymmärtää, että käytettäessä Item Shortcode määrittelyä, täytyy määrittelyn edetä peräkkäisessä järjestyksessä.

Esimerkissä on ensin komponentin päätietue "Blind Flange" (umpilaippa) sen jälkeen päätietuetta tarkemmin määrittelevät tietueet, Options="". Tarkemmilla määrittelyillä voidaan viitata esimerkiksi laipan paineluokitukseen, specifikaatio, Spec="A3N". Tällöin valitun laipan paineluokitus olisi 300 lbs.

Kuitenkin lähes aina, putkistosuunnittelussa joudutaan määrittelemään putken putkiluokka, joka käsittää komponentin materiaalin, paineluokan sekä aakkosellisen tiedon, milloin putkiluokka on otettu käyttöön. Tämä tarkoittaa sitä, että parametri `ClosingItemOption` jää tällöin pois ja sen tiedot sisältyvät seuraavanlaiseen määrittelyyn.

```
<Item ShortCode="Flange" GeometryType="RUNCHANGE" Spec="A3N"/>
```

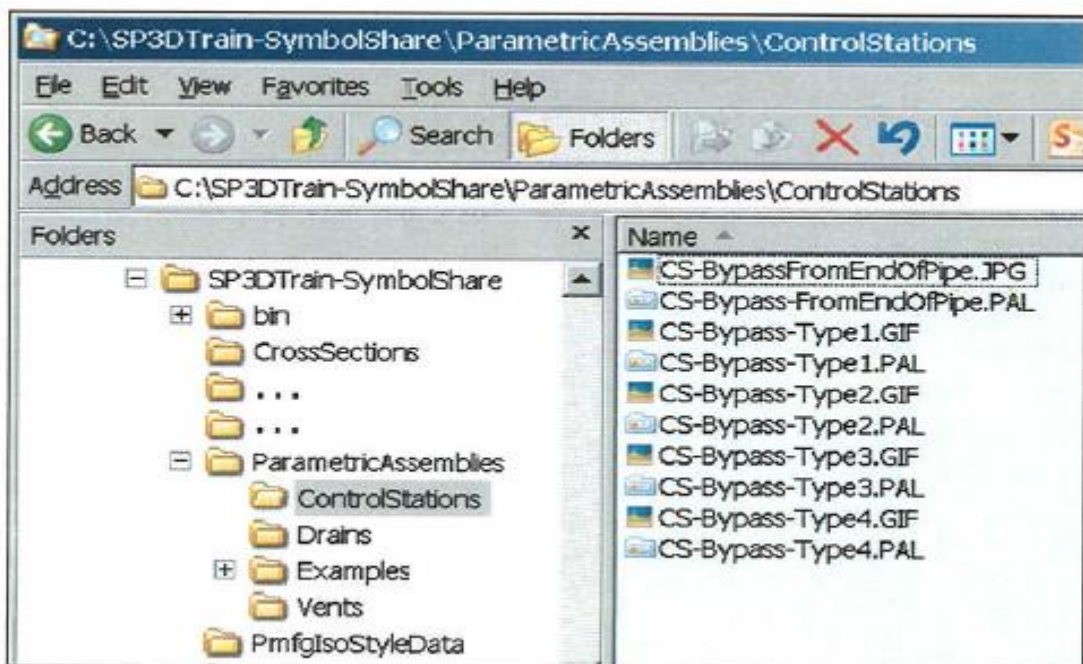
Tässä määrittelyssä on käytettävä myös geometriatyyppiä (`GeometryType`), jotta SP3D:n ehdot putkispesifikaation muutokselle täyttyvät. Toinen syy geometriatyypin määrittelyyn on se, että jos tehdään esim. tyhjennystä, johon sisältyy laippa ja umpilaippa, on molemmat mainittava ohjelmakoodissa allekkain, ja laippa määriteltävä spesifikaation muutoksella, jotta kokoonpanon spesifikaation muutos tulee isometriin oikein. Umpilaippaa ei kuitenkaan tarvitse erikseen määritellä vaan ohjelma lukee tiedon spesifikaation muutoksesta edellisestä komponentista, jos komponentit ovat ohjelmakoodissa peräkkäin eikä spesifikaatiota muuteta uudelleen niiden (komponenttien) välissä.

## 8 PUTKISTON OLIOIDEN TIETOKANTAOMINAISUUKSIIN VIITTAAMINEN

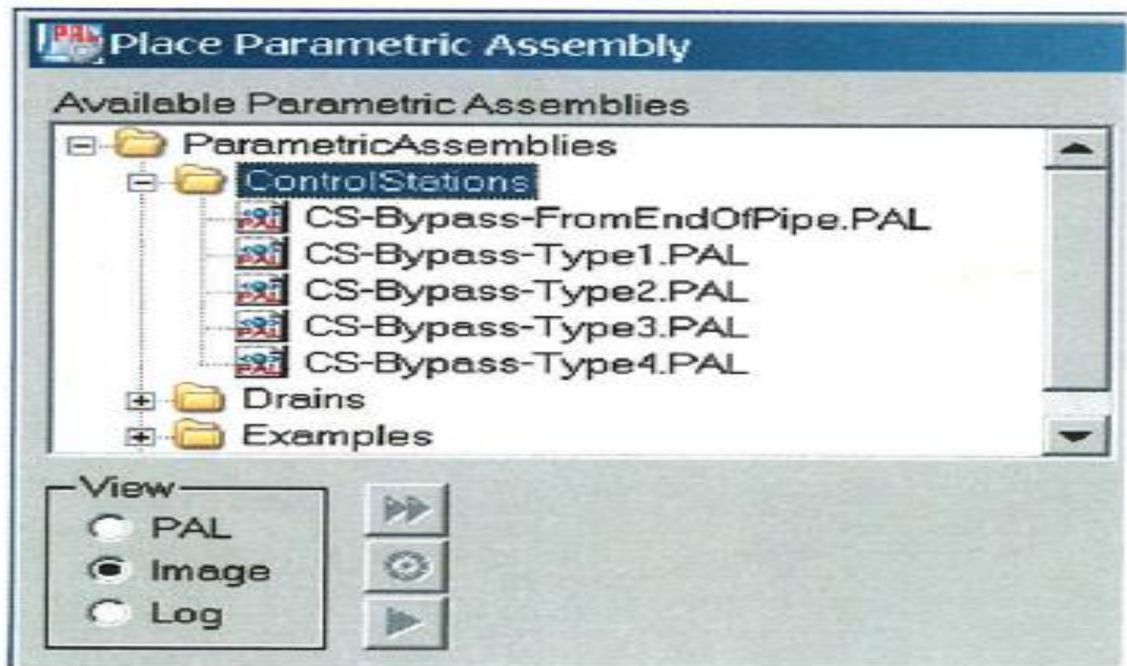
Tässä luvussa tarkastellaan parametrusten kokoonpanojen tietokannan rakennetta ja ohjelmakoodissa esiintyvien komponenttien ja niiden ominaisuuksien määrittelyä. Luku alkaa SymbolShare-tiedostokansion esittelyllä SmartPlant 3D:n tietokannan tiedostokansiohakemiston esittelyllä. Luvun lopussa käydään lävitse kuinka eri putkiston osat kuten suora putkiosuus tai putkimutka on määritelty SP3D:n tietokannassa.

### 8.1 Kokoonpanokansion luominen

SymbolShare on eräs laitokseen liittyvistä jaetuista, useammalle käyttäjälle näkyvistä hakemistoista. Tämän hakemiston käyttö edellyttää siihen liittyviä käyttöoikeuksia. Kokoonpanokansio on eräs SymbolSharen alikansioista. Kokoonpanokansioon luodaan tiedostoja sisällään pitäviä uusia kansioita, jotka nimetään käyttäjien kannalta selkeästi. Luotaessa omannäköistä kokoonpanokansiota löytyy SymbolSharesta valmis mallikansio nimellä ParametricAssemblies. Tässä kansiossa on aluksi vain muutama Intergraphin esimerkki muun muassa säätöventtiiliryhmistä, tyhjennyksistä ja ulospuhalluksista.



Kuva 31. ParametricAssemblies-kokoonpanokansio. (Intergraph, PAL. 2009)



Kuva 32. Säätoventtiiliryhmän mallikansion sisältämät kokoonpanot.

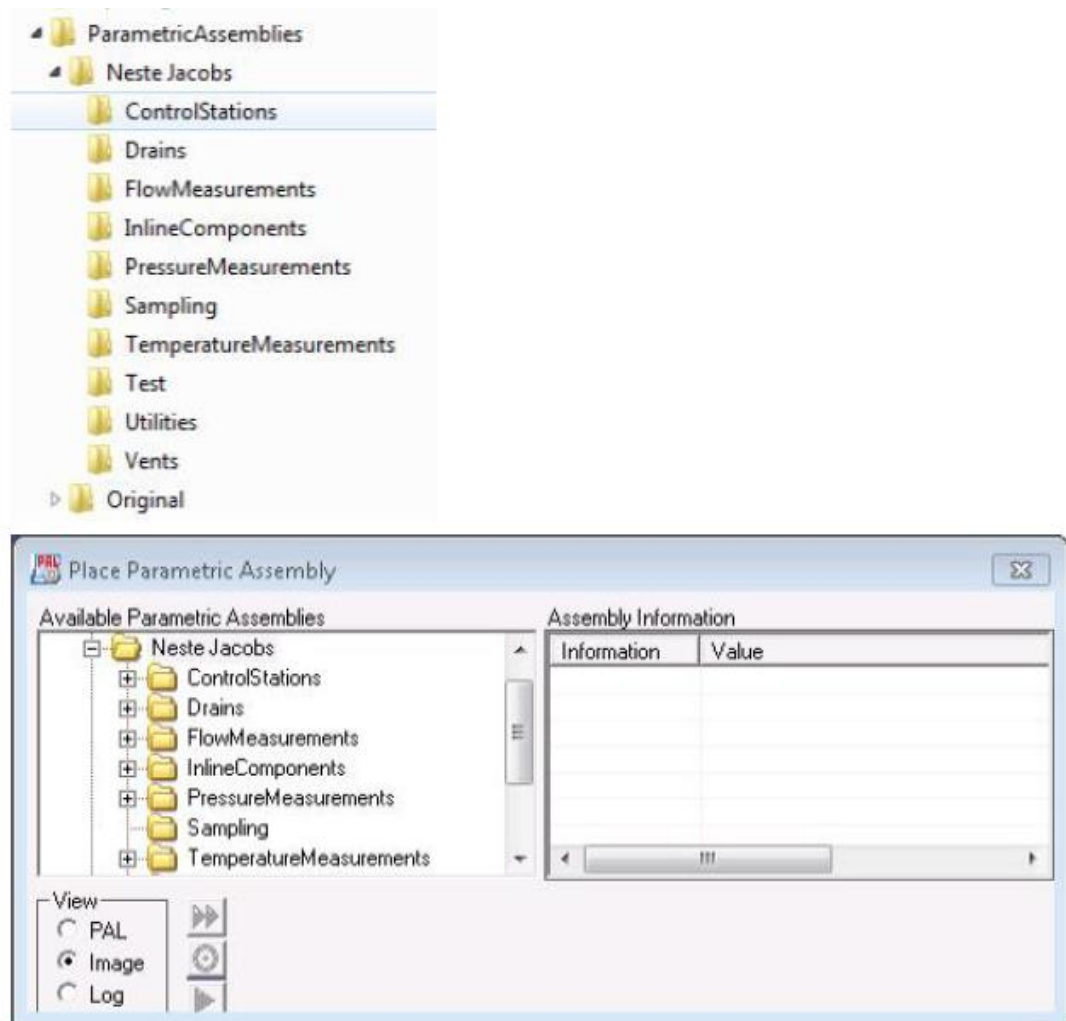
(Intergraph, PAL. 2009)

Kuvan 32 kansioon luodut tiedostokansiot tulevat näkyviin otettaessa mallista kokoonpanojen sijoitus käyttöön valitsemalla Tools -> Automation. Kuvassa 33 on vastaavasti esitetty tämän työn tuloksena luotu kokoonpanokansio.

Luomalla SymbolSharen ParametricAssembly-kansioon uusia alikansioita, joiden halutaan näkyvän kokoonpanojen sijoitusikkunassa niitä käytettäessä, täytyy kansiot nimetä selkeästi. Jos halutaan vielä lisätä esimerkiksi kuva, joka näkyy kokoonpanon sijoitusikkunassa, täytyy se tällöin tallentaa samaan kansioon kokoonpanon ohjelmakoodin sisällään pitämän tiedoston kanssa.

SymbolSharesta löytyy myös laitoksen putkistohierarkiarakenne. Putkistohierarkialla tarkoitetaan tapaa, jolla putkistokomponentit halutaan saada esille SP3D-ohjelman WSE:ssä, Catalogissa, Schema Browserilla tai Assembly tietokannassa. Seuraavassa on esitetty hierarkia malli Assembly-kokoonpano tietokannasta.

Available Parametric Assemblies (käytettävissä olevat Parametriset Assemblyt)



Kuva 33. Kokoonpanokansion sisältö.

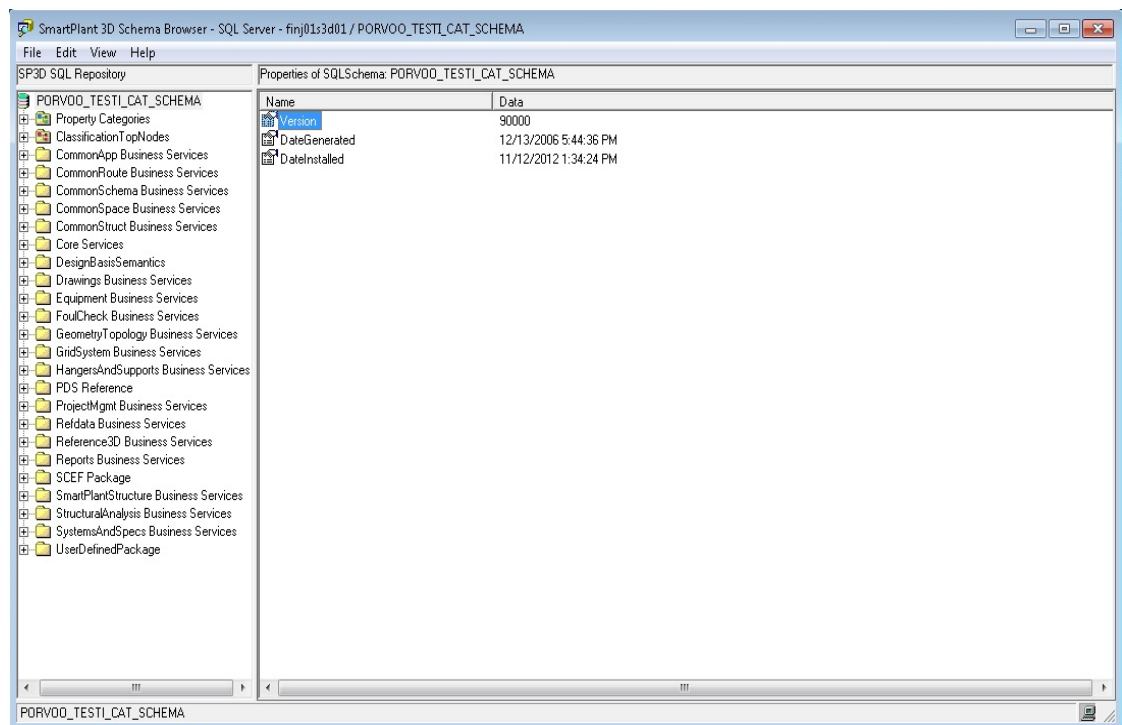
Työkaluohjelmassa hierarkia rakentuu ylhäältä alaspäin, niin että avattaessa hierarkiassa alempi laatikko päästään käsiksi yksityiskohtaisempiin putkistosuunnittelun kokoonpanomalleihin.

## 8.2 Tietokantaan viittaaminen työkaluohjelmalla

Tässä luvussa käsitellään tietokannan linkittymistä parametriseen ohjelmointiin, sekä tapoja viitata tietokannan tietueisiin SP3D:ssä, eritoten NJ:n käyttämässä tietokantatyökalusovelluksessa - Schema Browserissa. Parametrinen ohjelmointi perustuu tietueiden poimimiseen geometrisille objekteille ohjelmakoodista. Tästä syystä tarvitaan ohjelma, jonka kautta tietueisiin on helppo viitata.

Tietokantatyökalu Schema Browser (SB) on Intergraphin Smart 3D -tuoteperheen työkalu metadatan hallintaan. SB:llä pystyy tarkastelemaan SP3D:stä löytyvien olioiden määrittelyjä luokka-käyttöliittymä-attribuutihierarkia -näkyssä. (Intergraph, 2008)

Hierarkiasta on alla esitetty päänäkymä:



Kuva 34. Schema Browser - päänäkymä.

Kuvan 34 otsikkoalueen teksteistä voidaan havaita, että tietokanta käyttää hyväkseen Microsoftin SQL -etäserveriä. SQL-serveri on kapasiteetiltaan lähes rajattoman kokoinen tietokannan etäserveri, johon kaikki SP3D:n olioihin liittyvät luokka-, tyyppi- ja muut ominaisuustiedot on säilötty. Koska tietokannat tarvitsevat paljon ylläpitoa sekä suuret sähköisen tiedon säilöntäalueet, käyttää SP3D hyväkseen Microsoftin palvelimia. Jos SQL on datavarasto, tällöin tietokantasovellus Schema Browser on sen hallinnointiin kehitetty ohjelma.

Schema Browserissa olioluokkaviittaukset löytyvät tietokannan kohdasta Class. Viitattaessa luokan sisältämiin ”kapseloituihin” tietoihin tulee viitata luokan käyttöliittymä nimikkeeseen IJ..Interface. Aina, kun ohjelmakoodissa viitataan olion

osan tai sen ominaisuuden tietoon aloitetaan viittaus käyttöliittymästä. Tämän jälkeen avataan Property kuutio, josta päästään kiinni tiedon attribuutti-arvoihin, joita ovat esimerkiksi virtaussuunta, tarkastusluokka ja paineluokka. Avaamalla attribuuttiarvon alta löytyvän Codelist nimikkeen tulevat esille koodilistan mukaiset arvot, jotka ovat attribuuttien luku/tekstuaalisia arvoja. Esimerkiksi tarkastusluokan koodiarvo :10 vastaa tiettyä tarkastusluokkaa (0).

Schema Browserin päänäkymästä (kuva 34) päästään tarkempiin oliikohtaisiin tietoihin joko hakemalla valikosta Edit -> Find tiettyä datatietoa, tai avaamalla kansioita plus-merkeistä. SQL Repositoryn rakennehierarkia esitetään seuraavassa yksityiskohtaisemmin.

#### SP3D SQL Repository (Composition)

```

+ SP3DTrain_Cat_SCHEMA
+ Property Categories
Classification TopNodes (Package)
- CommonApp Business Services
    + CADefinition
    + CAssocPoint3d (Class)
    - CGeneralNote
        + IJCommonObjects (Interface)
        - IJGeneralNote
            + Name
            + Dimensioned (Property)
            - Purpose
                IJGeneralNote
                o NotePurpose (Codelist)
  
```

Yllä esitetyssä SP3D:n SQL-hierarkiassa on varattu tiedon säilöntäalueet olioiden luokkakohtaisille (Class), mallissa tehtäville käytönaikaisille (Interface) sekä attribuuttina muokattaville (Property) tiedoille. Codelist kohta on vielä tarkempia määrittelyjä varten ja se voi olla esimerkiksi tarkastusluokan Default-arvo.



Ohjelmoinnin yhteydessä tärkeimmät rakennetyypit ovat: Interface, Property ja Codelist. Edellä mainitut esiintyvät ohjelmakoodissa yleensä kohdassa **SetAttributesOnRunItems**, jossa viittaus tietokantaan tallennettuihin parametreihin voidaan muun muassa tehdä. Esimerkiksi putkiosuudella voidaan määrittellä paineensietokykyä, virtaussuuntaa ja käytettävää materiaalia. Tällaisessa määrittelyssä olion päätiedot löytyvät ensin putkiosuuden luokan alta (Class-CPMPiperun) kohdasta. Mentäessä yksityiskohtaisempiin määrittelyihin, löytyvät käyttöliittymässä käyttäjän määriteltävissä olevat tiedot useasta kymmenestä sinisellä merkitystä interface (IJ/UA, kuva 35) polusta. Edelleen tarkempiin määrittelyihin siirryttäessä esimerkiksi eristyksen käyttöliittymän IJRtePipeRunInsulation määrittelyt (attribuutit) löytyvät kohdasta Property. Codelist kohdassa on lopulta määrittelyjen luku/tekstiarvot, jotka tulevat ohjelmoinnissa kenttiin Default ja Value.

Viittaus parametriseen tietoon, joka oliolle halutaan antaa, voidaan tehdä muutamalla tavalla. Yleisin tapa on kuitenkin viitata käyttöliittymän attribuutin koodilistan arvoon:

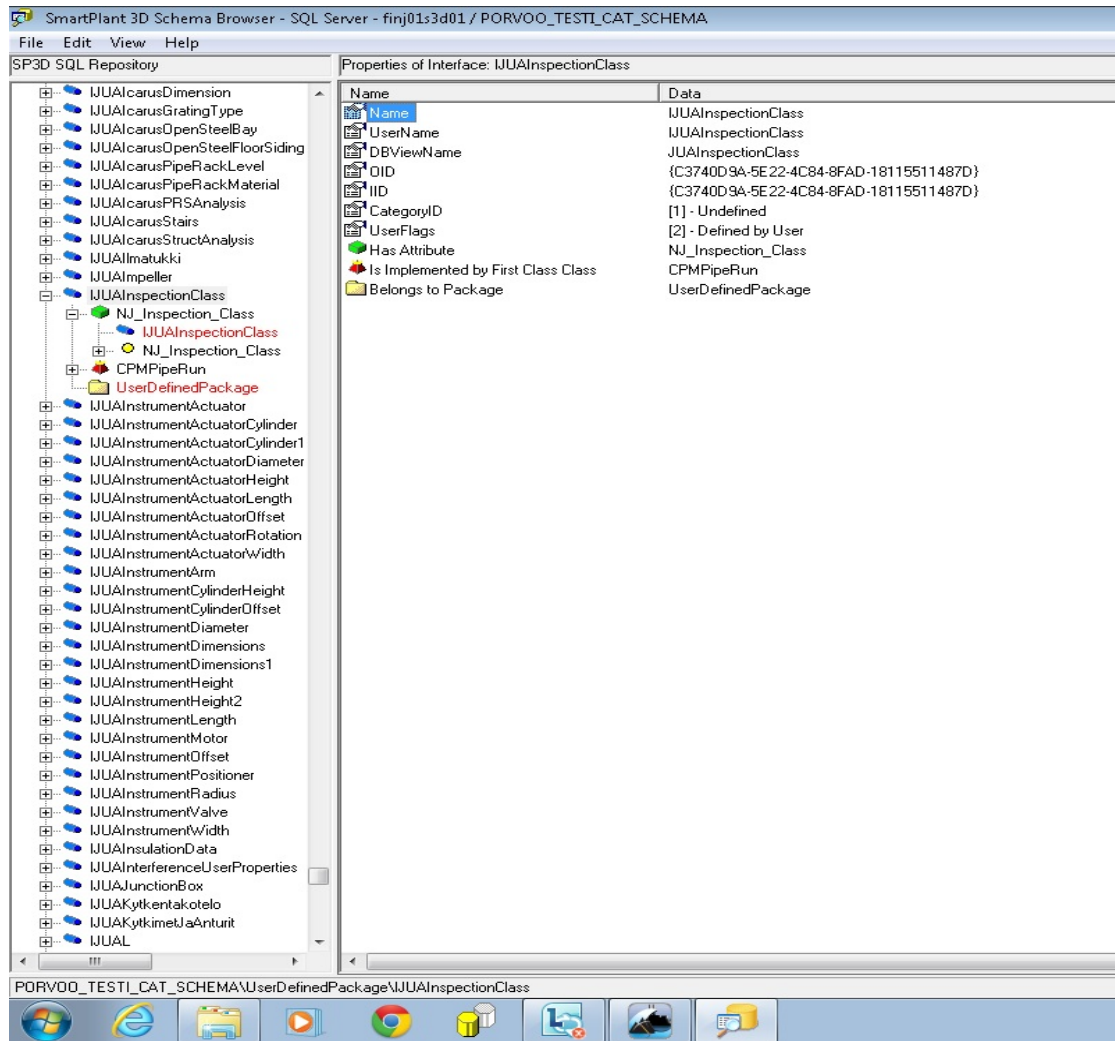
Esimerkki 1. Viittaus parametriseen tietoon

```
<Parameter Name="NJ_Inspection_Class" Default=":10"/>
```

```
<Network>
```

```
<PipeRun CreateLike="Element(1)" IJUAInspection  
Class.NJ_Inspection_Class="{NJ_Inspection_Class}"/>
```

```
<Network>
```



Kuva 35. Tarkastusluokka (InspectionClass) näkymä SB:ssä.

Kuvassa 35 käyttöliittymää kuvaa lieriö, jossa lukee IJUAInspectionClass. Attribuuttia taas kuvastaa kuutio, jossa lukee NJ\_Inspection\_Class. Koodilistan arvo ei sinänsä ole auki näkymässä, mutta se (esimerkiksi :10) löytyisi pallotunnuksella merkitystä kohdasta, jossa lukee NJ\_Inspection\_Class.

### 8.3 Putkiston ominaisuuksien määrittely ohjelmoinnissa

Tässä luvussa selvitetään kuinka putkisto jakaantuu osiin ja kuinka näihin osiin on mahdollista viitata ohjelmakoodissa.

SP3D:ssä kaikki putkiosuudet ja komponentit määritellään olioina. Ymmärtämällä miten oliot sijoittuvat vastaavat putkistoja, komponentteja ja yhteyksiä, pystyy suunnittelija sijoittamaan oikean tyyppisiä olioita SP3D malliin.

Ohjelmoinnissa täytyy usein viitata tietokannassa sijaitsevaan putkidataan. Tällöin viitataan putkiston osaolioiden tietoon.

Esimerkki 1.

```
<SetAttributesOnRunItems
PART-IJSequence.Id="NJ_NP3-6166E"
PART-IJDynamicPipePort1.PressureRating1="300"
PART-IJUIInstrumentDimensions.InstrumentDiameter="150mm"/>
```

Esimerkissä SetAttributesOnRunItems on käytetty tietokannan käyttöliittymään määriteltäviä olioita IJDynamicPipePort1 sekä IJUIInstrumentDimensions, joilla on määritetty ”komponentin” paineluokka yhdelle portille sekä instrumentin halkaisija. PressureRating1 sekä InstrumentDiameter ovat tietokannan attribuutteja.

Tietokanta on jaettu käyttötarkoitusten mukaisiin nimikkeisiin. Kun halutaan määritellä esimerkiksi putkiosuuden ominaisuuksia, annetaan määrittelyt ohjelma-koodissa (SetAttributesOnRunItems) olio-osakohtaisesti. Olio-osia ovat: PART, FEATURE, RUN, GASKET, WELD sekä BOLT (kuva 36).

Putkiston osaoliot määritellään tarkemmin seuraavalla tavalla (kuva 36):

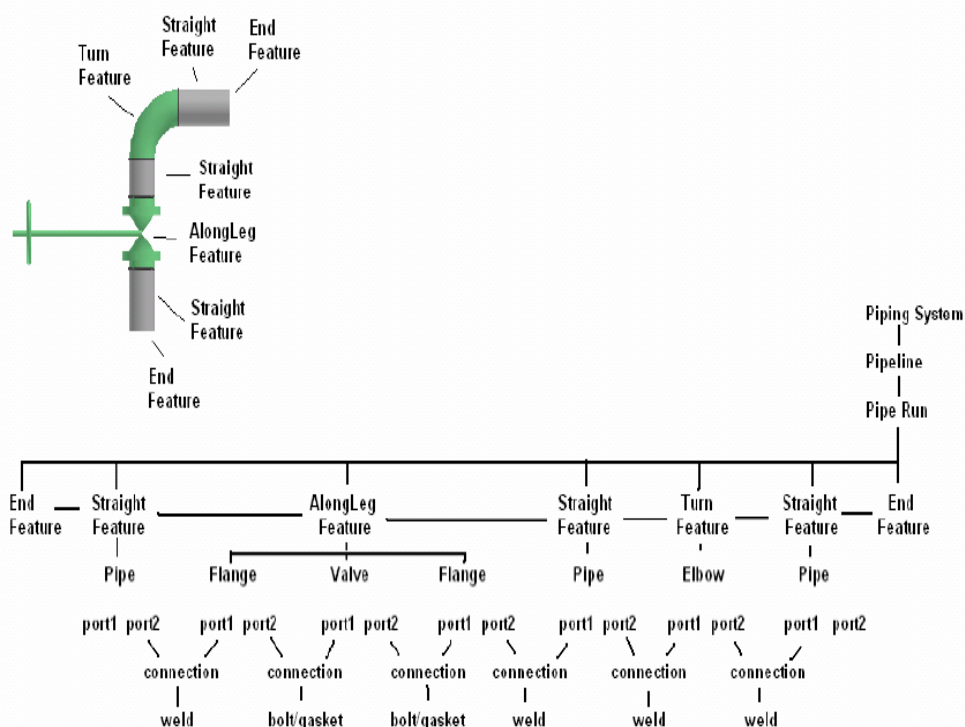
Turn feature - Kääntyvä olio

Run Change Feature - Putkikokoa muuttava olio

Along leg feature - Putkiosuudella oleva olio

Straight Feature - Suora putkiosuus

End Feature - Pääteolio



Kuva 36. Putkiston osaoliohierarkia yleiskuvana.

Määriteltäessä mm. putkiston ominaisuuksia, geometriaa tai valittaessa elementtiä putkiosuudelle, joudutaan usein viittaamaan putkistodataan. Myös portista lähtevät osuudet määritellään porttien ominaisuuksien mukaan.

Seuraavassa on esitelty putkiston osien määrittelyä tietokantatyökalulla ja ”varatuilla sanoilla”. Esimerkissä 1 käydään lävitse tapa muuttaa putkiosuuden (Pipe Run) ominaisuuksia (kuva 10.) lukemalla ne tietokantatyökalusta. Esimerkissä 2 taas esitellään yksi tapa muuttaa elementin kokoa - Run Change Feature.

Esimerkki 1. Tarkastusluokan luku Schema Browserista - Straight Feature.

```
<PipeRun CreateLike="Element(1)" Flow="{Flow}"
IJUAIInspectionClass.NJ_Inspection_Class="{NJ_Inspection_Class}" NPD="1in"
NameRule="PipeRunNJ" PART-IJSequence.Id="NJ_NP4-17434_6_H"/>
```

IJUAIInspectionClass on tietokannassa oleva käyttöliittymänimen tietokantavastine, jonka attribuutin tietuetta NJ\_Inspection\_Class koodissa luetaan.

Esimerkki 2. Geometriatyyppi - Run Change Feature.

```
<Item ShortCode="{ClosingItem}" OptionCode="{ClosingItemOption}"
  GeometryType="RUNCHANGE" Flow="NOFLOW" Spec="A3N" NPD="1in"/>
```

Geometriatyyppin määrittelyllä osoitetaan ohjelmalle, että kyseinen komponentti on eri kokoa kuin edellinen ja sen uusi koko on määritelty varatun Spec-sanan tietueen A3N mukaan.

Kun browserista etsitään tietoa ominaisuuksista, joiden arvot halutaan määrittellä esimerkiksi putkelle on hyvä ymmärtää kyseisen selaimen rakenne, josta seuraavassa.

Browserissa putkiosuusmäärittelyt ovat niin sanotun Parent-Child suhteen mukaisessa hierarkiassa.

Esimerkki 2. Mallinnus browser näkymästä Parent-Child suhteessa.

- CPMPipeRun
- IJRtePathRun
- IJSystemChild
  - SystemParent
    - SystemHierarchy
      - SystemParent
        - SystemChildren
          - + IJAllowableSpecs

Jos esimerkissä kaksi etsittäisiin tietoa jostakin eristykseen liittyvästä tiedosta, lukisi IJRtePathRun nimikkeen tilalla IJRteInsulation. Tällöin eristyksestä löytyisi saman verran alakenttiä, joissa olisi jaoteltu eristys aina tarkentuvien määrittelyjen mukaan. Havainnollistuksen vuoksi kuvassa 37 on esitetty luokan-putkiosuus alta löytyvät käyttöliittymämäärittelyt.

## PipeRun Interfaces



Kuva 37. Putkiosuuden sisältämät käyttöliittymämäärittelyt. (Intergraph. 2011)

### 8.4 Tietokannasta löytyvät komponentit ja niiden tärkeimmät määritteet

Putkiluokka: Putkiluokalle käyttöliittymänimi on IJUAInspectionClass, missä UA viittaa käyttäjän antamaan attribuuttiin ohjelman rajaamien arvojen välissä.

Putkiluokka määrittelee käytettävän putken sisällä virtaavan aineen, korroosio-olosuhteiden, paineen ja lämpötilan keston sekä putken koon ja materiaalin. Suunnittelijan on tiedettävä käytettävä putkiluokka, jotta putkistosuunnittelu voidaan aloittaa. Putkiluokka on looginen kokoelma komponenteista, jotka saavat määrittelynsä putken erityisominaisuuksien mukaan. Putkiosuuden reititystä tehtäessä, voidaan putkeen lisätä kyseisiä luokkaominaisuuksia; nämä ominaisuudet määrittelee ylemmän tason suunnittelutieto. (Joronen, O-P & Viertävä, J. 2008)

Putkiston halkaisija: Putkiston halkaisijaan viitataan ohjelmoinnissa lyhenteellä NPD. Nominal Pipe Diameter -termillä tarkoitetaan nimellistä putkiston halkaisijaa. Kaikki putkiosuudet mallissa säännellään saman putkiston erityisominaisuuksien mukaan. Kaikki putkiluokan ominaisuudet kuuluvat putkiosuuteen. Putkilinja voi sisältää useamman putkiosuuden.

Putkistoon sijoitettavia komponentteja: Putkistoon voidaan ohjelmoida erilaisia venttiileitä, mutkia, nippoja ja muita toimintoja tekeviä laitteita/elementtejä, joista seuraavassa tärkeimpiä on esitetty Excel-taulukon muodossa ohjelmointi nimikkeillään ja suunnittelu tarkoituksillaan.

Taulukko 1. Putkistosuunnittelun käytetyimmät oliot.

YLÄLUOKKA	ALALUOKKA	KÄYTTÖTARKOITUS SUUNNITTELUSSA
Venttiilit - Valves	Butterfly Valve	Virtauksen säätö
	Gate Valve	
	Check Valve	
	Water Check Valve	
	Need Valve	
	Vent-Drain Valve	
Laipat - Flanges	Flange	Sokeointi
	Orifice Flange	
	Orifice Flange 90deg	
	Orifice Flange 180deg	
Haaroitukset - Branches	Tee	Ohitukset
	Nipolet	
	Endolet	
	Weldolet	
Supistukset - Reducers	Eccentric Size Change	Putkikoon muutos
	Concentric Size Change	
Splits - Ositukset	Weld Joint	Putkien jatkaminen
	Butt Weld	
	Take DownJoint	
	Feature Break	
Tap - Jatkokohta	Tap 0,38-48 in	Itse aseteltavan jatkopisteen paikan valinta
Muita - Other	Spectacle Blind	
	Reinforcing Pad	
	Clamp On Sensors	
	Elbolet	
	Union	
	Nipple	
Elbow		

Taulukossa 1 on esitetty komponentteja, joita suunnittelijat käyttävät mallissa ja joista kaikkiin muihin paitsi Tap-olioon on mahdollista viitata myös parametrisen ohjelmoinnin koodissa.

## 9 TOTEUTETUT OHJELMOIDUT MALLIT

Tässä luvussa esitetään työssä tehdyt kokoonpano mallit nimeltä ja selvitetään miksi juuri kyseiset mallit valittiin tehtäviksi kokoonpanoiksi.

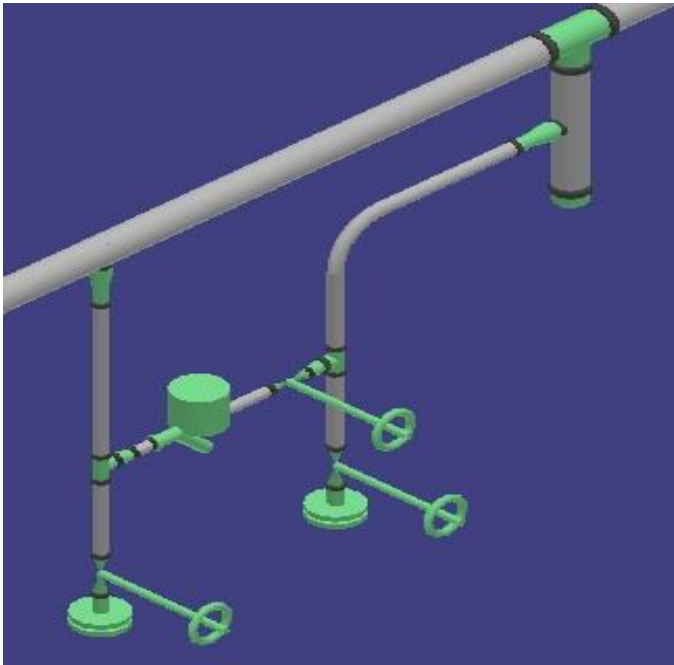
Malleja tehtiin kaikkiaan noin kolmekymmentä. Alla on lueteltu toteutettujen mallien tyypit:

- Erilaiset ilmanpoistot
- Erilaiset tyhjennykset
- Ulospuhallukset
- Lämpötilan- ja paineen mittaukset
- Orifice - laippa
- RO - Levy
- Korroosiosensori
- T- sihti
- Säätoventtiiliryhmä
- Lauhteenpoistokokoonpano

Mallityypit valittiin putkistosuunnittelijoiden kokemusten perusteella. Kyseiset mallityypit ovat yleisimmät putkistosuunnittelussa käytetyt putkistosuunnittelukokonaisuudet, ja sitä varten ne valittiin työn malleiksi.

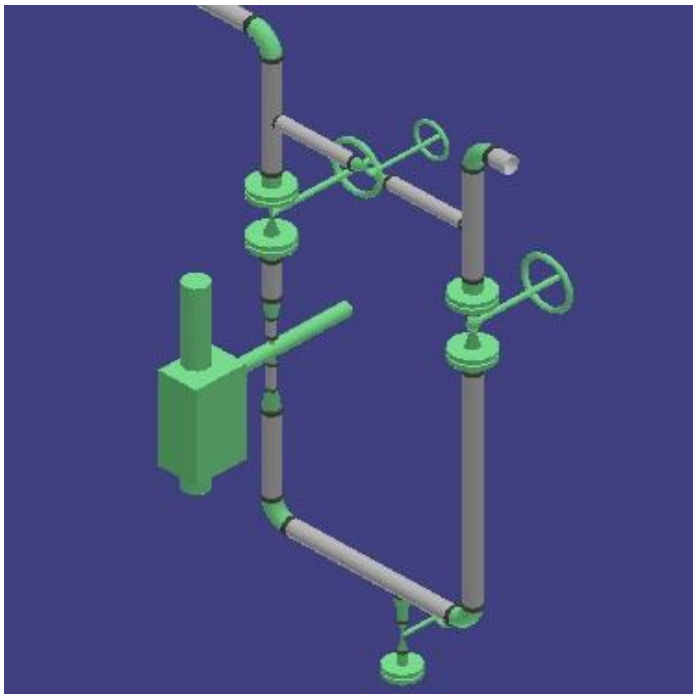
Seuraavassa on esitetty neljä vaativinta koodattua kokoonpanoa, niiden parametrejä ja käyttötarkoitusta.





Kuva 35. Lauhteenpoisto.

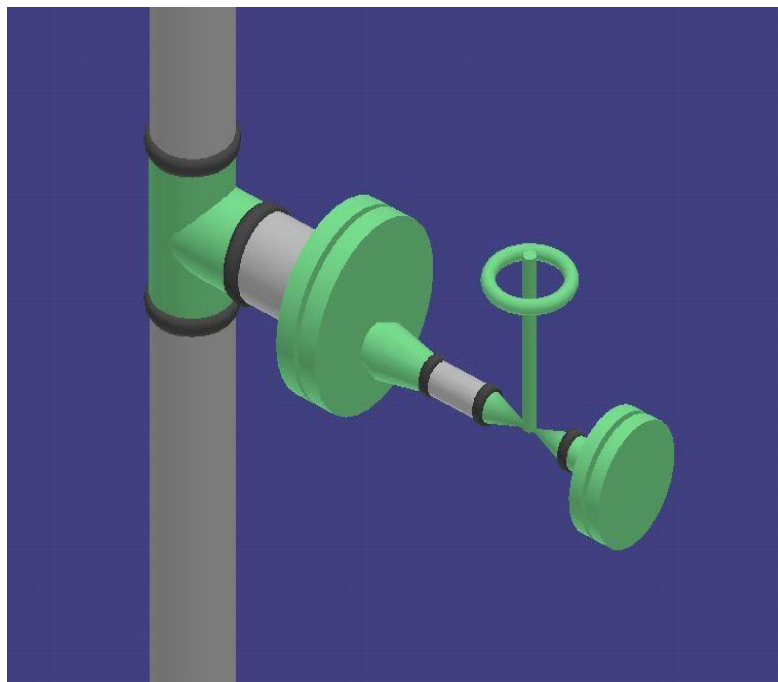
Lauhteenpoistokokoonpano sisältää useita venttiilejä ja lauhde "ansan". Sitä käytetään, jotta lauhtunut vesi saadaan ulos järjestelmästä tyhjennysansan avulla. Sen parametrejä ovat muun muassa kokoonpanon leveys, korkeus, venttiilien karan asento, komponenttityypit ja niiden suunnat.



Kuva 36. Säätöventtiiliryhmä.

Säätöventtiiliryhmä on monimutkainen kokonaisuus niin koodata, kuin suunnitella malliin. Se on jopa tärkein yksittäinen luotu kokoonpano. Säätöventtiiliryhmiä käytetään, kun virtausta halutaan säädellä.

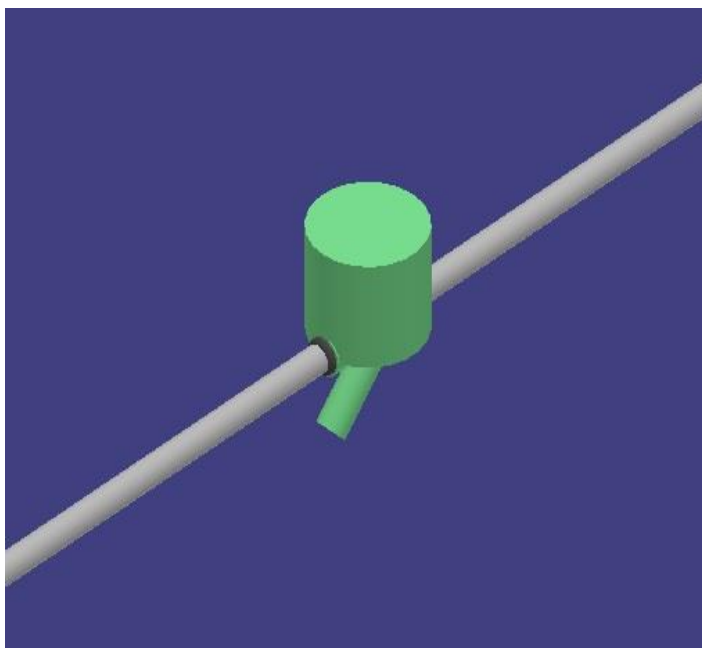
Säätöventtiiliryhmän parametrejä ovat muun muassa säätöventtiiliryhmän leveys, korkeus, venttiilientyypit, karan asento, toimilaitteen tyyppi sen ominaisuudet sekä kiinnityspiste.



Kuva 37. T-sihtti.

T-sihtti on kahdessa kokoonpanolla tehtävä kokonaisuus. Tämä johtuu siitä, että SP3D:n koodissa on mahdoton käyttää Tap-rakenneliota, jolla epäkeskeinen tyhjennyslähtö toteutetaan. Tap-olion lisääminen jää siis vielä kokoonpanon käyttäjälle.

T-sihtin parametrejä ovat muun muassa Venttiilin karan asento ja kokoonpanon ryhmitys suunta. T-sihtti tulee lähes aina poikkiputkeen. Lauhdeansa on lauhteenpoistoa varten. Sen parametrejä ovat geometria sekä muun muassa paineluokka porttien kiinnitystavat sekä lauhteenpuhallussuunta. Lauhdeansalla saadaan järjestelmään kertyvä haitallinen lauhtunut vesi poistettua kätevästi.



Kuva 38. Lauhdeansa.

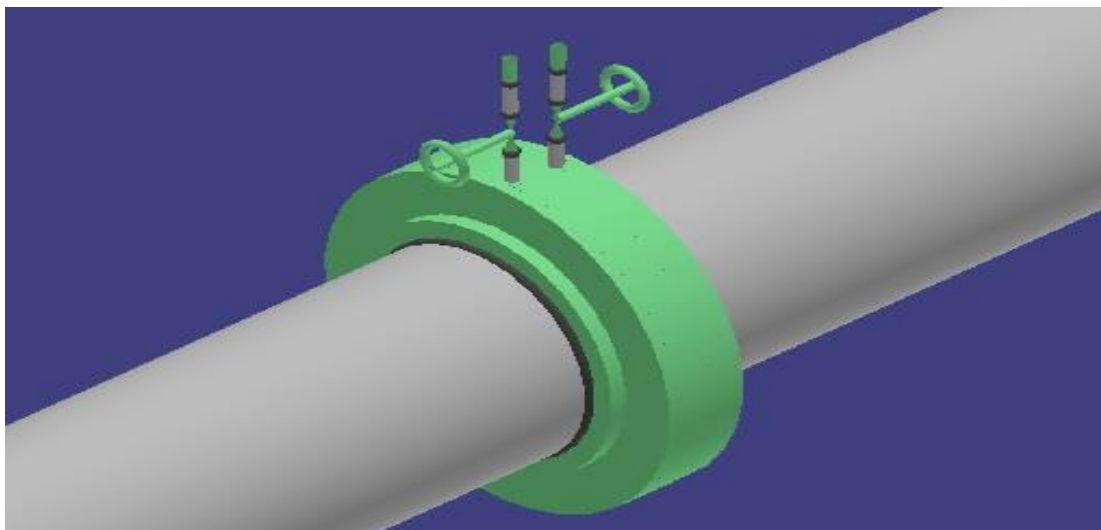
Seuraavassa on esitetty yhden luodun kokoonpanon kuva, isometri, sen osaluettelo sekä kokoonpanon täydellinen koodi, jonka Parameter Name-osiosta näkyvät kaikki käyttäjän valittavissa olevat parametrit.

### 9.1 Esimerkkiohjelma

Tässä luvussa esitellään esimerkkiohjelma. Työhön sisältyi useita erilaisia kokoonpanoja ja määrittelyjä. Näistä valittiin yksi kokoonpano, jonka avulla voidaan havainnollistaa mahdollisimman monipuolisesti, minkälaisia määrittelyjä ohjelmalla voi tehdä. Seuraavassa alaluvussa on esitetty virtausmittauksen ohjelmakoodi, siitä Isogen-ohjelmalla luotu isometri ja sen osaluettelo.

Periaatteessa jokainen kokoonpano oli oma aikaansaannos, koska NJ:n kokoonpanoissa käytettävät tyypit olivat hyvin erilaisia verrattuna intialaisen ekspertin (Prasad Mantraratnam - Staff Software Scientist at Intergraph) tekemiin malleihin. Monia asioita jouduttiin päättämään omin avuin, sillä esimerkkejä ei ollut olemassa. Yhteensä ohjelmakoodia tuli noin 50 sivua ja valmiita kokoonpanoja noin kolmekymmentä.

Seuraavassa on esitelty yksi työn tuloksena valmistuneista kokoonpanoista. Esittelyssä on hyvä kiinnittää huomio kokoonpanossa esiintyvään määrittelymäärään. Nämä suunnittelija joutuisi tekemään manuaalisesti kokoonpanolle, mikäli kokoonpano luotaisiin ilman valmista makro-ohjelman sijoitus mahdollisuutta.



Kuva 38. Orifice - laippa kokoonpano virtausmittauksiin.

Seuraavassa on esitetty virtausmittauskokoonpanon ohjelmakoodi:

```
<ParametricAssemblyDefinition>
```

```
<Information Name="Orifice Plate assembly with Orifice Flanges"
```

```
Description="Orifice Plate assembly with Orifice Flanges" CreatedBy="Ville Kerttula"  
Date="20-Apr-2014" Version="Revision1"/>
```

```
<InputElements>
```

```
<Element Prompt="Select Pipe / End to Place Orifice Assembly"  
Type="IJRtePipePathFeat"/>
```

```
</InputElements>
```

```
<GIFFile Name="NJ_NP4-21635.jpg"/>
```

```
<Parameters>
```

```
<Parameter Name="OrificePlate" Default="IOP4"/>
```

```
<Parameter Name="Angle" Default="1deg"  
Options="45deg|60deg|90deg|120deg|180deg"/>
```

```

<ParameterFromReferenceElement Name="NPD"
Value="NominalDiameter@Element(1)"/>
<Parameter Name="StraightPipeFeature1" Default="7*{NPD}" Options=""/>
<Parameter Name="StraightPipeFeature2" Default="3*{NPD}" Options=""/>

```

Yllä esitetyllä yliviivatulla ohjelmakoodilla luotiin kokoonpanoon Orifice - laipan molemminpuoliset putkiosuudet. Laipalle on tehtävä tietyn mittaiset putket molemmin puolin, koska kokoonpanon putkien on oltava tiettyä tarkastusluokkaa. Kyseinen yliviivattu ilmaisu päätettiin kuitenkin muuttaa ohjelmakoodissa niin (lihavoidut kohdat), että kokoonpanon tarvitsemien tiettyä tarkastusluokkaa olevien putkien päihin tehtiin ("butt weld") päittäin hitsaukset.

```

<Parameter Name="NJ_Inspection_Class" Default=":10"/>
<Parameter Name="Flow" Default="Undefined"
Options="Up|Down|Both|No|Undefined"/>
<Parameter Name="ShortCode" Default="butt weld"
Options="butt weld|Flange"/>
<Parameter Name="Type" Default="SPLIT"/>
</Parameters>
<Items>
<Instrument Location="0,0,0" PartNumber="{OrificePlate}"
PART-IJFaceToFace.FaceToFace="15mm" SavePort1LocationAs="OrificePlatePort1"
SavePort2LocationAs="OrificePlatePort2"
PART-IJDynamicPipePort1.TerminationClass1="Bolted"
PART-IJDynamicPipePort2.TerminationClass2="Bolted"
PART-IJDynamicPipePort1.TerminationSubClass1="Thru-Bolted Without Bolt Holes"
PART-IJDynamicPipePort2.TerminationSubClass2="Thru-Bolted Without Bolt Holes"
PART-IJDynamicPipePort1.EndPreparation1="FFTBE"
PART-IJDynamicPipePort2.EndPreparation2="FFTBE"
PART-IJDynamicPipePort1.PressureRating1="300"
PART-IJDynamicPipePort2.PressureRating2="300"
PART-IJDynamicPipePort1.EndStandard1="Default"

```

```

PART-IJDynamicPipePort2.EndStandard2="Default"
PART-IJDynamicPipePort1.ScheduleThickness1="Undefined"
PART-IJDynamicPipePort2.ScheduleThickness2="Undefined"
PART-IJDynamicPipePort1.PipingPointBasis1="DualFlow"
PART-IJDynamicPipePort2.PipingPointBasis2="DualFlow"
PART-IJUAIInstrumentDimensions.InstrumentDiameter="150mm"
PART-IJUAIInstrumentDimensions.InstrumentHeight="150mm"
PART-IJUAIInstrumentDimensions.InstrumentLength="10mm"
PART-IJUAIInstrumentDimensions.InstrumentWidth="10mm"
PART-IJSequence.Id="NJ_NP4-21635"/>
<Item Location="OrificePlatePort1" ShortCode="Orifice Flange"
SavePort3LocationAs="OrificeFlangePort1" SavePort2LocationAs="Flange1Port2"
OrientationAngle="{Angle}"/>

<Item Location="OrificePlatePort2" ShortCode="Orifice Flange"
SavePort3LocationAs="OrificeFlangePort2" SavePort2LocationAs="Flange2Port2"
OrientationAngle="-{Angle}"/>

</Items>

<Network>

<PipeLine Use="Element(1)"/>

<PipeRun CreateLike="Element(1)" Flow="{Flow}"
IJUAIInspectionClass.NJ_Inspection_Class="{NJ_Inspection_Class}"
NPD="OrificeFlangePort1" NameRule="PipeRunNJ"/>

<Point From="OrificeFlangePort1" Direction="UP"/>

<Pipe Length="70mm"/>

<Item ShortCode="Gate Valve" Secondary="Secondary" GeometryType="INLINE"/>

<Pipe Length="90mm"/>

<Item ShortCode="Tube Connection"
OptionCode="Swagelok compression tubing end" GeometryType="End"/>

<PipeRun CreateLike="Element(1)" NPD="OrificeFlangePort2"/>

<Point From="OrificeFlangePort2" Direction="UP"/>

```

```

<Pipe Length="70mm"/>
<Item ShortCode="Gate Valve" Secondary="Primary" GeometryType="INLINE"/>
<Pipe Length="90mm"/>
<Item ShortCode="Tube Connection"
OptionCode="Swagelok compression tubing end" GeometryType="End"/>

</Network>

<Items>

<Item Location="-7*{NPD},0,0" ShortCode="{ShortCode}" Type="{Type}"/>
<Item Location="3*{NPD},0,0" ShortCode="{ShortCode}" Type="{Type}"/>

</Items>

<Overrides>

</Overrides>

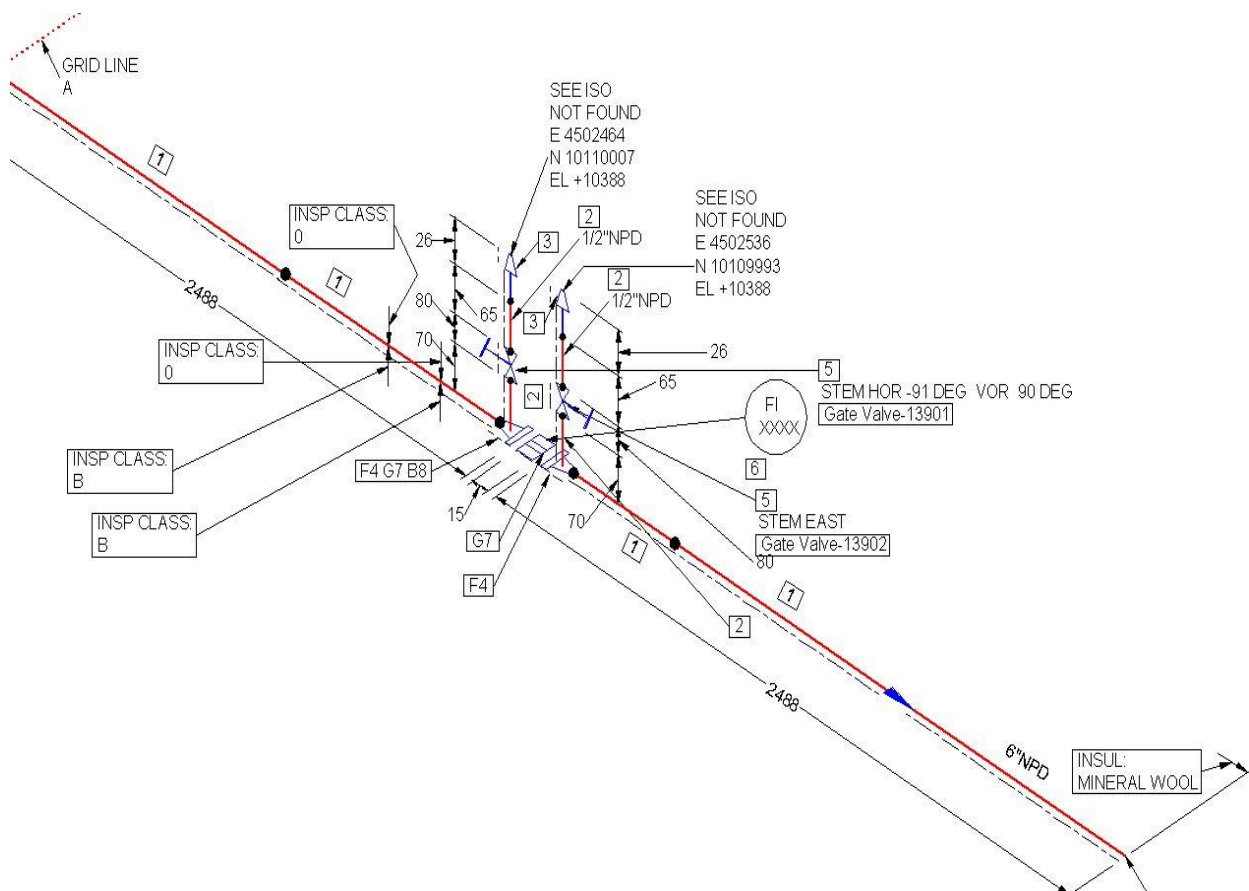
</ParametricAssemblyDefinition>

```

Yllä olevassa ohjelmakoodissa on käytetty Items osiota kahteen kertaan. Näin on pystytty antamaan items määrittelyjä eri ”rakentamisen vaiheissa” muodostettaessa kokoonpanoa malliin. Kokoonpanon rakentuessa pääosin ylhäältä alaspäin (ohjelmakoodissa) voidaan vaihtamalla osioiden järjestystä vaikuttaa kokoonpanon syntymiseen malliin, kokoonpanoa luodessa.

Seuraavalla sivulla on esitetty virtausmittauskokoonpanon luonnista aikaan saatu Isogen-ohjelmalla isomeroitu isometri ja sen osaluettelo.

## Isometri



Kuva 39. Isogen ohjelmalla generoitu isometri Orifice laipan ohjelmakoodista.

## Osaluettelo Isometrasta

Osa Pt.No.	Repro koodi Client Commodity Code	Putkiluokka Pipe Class	Selitys Description	Koko N.S.	Määrä Quantity
1	106004-06	A1N	Pipe BLE SMLS S-40 B36.10M SA333-6	6	5.0 M
2	106008-004/8	A1N	Pipe SQE SMLS S-80 B36.10M SA333-6	1/2	0.3 M
3	CI6921	A1N	Male connector Swagelok A316, SS -12MO-1-8W, NP4-17455 BW/SWG	1/2	2
4	506352-06	A1N	Orfc flg Slip-on CL300 RF - B16.36 SA350-LF2 CL1	6	2
5	300608-004/8	A1N	GATE CL800 BW S80 SA105N API602 OS&Y BB/WB RB Sol.wedge Trim F6 HW	1/2	2
6			FI-XXXX	6	1
7	770315-06		Spir w.gask CL300 NOS5403 (B16.20 for B16.5 flg) 316L/graph-F, CS IR & OR	6	2
8	501816-M20-160		Studb w/2 hex nut NOS5101 B7 / 2H	20	12

Kuva 40. Ohjelmakoodin perusteella luodun isometrin generoima osaluettelo Orifice laipan osista.



## 9.2 Havaittuja ongelmakohtia

Smartplant 3D-ohjelman parametointia tehdessä huomattiin muutamia epäkohtia, joista yksi pakotti mm. tekemään T-sihtin kokoonpanon kahdella makrolla. T-sihtissä, tämä toi esille tilanteen, jossa olisi tehtävä Tap-niminen epäkeskeinen jatkopaikka laippaan. Tulevaisuudessa näitä "TAP" rakenneolioita tulisi koodata kokoonpanojen parantamiseksi mm. epäkeskeisiä T-sihtejä varten SP3D:hen. PDS-laitossuunnitteluohjelmassa TAP-haaroituksia vastaavasti pystytään ohjelmoimaan makroiin.

Toinen mielenkiintoinen asia, joka selvisi vasta ohjelman (SP3D) ohjelmointikehittäjän (Prasad Mantraratnam) konsultoinnin jälkeen, oli ohjelman kyvyttömyys tehdä Note Text tyyppisiä merkintätekstejä isometrin elementeille. Ohjelmointikehittäjältä tuli kuitenkin ohje, että Note Text osion puutteellisuuden pystyy korvaamaan käyttämällä Sequence Id -nimistä osan (PART) ominaisuusmäärittelyä koodissa. Näin olioiden tunnuksia saatiin tulemaan isometriin oikein.

## 10 POHDINTA JA YHTEENVETO

Tämän työn tarkoituksena oli tuottaa NJ:lle tarvittava määrä putkistosuunnittelussa käytettäviä, automatisoivia makroja. Kokoonpanojen ja laitossuunnittelun 3D-mallintamisen idea on havainnollistaa mallia aidon näköisesti sekä antaa isomerisointiohjelmalle arvot, joiden perusteella kyseinen linja pystytään toteuttamaan. Siihen tarvittavat osat pitää pystyä hankkimaan ja linja rakentamaan toimivaksi tämän 3D-mallin isometripiirustuksen perusteella.

Intergraphin asiantuntijan tekemät esimerkit eivät olleet kovinkaan syvällisiä. Esimerkit oli selvästikin suunniteltu vain havainnollistamaan PAL-kielistä ohjelmointia, ei niinkään käytettäväksi (edes osina) sellaisenaan minkään kokoonpanon tekoon. Idea tavasta käyttää PAL-kieltä kokoonpanojen ohjelmointiin tuli kuitenkin selväksi. Ja esimerkkeihin perehtymällä saatoinkin omaksua valmiudet alkaa soveltamaan PAL-kielen ”hahmotelmia” NJ:n vaatimiin huomattavasti monimutkaisempiin malleihin.

Suurin ongelma Intergraphin toimittaman teoria-aineiston ja laitossuunnittelun todellisuuden välillä oli esimerkkien toimimattomuus isomerisointi-ohjelmassa. Tämä ilmeni siten, että esimerkkien pohjalta rakennetut kokoonpanot eivät tulleet oikein isometriin. Monia osia katosi jäljettämiin, kuten tyhjennyksen umpilaippa. Todellisuudessa laitossuunnittelussa 3D-malleista ajetaan isometrit ”valmistuspiirustukset”, joiden perusteella linja pystytään rakentamaan. Tällöin, jos 3D-malliohjelma ei tule isometriin oikein, se ei ole toteutuskelpoinen.

Työssä sovellettiin esimerkkinä ollutta PAL-kielistä ohjelmakoodia niin, että sen lopputuotteet olivat NJ:n spesifikaatioiden mukaisia ja tulivat isometreihin ja osaluetteloihin tyyppikuvien mukaisina.

Voidaan vielä todeta, että NJ:n spesifikaatiot sisälsivät huomattavan määrän tietoa, joka tuli saada toimimaan keskenään. Usein, kun jonkin osa-alueen sai toimimaan, huomasi ettei kyseinen toteutustapa tuekaan jotain toista spesifikaation osa-aluetta, joka piti saada toimimaan mallissa.

Kokoonpanojen tuli muun muassa sisältää kyky lukea tietoja 3D-mallista siitä elementistä (element(1)), jonka putkistosuunnittelija halusi valita kokoonpanon rakentamisen aloituskohdaksi. Tällaisia tietoja olivat esimerkiksi eristyspaksuus, tarkastusluokka, spesifikaatiotunnus, nimeämissääntö, materiaali, halkaisija jne. Usein myös kokoonpano tuli rajata toimivaksi vain johonkin tiettyihin arvoihin. Esimerkiksi säätöventtiiliryhmä (NP4-17448\_E) sai rakentua vain, jos sen kokoonpanon luonnin yhteydessä osoitettava putkilinja olisi halkaisijaltaan 2, 3, 4 tai 6 tuumaa. Tällaisten kokoonpanojen ohjelmakoodien tekeminen on hidasta, mutta palkitsevaa.

Kokonaisuutena opinnäytetyö onnistui mallikkaasti. Tehdyt reilut kolmekymmentä makroa otettiin käyttöön nyt aluksi tuotantolaitoksien TL1 ja TL2 puolella ja NJ:n System Manager laati niiden käytöstä NJ:lle sisäisen laatuohjeen. Smartplant 3D-projektien kasvaessa tulevaisuudessa luotujen kokoonpanojen määrää tultaneen kasvattamaan. PDS tulee jäämään pois ja tässä opinnäytetyössä tehtyjen makrojen tulevaisuuden taloudellisen arvon tulee määrittelemään se, mihin laitossuunnitteluohjelmaan PDS:stä luopumaan tulevat putkistosuunnittelua tarvitsevat yritykset siirtyvät.

## 11 LÄHDELUETTELO

Graves, M. 2002. Designing XML databases. Prentice Hall books. 2. painos.

Heikkilä, A-M & Malmén, Y. VTT.

Turvallisuus prosessien suunnittelussa ja käyttöönotossa.

<http://virtual.vtt.fi/virtual/proj3/alarp/aineisto/luento-06-moduuli-02.pdf>.

[viitattu 2014-07-20].

Hietanen, P. 2001. C++ ja olio-ohjelmointi. Docendo Finland Oy. 7. painos.

INSKO- Seminaarin materiaali. 2003. Prosessilaitoksen layout suunnittelu.

Helsinki: AEL.

Intergraph. 2008. A Gentle Introduction to SmartPlant Schema. Saatavilla Intergraphilta.

Intergraph. 2009. PAL Language Format Reference. Saatavilla Intergraphilta.

Intergraph. 2011. Plant Design System Reference Data Manager (PD\_Data) (V12).

June 2011: DPDS3-PB-200034H. Saatavilla Intergraphilta.

Intergraph. 2009. User Guide. Saatavilla Intergraphilta.

Jaatinen, T. 2009. Putkistorakenteiden suunnittelun perusteet. (H053001/13):

Venttiilien ja varusteiden sijoitus. Luentokansio. Helsinki: AEL.

Joronen, O-P & Viertävä, J. 2008. Putkiluokilla tehoa putkistosuunnitteluun.

Saatavissa: <http://www.promaint.net/downloader.asp?id=2988&type=1>.

[viitattu 2014-07-06].

Kesti, M. 1992. Teollisuusputkistot. Helsinki: VAPK-kustannus.

Myyry, J-M. 2007. Laitossuunnittelun projektiaikataulu. Koulutusmateriaali

Rintekno Oy.

Saatavissa: <http://www.doria.fi/bitstream/handle/10024/5332/stadia-1176715576-2.pdf?sequence=1>.

Neste Jacobsin henkilökunnan kanssa käydyt keskustelut aikavälillä 2.5.2014 - 20.9.2014. Ei saatavilla.

Neste Jacobs Oy. Sisäinen materiaali.

Ei saatavissa: [http://finj01vsha01/handbookJulkaisu/Handbook/Virtausmittauskohdat\\_ja\\_mittalevyt.doc](http://finj01vsha01/handbookJulkaisu/Handbook/Virtausmittauskohdat_ja_mittalevyt.doc). Laatija/Issued by: Vesa Mikkolainen. [viitattu 2014-04-08].

Neste Jacobs. 2014. Yritysesittely [online].

saatavissa: <http://www.nestejacobs.com/index.php?page=2&lang=en> [viitattu 2014-07-18].

Niemi, R. 2014. Neste Oil putkisto specifikaatiot. H102 Annex A: Pipe Classes and Colour markings -table. Saatavuus: Neste Jacobsin sisäistä materiaalia.

Pajuniemi, K. 2009. Putkistorakenteiden suunnittelun perusteet (H053001/5): Teollisuusputkistot: Olemassaolevat standardit ja niiden hyödyntäminen. Luentokansio. Helsinki: AEL.

Pere, A. 2009. Koneenpiirustus 1&2. Espoo: Kirpe Oy.

PSK 2401 Putkiston virtausnopeudet. 2009. Helsinki: PSK Standardisointiyhdistys ry. Saatavissa: <http://www.psk-standardisointi.fi/Alasivut/Standardiluettelo.htm>. [viitattu 2014-08-15].

PSK 0201 Teollisuusventtiilit. 2011. Valinta ja käyttösuositus 8.9.2011. Helsinki: PSK Standardisointiyhdistys ry. Saatavissa: <http://www.pskstandardisointi.fi/Alasivut/Standardiluettelo.htm>. [viitattu 2014-07-08].

Ranta, R. 1997. Yleissuunnittelun ohjekirja: NP4-21600. Kotka: Neste Engineering.

Rintekno Oy. 2007. Laitossuunnittelun projekti aikataulu. Koulutusmateriaali.

Saatavissa: <http://www.doria.fi/bitstream/handle/10024/5332/stadia-1176715576-2.pdf?sequence=1>. [viitattu 2014-07-18].

Toivonen, R. 2011. H101 ANNEX 4. Putkistopiirustusten nimeäminen ja sisältö. Saatavuus: Neste Jacobsin sisäistä materiaalia.

Tähti, M. Neste Oil Naantalin vetyverkon virtauskaavio. Opinnäytetyön tulos. Helsingin Ammattikorkeakoulu. Stadia. [online].

Saatavissa: [https://www.theseus.fi/bitstream/handle/10024/26911/Tahti\\_Matti.pdf?sequence=1](https://www.theseus.fi/bitstream/handle/10024/26911/Tahti_Matti.pdf?sequence=1). [viitattu 2014-07-25].

## 12 KUVALUETTELO

<i>Kuva 1.</i> Kokoonpanon käyttö- ja määrittelyikkunat.....	10
<i>Kuva 2.</i> Laitosinvestointiprojektin vaiheet.....	14
<i>Kuva 3.</i> Laitossuunnittelun tiedonkulku.....	15
<i>Kuva 4.</i> Dokumenttien esiintyminen prosessisuunnittelussa vaiheittain.....	18
<i>Kuva 5.</i> PI-kaavio prosessisäiliöstä.....	19
<i>Kuva 6.</i> Putkistoisometri.....	22
<i>Kuva 7.</i> Suodatuksen eli Filtrerin valinta.....	24
<i>Kuva 8.</i> Clippaus rajatun alueen mukaan. Ylemmässä kuvassa näkyy koko rakennus ja alempaan kuvaan on rajattu siitä ensimmäisen kerroksen etuosa.....	25
<i>Kuva 9.</i> Grid-viivastot, Kymenlaakson ammattikorkeakoulun koulutusmateriaali. ....	26
<i>Kuva 10.</i> Structure-tason toiminnoilla rakennettu yksikkö.....	27
<i>Kuva 11.</i> Lämmönvaihtimen valinta malliin, Place Equipment toimintolaatikolla. ....	28
<i>Kuva 12.</i> Systems and Specifications-osion näkymä 1.....	29
<i>Kuva 13.</i> Systems and Specifications-osion näkymä 2.....	29
<i>Kuva 14.</i> Smartplant 3D-ohjelman Drawings & Reports-osion yhteys isometrin generointiin.....	30
<i>Kuva 15.</i> Komponentin lisääminen putkiosuuteen Place Component-toiminnolla. Kuvassa näkyy sama putkilinja kolmesta eri kuvakulmasta.....	31
<i>Kuva 16.</i> Kokoonpanon sijoitusikkuna.....	39
<i>Kuva 17.</i> Kokoonpanon sijoittamisen ohjeistusta.....	40
<i>Kuva 18.</i> PDS-ohjelman moduulit.....	42
<i>Kuva 19.</i> Kokoonpanon sijoituslomake eli formi.....	43
<i>Kuva 20.</i> Luotu kokoonpanokirjasto.....	44

<i>Kuva 21.</i> Parametroitavia asioita. ....	48
<i>Kuva 22.</i> Kokoonpanon sijoitusikkuna PDS. ....	49
<i>Kuva 23.</i> Mallikuvan täydennys kuvankäsittelyohjelmalla. ....	54
<i>Kuva 24.</i> Säätoventtiiliryhmän validointi sijoitusikkunassa. ....	58
<i>Kuva 25.</i> Leikkaus H102 putkiluokkastandardista – Ilmanpoistolle.....	82
<i>Kuva 26.</i> Leikkaus - Standardin mukaisista tyyppiirustuksista - H104 (Yksinkertainen ulospuhallus kokoonpano). ....	83
<i>Kuva 27.</i> Piirto-ohjelma hahmotelma tyyppiirroksen H104 TYPE A kokoonpanosta.....	83
<i>Kuva 28.</i> Kokoonpanon luonti-ikkuna. ....	88
<i>Kuva 29.</i> Valmis kokoonpano kahdesta eri kuvakulmasta. ....	88
<i>Kuva 30.</i> Lokitiedot ovat nähtävissä sijoituksen jälkeen Log-ikkunassa (View-Log). ....	89
<i>Kuva 31.</i> ParametricAssemblies-kokoonpanokansio. ....	92
<i>Kuva 32.</i> Säätoventtiiliryhmän mallikansion sisältämät kokoonpanot.....	93
<i>Kuva 33.</i> Kokoonpanokansion sisältö.....	94
<i>Kuva 34.</i> Schema Browser – päänäkymä. ....	95
<i>Kuva 35.</i> Tarkastusluokka (InspectionClass) näkymä SB:ssä.....	98
<i>Kuva 36.</i> Putkiston osaolihierarkia yleiskuvana .....	100
<i>Kuva 37.</i> Putkiosuuden sisältämät käyttöliittymämäärittelyt.....	102
<i>Kuva 38.</i> Orifice laippa kokoonpano virtausmittauksiin .....	108
<i>Kuva 39.</i> Isogen ohjelmalla generoitu isometri Orifice laipan ohjelmakoodista. ....	112
<i>Kuva 40.</i> Ohjelmakoodin perusteella luodun isometrin generoima osaluettelo Orifice - laipan osista.....	112



## LIITTEET

## Liite 1. Olio-ohjelmointi sanasto 1(2)

<i>Aliluokka - Sub Class</i>	Periytymishierarkian luokka, jolla on omien ominaisuuksiensa lisäksi ylläluokassa määritellyt ominaisuudet (tiedot).
<i>Attribuutti - Attribute</i>	On toteutettuna: Tietoa, olio tai metodi, joka sisältää luokkansa ominaisuudet.
<i>Kirjasto - Library</i>	Käyttäjän nimeämä "arkisto" uudelleenkäytettävistä aliohjelmista tai luokista.
<i>Komponentti - Component</i>	Ohjelmiston valmisosa; luokkakokoelma tai aliohjelma, jolla on joitain ennalta määriteltäviä parametrejä.
<i>Lohko - Block</i>	Ilmaisee lohkosulkeilla toisiinsa liitetyt lauseet.
<i>Luokkakirjasto - Class Library</i>	Uudelleen käytettävien luokkien nimetty kokoelma.
<i>Luokkatyyppi - Class Type</i>	Ohjelmoijan määrittelemä tietue, joka sisältää yhteen kuuluvien tietojen sekä aliohjelmien määrittelyn.
<i>Malli - Template</i>	Tyyppiriippumaton ohjelmakoodi.
<i>Määrittelyluokka - Definition Class</i>	Oliolle tehty tilanvaraus luokka.
<i>Olio - Object</i>	'Tietojen ja palvelujen kokonaisuus. Luokan ilmentymä. Olio on ohjelman toteutuksen kannalta ajonaikainen keskusmuistista tehty tilanvaraus'.
<i>Olio-ohjelmointi</i>	❖ C++ ja olio-ohjelmointi, Päivi Hietanen Ohjelmointitekniikoihin perustuva ohjelmointityyli, joka sisältää kapselointia, periytymistä sekä monimuotoisuutta.

<i>Operaattori - Operator</i>	Merkintätapa/Tilanvaraus, jota käytetään (pohjautuen C++ kielen yleisiin kirjoitus sääntöihin). Lähes kaikille matemaattisille toiminnoille löytyy tällainen symbolinen merkintätapa ohjelma koodiin.
<i>Parametri - Parameter</i>	Tiedonvälitysmuuttuja, esimerkiksi: mitta, komponenttinimi, määrittely. Ohjelmasta toiseen välittyvä tieto.
<i>Vakio parametri</i>	Muuttumattomaksi määritelty parametri.
<i>Viittaus - Reference</i>	Muuttujalle määritelty nimi, johon voidaan viitata epäsuorasti uudelleen.
<i>Vertailulause - Comparing sentence</i>	If-Then-vertailulause; jos joku on jotain, tällöin jotain tapahtuu.
<i>Varattu sana</i>	On sana, jolla on ennalta määritelty merkitys ohjelmassa. Se voi olla lause määre tai operaattori. Kaikille edellä mainituille on kuitenkin yhtenäistä se, että niille löytyy koodattu tietue, joka pitää sisällään ”varatun sanan” palvelun, jossakin tietokannassa.

## Liite 2. PDS-assembly, ilmanpoisto

```
# BW_ILMANPOISTO_NP4_17433_SH2_TYPE_UVA
PAL 'VEN_TYP_A_BW'
DISPLAY_TUTORIAL = 'PAS000'
SECOND_SIZE = 0.75
PLACE BRANCH , 90_DEG , BY CENTER
IF ( NUMBER_CPS .EQ. 2 ) THEN
    REF_PNT_A = CP2
ELSE
    REF_PNT_A = CP3
ENDIF
CONNECT TO REF_PNT_A
PROMPT_MESSAGE = 'ENTER NEW PIPING O-SPEC'
DISPLAY = PROMPT_MESSAGE
MATERIALS_CLASS_B = USER_INPUT
ACTIVE_MATERIALS_CLASS = MATERIALS_CLASS_B
PLACE FITTING , 6Q2C01 , BY CP2
T34C46 = 'SEE_NP4_17033_SH_2_TYPE_A'
PLACE FITTING , 6Q2C08 , BY CP1
END
```

## Liite 3. SP3D-assembly, ilmanpoisto 1(2)

```

<ParametricAssemblyDefinition>

<Information Name="Vent - Pipe + Valve + ClosingItem" Description="Piping Vent As-
sembly with Closing Item" CreatedBy="Ville Kerttula" Date="7-Apr-2014" Ver-
sion="Revision1"/>

<ReferenceElements>

<Element Prompt="Select Pipe Feature / Pipe Part to Place Vent / Pressure Rating
Must Be Between 150# - 600#"/>

</ReferenceElements>

<GIFFile Name="NJ_NP4-17433_2_A.jpg"/>

<Parameters>

<Parameter Name="D" Default="100MM" UnitType="DISTANCE" />

<Parameter Name="NJ_Inspection_Class" Default=":10" />

<Parameter Name="Flow" Default="Down" Op-
tions="Up|Down|Both|No|Undefined"/>

<ParameterFromReferenceElement Name="Spec" Value="Spec@Element(1)" />

<Parameter Name="Branch" Default="Nipolet" Options="Nipolet"/>

<Parameter Name="ClosingItem" Default="Blind Flange" Options="Blind Flange"/>

<Parameter Name="ClosingItemOption" Default="Default" Options="300|+300"/>

</Parameters>

<Network>

<PipeLine Use="Element(1)"/>

<ConditionallyExecute IF='("{Spec}".EQ."A1N") OR ("{Spec}".EQ."A3N") OR
("{Spec}".EQ."A6N") OR ("{Spec}".EQ."K1N") OR ("{Spec}".EQ."K3N") OR
("{Spec}".EQ."K6N") OR ("{Spec}".EQ."L1N") OR ("{Spec}".EQ."L3N") OR
"{Spec}".EQ."L6N"' Prompt="Assembly Valid only for Specs - A1...6N OR K1...6N OR
L1...6N">

<PipeRun CreateLike="Element(1)" Flow="{Flow}" IJUAInspection-
Class.NJ_Inspection_Class="{NJ_Inspection_Class}" NPD="0.75in"
NameRule="PipeRunNJ" />

```

```
<Point X="0" Y="0" Z="0" />
```

```
</ConditionallyExecute>
```

```
<Point Distance="{D}" Direction="UP" Flush="true" />
```

```
<Point Distance="2mm" Direction="UP" SaveLocationAs="A"/>
```

```
<PipeRun CreateLike="Element(1)" Type="RUNCHANGE" Spec="A3N" NPD="1in"  
Flow="{Flow}" IJUAInspectionClass.NJ_Inspection_Class="{NJ_Inspection_Class}"  
NameRule="PipeRunNJ"/>
```

```
<Point Location="A"/>
```

```
<Point Distance="2mm" Direction="UP"/>
```

```
<Item ShortCode="Flange"/>
```

```
<Item ShortCode="Blind Flange" GeometryType="END" PART-  
IJSequence.Id="NJ_NP4-17433_2_A"/>
```

```
</Network>
```

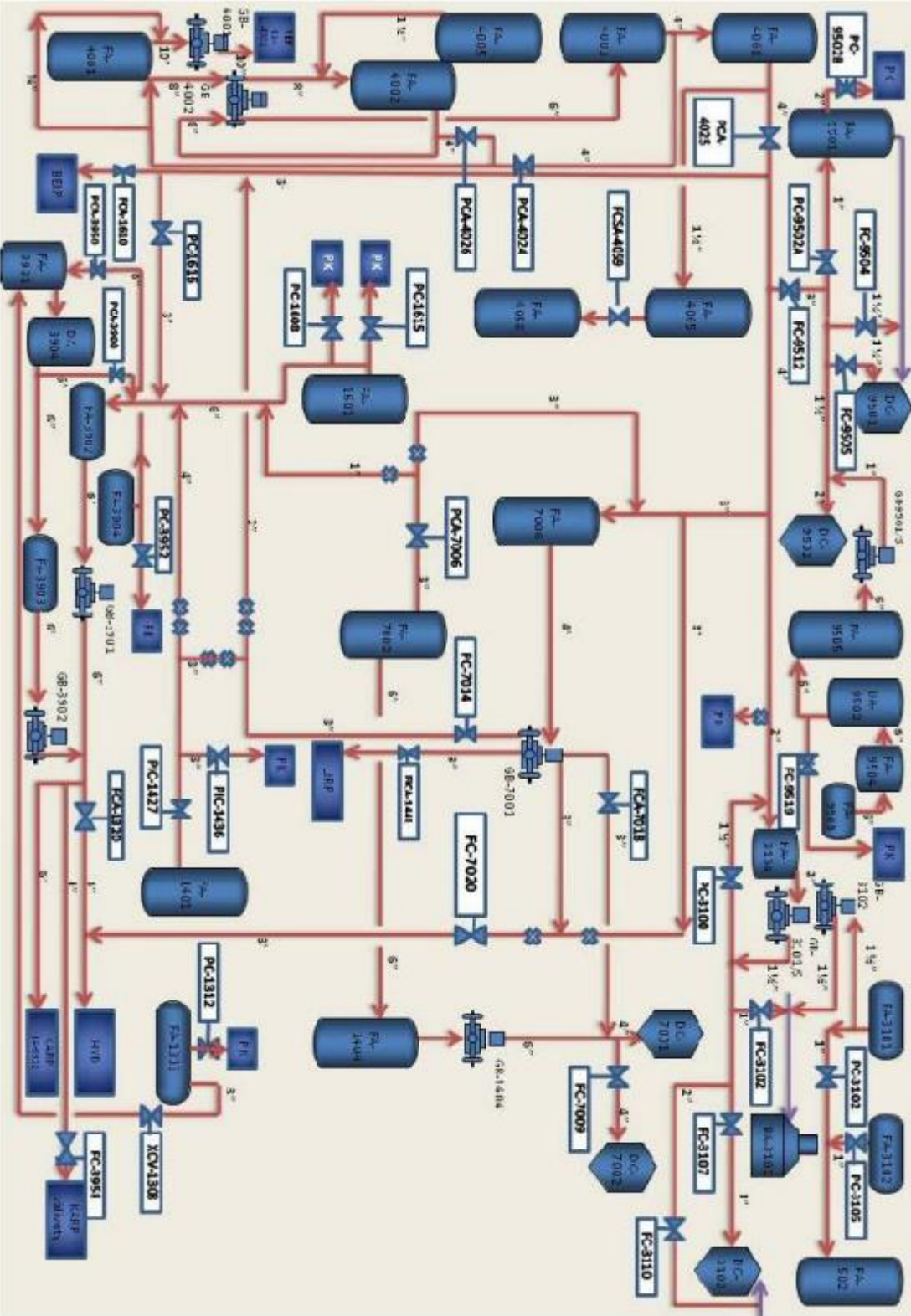
```
<Overrides>
```

```
</Overrides>
```

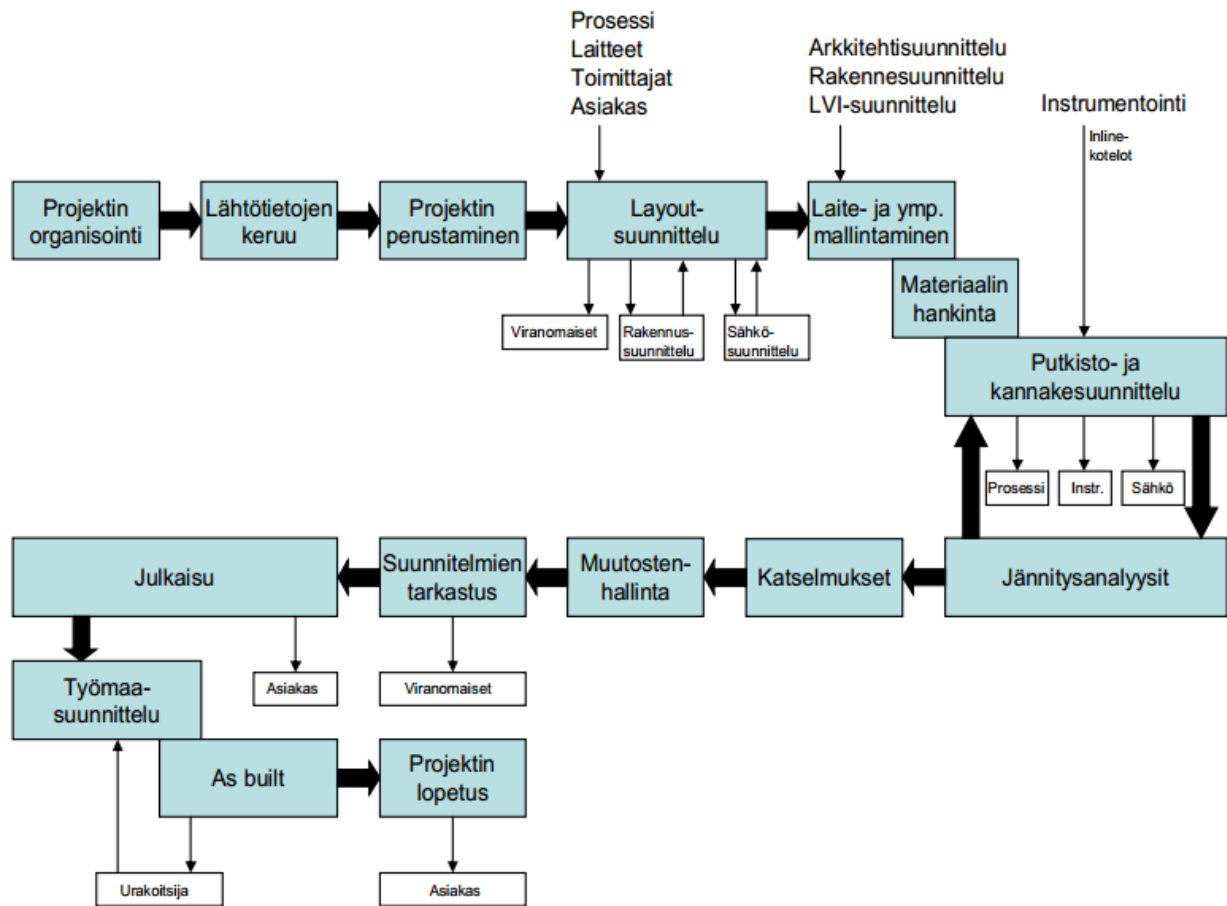
```
</ParametricAssemblyDefinition>
```

- Havainnollistuksen vuoksi varsinainen toimintakäskyalue on **tummennettuna**, SP3D mallissa ja kokoonpanon sijoitusta rajaavat parametrit *kursivoituina*.

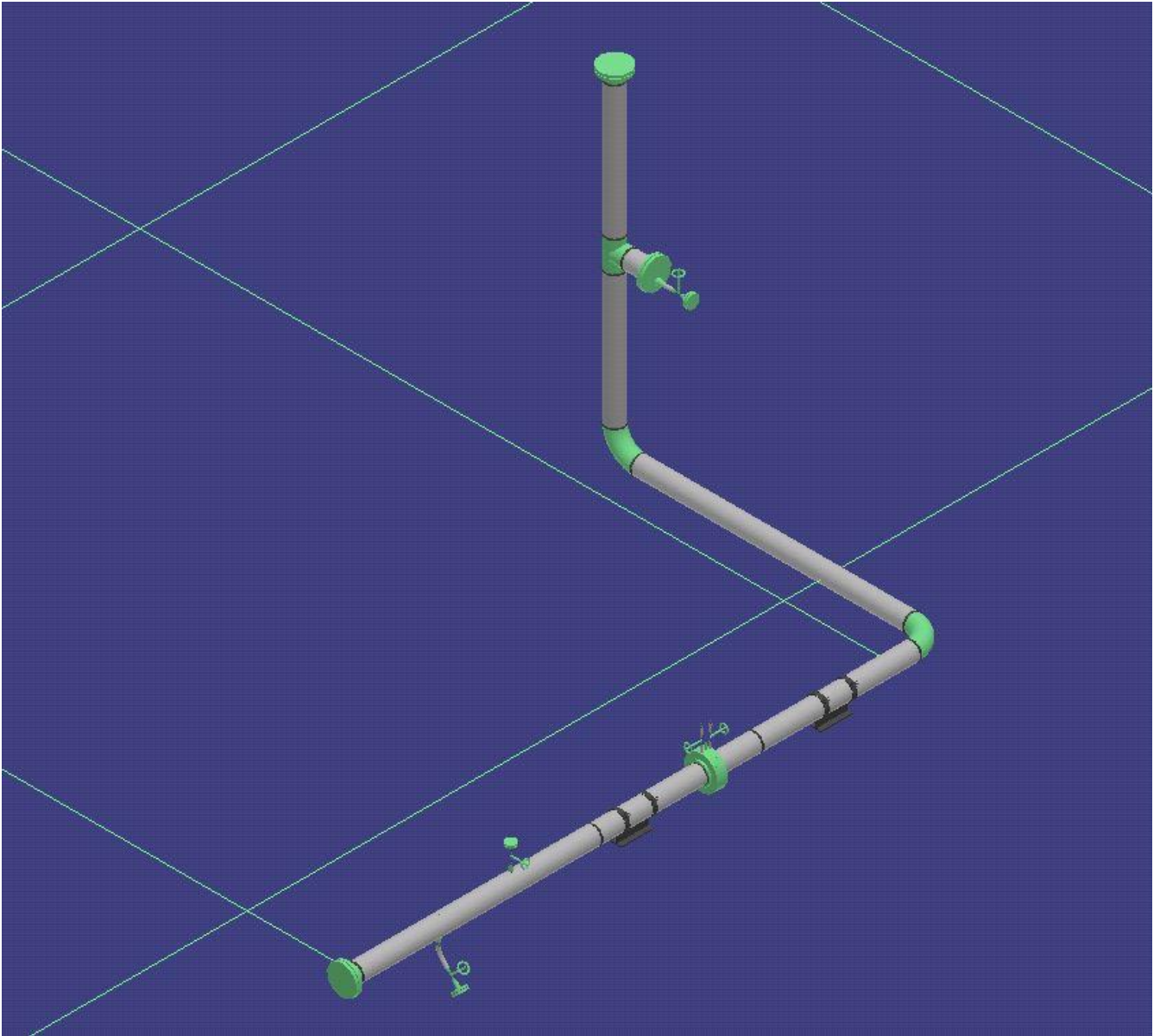
Liite 4. Vetyverkon virtauskaavio



## Liite 5. Laitossuunnittelun kierto



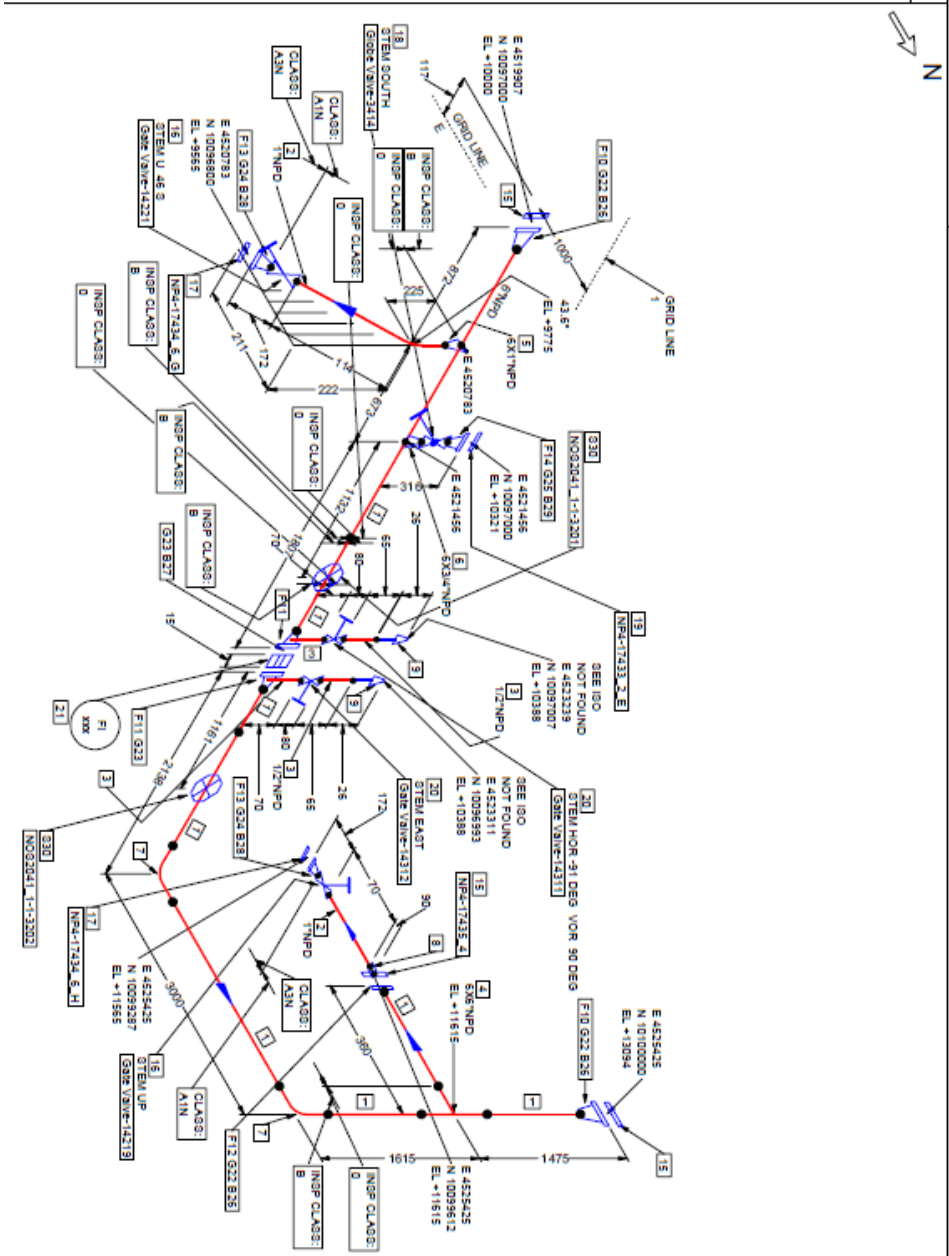
## Liite 6. Putkistolinja, isometri ja osaluettelo 1(2)



\*Yllä esitetyn mallin osaluettelo ja isometri löytyvät seuraavalta sivulta.



Qty	Part Used	Part Name	Notes	Unit	Material
1	108004-06	AIN	Pipe BLE SMLS S-40 B36 10M SA333-6	6	10.5 M
2	108008-01	AIN	Pipe SOE SMLS S-80 B36 10M SA333-6	1	0.3 M
3	108008-0048	AIN	Pipe SOE SMLS S-80 B36 10M SA333-6	1/2	0.3 M
4	806204-06	AIN	TEE BW S-40 SMLS B16.5 SA420-WPL6	6X6	
5	806873-01	AIN	Nipolet Length 3.5' BW S-80 Manufacturer	6X1	
6	806873-0068	AIN	Nipolet Length 3.5' BW S-80 Manufacturer	6X3/4	
7	808004-06	AIN	90 EILR BW S-40 SMLS B16.5 SA420-WPL6	6	2
8	808873-01	AIN	Nipolet Length 3.5' BW S-80 Manufacturer	1	1
9	CI9821	AIN	Male connector Swagelok A316 SS	1/2	2
10	508004-06	AIN	-12MC-18W, NP4-17455 BW/SWVG	6	2
11	508352-06	AIN	Orfe flg Slip-on CL300 RF - B16.36	6	2
12	508310-06	AIN	SA350-LF2 CL1	1	1
13	508308-01	ASN	Fig Slip-on CL150 RF - B16.5 SA350-LF2 CL1	1	2
14	508108-0068	AIN	Fig WN CL300 RF S-80 B16.5 SA350-LF2 CL1	3/4	1
15	508140-06	AIN	Blind flg CL150 RF - B16.5 SA350-LF2 CL1	3	3
16	300808-01	AIN	GATE CL800 BW S80 SA105N API602 OS&Y BBWB 1	2	2
17	508342-01	A2N	Blind flg CL300 RF - B16.5 SA350-LF2 CL1	1	2
18	300839-0068	AIN	GL08B B16.5 SA105N ISO19761 OS&Y	3/4	1
19	508140-0068	AIN	BBWB Plug type disc Trim F8 HW	3/4	1
20	300808-0048	AIN	Blind flg CL150 RF - B16.5 SA350-LF2 CL1	2	2
21	770115-06	AIN	GATE CL800 BW S80 SA105N API602 OS&Y BBWB 1/2 RB Solwedge Trim F8 HW	1	1
22	770115-06	AIN	Floor	6	3
23	770315-06	AIN	Spr w gask CL150 NOS5403 (B16.20 for B16.5 flg) 310L/graph-F, CS IR & OR	6	2
24	770815-01	AIN	Spr w gask CL300 NOS5403 (B16.20 for B16.5 flg) 310L/graph-F, CS IR & OR	1	2
25	770115-0068	AIN	Spr w gask CL150 NOS5403 (B16.20 for B16.5 flg) 310L/graph-F, CS IR & OR	3/4	1
26	501816-M2C-120	AIN	Studs w/2 Hex nut NOS5101 B7 / 2H	20	24
27	501816-M2C-180	AIN	Studs w/2 Hex nut NOS5101 B7 / 2H	20	12
28	501816-M12-070	AIN	Studs w/2 Hex nut NOS5101 B7 / 2H	16	8
29	501816-M12-070	AIN	Studs w/2 Hex nut NOS5101 B7 / 2H	12	4
30	NOS2041_1	AIN	300 - 100 - 0 - NOS2041	6	2



## Liite 7. Luotuja kokoonpanoja (17/32) 1(2)

