

Juho Uusi-Luomalahti

LIITOSKOMPONENTTIEN AUTOMATISOINTI

Kone- ja tuotantotekniikan koulutusohjelma
2013

LIITOSKOMPONENTTIEN AUTOMATISOINTI

Uusi-Luomalahti, Juho
Satakunnan ammattikorkeakoulu
Kone- ja tuotantotekniikan koulutusohjelma
Huhtikuu 2013
Ohjaaja: Reunamo, Petteri
Sivumäärä: 38
Liitteitä: 0

Asiasanat: laippaliitos, suunnitteluprosessi, automatisointi, putkiliitos

Työn tavoitteena oli automatisoida suunnitteluprosessia. Automatisoitava asia oli liitoskomponenttien lisääminen instrumenteille. Alfa Laval Rauma suunnittelee pako-
kaasun lämmön talteenottokattiloita ja erilaisia höyryjärjestelmiä. Laitteet sisältävät
useita putkistojen ja instrumenttien (venttiilit, lauhteenpoistimet yms.) välisiä laippa-
liitoksia, jotka sisältävät tietyt vakiokomponentit. Tällaisia vakiokomponentteja ovat
tiivisteet, aluslaatat, mutterit ja ruuvit. Nykyisellään kokoonpanopiirustusta tehdessä
suunnittelija joutuu valitsemaan manuaalisesti vakiokomponentit komponenttikirjas-
tosta, laskemaan niiden lukumäärät ja tekemään vastaavat osanumeroinnit kokoon-
panopiirustukseen. Suurin osa kokoonpanopiirustusten tekemiseen kuluva työajasta
menee tähän työvaiheeseen, joten järjestelmän ottaminen kesän aikana käyttöön
tulee nopeuttamaan suunnitteluprosessia merkittävästi. Näin ollen suunnittelijat voi-
vat käyttää työaikaansa olennaisempiin asioihin kuin yksinkertaisen ja toistuvan työ-
vaiheen toteuttamiseen. Järjestelmä vähentää myös inhimillisten virheiden määrää,
jotka voivat pahimmillaan viivästyttää asiakkaalle toimitettavan laitteiston käyttöö-
ottoa. Alfa Lavalin toimittamat järjestelmät sijaitsevat usein ei asutuilla seuduilla,
joten väärin komponenttien korvaaminen oikeilla voi viedä useita päiviä.

AUTOMATIZATION OF CONNECTION COMPONENTS

Uusi-Luomalahti, Juho

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in mechanical engineering

April 2013

Supervisor: Reunamo, Petteri

Number of pages: 38

Appendices: 0

Keywords: flange connection, designing process, automatization, pipe connection

Goal of the thesis was to automate one stage of a designing process. The automated stage was addition of connection components for instruments. Alfa Laval Rauma is enterprise which has focused to steam systems. Steam systems include connections between pipes and instruments (valves, strainers etc.). This kind of connections includes certain standard parts, such as gaskets, nuts, screws and washers. At the moment designer has to pick manually these parts from a component library, has to calculate correct amounts of the parts and has to make corresponding part numbers to an assembly drawing. The most of working time with assembly drawings is spent to this stage, thus commissioning of a system during next summer will enhance the designing process significantly. That will release designers to conduct more important assignments than carrying out simple and repetitive task. The system will also reduce amount of mistakes occurred during the designing process. In the worst-case scenario these mistakes may delay commissioning of a system delivered for a customer. Systems provided by Alfa Laval are usually located to remote areas, so replacing incorrect components might take many days.

KÄSITELUETTELO

VBA	Visual Basic for Applications –ohjelmointikieli
PaavoCAD	Alfa Laval Aalborg Oy:n Rauman yksikössä käytetty toiminnanohjausjärjestelmä
Makro	Sovellusohjelman ohjaamiseen käytetty ohjelmatyyppi
Komponenttiluettelo	Luettelo, joka sisältää laitteelle kuuluvat instrumentit
Instrumentti	Laitteeseen tuleva komponentti, yleensä venttiili, lämpömittari yms.
Cluet	PaavoCAD:n osio, jolla tehdään komponenttiluettelot
Kluet	PaavoCAD:n osio, jolla tehdään osaluetelot
Liitoskomponentti	Laippaliitoksessa käytettävä osa, kuten ruuvi, aluslaatta, tiiviste tai mutteri
Liitosyhdistelmä	Liitosyhdistelmä tarkoittaa instrumenttien ja laippojen muodostamaa liitoskokonaisuutta
Liitostyyppi	Määritelty liitosyhdistelmä, esimerkiksi laippa, istukka-venttiili, laippa – liitosyhdistelmä on yksi liitostyyppi
Detail-kuva	Konepiirustuksissa käytetty yksityiskohtakuva
DN-koko	Putkistoissa käytettävä kokomerkintä

SISÄLLYS

1	ALFA LAVAL AALBORG OY RAUMA	7
1.1	Yrityksen tuotteita.....	8
2	OHJELMOINNIN TEORIAA	8
2.1	Ohjelmoinnin käsitteistöä	9
2.2	Kommunikointihierarkia.....	10
2.3	Tietorakenteet ja algoritmit.....	10
2.4	Ohjelman perusrakenteet	11
2.4.1	Peräkkäisrakenne	11
2.4.2	Valintarakenne.....	11
2.4.3	Toistorakenne	12
2.5	Opinnäytetyössä käytetyt ohjelmointityökalut	13
2.6	Excel VBA ohjelmointi.....	13
3	NYKYTILANTEEN KUVAUS.....	18
3.1	Suunnitteluprosessin eteneminen nykytilanteessa	18
4	KEHITTÄMISTEHTÄVÄ JA TAVOITTEET	20
4.1	Kehittämistehtävän kuvaus ja tavoitteet	20
4.2	Analyysi saavutettavista hyödyistä ja mahdollisista haasteista	21
5	JÄRJESTELMÄN SUUNNITTELU	23
5.1	Liitosyhdistelmät.....	23
5.2	Liitosten ruuvipituudet.....	25
6	TULOKSET	25
6.1	Suunnitteluprosessi järjestelmän käyttöönoton jälkeen	26
6.2	Liitoskomponentti-ohjelman toimintalogiikka	27
6.3	Liituskoodi	28
6.4	Käyttöliittymä	29
6.5	Liitoskomponentti-tietokanta	29
6.6	Liituskoodi-tietokanta	30
6.7	Liitostyyppien detail-kuvat	30
7	ARVIOINTI	31
7.1	Arviointi tämänhetkisten tuloksien perusteella.....	31
7.2	Koekäyttö ja käyttöönotto.....	32
7.3	Laajennettavuus	32
7.4	Tavoitteiden täyttyminen	33

7.5 Ennen ja jälkeen – vertailu.....	34
8 LOPPUSANAT.....	35
LÄHTEET.....	37
LIITTEET	

1 ALFA LAVAL AALBORG OY RAUMA

Alfa Laval – konserni on kansainvälinen yritys, jonka pääosaamisaluetta ovat lämpö-, erotus-, ja virtaustekniikka. Alfa Laval toimii sadassa eri maassa ja yhtiön palveluksessa työskentelee 11500 työntekijää. (Alfa Lavalin [www-sivut](http://www.alfalaval.com) 2013)

Alfa Laval Aalborg Oy Rauma on osa Alfa Laval – konsernin Marine & Diesel divisioonaa. Yritys toimittaa höyryjärjestelmiä ja pakokaasun lämmön talteenottojärjestelmiä meriteollisuudelle sekä maavoimalaitoksiin. Yritys vastaa järjestelmien suunnittelusta, projektinhoidosta ja käyttöönotosta. Päälaitteiden valmistus tapahtuu omilla tehtailla Euroopassa, Aasiassa ja Etelä-Amerikassa. Kuormitustilanteen mukaan lisäkapasiteettia ostetaan sopimusvalmistajilta.

Alfa Laval Aalborg Oy:n historia juontaa juurensa vuoteen 1964, jolloin uudenkaupungin telakka aloitti valmistamaan höyrykattiloita laivoihin. Kun Finnyards osti yhtiön, se sai nimekseen Pipemasters Oy. Vuonna 1994 yhtiö aloitti kattilanvalmistuksen myös teollisuuden tarpeisiin. Aalborg Industries Oy perustettiin vuonna 1997, kun tanskalainen Aalborg Industries osti Pipemasters Oy:n. Vuonna 2011 Alfa Laval Oy osti koko Aalborg Industries yhtiön, ja näin ollen nykyinen Alfa Laval Aalborg Oy perustettiin.

1.1 Yrityksen tuotteita

UNEX™ H / HW (/AV-4)	AV-6N (finned) AV-8N (bare)	AV-6H
		
No limit dep. on engine	No limit dep. on engine	No limit dep. on engine
• Cylindrical • Horizontal or vertical • Smoke tubes - bare type • One- or two-pass • Natural circulation • Pinch point > 20...25 °C	• Vertical • Water tubes • Gilled tubes • Natural or forced circulation • High-efficient, <i>engine-running</i> on-line cleaning	• Horizontal • Water tubes • Gilled tubes • Forced circulation
• < 20 bar(g) • HFO heating steam • (Process steam) • District heating hot-water ($T_{out} < 180\text{ °C}$) (UNEX™ HW)	• Typically 5 - 30 bar(g), max. 65 bar(g) • Superheat steam for turbine • Process steam	• Typically 5 - 30 bar(g) • District heating hot-water • (Steam)

Kuvio 1. Alfa Laval Aalborg Oy:n tuotteita (Alfa Laval Aalborgin markkinointimateriaali 2013)

Kuviossa 1 on esitetty Alfa Laval Aalborg Oy:n voimalaitossovellusten tyypillisimmät laitteet. Laitteet ovat pakokaasun lämmön talteenottokattiloita. Kattila kytketään kiinni diesel- tai kaasumoottoriin, jonka tuottaman pakokaasun lämpöenergialla kattila tuottaa höyryä. Kattilalla tuotetulla höyryllä voidaan käyttää turbiinia ja generaattoria, jolloin saadaan sähköenergiaa. Tuotettu höyry voidaan käyttää myös prosessihöyrynä erilaisiin käyttökohteisiin.

2 OHJELMOINNIN TEORIAA

Tässä luvussa käsitellään ohjelmoinnin perusteita, koska opinnäytetyö sisälsi huomattavan määrän ohjelmointityötä.

2.1 Ohjelmoinnin käsitteistöä

Tietokoneohjelma on luettelo käskyistä, joiden tarkoituksena on toteuttaa jokin tietojenkäsittelytehtävä. Tietokoneohjelmat kirjoitetaan ohjelmointikielellä. Ohjelmointikielellä kirjoitettu tietokoneohjelma käännetään konekielelle, jotta se voitaisiin suorittaa tietokoneessa. Tietokoneen suoritin kykenee käsittelemään vain konekielisiä ohjelmia. Konekieli koostuu yksinkertaisista käskyistä, joilla käsitellään tietokoneen muistipaikkojen sisältöä. Tietokoneessa kaikki käsiteltävä tieto esitetään bittien avulla, joten konekieliset ohjelmat ovat myös vain ykkösiä ja nollia. (Silander 2000, 15).

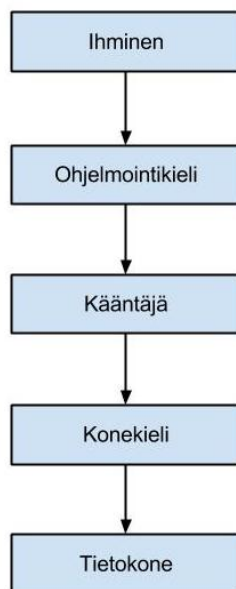
Konekieli on ihmisen näkökulmasta hyvin vaikeaselkoista ja hankalasti ohjelmoitavaa, joten ohjelmointityötä varten on kehitetty erilaisia ohjelmointikieliä. Ohjelmointikielet ovat konekieltä korkeammalla tasolla toimivia ohjelmoinnin apuvälineitä. Ohjelmointikielellä kirjoitetut ohjelmat käännetään kääntäjällä tietokoneen ymmärtämään konekieliseen muotoon. (Silander 2000, 15).

Ohjelmaa, joka suorittaa käynnöksen konekielelle, kutsutaan kääntäjäksi. Kääntäjä hoitaa ohjelmointikielellä kirjoitetun ohjelman käynnöksen konekieliseksi ihmisen puolesta. Kun ohjelmointikielellä kirjoitettua ohjelmaa käännetään, kääntäjä tarkastaa samalla, ettei ohjelmassa esiinny ohjelmointikielen sääntöjen vastaisia piirteitä. Vaikka ohjelma saataisiinkin käännettyä virheettömästi konekielelle, siinä saattaa silti olla loogisia virheitä, joiden vuoksi ohjelma ei toimi halutulla tavalla. (Silander 2000, 15).

Ohjelmointikielellä kirjoitettu ohjelma voidaan myös tulkata. Ohjelmointikielen tulkilla ohjelmakoodi käännetään konekieliseksi ja suoritetaan lause kerrallaan. Tulkin ero kääntäjään on se, että kääntäjällä ohjelmakoodi muutetaan kerralla konekieliseksi ja sen jälkeen suoritetaan, kun taas tulkilla ohjelmakoodi suoritetaan lause kerrallaan. Ohjelmakoodin suorittaminen tulkkaamalla on hitaampaa kuin käännetyn ohjelman ajaminen, koska tulkin pitää ensin kääntää ohjelmakoodi ja sen jälkeen suorittaa se, kun taas käännetty konekielinen ohjelma tarvitsee vain suorittaa. (Wikipedia 2013).

2.2 Kommunikointihierarkia

Tietokoneohjelman tuottaminen ja suorittaminen muodostavat kuviossa 2 esitetyn hierarkian.



Kuvio 2. Ihmisen ja tietokoneohjelmoinnin kommunikointihierarkia (Silander 2000, 16).

2.3 Tietorakenteet ja algoritmit

Tietorakenteet ovat ohjelmointikielissä käytettyjä informaatioyksiköitä, jotka sisältävät yksittäisen tiedon tai tietojoukon. Tietorakenne voi sisältää esimerkiksi päivämäärän, yhden kokonaisluvun tai vaikka koko pääkaupunkiseudun puhelinluettelon. (Silander 2000, 22).

Tietokoneohjelmat käsittelevät aina erilaisia tietorakenteita. Metodeita, joilla tietorakenteita käsitellään, kutsutaan algoritmeiksi. Algoritmi on menettelytapojen ja sääntöjen kokoelma. Algoritmeilla käsitellään niihin syötettyä tietoa niin, että syntyy uutta tietoa.

”On myös huomattava, että algoritmi ei ole yksinomaan ohjelmointiin liittyvä käsite. Esimerkiksi tavallista keittokirjaa voidaan pitää algoritmikokoelmana. Reseptit voidaan tällöin nähdä algoritmeina ja raaka-aineet tietorakenteina.” (Silander 2000, 23).

2.4 Ohjelman perusrakenteet

Useimmilla korkeamman tason ohjelmointikielillä on olemassa tiettyjä yhteisiä perusrakenteita, jotka käsitellään tässä luvussa. Näitä rakenteita kutsutaan ohjausrakenteiksi tai kontrollirakenteiksi. Ne ohjaavat nimensä mukaisesti ohjelman kulkua. Rakenteiden käyttämisen etu on se, että niitä voidaan yhdistellä mielivaltaisesti. Niitä voidaan kirjoittaa peräkkäin ja kunkin rakenteen sisälle voidaan sijoittaa muita rakenteita. Ohjelmointikielen ohjausrakenteiden yhdistely antaa mahdollisuuden luoda rajattomasti erilaisia ohjelmia tietotekniikan mahdollistamissa puitteissa. (Silander 2000, 28).

2.4.1 Peräkkäisrakenne

”Peräkkäisrakenne on yksinkertaisin ohjelmassa käytettävistä rakenteista. Siinä toteutetaan sarja toimenpiteitä peräkkäin.” (Silander 2000, 29).

Peräkkäisrakenne tarkoittaa rakennetta, joka suoritetaan ilman toistoa tai ehtolauseita. Esimerkkinä peräkkäisrakenteesta voisi olla voileivän syönti:

1. Avaa suu
2. Syötä pala voileipää suuhun
3. Pureskele
4. Niele

2.4.2 Valintarakenne

Valintarakenteessa on jokin ehtolause, jonka on oltava tosi, jotta rakenne suoritetaan. Jos edelliseen voileipäesimerkkiin liitetään ehtorakenne, se voisi näyttää seuraavalta:

JOS (jääkaapissa on voita)

```
{
  1. Avaa suu
  2. Syötä pala voileipää suuhun
  3. Pureskele
  4. Niele
}
```

Eli voileipä syödään vain silloin, kun jääkaapissa on voita, jos voita ei ole niin voileipää ei syödä.

Valintarakenteiden ehtolauseisiin voidaan tarpeen vaatiessa lisätä loogisia- ja vertailuoperaattoreita. Loogisia operaattoreita ovat esimerkiksi TAI, NEGAATIO, JA. Vertailuoperaattoreita ovat esimerkiksi PIENEMPI KUIN, SUUREMPI KUIN tai YHTÄSUURI KUIN. Edellinen voileipäesimerkki vertailuoperaattorin kanssa:

JOS (jääkaapissa on voita) JA (jääkaapissa on kinkkua)

```
{
  1. Avaa suu
  ...
  6. Niele
}
```

Eli rakenne suoritetaan vain kun molemmat ehdot ovat tosia. Toisin sanoen, molemmat sekä voi ja kinkku vaaditaan, jotta voileipä voidaan syödä.

2.4.3 Toistorakenne

Toistorakenteella voidaan toistaa jotain määrättyä tehtävää useita kertoja. Toistorakenteessa määritellään jatkamisehto jonka perusteella päätellään kuinka kauan toistorakennetta toistetaan. (Silander 2000, 31). Esimerkkinä toistorakenteesta:

NIIN KAUAN KUIN (voileipiä on jäljellä)

```
{
```

```

1. Voitele leipä
...
6. Niele
}

```

Eli toistorakennetta suoritetaan niin kauan kuin ehto on tosi. Esimerkissä voileivän syöntiä toistetaan niin kauan kuin voileipiä on jäljellä.

2.5 Opinnäytetyössä käytetyt ohjelmointityökalut

Järjestelmä ohjelmoitiin käyttäen VBA-ohjelmointikieltä (Visual Basic for Applications). VBA-ohjelmointikieli on Microsoftin Visual Basiciin pohjautuva tapahtumapohjainen makrokieli. Sillä voidaan ohjata erilaisia toimintoja isäntäohjelman sisällä. VBA eroaa tavanomaisesta Visual Basicista siinä, että sillä ei pysty rakentamaan itsenäisesti toimivia ohjelmia vaan se toimii aina isäntäohjelman sisällä. VBA on käytössä mm. Microsoft Office – tuotteissa. (Wikipedia 2013)

Järjestelmän vaatimat tietokannat oli luontevinta toteuttaa Excel – ympäristössä, koska PaavoCAD ja täten myös kehitetty järjestelmä toimii Excelissä. Koska tietokannat sisältävät vain muutamia satoja artikkeleita, niiden toteuttaminen Excelissä ei aiheuttanut ongelmia. Jos tietokannoista olisi tullut suurempia, olisi tietojen hallinnan kannalta ollut viisaampaa toteuttaa tietokannat jollakin tietokantaohjelmistolla.

2.6 Excel VBA ohjelmointi

Microsoft Excelin VBA-ohjelmoinnissa hyödynnetään makroja. Makrot ovat koodinpätkiä, joilla suoritetaan jokin haluttu tehtävä. Makroilla voidaan luoda Excel-taulukoihin erilaisia toimintoja tiettyjä tarkoituksia varten. VBA-ohjelmointikielen hyödyntäminen osana taulukoita antaa mahdollisuuden rakentaa monimutkaisiakin ohjelmia toimimaan Excel-ympäristössä.

Exceliin on sisällytetty valmiiksi kokoelma funktioita, joita voidaan käyttää osana taulukoita. Nämä Excelin sisäänrakennetut funktiot kuitenkin ovat hyvin rajoittuneita

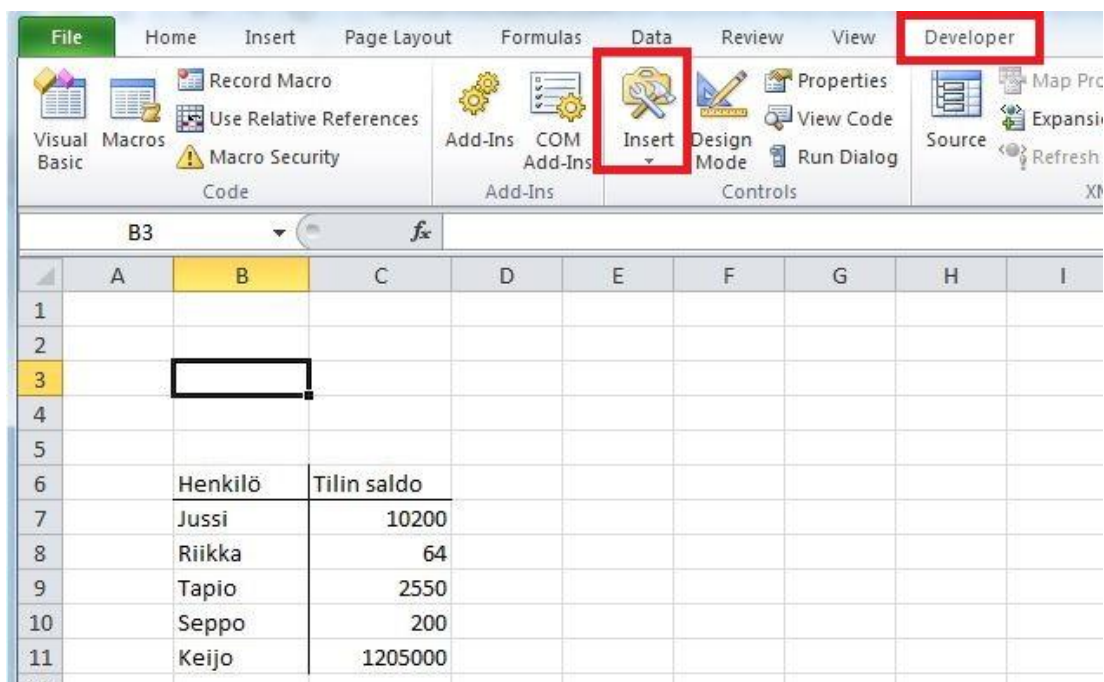
ja niillä ei ole mahdollista toteuttaa taulukoihin monimutkaisia ja älykkäitä toimintoja.

Alla esitetään esimerkki makron luomisesta Excel-tilukoon. Esimerkissä luodaan taulukoon painike, jota painamalla taulukossa olevien henkilöiden tilien saldot väritetään määrätillä värillä. Jos henkilön tilin saldo on alle 10000, niin solu väritetään punaisella. Jos tilin saldo on 10000–100000, niin solu väritetään keltaisella. Jos tilin saldo on yli 100000, niin solu väritetään vihreällä.

Henkilö	Tilin saldo
Jussi	10200
Riikka	64
Tapio	2550
Seppo	200
Keijo	1205000

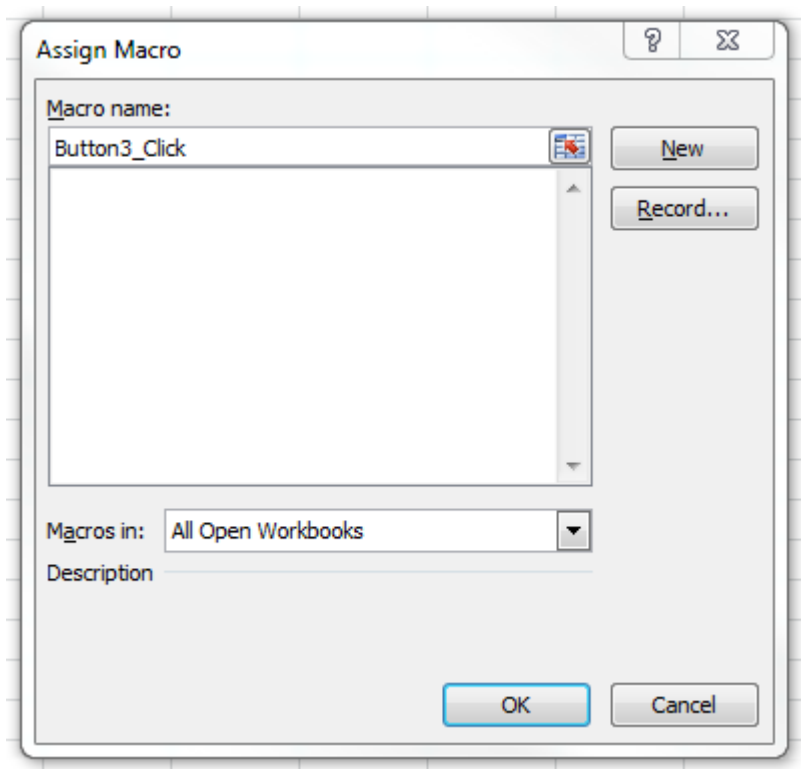
Kuvio 3. Henkilöiden tilien saldot.

Ensin taulukoon on luotava painike, josta makro käynnistetään. Painike luodaan Developer – välilehden Insert – painikkeesta.



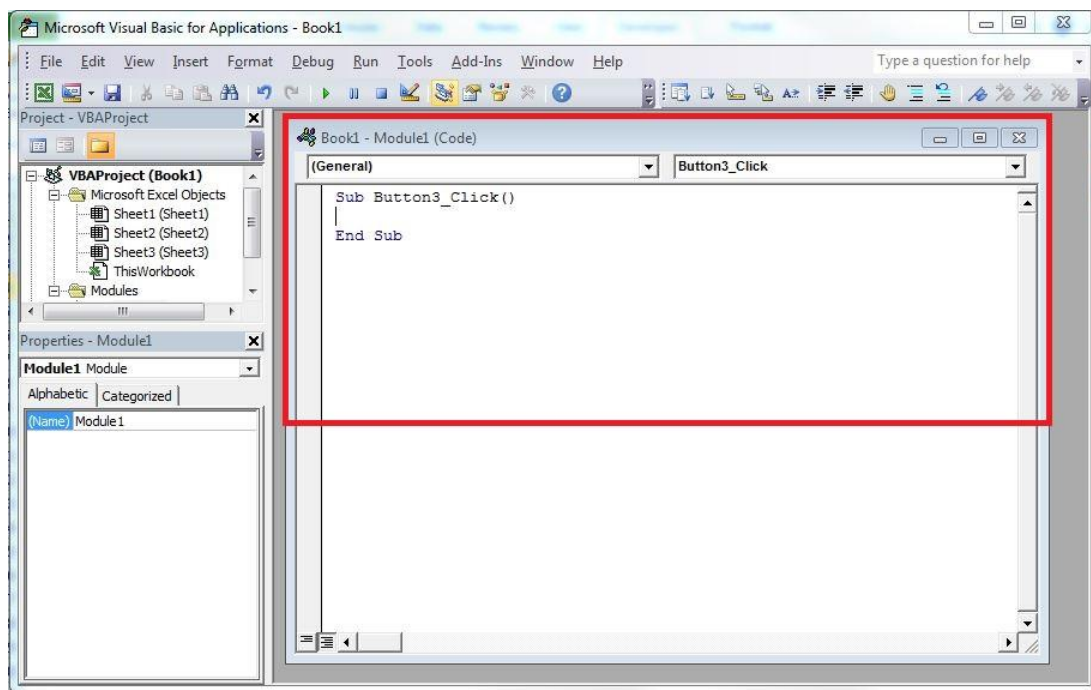
Kuvio 4. Developer – välilehti

Insert – painikkeesta aukeaa valikko, josta painike valitaan. Painike sijoitetaan taulukkoon maalaamalla se haluttuun kohtaan. Tämän jälkeen ohjelma aukaisee kuviossa 5 esitetyn ikkunan.



Kuvio 5. Ikkuna, jossa makro luodaan

Kuvion 5 ikkunaan syötetään makrolle haluttu nimi, jonka jälkeen painetaan New – painiketta. Tässä tapauksessa makron nimeksi on annettu Button3_Click. New – painikkeen klikkaaminen avaa kuviossa 6 esitetyn kehitysympäristön.



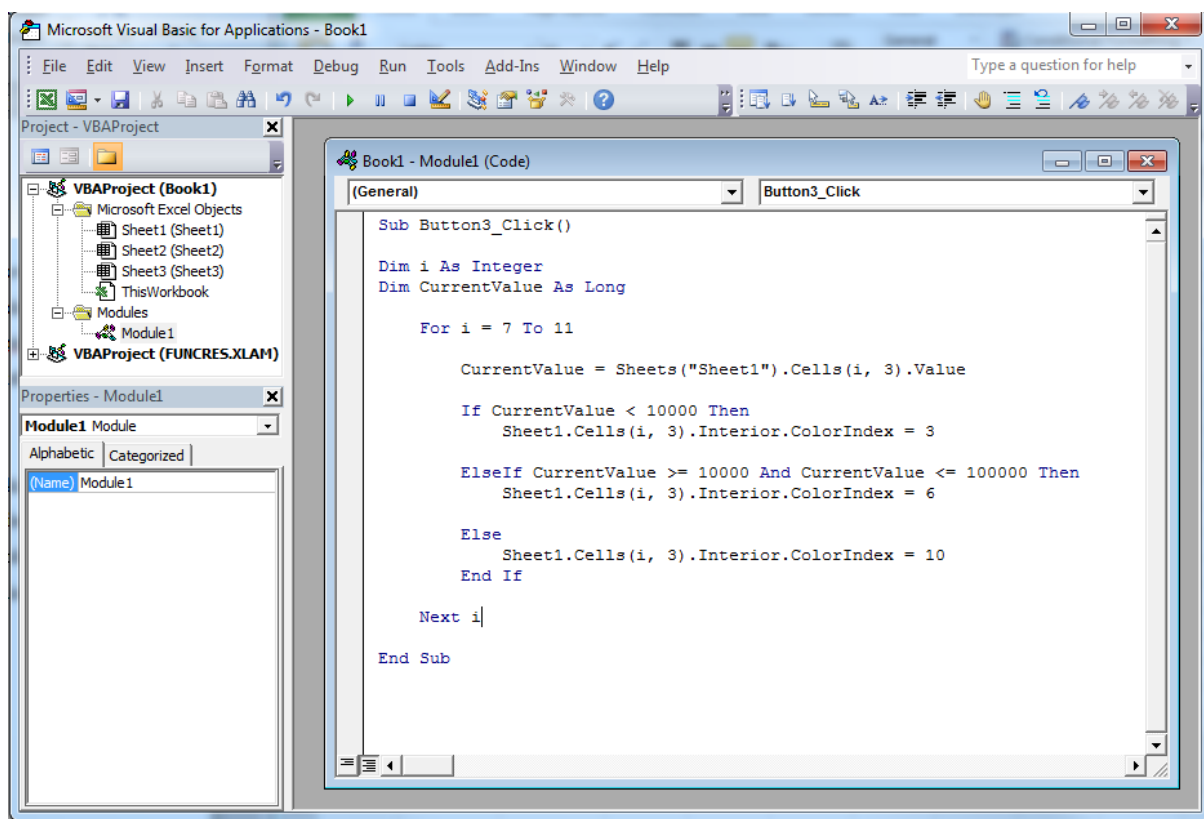
Kuvio 6. Kehitysympäristö

Kuviossa 6 esitetyn punaisen laatikon sisällä olevaan ikkunaan kirjoitetaan ohjelmakoodi. Excel on luonut automaattisesti aliohjelman nimeltä ”Sub Button3_Click()”. Excel suorittaa kyseisen aliohjelman, kun tapahtuma Button3_Click toteutuu. Eli kun luotua painiketta klikataan.

	A	B	C
1			
2		Button 3	
3			
4			
5			
6		Henkilö	Tilin saldo
7		Jussi	10200
8		Riikka	64
9		Tapio	2550
10		Seppo	200
11		Keijo	1205000
12			

Kuvio 7. Tilien saldot ja painike

Nyt taulukko näyttää kuviossa 7 esitetyltä. Painikkeen klikkaamisesta ei toistaiseksi vielä tapahdu mitään, koska sille ei ole luotu koodia.



Kuvio 8. Ohjelmakoodi

Kuviossa 8 on kirjoitettu ohjelmakoodi, joka suoritetaan kun luotua painiketta klikataan. Tässä tapauksessa kyseessä oli solujen värjääminen tilin saldon mukaan.

	A	B	C
1			
2			
3			
4			
5			
6		Henkilö	Tilin saldo
7		Jussi	10200
8		Riikka	64
9		Tapio	2550
10		Seppo	200
11		Keijo	1205000
12			

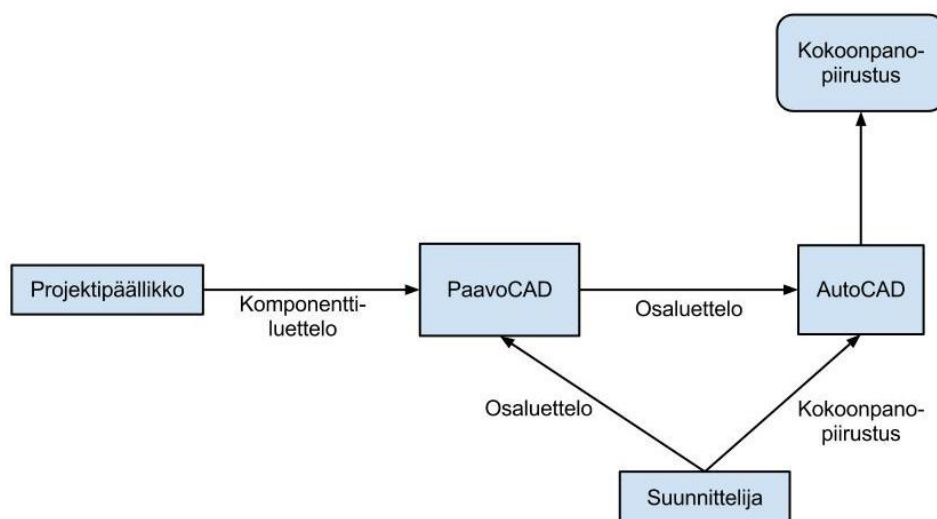
Kuvio 9. Taulukko painikkeen klikkaamisen jälkeen

Kun painiketta klikataan, niin ohjelma värjää saldot ohjelmakoodissa määritellyillä väreillä. Kuviossa 9 esitetään taulukko painikkeen klikkaamisen jälkeen.

3 NYKYTILANTEEN KUVAUS

3.1 Suunnitteluprosessin eteneminen nykytilanteessa

Tällä hetkellä yrityksessä laitteen suunnitteluprosessi etenee kuvion 10 mukaisella tavalla.

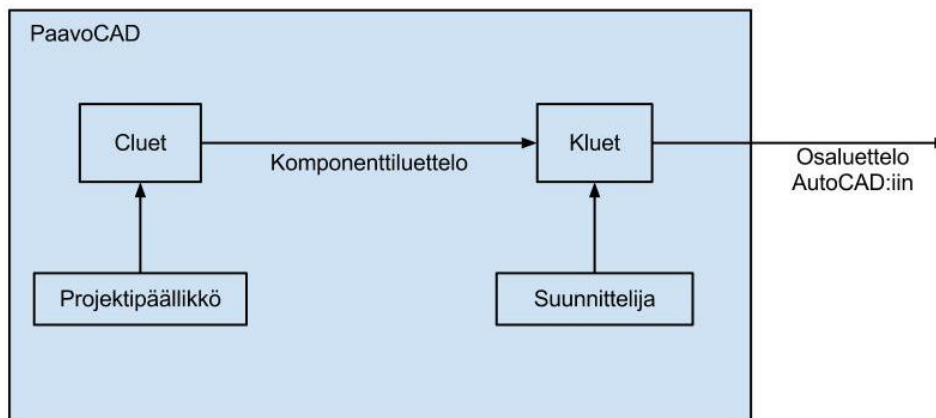


Kuvio 10. Suunnitteluprosessin eteneminen yrityksessä

PaavoCAD on yrityksen Excel-pohjainen toiminnanohjausjärjestelmä, jonka avulla laaditaan suunniteltaville laitteille mm. komponenttiluettelot ja osaluettelot piirustuksiin.

Laitteen suunnittelu alkaa, kun projektipäällikkö tekee laitteelle komponenttiluettelon PaavoCAD:lla. Suunnittelija tekee osaluettelon PaavoCAD:lla projektipäällikön tekemän komponenttiluettelon pohjalta. PaavoCAD:sta tehty osaluettelo viedään AutoCAD:iin, jolla suunnittelija piirtää laitteesta tarvittavat piirustukset.

Tarkastellaan kuviossa 11 tarkemmin suunnitteluprosessia PaavoCAD:n sisällä:



Kuvio 11. Suunnitteluprosessi PaavoCAD:n sisällä

- Cluet on PaavoCAD:n osio, jolla tehdään ja muokataan laitteiden komponenttiluetteloita. Projektipäällikkö tekee laitteelle komponenttiluettelon käyttäen Cluet – osiota.
- Kluet on PaavoCAD:n osio, jolla tehdään ja muokataan laitteiden piirustuksien osaluetteloita. Suunnittelija käyttää Kluet – osiota tehdessään laitteen piirustukselle osaluettelon.

PaavoCAD:n sisällä komponenttiluettelon sisältämä tieto kulkee Cluet – osiosta Kluet – osioon, jolla suunnittelija luo osaluettelon piirustukselle.

Nykytilanteessa prosessin pullonkauloja ovat suunnittelija ja Kluet - osio. Komponenttiluettelo sisältää vain informaation laitteeseen tulevista instrumenteista (venttiilit yms). Tästä johtuen suunnittelijan tarvitsee hakea PaavoCAD:n komponenttikirjastosta instrumenttien välisiin liitoksiin vaaditut liitoskomponentit kuten ruuvit, aluslaatat, mutterit ja tiivisteet. Suunnittelijan tarvitsee myös laskea liitoskomponenttien oikeat määrät, antaa niille osanumeroinnit ja merkata jokaiselle instrumentille osanumeropallot piirustukseen. Tämä työvaihe on aikaa vievä ja altis inhimillisille virheille.

Liitoskomponentit ovat vakio-osia, jotka ovat tietyille liitostyypeille aina samanlaisia. Tästä johtuen tietotekniikka tarjoaa mahdollisuuden automatisoida tämä työvaihe suunnitteluprosessista. Kun eri liitostyypeille ja liitoskomponenteille on kerran tehty

kattava tietokanta, niin liitosinformaatio on mahdollista tuoda automaattisesti samalla, kun suunnittelija tuo komponenttiluettelon Cluet – osiosta Kluet – osioon.

4 KEHITTÄMISTEHTÄVÄ JA TAVOITTEET

4.1 Kehittämistehtävän kuvaus ja tavoitteet

Kehittämistehtävänä oli suunnitella ja toteuttaa järjestelmä, joka toimii osana yrityksen PaavoCAD – toiminnanohjausjärjestelmää. Järjestelmän tarkoitus on automatisoida instrumenttien välisiin liitoksiin käytettävien liitoskomponenttien tuominen piirustuksiin.

Tavoitteena oli päästä eroon seuraavista suunnittelijan tekemistä työvaiheista:

- Instrumenttien välisiin liitoksiin tulevien liitoskomponenttien manuaalisesta noutamisesta komponenttikirjastosta
- Liitoskomponenttien oikean määrän laskemisesta manuaalisesti
- Liitoskomponenttien osanumeroinnista PaavoCAD:ssa
- Liitoskomponenttien osanumeropallojen lisäämisestä piirustukseen

Luetellut työvaiheet ovat toistuvia ja rutiininomaisia, joten ne soveltuvat hyvin suoritettavaksi ohjelmallisesti.



Kuvio 12. Laippa, istukkaventtiili, laippa – liitokseen sisältyvät liitoskomponentit



Kuvio 13. Laippa, istukkaventtiili, laippa – liitos koottuna



Kuvio 14. Laippa, istukkaventtiili, laippa – liitos koottuna

Kuvioissa 12–14 on esitetty esimerkki yhdestä liitosyhdistelmästä, kyseessä on laippa, istukkaventtiili, laippa – liitostyyppi. Kuten kuvioista 12 näkyy, jo näinkin yksinkertainen liitos sisältää 34 liitoskomponenttia. Kuvioissa esitetyn kaltaisia liitoksia saattaa olla yhdessä kokoonpanopiirustuksessa kymmeniä, joten liitoskomponenttien lisäämiseen suunnittelijalta kuluva työmäärä on huomattava.

4.2 Analyysi saavutettavista hyödyistä ja mahdollisista haasteista

Järjestelmästä saavutettavia hyötyjä:

- Inhimillisten virheiden määrä vähenee, kun toistuvia työvaiheita automatisoidaan koneiden tehtäväksi. Tämä parantaa suunnittelun laatua.
- Suunnittelutyö nopeutuu ja säästetty aika voidaan käyttää esimerkiksi tuotekehitystyöhön tai useampien laitteiden suunnitteluun samassa ajassa.
- Suunnittelijat kokevat työnsä miellyttävämmäksi, koska toistuvaa työtä on vähemmän.
- Kokoonpanokuvat tulevat selkeämmiksi, koska osanumeropallojen määrä vähenee ja liitoksista esitetään selkeät detail-kuvat.

Edellä mainitut asiat säästävät ennen kaikkea yrityksen rahaa, kun samat asiat voidaan toteuttaa tehokkaammin kuin ennen. Virheiden määrän väheneminen parantaa tuotteiden laatua ja säästää myös rahaa. Esimerkiksi liian vähäinen ruuvien lukumäärä toimitetun laitteen lähetyksessä aiheuttaa viivästyksiä ja ongelmia, jos laitteen asennuspaikka sijaitsee kaukana lähimmästä kiinnityselimiä toimittavasta tahosta.

Mahdollisia haasteita järjestelmälle:

- Järjestelmästä tulee liian hidas- ja vaikeakäyttöinen, eikä henkilöstö halua käyttää sitä.
- Ihmisen perusluonteeseen kuuluva uusien asioiden vastustus, vaikka ne olisivatkin hyödyllisiä.
- Ohjelmointivirheet
- Järjestelmän kehittäjältä saattaa jäädä huomioimatta olennaisia asioita, jotka vaikuttavat järjestelmän toimintaan osana yritystä

Järjestelmälle aiheutuvat haasteet ovat ihmisistä ja järjestelmän kehittäjän taidoista riippuvaisia. Ihmisistä aiheutuvia haasteita pystytään torjumaan riittävällä koulutuksella, järjestelmästä saatavien hyötyjen huolellisella perustelulla ja sillä, että järjestelmä päätetään ottaa määrätietoisesti osaksi yrityksen toimintatapoja. Olennaisin ja merkitsevin haaste järjestelmälle on se, että sen käyttöliittymästä tehdään riittävän selkeä, yksinkertainen ja nopeakäyttöinen. Jos käyttöliittymä koetaan käyttäjien puolelta huonoksi, niin mikään ei pysty pelastamaan järjestelmää, vaikka ajatus sen taustalla olisi kuinka hyvä tahansa.

5 JÄRJESTELMÄN SUUNNITTELU

Järjestelmän suunnittelu lähti liikkeelle aloituspalaverista, jossa määriteltiin tavoitteet järjestelmälle, rajaehdot ja sen kuinka järjestelmän tulee toimia, jotta se istuu saumattomasti yrityksen toimintatapoihin. Jatkossa liitoskomponentit osanumeroidaan automaattisesti joten liitoskomponenteille sovittiin kiinteät osanumerot, jotka eivät riipu piirustuksen muista komponenteista. Tavallisesti konepiirustuksien osanumerot noudattavat juoksevaa numerointia, mutta tässä tapauksessa se olisi vaikeuttanut järjestelmän toteutusta huomattavasti. Kiinteät osanumerot mahdollistavat erillisten detail-kuvien käyttämisen liitoksista. Detail-kuvissa esitetään kuva liitoksesta ja liitokseen tulevien liitoskomponenttien osanumerot. Detail-kuvien osanumerointia on vaikea muuttaa ohjelmallisesti AutoCAD:n päässä, joten tästä syystä päädyttiin kiinteään osanumerointiin. Päätettiin myös, että alkuvaiheessa järjestelmä kattaa vain laippaliitokset, jottei päättötyö ei laajene liian suuritöiseksi.

5.1 Liitosyhdistelmät

Ensimmäisessä järjestelmän suunnitteluvaiheessa kartoitettiin erilaisten liitosyhdistelmien määrä ja niiden esiintyminen yrityksen suunnittelemissa laitteissa. Yksinkertaisimmillaan liitos sisältää vain kaksi laippaa, kun taas monimutkaisimmillaan liitoksessa voi esiintyä esimerkiksi laippa, istukkaventtiili, takaiskuventtiili ja laippa. Harvinaisimmat liitostyypit jätettiin järjestelmän ulkopuolelle, koska se olisi kasvattanut erilaisten liitosyhdistelmien määrää huomattavasti. Vähän käytettyjen liitostyyppien jättämien järjestelmän ulkopuolelle ei aiheuta ongelmia laitteiden suunnittelussa, koska ne voidaan edelleen suunnitella manuaalisesti vanhaa käytäntöä noudattaen.

Flanges + flange									
(mikä tahansa laippatyyppi)									
	Plate flange	Flange	Blind flange	Flange	Set-in flange	Flange	Neck flange	Neck flange	
1	1	1							Plate flange - Plate flange
2			1	1					Plate flange - Blind flange
3					1	1			Plate flange - Set-in flange
4							1	1	Neck flange - Neck flange

Kuvio 15. Laippa, laippa – liitosyhdistelmät

Flange + component									
	Flange	Ball valve	Globe valve		Butterfly valve	Non-return valve	Steam trap	Strainer	
5	1	1							Flange - Ball valve
6	1		1						Flange - Globe valve
7	1				1				Flange - Butterfly valve
8	1					1			Flange - Non-return valve
9	1						1		Flange - Steamtrap
10	1							1	Flange - Strainer

Kuvio 16. Laippa, komponentti – liitosyhdistelmät

Component + component									
	Ball valve	Globe valve	Vexve valve		Non-return valve	Steam trap	Strainer		
11	1				1				Ball valve - Non-return valve
12	1						1		Ball valve - Strainer
13		1			1				Globe valve - Non-return valve
14		1				1			Globe valve - Steamtrap
15		1					1		Globe valve - Strainer
16					1	1			Non-return valve - Steamtrap
17						1	1		Steamtrap - Strainer

Kuvio 17. Komponentti, komponentti – liitosyhdistelmät

Kuvioissa 15–17 on esitetty mahdollisia liitosyhdistelmiä laippaliitoksille. Taulukot ovat kuvakaappauksia aputyökalusta, jonka avulla selvitin tarpeelliset liitosyhdistelmät. Kuvioissa 15–17 esiintyvät liitosyhdistelmät kattavat useimmat yrityksessä käytetyt laippaliitokset. Jokainen kuvioissa esitetty liitosyhdistelmä jakautuu lisäksi eri

DN-kokoihin ja paineluokkiin. Erilaisten liitosyhdistelmien kokonaismääräksi tulee näin ollen noin 40 kappaletta. Näiden taulukoiden pohjalta järjestelmään luotiin tietokannat laippaliitoksista ja niihin kuuluvista liitoskomponenteista. Tietokantoihin on mahdollista lisätä myöhemmin uusia liitostyyppejä, jos tarvetta esiintyy.

5.2 Liitosten ruuvipituudet

Kun erilaisten liitosyhdistelmien määrä oli selvitetty, tuli jokaiselle liitostyypille määrittellä oikeat ruuvipituudet. Tätä varten loin Exceliin aputyökalun, jolla pystyin määrittelemään jokaiselle liitostyypille oikeat ruuvipituudet. Kuviossa 18 alin rivi kertoo liitokseen tulevan ruuvin minimipituuden.

Plate flange + plate flange			BACK		
PN40					
Ruuvikoko	m12	m12	m12	m12	m16
DN	10	15	20	25	32
Washer	2,5	2,5	2,5	2,5	3
Plate flange PN40	14	14	16	16	18
Flange gasket	2	2	2	2	2
Plate flange PN40	14	14	16	16	18
Washer	2,5	2,5	2,5	2,5	3
Sum	35	35	39	39	44
mutterin paksuus	10	10	10	10	13
sum	45	45	49	49	57

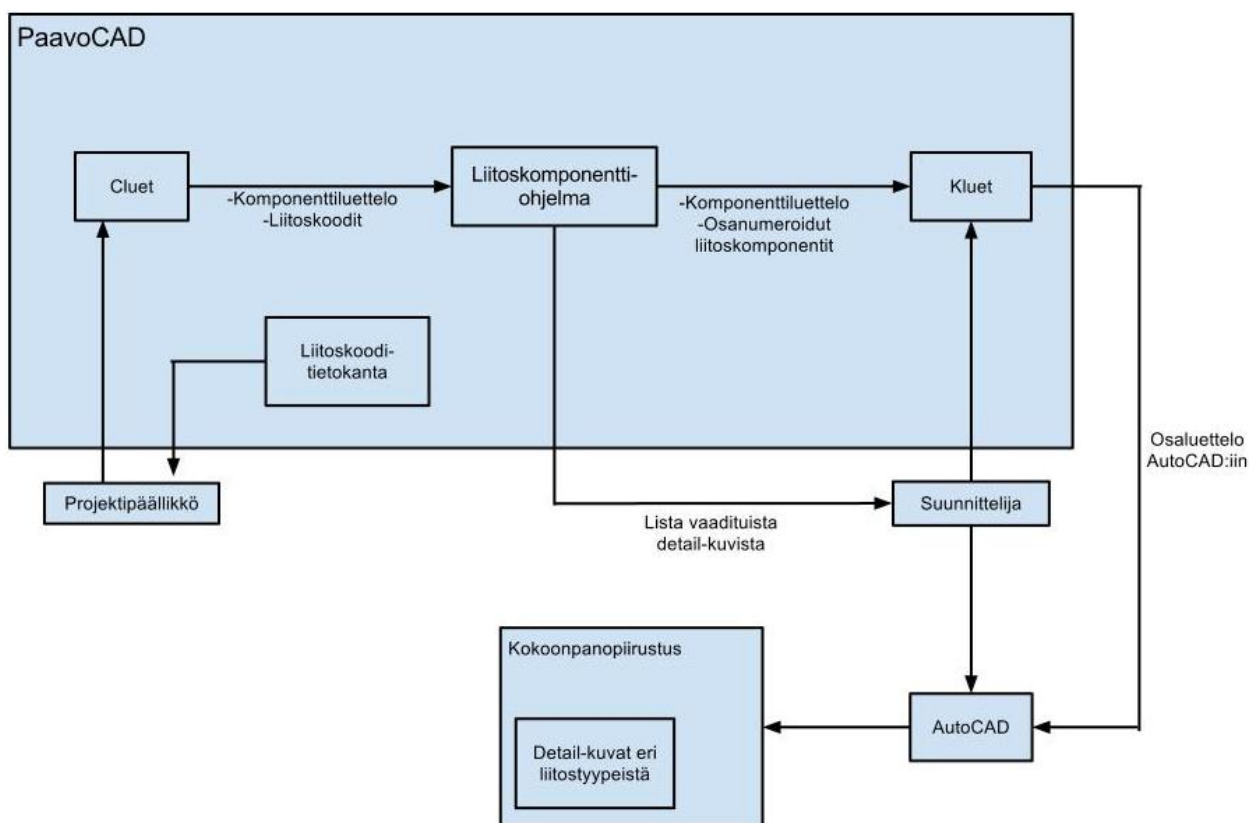
Kuvio 18. Kuvakaappaus aputyökalusta erilaisten liitostyyppien ruuvipituuksien määrittämiseksi

6 TULOKSET

Tuloksena opinnäytetyöstä on järjestelmä, joka toimii osana yrityksen PaavoCAD -toiminnanohjausjärjestelmää. Järjestelmä ei ole vielä täysin valmis, koska sitä ei ole käyttöönotettu eikä testattu riittävästi. Testejä järjestelmälle tullaan suorittamaan kevään ja alkukesän aikana, jotta järjestelmä voidaan käyttöönottaa kesän aikana. Itse ohjelma toimii suunnitellusti ja täyttää sille asetetut tavoitteet.

Järjestelmän toimintalogiikka on kuvattu yleisellä tasolla tässä luvussa. Selkeyden vuoksi ohjelman ja suunnitteluprosessin toimintaa ei ole kuvattu yksityiskohtaisesti.

6.1 Suunnitteluprosessi järjestelmän käyttöönoton jälkeen



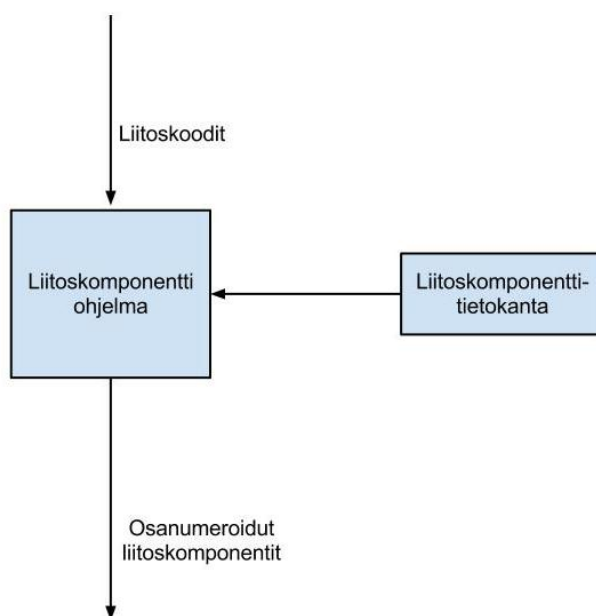
Kuvio 19. Kaavio suunnitteluprosessista järjestelmän käyttöönoton jälkeen

Liitoskomponenttiohjelma on yksi tämän päättöyön tuloksista. Se on PaavoCAD:n sisällä toimiva lisäosa, joka vastaa liitoskomponenttien lisäämisestä osaluetteloon automaattisesti. Kuviossa 10 on esitetty kaavio suunnitteluprosessista nykytilanteessa. Kuviossa 19 on esitetty kaavio suunnitteluprosessista, kun liitoskomponenttien lisääminen on hoidettu automaattisesti. Liitoskomponentti-ohjelma noutaa liitoskoodin perusteella liitokseen kuuluvat komponentit tietokannasta, laskee komponenttien lukumäärät ja antaa komponenteille oikeat osanumerot.

Suunnitteluprosessi alkaa projektipaallikon laatiessa komponenttiluettelon laitteen instrumenteista. Komponenttiluetteloon liitetään myös jokaiselle instrumentille lii-

toskoodi. Liitoskoodi kertoo Liitoskomponentti-ohjelmalle mitä liitoskomponentteja kyseisen instrumentin liitokseen sisältyy. Projektipäällikkö lisää jokaiselle instrumentille tarvittavat liitoskoodit Liitoskoodi-tietokannasta. Liitoskomponentti-ohjelma lukee liitoskoodit ja tuottaa niiden perusteella osanumeroidut liitoskomponentit jokaiselle instrumentille. Komponenttiluettelo ja osanumeroidut liitoskomponentit lähtevät Liitoskomponentti-ohjelmasta Kluet - osioon, jolla suunnittelija tekee osaluettelon piirustukselle. Liitoskomponentti-ohjelmalla muodostetaan myös tekstitiedosto, joka sisältää listan piirustukseen vaadittavista detail-kuvista. Tämän listan avulla suunnittelija pystyy noutamaan AutoCAD:n kirjastosta laitteen liitoksille tarvittavat detail-kuvat. Detail-kuvissa esitetään kuva liitostyypistä ja vastaavat osanumeroinnit liitoskomponenteille.

6.2 Liitoskomponentti-ohjelman toimintalogiikka



Kuvio 20. Kaavio Liitoskomponenttiohjelman toimintalogiikasta

Kuviossa 20 on esitetty Liitoskomponenttiohjelman toimintalogiikka. Liitoskomponentti ohjelma noutaa siihen saapuvien liitoskoodien perusteella tarvittavat liitoskomponentit Liitoskomponentti-tietokannasta, laskee liitoskomponenttien lukumäärät ja antaa liitoskomponenteille oikeat osanumerot.

6.3 Liituskoodi

Liituskoodi sisältää tiedon tietylle liitostyypille kuuluvista liitoskomponenteista. Jokaiselle laitteen instrumentille voidaan määritellä tarvittaessa liituskoodi.

Liituskoodi noudattaa seuraavaa muotoa:

Liitostyyppi / komp1 # lukumääräkomp1 / komp2 # lukumääräkomp2 /.../ komp7 #
lukumääräkomp7 / Detail-kuva /

- Liituskoodissa on määritelty ensin liitostyyppi, joka voi olla esimerkiksi laippa, istukkaventtiili, laippa.
- Liitostyyppin jälkeen liituskoodissa määritellään liitoskomponentit ja niiden lukumäärät. Liituskoodiin on mahdollista määrittää yhteensä seitsemän erilaista liitoskomponenttia. Tarpeen vaatiessa liitoskomponenttiohjelma ja liituskoodi ovat laajennettavissa useammallekin kuin seitsemälle liitoskomponentille.
- Viimeisenä liituskoodissa on liitostyypille kuuluvan detail-kuvan nimi.

Liituskoodissa eri osiot on eroteltu kauttaviivalla. Liitoskomponenttien tyyppi ja lukumäärä on eroteltu risuaidalla. Erottelumerkkien avulla liitoskomponenttiohjelma pystyy erottelemaan liituskoodista eri osat ja noutamaan tarvittavat komponentit tietokannasta.

6.4 Käyttöliittymä



Kuvio 21. Käyttöliittymä

Liitoskomponentti-ohjelman käyttöliittymä on kuviossa 21 esitetyn mukainen.

Kohdassa 1. käyttäjä valitsee liitostyyppin, jonka hän haluaa instrumentille lisätä.

Kohdassa 2. ohjelma listaa eri liitosmahdollisuudet kyseiselle liitostyypille. Kohdasta 2. käyttäjä valitsee sopivan liitostyyppin ja painaa OK-painiketta, jolloin ohjelma lisää valitulle instrumentille vastaavan liitoskoodin.

6.5 Liitoskomponentti-tietokanta

Liitoskomponentti-tietokannassa on määritelty jokaiselle liitoskomponentille tarvittavat tiedot, jotka vaaditaan, että liitoskomponentteja pystytään käsittelemään yrityksen järjestelmissä.

No.	POS	ITEM No.	NAME	SIZE	STD	MATERIAL	UNIT	UNIT WEIGHT
S1	501	30103520290RA	HEXAGONAL SCREW	M12x55	DIN 931 / 8.8	FE-ZN HOT	PCS	0,06
S2	502	30103520304RA	HEXAGONAL SCREW	M12x70	DIN 931 / 8.8	FE-ZN HOT	PCS	0,07

Kuvio 22. Kuva liitoskomponentti-tietokannasta

- No. – sarake on liitoskoodissa käytettävä tunnus.

- POS – sarake on liitoskomponenttia vastaava osanumero piirustuksessa.
- Lopuissa sarakkeissa on esitetty tiedot, jotka vaaditaan piirustuksen osaluettelon muodostamista varten.

6.6 Liituskoodi-tietokanta

Liituskoodi-tietokantaan on listattu yrityksessä yleisesti käytetyt liitostyypit. Liituskoodi-tietokannan perusteella muodostetaan instrumenteille liituskoodit.

Con. Type	Qty screw	Screw	Nut	Washer
Neck Flange + Neck Flange DN10 PN40	4	S1	N1	W1
Neck Flange + Neck Flange DN15 PN40	4	S1	N1	W1

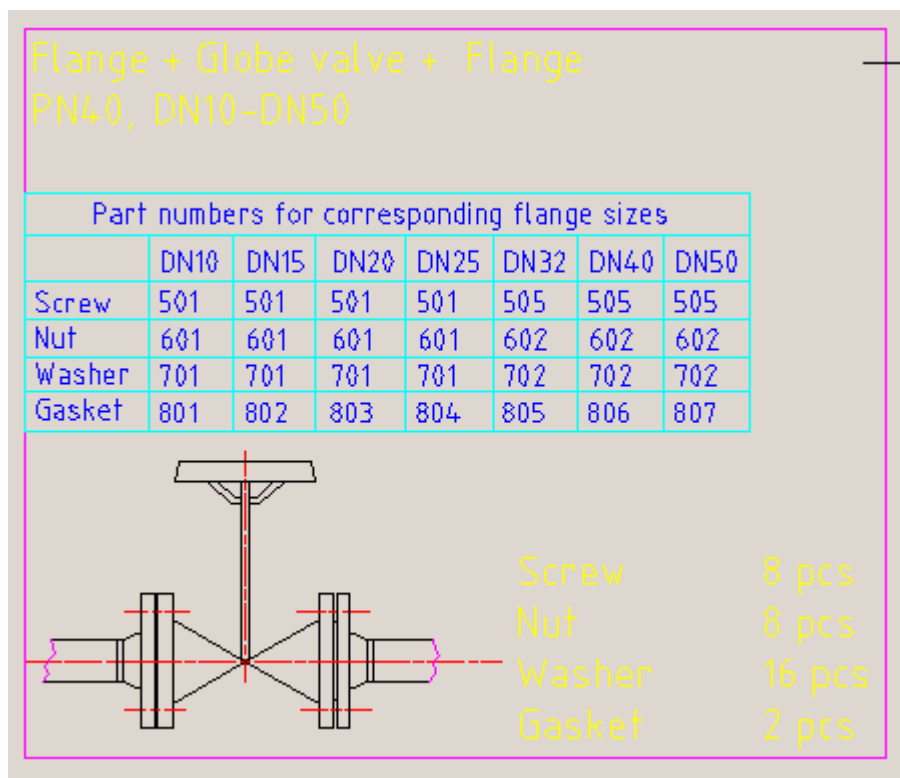
Kuvio 23. Kuva liituskoodi-tietokannasta

- Con. Type – sarakkeessa on määritelty liitostyyppi, DN-koko ja PN-paineluokka.
- Qty screw – sarakkeessa on määritelty liitoksen ruuvien lukumäärä. Mutterien ja aluslaattojen lukumäärät Liitoskomponentti-ohjelma laskee ruuvien lukumäärän perusteella.
- Screw-, Nut- ja Washer – sarakkeissa on määritelty liitoksessa käytettävien liitoskomponenttien tunnuksat. Tunnusten avulla Liitoskomponentti-ohjelma noutaa oikeat liitoskomponentit liitoskomponentti-tietokannasta.

Kuviossa 23 on näytetty vain osa liituskoodi-tietokannan sarakkeista. Muissa sarakkeissa voidaan määritellä lisää liitoskomponentteja. Sarakkeet ovat vastaavaa muotoa kuin kuviossa 23 näkyvät.

6.7 Liitostyyppien detail-kuvat

Eri liitostyyppien detail-kuvista tehtiin AutoCAD:iin kirjasto, josta suunnittelija voi hakea tarvittavat detail-kuvat liitostyypeistä tehdessään piirustusta laitteesta. Kirjoitushetkellä liitostyyppien detail-kuvista ei ole vielä täydellistä kirjastoa, vaan se tul- laan toteuttamaan valmiiksi myöhempänä ajankohtana. Detail-kuvien piirtoon käytettiin AutoCAD:a.



Kuvio 24. Laippa, istukkaventtiili, laippa – liitostyyppin detail-kuva

Kuviossa 24 on esitetty esimerkki detail-kuvasta. Detail-kuva sisältää osanumerot ja lukumäärät liitoskomponenteille sekä sivukuvannon liitostyyppistä.

7 ARVIOINTI

Opinnäytetyön tuloksena kehitettyä järjestelmää ei ole raportin kirjoitushetkellä käytönotettu, joten lopullisia arvioita järjestelmän toiminnasta osana yritystä ei pystytä antamaan.

7.1 Arviointi tämänhetkisten tuloksien perusteella

Tähän mennessä järjestelmää on koekäytetty tekemällä jo kerran suunniteltujen laitteiden kokoonpanopiirustuksia uudestaan. Tämän jälkeen järjestelmän avulla suunniteltua kokoonpanopiirustusta on verrattu täysin manuaalisesti suunniteltuun vastaavaan. Järjestelmän avulla suunnitellun kokoonpanopiirustuksen ja manuaalisesti suunnitellun kokoonpanopiirustuksen osaluetteloita on verrattu keskenään ja tutkittu,

että esiintyykö järjestelmän avulla suunnitellun piirustuksen osaluettelossa virheitä. Näiden testauksien perusteella virheitä ei ole esiintynyt ja tulevat testit sekä käyttöönotto antavat lopullisen varmuuden järjestelmän luotettavuudesta.

Kriittisiä kohtia järjestelmässä ovat Liitoskoodi-tietokannan tietojen oikeellisuus ja ohjelmakoodi. Tiedot on syötetty käsin liitoskoodi-tietokantaan, joten satunnaisilta inhimillisiltä virheiltä on vaikea välttyä. Tietokanta sisältää noin 400 erilaista liitoskoodia, joten ainakin yksittäisten virheiden löytyminen on todennäköistä. Ohjelmakoodille on suoritettu koekäyttöä ohjelmoinnin aikana useita kertoja ja se on toiminut oletetulla tavalla, joten suuria virheitä ei ole odotettavissa.

7.2 Koekäyttö ja käyttöönotto

Koekäytön tarkoituksena on löytää mahdolliset virheet järjestelmästä ja varmistua sen luotettavuudesta siinä määrin, että järjestelmä voidaan käyttöönottaa. Järjestelmää on tarkoitus koekäyttää oikealla projektilla. Koekäyttöön osallistuvat henkilöt eivät ole olleet osallisina järjestelmän kehittämiseen, joten sen perusteella tullaan toteamaan järjestelmän käyttöliittymän käyttökelpoisuus, sekä järjestelmän soveltuvuus todelliseen suunnittelutyöhön.

Järjestelmän käyttöönotto on tarkoitus toteuttaa kesän aikana. Ennen järjestelmän yleistä käyttöönottoa sille tulee tehdä riittävän kattava koekäyttö, siitä tulee laatia käyttöohjeet sekä henkilökunta on perehdytettävä sen käyttöön.

7.3 Laajennettavuus

Kehitettyä järjestelmää on mahdollista laajentaa kattamaan muitakin yrityksen toimintoja kuin laippaliitoksien liitoskomponenttien automatisoinnin. Laajennettavuus lisää järjestelmän käyttöarvoa ja laajennuksilla on mahdollista tehostaa yrityksen sisäisiä toimintoja. Alla on lueteltu esimerkkejä mahdollisista laajennuskohteista, jotka jätettiin opinnäytetyön ulkopuolelle jotta se ei laajenisi liian suuritöiseksi.

Järjestelmän avulla voidaan tehostaa varaston toimintaa. Tällä hetkellä kaikkien liitoskomponenttien oikeat määrät lasketaan varastossa käsin. Järjestelmään on mahdollista luoda jokaiselle liitoskomponentille tietokanta, joka sisältää tiedon varastossa olevista pakkauskoista. Tämän tiedon perusteella järjestelmä voidaan ohjelmoida pyöristämään liitoskomponenttimäärät ylöspäin seuraaviin mahdollisiin pakkauskoihin. Näin ollen varaston henkilöstön tarvitsee laskea vain oikea määrä pakkauksia, eikä yksittäisiä liitoskomponentteja.

Piirustukset, jotka sisältävät liitostyyppien detail-kuvat, on mahdollista luoda ohjelmallisesti. Tällä hetkellä Liitoskomponentti-ohjelma luo tekstitiedoston, joka sisältää listan piirustukseen vaadittavista detail-kuvista. Tämän listan perusteella suunnittelija lisää manuaalisesti detail-kuvat AutoCAD:n kirjastosta piirustukseen. AutoCAD:n LISP – ohjelmointikieli tarjoaa mahdollisuuden ohjelmoida AutoCAD:iin aputyökalun, jolla pystytään luomaan automaattisesti erillinen piirustus detail-kuville edellä mainitun tekstitiedoston perusteella.

Liitoskomponentti-ohjelman käyttöliittymää on mahdollista parantaa. Nyt käyttäjän tarvitsee valita manuaalisesti instrumentille sopiva liitostyyppi painamalla painikkeita. Ohjelmallisesti on mahdollista lukea valitun instrumentin koko ja paineluokka, jonka perusteella käyttöliittymä voi tarjota automaattisesti vain kyseiselle instrumentille sopivat liitostyyppivaihtoehdot.

Järjestelmä on mahdollista laajentaa kattamaan kaikki muutkin yrityksen suunnitteluosastolla käytössä olevat vakio-osat. Tällä hetkellä järjestelmä kattaa vain laippaliitokset ja muutaman erikoistapauksen muunlaisista liitoksista.

7.4 Tavoitteiden täytyminen

Kehitetty järjestelmä pystyy toteuttamaan sille asetetut tavoitteet. Järjestelmän toiminta on alustavassa koekäytössä osoittautunut luotettavaksi, joten voidaan olettaa, että tulevaisuudessa testauksissa saavutetaan myös myönteisiä tuloksia. Järjestelmän kehitys on vielä toistaiseksi kesken, koska sitä ei ole käyttöönotettu yrityksessä. Käyttöönotto tulee sisältämään omat haasteensa ja järjestelmän tuottamat positiiviset vaikutukset.

set yrityksen toimintaan alkavat näkyä vasta jonkin ajan kuluttua sen käyttöönottamisesta. Vasta kun järjestelmää on käytetty jonkin aikaa osana yrityksen toimintaa, voidaan saada lopullinen varmuus sen tuottamista hyödyistä. Järjestelmän kehittäminen on sujunut toivotusti ja järjestelmästä on odotettavissa tulevaisuudessa positiivisia tuloksia.

7.5 Ennen ja jälkeen – vertailu

Järjestelmä tulee vähentämään suunnittelijan työmäärää oleellisesti, koska sen avulla päästään eroon monista toistuvista työvaiheista. Projektipäällikön tekemä työmäärä tulee lisääntymään hieman, koska hänen pitää lisätä instrumenteille liituskoodit. Kokonaistyömäärä laitetta kohden tulee kuitenkin vähenemään merkittävästi. Ihmisenäkökulmasta katsottuna järjestelmän aiheuttama työtapojen muutos ei ole suuri ja siihen sopeutumisen voi olettaa tapahtuvan helposti.

PaavoCAD:n toimintaan vaikuttavia teknisiä muutoksia on järjestelmän myötä tullut useita. Alla olevaan kuvioon 25 on merkattu punaisilla laatikoilla järjestelmän aiheuttamat muutokset PaavoCAD:iin. Merkittävin uusi asia PaavoCAD:ssa on Liitoskomponentti-ohjelma, joka prosessoi liituskoodien sisältämän informaation ja lisää tämän perusteella osaluetteloon tarvittavat liitoskomponentit osanumeroineen.

antava työskentelytapa sopii hyvin luonteelleni ja tästä johtuen myös opinnäytetyön tekeminen oli mielekästä.

Lopuksi haluan kiittää Juha Mikkilää, opinnäytetyöni ohjaajaa Alfa Lavalin puolelta, ja Petteri Reunamoa, opinnäytetyöni ohjaajaa koulun puolelta, sekä kaikkia muita, jotka ovat jollain muotoa joutuneet olemaan opinnäytetyön kanssa tekemisissä. Jokaisen suuren taiteilijan tulee myös kiittää samassa taloudessa majailevaa naishenkilöä, joten kiitän tässä myös tyttöystävääni Jennaa henkisenä innoittajana olemisesta.

LÄHTEET

Alfa Lavalin www-sivut. 2013. Viitattu 4.4.2013.

http://local.alfalaval.com/fi-fi/tietoa_alfa_lavalista/pages/tietoa-alfa-lavalista.aspx

Wikipedia. 2013. Visual Basic for Applications. Viitattu 6.4.2013.

http://en.wikipedia.org/wiki/Visual_Basic_for_Applications

Silander, S. 2000. Ohjelmointi – Pro Training. Jyväskylä: Gummerus.

Wikipedia. 2013. Ohjelmointikielen tulkki. Viitattu 10.4.2013.

http://fi.wikipedia.org/wiki/Ohjelmointikielen_tulkki

Alfa Laval Aalborgin oma markkinointimateriaali. 2013.

LIITE 1