

LAPPEENRANNAN TEKNILLINEN YLIOPISTO

Tietotekniikan osasto

DIPLOMITYÖ

**TIETOJÄRJESTELMÄN KEHITTÄMINEN MARKKINAEHTOISTEN
SÄHKÖSOPIMUSTEN HALLINTAAN**

Lappeenrannassa 23. toukokuuta 2007

Antti Reinikka

Konstunkaari 1 D 19

54915 Saimaanharju

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Tietotekniikan osasto

Antti Reinikka

Tietojärjestelmän kehittäminen markkinaehtoisten sähkösopimusten hallintaan

Diplomityö

2007

54 sivua, 17 kuvaa, 1 taulukko ja 2 liitettä

Tarkastajat: Professori Esa Kerttula
Johtaja Marko Pollari

Hakusanat: tietojärjestelmä, tietojärjestelmien kehitys, dynaamiset sähkösopimukset

Keywords: information system, development of information systems, time limited electricity contracts

Tässä työssä toteutettiin Lappeenrannan Energia Oy:lle tietojärjestelmäratkaisu, jolla yritys kykenee helposti ja vaivattomasti hallitsemaan asiakkaiden dynaamiset sähkösopimukset. Työn teoriaosa antaa lukijalle kattavan kuvan tietojärjestelmien kehitysprosessista, sekä esittelee työmenetelmiä ja mahdollisia ongelmia. Myös sähkömarkkinoista esitellään kompakti tietopaketti lukijalle, jotta toteutusosan asiat olisivat selkeämpiä.

Käytännön osuus esittelee Lappeenrannan Energia Oy:n dynaamisten sähkösopimusten nykyistä toimintamallia ja uuden toteutettavan tietojärjestelmän vaatimukset. Uuden järjestelmän tavoitteena oli tuoda yritykselle kilpailuetua, parantamalla työn tehokkuutta ja sopimuksien hallintaa. Uusi tietojärjestelmä käydään läpi ominaisuuksittain, jotta saadaan hyvä kokonaiskäsitys sen kaikista ominaisuuksista.

Työ käsittää siis toteutuksen aina suunnitteluasteelta lopputulokseen asti. Myös tulevaisuuden kehityksen ensiaskeleet esitellään työn lopuksi.

ABSTRACT

Lappeenranta University of Technology
Department of Information Technology

Antti Reinikka

Development of information system for controlling market based electricity contracts

Masters Thesis for Degree of Science in Technology

2006

54 pages, 17 figures, 1 table and 2 appendices

Examiners: Professor Esa Kerttula
Director Marko Pollari

Keywords: information system, dynamic electricity contracts, development of information systems

An information system was produced for Lappeenrannan Energia Oy during this masters thesis. The company had need for this information system, because they needed a simple and an easy-to-use system to control their customers dynamic electricity contracts. The theory part of this thesis gives a wide prospect to development of information systems and it also points out few working methods and possible problems. Also there is an overview to electrical markets.

The practical part of this thesis describes the company's current principles for dynamic contracts and then it shows the demands for the new information system. Main goal for the new information system, was to give advantage for the company when comparing to other electric companies, by improving working efficiency and improving contract controlling. Thesis shows the new systems properties one by one. So this report includes the whole accomplishment from design to result.

ALKUSANAT

Tämä työ tehtiin Lappeenrannan Energia Oy:ssä vuonna 2006, vaikkakin loppuraportin työstäminen venytti diplomityöraportin reilusti vuoden 2007 puolelle.

Haluan kiittää työni onnistumisen johdosta muutamia ihmisiä, joita ilman tämä työ ei olisi valmistunut. Lappeenrannan Energia Oy:n myyntiosaston johtajalle ja tämän työn ohjaajalle Marko Pollarille haluan antaa erityiskiitokset kärsivällisyydestä ja ansiokkaasta opastuksesta. Yliopiston puolelta haluan kiittää alkuperäistä tarkastajaa Arto Kaarnaa erinomaisista ohjeista, sekä Esa Kerttulaa työn loppuvaiheen ohjeista. Avopuolisoani Katria ja lapsiani haluan kiittää kärsivällisyydestä sekä ajoittaisesta työskentelymotivaation kohottamisesta.

SISÄLLYSLUETTELO

TIIVISTELMÄ	
ABSTRACT	
ALKUSANAT	
SISÄLLYSLUETTELO	1
1 JOHDANTO	3
1.1 Yleistä tietojärjestelmistä	3
1.2 Tietojärjestelmän perusosat	4
2 TIETOJÄRJESTELMIEN KEHITTÄMINEN	5
2.1 Yleistä	5
2.2 Kehityskaaren eri vaiheet	6
2.2.1 Esitutkimus	7
2.2.2 Vaatimusmäärittely	8
2.2.3 Järjestelmäanalyysi	10
2.2.4 Suunnittelu	10
2.2.5 Toteutus	12
2.2.6 Testaus	13
2.2.7 Käyttöönotto ja ylläpito	14
2.3 Kehittämisen ongelmakohtia	15
2.4 Elinkaarimallit	16
2.4.1 Vesiputousmalli	16
2.4.2 Prototyyppimalli	17
2.4.3 Spiraalimalli	19
3 KEHITTÄMISMENETELMÄT JA APUVÄLINEET	20
3.1 Kuvaustavat	21
3.1.1 Oliomallit	22
3.1.2 Tietomallit	23
3.1.3 Prosessimallit	24
3.1.4 Tilamallit	25
3.1.5 Tietovirtakaaviot	26
3.2 Menetelmät	26
3.3 Systeemyömenetelmien edut ja haitat	27
4 SÄHKÖMARKKINAT	28
4.1 Perusteet	28
4.2 Pörssituotteet	29
4.3 SPOT-hinnat	30
4.4 Dynaamiset sähkö sopimukset	31
5 LRE OY:N TIETOJÄRJESTELMÄN SUUNNITTELU	32
5.1 Tietojenkäsittelyn tila ennen	32
5.2 Uuden tietojärjestelmän vaatimukset	33
5.3 Tietojärjestelmä ratkaisun esittely	34
6 TIETOJÄRJESTELMÄN TOTEUTUS	36
6.1 Valmistelu	36

6.2	Toteutus.....	36
6.2.1	Ohjelmointikieli	37
6.2.2	Tiedostojärjestelmä	37
6.2.3	Tietokantaratkaisu.....	38
6.2.4	Markkinatietojen haku	38
6.2.5	Valikkorakenne	40
6.2.6	Perustietojen kerääminen	41
6.2.7	Energiatiedot	43
6.2.8	Hinnankiinnitykset.....	44
6.2.9	Tietojen katselu.....	46
6.2.10	Raportointi	48
6.3	Testaus	51
7	JOHTOPÄÄTÖKSET JA YHTEENVETO	51
	LÄHDELUETTELO.....	53

1 JOHDANTO

Työn aiheena oli dynaamisten sähköasiakkaiden sopimusten hallintajärjestelmä Lappeenrannan Energia (LRE) Oy:lle. Yhtiöllä oli tarvetta erilliselle sovellukselle, jolla myyjät kykenevät hoitamaan asiakkaiden dynaamiset sähkö sopimukset. Sovellus toteutettiin pohjautuen olemassa olevaan Excel- taulukossa sijaitsevaan kirjanpitoon. Uuteen järjestelmään toteutettiin lisäksi uusia ominaisuuksia, jotka antavat lisäarvoa niin myyjille kuin asiakkaille. Työn tavoitteena oli siis luoda tietojärjestelmä, jonka avulla myyntihenkilöstön on helppo ylläpitää ja hallita dynaamisia sähkö sopimuksia. Toisena tavoitteena oli nopeuttaa yrityksen myyntihenkilöstön päivittäisiä rutiineja, jotka liittyvät kyseisten sopimusten kanssa työskentelemiseen.

Tämän työn tietojärjestelmä koostuu sopimusten hallintasovelluksesta sekä siihen liittyvästä ftp-liikennöinnistä pohjoismaisen sähköpörssin ftp-palvelimen ja sovelluksen välillä. Tietoliikennettä tapahtuu myös ohjelmiston sekä tietovaraston välillä. Tietovarastona toimii yrityksen olemassa oleva tiedostopalvelin. Tietojärjestelmä sisältää automatisoituja sekä manuaalisia osia.

Työssä toteutettavan järjestelmän vaatimukset ja toiminnot käytiin läpi yrityksen myyntihenkilöstön kanssa yhdessä. Muu suunnittelu työ ja toteutuksen suunnittelu sekä itse ohjelmointi oli täysin tämän diplomityön tekijän vastuulla.

1.1 Yleistä tietojärjestelmistä

Sana tietojärjestelmä viittaa johonkin suurempaan kuin pelkkään ohjelmistoon. Se kuvaa jotain suurempaa kokonaisuutta. Sanakirjan määritelmän mukaan tietojärjestelmä tarkoittaa seuraavaa /8/ : ”Ihmisistä, tietojenkäsittelylaitteista, tiedonsiirtolaitteista ja ohjelmistoista koostuva järjestelmä, jonka tarkoituksena on tietojen käsittelyn avulla tehostaa tai helpottaa jotain toimintaa tai tehdä se ylipäätään mahdolliseksi.” /

Vaikka sanakirjamääritelmä antaakin jossain määrin kuvan tietojärjestelmien automaattisuudesta, ei näin välttämättä ole. Tietojärjestelmä voi olla täysin manuaalinen tai mitä vain automaattisen ja manuaalisen väliltä. Käytäntö onkin osoittanut, että useimmat automaattiset tietojärjestelmät ovat alun perin kehittyneet manuaalisista tietojenkäsittelykäytännöistä, jotka on tekniikan kehittyessä kyetty automatisoimaan. Sekä automaattisella, että manuaalisella osalla tietojärjestelmää on omat rajapintansa toistensa ja ympäristön kanssa. Rajapinnan tehtävä on määritellä mitä syötteitä ja mitä tulosteita tietojärjestelmä pystyy käsittelemään. Näin ollen tietojärjestelmän käsite on ohjelmiston käsitettä laajempi, koska se käsittää myös tietojenkäsittelyn ympäristön erilaiset organisaatiot sekä sosiaaliset ja inhimilliset ulottuvuudet.

1.2 Tietojärjestelmän perusosat

Tietojärjestelmä koostuu siis tietyistä perusosista, joita ovat ihminen, tietojenkäsittelylaitteet, tiedonsiirtolaitteet ja ohjelmistot /1/.

Ihminen tuo tietojärjestelmään manuaaliset syötteet, joita hän antaa järjestelmälle, mutta ihminen tuo mukanaan myös inhimillisen erehdyksen mahdollisuuden. Usein ohjelmistot on suunniteltu siten, että ne eliminoivat ihmisen virheen mahdollisuuden tarkastamalla kaikki ihmisen antamat syötteet. Tämä onkin ainoa tapa suojautua ihmisen mahdollisia virheitä vastaan. Ohjelmistoja taas kehittävät ihmiset, joten myös niiden kehitysvaiheessa voi jäädä jotain huomaamatta ja näin ollen tietojärjestelmään on mahdollista aiheuttaa virhe esimerkiksi vääränlaisella syötteellä.

Tietoliikenteen näkökulmasta ajatellen tietojärjestelmän tietojenkäsittelylaitteita ovat fyysiset tietokoneet, joko palvelin tai työasema, sekä verkon kaikki laitteet jotka osallistuvat tiedon välittämiseen. Tämän työn osalta tietojenkäsittely tapahtuu työasemalla, mutta tiedot tallennetaan palvelimelle.

Tiedonsiirtolaitteet eli kytkimet, verkkokortit ja keskittimet ovat yleensä organisaatioilla jo hallussaan ennen tietojärjestelmän kehitystä. Jos kuitenkin tietojärjestelmä kehitetään esimerkiksi vaiheessa, jossa yrityksellä ei ole olemassa olevaa verkkoinfrastruktuuria, on tietojärjestelmän suunnittelussa pohdittava mahdollisia verkon rakennevaihtoehtoja. Mikäli verkkoinfrastruktuuri on jo olemassa, on tietojärjestelmä helpoin toteuttaa käyttäen jo olemassa olevaa rakennetta.

Ohjelmistot luodaan usein tietojärjestelmän kehityksen aikana ja kuuluvat olennaisena osana siihen /8/. Joskin on mahdollista käyttää jo olemassa olevia ohjelmistoja tai ohjelmistokokonaisuuksia, mikäli niistä katsotaan olevan hyötyä. Näiden kaikkien edellä mainittujen osien kehittämiseen paneudumme tarkemmin luvussa 2.

2 TIETOJÄRJESTELMIEN KEHITTÄMINEN

2.1 Yleistä

Kun tietojärjestelmiä kehitetään, on se sitä suorittavalle organisaatiolle sen oman toiminnan kehittämistä. Kehityksen tarkoitus on edesauttaa jonkin halutun toimintatavan muutosta. Tälle voidaan asettaa seuraavanlaisia tavoitteita:

- Auttaa toimintayksikköä suuntautumaan tavoitteisiinsa entistä paremmin
- Mahdollistaa entistä vaativampien tavoitteiden asettamisen
- Tehdä mahdolliseksi jonkin uuden toiminnon
- Tehostaa jo olemassa olevia toimintatapoja.

Toiminnan kehittämisen täytyy kohdistua joko ihmisiin, teknologiaan tai toimintoihin. Usein tietotekniikka-alan tietojärjestelmäkehitys kohdistuu pääosin teknologiaan ja

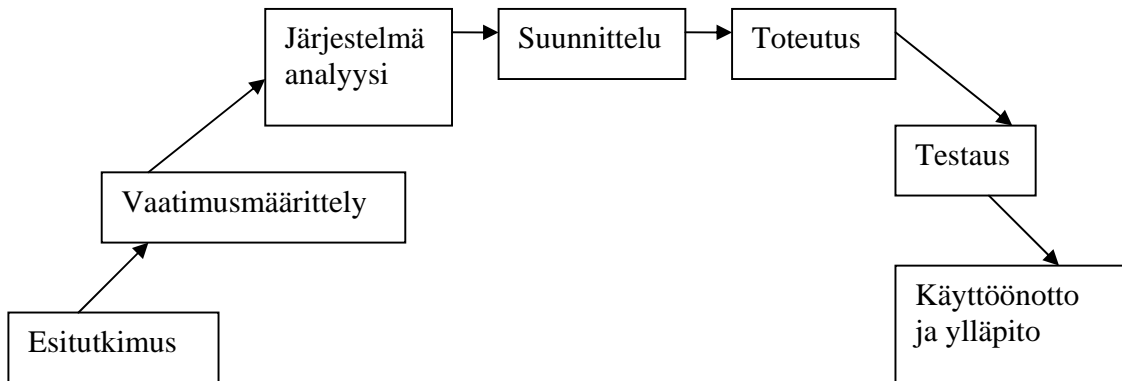
toimintoihin. Teknologiaan kehitys kohdistuu siksi, koska se kehittyy edelleen valtavalla nopeudella ja yrityksille on tärkeää kestää tämän kehityksen mukana. Teknologian kehittyessä tulee tietysti mahdolliseksi monet uudet toiminnot ja näin ollen kehitys kohdistuu myös niihin. Tietojärjestelmien kehitys kohdistuu toki myös ihmisiin esimerkiksi koulutuksena.

Sanakirjamääritelmä tietojärjestelmien kehittämiseksi on uusien tietojärjestelmien laatimista tai nykyisten oleellista muuttamista. Sille rinnakkainen käsite systeemyö, on tietojärjestelmien suunnittelua, ohjelmointia ja ylläpitoa. Risto Pohjosen kirja *Tietojärjestelmien kehittäminen* antaa huomattavasti paremman määritelmän tietojärjestelmien kehittämisestä. Määritelmä kuuluu näin:

Tietojärjestelmien kehittäminen on kehitysryhmän tietyssä ympäristössä kohdejärjestelmälle suorittama muutosprosessi, joka tapahtuu kohdejärjestelmälle asetettujen tavoitteiden mukaisesti /8/.

2.2 Kehityskaaren eri vaiheet

Itse tietojärjestelmien kehitystyö on hyvin systemaattista toimintaa. Tehtävät ja tehtäväkokonaisuudet seuraavat loogisesti toisiaan. Yhteensä näitä toisiaan seuraavia tehtäviä ja vaiheita kutsutaan tietojärjestelmän elinkaareksi /2/. Jako vaiheisiin tekee työstä kuitenkin selkeämpää ja loogisempaa. Myös projektin johtajille vaiheet antavat helpon tavan kontrolloida työn etenemistä, koska on helppo määritellä projektin tarkastelupisteet. Allaoleva kuva 2.1 esittää edellä mainittua elinkaarta. Seuraavaksi tarkastellaan elinkaaren eri vaiheita lähemmin.



Kuva 2.1. Tietojärjestelmän elinkaari /2/

2.2.1 Esitutkimus

Elinkaaren ensimmäinen vaihe on esitutkimus, jonka tarkoituksena on antaa käsitys siitä onko tietojärjestelmän toteutus mahdollista ja mielekästä /8/. Tässä vaiheessa ei vielä ohjelmoida tai rakenneta mitään, eikä myöskään tehdä minkäänlaisia teknisiä ratkaisuja. Erilaisia teknisiä ratkaisumenetelmiä voidaan poimia esiin ja pohtia niiden soveltuvuutta kyseiseen tietojärjestelmään, mutta valintoja ei tehdä elinkaaren tässä vaiheessa /3/. Toisin sanoen esitutkimus kertoo seuraavat asiat:

- Miksi uusi järjestelmä tulisi rakentaa,
- Mitkä ovat sille asetetut päätavoitteet,
- Mitkä ovat järjestelmän viiteryhmät sekä
- esitellä toteutuksen erilaisia ratkaisuvaihtoehtoja.



Esitutkimus onkin lähinnä tarkoitettu järjestelmän kehityksen aloituksesta päättävälle organisaatiolle /3/. Oikein laadittu esitutkimusraportti sisältää seuraavia asioita:

- Organisaation nykyisen tietojärjestelmän kuvauksen
- Nykyiset ongelmat, joihin uudella järjestelmällä haetaan ratkaisua
- Kuvaukset hankkeen sidosryhmistä

- Järjestelmän alustavat tavoitteet ja rajaukset
- Kehittämistavoitteiden määrittelyt
- Erilaisten toimintatapojen kuvaukset
- Jonkinlaisen suunnitelman tietojärjestelmän kehityksen elinkaaresta.

Esitutkimuksen aikana syntyvä aineisto olisi kannattavaa järjestellä ja arkistoida myöhempää käyttöä ajatellen. Mikäli myönteinen päätös tietojärjestelmän kehittämisestä annetaan, on seuraavana vaiheena järjestelmän elinkaaressa vaatimusmäärittely.

2.2.2 Vaatimusmäärittely

Tämän dokumentin tarkoituksena on koota yhteen eri sidosryhmien järjestelmälle asettamat vaatimukset. Ne määrittelevät ryhmien tarpeet järjestelmän suhteen. Vaatimuksia on kahdenlaisia, toiminnalliset ja ei-toiminnalliset vaatimukset /18/. Tässäkään vaiheessa ei kuitenkaan oteta kantaa teknisen toteutuksen suhteen.

Toiminnalliset vaatimukset ovat niitä, jotka kuvaavat mitä ja miten järjestelmän odotetaan toimivan /8/. Ne kuvaavat järjestelmää ulkoapäin tarkasteltuna ja sen kommunikointia oman ympäristönsä kanssa /3/.

Ei-toiminnalliset vaatimukset kuvaavat järjestelmän reuna-ehtoja eli siis missä ympäristössä järjestelmä pystyy toteuttamaan toiminnalliset vaatimuksensa /8/. Erilaisia ei-toiminnallisia vaatimuksia ovat esimerkiksi kapasiteetti ja käytettävyys. Järjestelmän on esimerkiksi toimittava Windows XP- työasemissa ja erilaisten hakujen kesto tietokannasta ei saa ylittää 10 sekunnin raja-arvoa.

Vaatimusmäärittelydokumentti voi sisältää myös rajoitteita, jotka ovat eräänlaisia ei-toiminnallisten vaatimusten erikoistapauksia /18/. Niillä voidaan määritellä rajoituksia järjestelmän toiminnallisille vaatimuksille, kuten esimerkiksi että käyttäjä ei voi tehdä enempiä, kuin viisi hakua yhtä aikaa. Dokumenttiin tulee myös priorisoida kaikki

vaatimukset ja rajoitteet, jotta järjestelmän toteuttamisvaiheessa voi nähdä mitä pitää tehdä ensin, ja mikä taas toisarvoisempaan on toteutettavissa myöhemmin.

Erilaisten asiakasvaatimusten kerääminen on elintärkeää tietojärjestelmiä kehitettäessä, koska muuten ei pystytä tuottamaan sitä, mitä asiakas pyytää. Näiden asiakasvaatimusten kerääminen on kuitenkin erittäin haasteellista työtä. Haaste on siinä, että ei ole olemassa yhtä tiettyä kaiken kattavaa keräysmenetelmää, jolla saadaan riittävän hyvä lopputulos. Useista eri keräysmenetelmistä suosituimmat ja parhaat ovat /19/:

- Eri sidosryhmien haastattelemine
- Aivoriihet ja ideointipalaverit
- Myös markkinatutkimuksia kannattaa hyödyntää

Voi olla myös niin, että erilaiset standardit ja lainsäädäntö asettavat erilaisia vaatimuksia järjestelmälle /19/. Oleellisinta on se, että tässä elinkaaren todella tärkeässä vaiheessa järjestelmän kaikki vaatimukset saadaan selville. Mahdollisten puutteiden korjaaminen myöhemmin voi tulla kalliiksi.

Yhteenvedon vaatimusmäärittelystä voidaankin todeta, että se on yksi tärkeimmistä, ellei tärkein kehityksen elinkaaren vaiheista ja sen dokumentointiin on kiinnitettävä erityistä huomiota. Hyvästä vaatimusmäärittelydokumentista löytyy seuraavat asiat /19/:

- Kuvaus kehityshankkeen toimeksiannosta
- Yleiskuvaus organisaation nykytilasta järjestelmän osalta
- Kuvaus kohdejärjestelmästä ja sen päätavoitteista
- Jokaisen toiminnallisen vaatimuksen kuvaus
- Jokaisen ei-toiminnallisen vaatimuksen kuvaus
- Jokaisen rajoitteen kuvaus
- Vaatimukset ja rajoitteet numeroituna ja priorisoituna
- Mahdolliset lisäselvitykset

2.2.3 Järjestelmäanalyysi

Järjestelmäanalyysin tavoitteena on selvittää, mitä tulevan järjestelmän tulee tehdä. Tämän saavuttamiseksi analysoidaan vaatimusmäärittelyvaiheessa laadittuja vaatimuksia ja johdetaan niistä järjestelmän toiminnallinen määrittely. Toisin sanoen luodaan eri sidosryhmien kesken yleisen tason kuvaus järjestelmän toiminnoista, sen tiedoista, liittymistä ja yhteyksistä ympäristöön /3/. Yhtenä keskeisenä seikkana voidaan myös mainita järjestelmälle asetettujen rajoitteiden tarkentaminen.

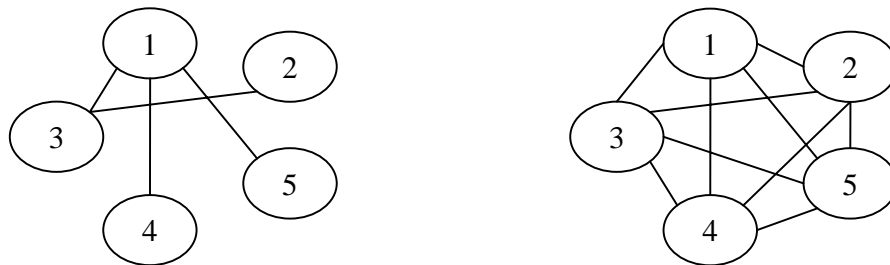
Mainittakoon myös, että järjestelmäanalyysin tuottamien dokumenttien pohjalta voidaan järjestelmän kehitys jakaa useaan yhtäaikaiseen kehityslinjaan. Dokumentin pohjalta voidaan aloittaa suunnittelun lisäksi esimerkiksi testauksen suunnittelu ja käyttöohjeiden laatiminen.

2.2.4 Suunnittelu

Kun kaikki aikaisemmat vaiheet kehityksen elinkaaresta on toteutettu, päästään jo itse asiaan eli tulevan tietojärjestelmän suunnitteluun. Suunnittelun tarkoituksena on muuttaa järjestelmän toiminnallinen määrittely tekniseksi määrittelyksi, joka kuvaa järjestelmän lopullisen toteutuksen. Suunnittelu voidaan jakaa arkkitehtuuri- ja moduulisuunnitteluun, riippuen toteutettavan järjestelmän koosta. Myöskin tietojärjestelmien suunnittelussa voidaan erottaa kolme näkökulmaa; organisaatiokonteksti ja käyttäjät eli organisaatiotaso, käyttäjien mielenkiinnon kohteet eli kielitaso ja laitteet eli teknologiataso. /9/

Arkkitehtuurisuunnitelma määrittelee järjestelmän yleisen rakenteen ja se pyrkii osittamaan järjestelmän sellaisiin osiin, joita voidaan kehittää yhden ihmisen tai ryhmän toimesta. Moduuli on järjestelmän osa, joka sisältää kaikki siihen määritellyt toiminnot ja sillä on määritelty rajapinta, jonka kautta se kommunikoi ympäristönsä kanssa. Moduuli saa syötteitä ympäristöstä ja käsittelee niitä, antaen sitten palautteen syötteestä.

Arkkitehtuurisuunnitelma siis jakaa järjestelmän moduuleihin. Nämä moduulit tulisi suunnitella sellaisiksi, että niillä on mahdollisimman vähän kytkentöjä etenkin toisiin moduuleihin. Mitä vähemmän kytkentöjä on toisiin moduuleihin, sen yksinkertaisempi järjestelmä on. Alla oleva kuva 2.2 havainnollistaa asiaa siten, että numeroidut pallo kuvaavat moduuleja ja pallojen väliset viivat moduulien välisiä kytkentöjä. Kuvasta näkee, että vasemman puoleisessa tapauksessa moduulit on suunniteltu huomattavasti paremmin, koska eri moduulien välisiä yhteyksiä on huomattavasti vähemmän ja näin ollen järjestelmä on selvästi yksinkertaisempi.



Kuva 2.2. Moduulien välisiä kytkentöjä.

Moduulisuunnittelu on nimensä mukaisesti moduulien sisäisen rakenteen suunnittelua. Prioriteettina suunnittelussa voidaan pitää moduulien koon pitämistä mahdollisimman pienenä. Tämä siksi, että suurien moduulien ylläpitäminen on huomattavasti vaivalloisempaa kuin pienten. Pohjosen *Tietojärjestelmien kehittäminen* -kirjaa lainaten, teknisen määrittelyn dokumentaation tulee sisältää seuraavat osa-alueet:

- Tiivistelmä järjestelmän tarkoituksesta
- Kuvaus järjestelmän sovellusalueesta ja järjestelmän osuudesta siinä
- Kuvaus järjestelmän laitteisto- ja ohjelmistoympäristöstä
- Kuvaus järjestelmän toteutuksen keskeisistä reunaehdoista
- Kuvaus järjestelmän ja sen ympäristön välisestä vuorovaikutuksesta
- Kuvaus järjestelmän arkkitehtuurista
- Kuvaus jokaisesta yksittäisestä järjestelmän moduulista ja alijärjestelmästä
- Toteutusrajoitteiden määrittelyt

- Mahdollisten erityisten teknisten ratkaisujen kuvaukset
- Kuvaukset mahdollisista vaihtoehtoista tai hyläytyistä ratkaisuista
- Mahdolliset muut toteutukseen vaikuttavat seikat

Suunnitelman tulee kaikenlaisäksi olla selkeä, ymmärrettävä, tehokas, luotettava ja ylläpidettävä. Suunnitelman pohjalta aloitetaan järjestelmän toteuttaminen, johon paneudutaan seuraavassa kappaleessa.

2.2.5 Toteutus

Ohjelmisto toteutetaan jollain ohjelmointikielellä tai sovelluskehittimellä. Moduulit toteutetaan usein irrallaan toisistaan ja lopuksi kun kaikki on valmista, ne integroidaan yhdeksi toimivaksi järjestelmäksi. Toteutus on sitä suoraviivaisempaa mitä paremmin aiemmat vaiheet on toteutettu. Näin ollen ensimmäisiä vaiheita ei voi liikaa korostaa. Työ on helppoa, koska kaikki tekniset päätökset ja ratkaisut on tehty jo ennen toteutuksen aloittamista. Voi olla tilanteita, joissa ei pystytä etukäteen tekemään ratkaisua siitä, miten jokin toiminto toteutetaan, vaan se on jätettävä toteutuksen aikaiseksi *protoiluksi* eli kehitetään pari vaihtoehtoa ja kokeillaan kumpi toimii paremmin. Tätä tulisi kuitenkin välttää, koska se monimutkaistaa työtä ja virheiden mahdollisuus moninkertaistuu.

Tällä hetkellä on olemassa todella suuri joukko erilaisia kehitystyökaluja tietojärjestelmien kehittämiseen. Näin ollen valinta näiden välillä ei aina ole niin yksinkertaista. Siihen vaikuttaa muun muassa sovellusalue, käytetyt menetelmät ja ohjelmistotuotannon malli, tehokkuusvaatimukset sekä toteutus- ja käyttöympäristö.

Järjestelmän siirrettävyys ja ylläpidettävyys ovat seikkoja, jotka hankaloittavat toteutusvälineen valintaa. Tämän vuoksi selainpohjaiset järjestelmät ovat kasvattaneet suosiotaan, koska niitä voidaan käyttää millä tahansa käyttöjärjestelmällä.

Koska toteutus ja suunnittelu ovat selkeästi hyvin läheisiä vaiheita toisilleen, on tieto järjestelmien toteuttamisesta ja siihen liittyvistä seikoista hyvin tärkeää suunnittelijalle. Yleensä suunnittelija on myös toteuttaja tai ainakin suunnittelijat tekevät hyvin läheistä yhteistyötä toteuttajien kanssa.

2.2.6 Testaus

Ilman testausta ei toteutettua tietojärjestelmää voi ottaa käyttöön tai se ei ainakaan ole suositeltavaa. Testaus on syytä tehdä huolella ja siitä on laadittava oma suunnitelma. Testaamisen tarkoitus on löytää järjestelmästä mahdollisia vikoja. Yleisin testausmalli on V-malli, jossa testaaminen jaetaan moduuli-, integrointi- ja järjestelmätestaukseen. V-mallin mukaisessa testauksessa, järjestelmätestaus suunnitellaan osana järjestelmäanalyysiä ja testatessa valmista järjestelmää, sitä verrataan sen toiminnalliseen määrittelyyn. Integrointitestaus suunnitellaan arkkitehtuurisuunnittelun yhteydessä ja moduulitestaus moduulisuunnittelun yhteydessä. Toisin sanoen testaaminen ei saa koskaan perustua ohjelmakoodiin, vaan laadittuihin suunnitelmiin ja määritelmiin.

Sisäinen ja ulkoinen testaus ovat moduulitestauksen strategioita. Sisäisellä testauksella varmistetaan eri komponenttien virheettömyys, kun taas ulkoinen testaus varmistaa eri komponenttien tuottamien tulosten oikeellisuuden, suhteessa syötearvoihin. Muita mahdollisia testausstrategioita on monia, joista muutamia on esitelty alla olevassa taulukossa 3.1. Käytännössä sellaista testausta ei voida järjestää, joka paljastaisi kaikki järjestelmän virheet, mutta kun mahdollisimman kattava testaus järjestetään, päästään tulokseen, jossa järjestelmästä on suurimmat ja ongelmallisimmat virheet löydetty. Loput virheet ovat yleensä pieniä ja ilmaantuvat käytön aikana, jolloin ne kuuluvat ylläpidon piiriin.

Taulukko 3.1. Testausstrategioita

Suorituskykytestaus	Varmistetaan järjestelmälle asetetut suorituskykytavoitteet
Palautumistestaus	Testataan järjestelmän kykyä selviytyä erilaisista poikkeustilanteista
Läpikäynnit	Ulkopuolisten asiantuntijoiden tai järjestelmän käyttäjien suorittamia tarkastuksia ja arvosteluja, jotka kohdistuvat järjestelmän toteutukseen, kuvauksiin ja määrittäisiin.

2.2.7 Käyttöönotto ja ylläpito

Järjestelmän läpäistyä testauksen menestyksekkäästi, on aika siirtyä käyttöönotto-vaiheeseen. Käyttöönotto-vaiheen tärkeimpiä huomioitavia seikkoja ovat olemassa olevien tietojen, tiedostojen ja tietokantojen siirtäminen sekä mahdollinen muuntaminen uuteen järjestelmään. On myös otettava huomioon mahdollisesti olemassa olevat rinnakkaiset tietojärjestelmät. Käyttäjien koulutusta uuden tietojärjestelmän toimintaan ei sovi unohtaa. Vähintään on syytä laatia kattava ja selkeä käyttöohjeistus, mutta sen läpikäynti käyttäjien kanssa on suositeltavaa mahdollisimman sujuvan käyttöönoton takaamiseksi.

Ylläpitovaihe on järjestelmäkehityksen pisin yksittäinen ja samalla viimeinen vaihe. Tämän vaiheen tarkoituksena on huolehtia käytössä olevan järjestelmän toimintakunnosta virheitä korjaamalla, sekä jatkokehittämällä järjestelmää. Erilaisia ylläpitotapoja on yleisesti katsoen neljä. Näitä ovat korjaava ylläpito, sopeuttava ylläpito, täydentävä ylläpito ja ennakoiva ylläpito.

Korjaava ylläpito käsittää lähinnä käyttöönoton jälkeiset virheiden korjaamiset. Sopeuttavalla ylläpidolla tarkoitetaan järjestelmän siirtämistä uusiin ympäristöihin. Täydentävä ja ennakoiva ylläpito keskittyvät molemmat järjestelmän kehittämiseen, poiketen toisistaan siinä, että täydentävässä ylläpidossa luodaan uusia ominaisuuksia

järjestelmään ja ennakoivassa ylläpidossa myös kehitetään olemassa olevia ominaisuuksia sekä dokumentaatiota paremman ylläpidon toivossa.

Ylläpidon kannalta kattava dokumentaatio on elintärkeää, koska tässä vaiheessa on hankalaa selvittää tietojärjestelmän kehitysvaiheessa tehtyjä suunnittelu- ja toteutusratkaisuja ilman asianmukaista dokumentaatiota. Suotavaa olisi käyttää standardimuotoista dokumentointitapaa, koska se helpottaa suuresti tiedon etsimistä. Myös uusille työntekijöille, joille yritys ei ole tuttu, mutta standardinmukainen dokumentaatio on tuttua, on helpompaa löytää tietoja.

2.3 Kehittämisen ongelmakohtia

Suurin ja yleisin ongelma tietojärjestelmien kehityksessä ovat epäonnistuneet kehityshankkeet ja –projektit. Oikeastaan suurin osa projekteista epäonnistuu jollain tavalla, mutta tällöin mukaan täytyy laskea pienimmätkin lipsahdukset ja puutteet alkuperäisestä suunnitelmasta. Tosielämässä erittäin monet ohjelmistoalan projektit näennäisesti epäonnistuvat, eli ei pysytä aikarajoissa, budjetti pettää tai resurssointi epäonnistuu. Kenen vika tämä sitten on? Yleisimmin syynä on puutteellinen projektin hallinta, tosin tekniset vaikeudet voivat olla osasyynä. Etenkin kehityshankkeiden koon kasvaessa eri kehitysryhmien välinen kommunikaatio voi olla ongelmallista, varsinkin jos selkeitä sääntöjä ei ole sovittu.

Tuottavuus ja kustannukset tietojärjestelmien kehityksessä luovat toisen ongelma-alueen. Ongelma johtuu nykyisten tietojärjestelmien kasvamisesta ja monimutkaistumisesta, jonka teknologian kehitys on luonut mahdolliseksi. Teknologian kehitys on tuonut paljon uusia mahdollisuuksia, joita tietojärjestelmiin voidaan toteuttaa. Kaiken tämän lisäksi ongelman vakavuutta lisää se, että nämä laajemmat ja monimutkaisemmat järjestelmät tulisi toteuttaa yhä nopeammin. Edes uusien työkalujen ja menetelmien kehitys ei auta tämän ongelman ratkaisussa tarpeeksi, jotta kehitystyön tuottavuutta voitaisiin kasvattaa ainakaan niin paljon kuin olisi tarpeen. /8/

Järjestelmän ylläpitoon ja yleisesti ottaen koko tietojärjestelmäkehitykseen liittyvä ongelmapiirre on se, että mitä aikaisemmassa kehityksen vaiheessa korjattava virhe on tehty, sen kalliimmaksi ja työläemmäksi sen korjaaminen tulee. Tämän takia kehitystyössä painotetaan alkuvaiheiden, kuten suunnittelun tärkeyttä ja huolellista dokumentaatiota niin paljon. Tähän liittyy myös asiakkaan uudet vaatimukset esimerkiksi toteutusvaiheessa, koska uusien vaatimusten toteuttaminen johtaa lähes automaattisesti siihen, että joudutaan palaamaan kehityksen elinkaarella taaksepäin, aina vaatimusmäärittelyn tasolle asti. Tämä tulee kalliiksi ja vie turhaa aikaa sekä korostaa esitutkimuksen ja vaatimusmäärittelyvaiheiden merkitystä, että hyvää kommunikaatiota asiakkaan kanssa. Sitä, voidaanko näitä tietojärjestelmän kehityksen ongelmia koskaan kokonaan ratkaista, ei kukaan voi tietää. Silti järjestelmällisellä työllä ja menetelmiä kehittämällä voidaan saada aikaan paljon hyödyllistä. Kaiken ratkaiseminen lienee kuitenkin mahdotonta, huomioonottaen kaikki edellä mainitut ongelmat ja sen, että paljon on kiinni ihmisten toiminnasta. Ihminen ei koskaan tule olemaan niin täydellinen, että inhimillisiä erehdyksiä ei tapahtuisi.

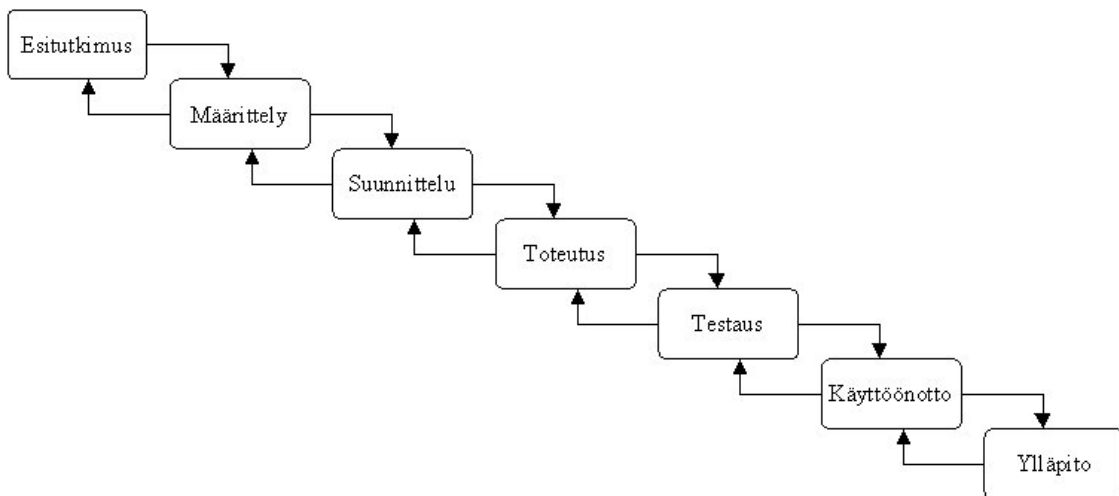
2.4 Elinkaarimallit

Tietojärjestelmiä kehitettäessä on havaittavissa erilaisia elinkaarimalleja. Yleisimmät niistä ovat vesiputousmalli, prototyypimalli sekä spiraalimalli. Ne periaatteessa eroavat toisistaan paljonkin, mutta silti jokaiseen voidaan katsoa pätevän edellisessä kappaleessa esitellyt kahdeksan elinkaarivaihetta. Vesiputousmallissa ne ovat selkeimmin nähtävissä, mutta silti jokaisen mallin katsotaan käsittävän kyseiset vaiheet. Seuraavaksi tarkastellaan lähemmin näitä erilaisia kehityksen elinkaarimalleja.

2.4.1 Vesiputousmalli

Vesiputousmalli kehitettiin jo 1960- ja 70-luvun vaihteessa. Sen kehittäjänä mainitaan usein W.W. Royce, joka julkaisi artikkelin aiheesta vuonna 1970. Mallissa

tietojärjestelmän kehitys nähdään tasaisesti alaspäin virtaavana vesiputouksena, jossa taaksepäin palaaminen on työlästä. Ideaalitapauksessa työvirta soljuu luontevasti läpi putouksen ja tietojärjestelmä on valmis, mutta näin hyvin harvoin tapahtuu. Vesiputousmalli ei kuvasta kehityksen iteratiivisuutta kovinkaan hyvin, sillä monesti asioita joudutaan korjaamaan eli palaamaan aiempiin vaiheisiin, koska virheitä paljastuu myöhemmissä vaiheissa. Mallin ongelma on myös se, että yleisesti tarkastuspisteet ja dokumentit kiinnitetään eri vaiheiden rajapintoihin, eli toisin sanoen vaiheen loppudokumentti toimii syötteenä seuraavalle vaiheelle /10/. Myöskään varsinaisia tuloksia ei voida asiakkaalle esitellä kuin vasta hyvin myöhäisessä vaiheessa, joka johtaa siihen, että asiakkaan huomattaessa virheitä järjestelmässä, niiden korjaaminen on kallista ja työlästä. Tämän vuoksi esimerkiksi prototyypimalli on huomattavasti parempi tässä suhteessa. Kuva 2.3 esittää vesiputousmallin rakenteen ja siitä käy ilmi hyvin edellä mainitut asiat.



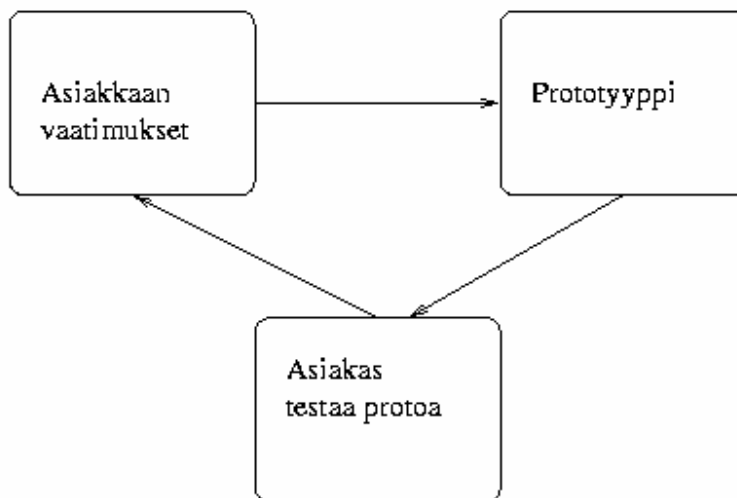
Kuva 2.3. Vesiputousmalli /3/

2.4.2 Prototyypimalli

Kuten edellä todettiin, vesiputousmalli soveltuu huonosti silloin, jos asiakas haluaa nähdä tuloksia nopeasti ja silloin, kun ei olla aivan täysin varmoja mitä halutaan. Tämän tyyppisiin tilanteisiin soveltuu prototyypimalli huomattavasti paremmin. Malli kierrättää

käytännössä elinkaaren kuutta keskimmäistä vaihetta iteratiivisesti läpi nopealla tahdilla /10/. Eli kun on tutkittu mitä asiakas haluaa, ryhdytään määrittelemään, suunnittelemaan ja toteuttamaan ensimmäistä prototyyppiä järjestelmästä.

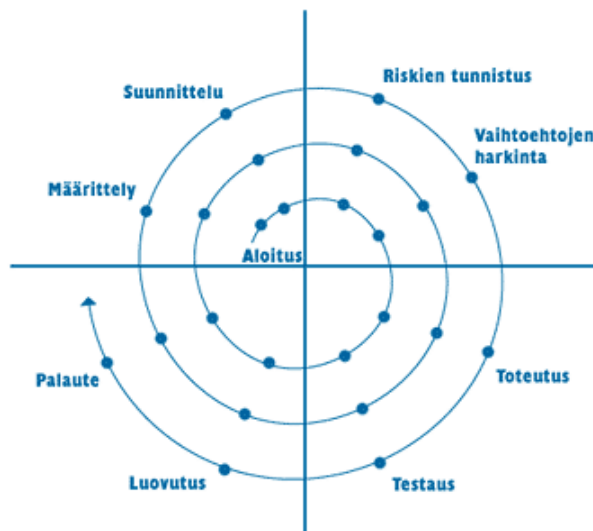
Kun ensimmäinen prototyyppi on valmistettu, voidaan se näyttää asiakkaalle ja keskustella, mitkä asiat ovat oikein, mitkä väärin ja mitä puuttuu. Tämän jälkeen aloitetaan kierros uudestaan ja luodaan toinen prototyyppi ja niin edelleen, kunnes tietojärjestelmä on valmis /12/. Prototyypimallikaan ei ole ongelmaton, koska voidaan katsoa, että monen prototyypin teko kuluttaa resursseja enemmän. On myös mahdollista, että prototyypit eivät paljasta kaikkia mahdollisia virheitä, vaikka niitä testattaisiinkin. Olennaisen tärkeää onkin testata lopullinen tuote läpikotaisin. Tosin tämä johtaa siihen, että mahdollisia virheitä löydetään vielä kehityksen loppuvaiheessa, joka tarkoittaa kustannusten kasvua kuten aiemmin on mainittu. Kuva 2.4 esittää prototyypimallin periaatteen.



Kuva 2.4. Prototyypimalli /12/

2.4.3 Spiraalimalli

Tässä työssä esitellyistä kolmesta elinkaarimallista spiraalimalli kuvaa parhaiten tietojärjestelmäkehityksen iteratiivisuutta. Kuten kuvasta 2.5 voidaan nähdä, mallissa toistuu neljä eri lohkoa peräjälkeen, kun kuljetaan spiraalin sisältä ulkokehälle päin. Lohkot voidaan nimetä esimerkiksi näin: suunnittelu, riskianalyysi, tuotanto, arviointi. Kuten kuvasta nähdään, jokainen lohko sisältää myös kaksi selvää omaa vaihetta. Suunnitteluvaiheeseen kuuluu määrittely sekä suunnittelu, riskianalyysiin riskien tunnistus sekä vaihtoehtojen harkinta. Tämän vaiheen jälkeen tehdään myös aina päätös siitä, jatketaanko vai lopetetaanko järjestelmän kehittäminen eli onko se enää kannattavaa. Tuotantovaiheeseen kuuluu toteutus ja testaus, kun sitten viimeiseen eli analyysivaiheeseen, kuuluvat järjestelmän ”luovutus” ja palautteen saaminen. Toisin sanoen järjestelmä näennäisesti luovutetaan asiakkaalle, joka käy sen läpi ja antaa palautteen siitä onko järjestelmä sellainen kuin haluttiin, ja mitä mahdollisia puutteita siinä on. Näitä vaiheita toistetaan, kunnes tietojärjestelmä on valmis.



Kuva 2.5. Spiraalimalli /13/

Spiraalimalli on näistä kolmesta elinkaarimallista nuorin ja sen vuoksi siitä on vähiten käytännön kokemusta. Mallin kehitti Barry Boehm vuonna 1988 ja se oli ensimmäinen elinkaarimalli, joka näytti miksi iteratiivisuus on tärkeää tietojärjestelmää kehitettäessä

/1/. Spiraalimallin heikkouksia tarkasteltaessa tulee ensimmäisenä ilmi se, että iteratiivisuus voi olla hyvin paljon aikaavievä prosessi. Myös asiakas on aktiivisesti saatava osallistumaan projektiin, koska asiakkaalla on iso rooli palautteen antajana ja sitä myöten kehittäjänä.

3 KEHITTÄMISMENETELMÄT JA APUVÄLINEET

Eri aloilla on eri nimiä kehitystyötä avustaville menetelmille ja apuvälineille. Tietojärjestelmäkehityksessä näitä kutsutaan systeemyömenetelmiksi. Niiden historia ei ulotu kovinkaan pitkälle, vain noin 30 vuotta taaksepäin, verrattaessa niitä muihin toimialoihin ja tästä johtuukin, että menetelmien kirjo on hyvin hajanainen. Tästä johtuen tietojärjestelmien kehittäminen on aina viime vuosiin saakka ollut hyvin paljon kokeellista toimintaa. Ehkä juuri siksi uusia, jopa hyvinkin radikaaleja menetelmiä, on esitelty hyvin nopealla tahdilla. Kukaan ei varmasti osaa sanoa, kuinka monta systeemyömenetelmää on olemassa tällä hetkellä, mutta arviot menetelmien määrästä liikkuu sadoissa, jollei jopa tuhansissa. Menetelmiä on syntynyt näin paljon siksi, että monet menetelmät lainaavat osia toisistaan ja ovat paljolti toistensa kaltaisia, pienine eroineen.

Teknologinen kehitys on tietysti osasyylinen siihen, että menetelmien kehitys on näin hajanaista, koska uusien mahdollisuuksien myötä on myös tullut uusia tarpeita systeemyömenetelmille. Tyyppiesimerkkinä voidaan pitää rakenteista ja olio-lähestymistapaa. Kummassakin tapauksessa on syntynyt ensin uudenlainen ohjelmointitapa, joka on sitten vaatinut tuekseen sen mukaisia suunnittelumenetelmiä. Nämä suunnittelumenetelmät ovat sitten taas vaatineet omanlaisensa määrittely- ja analyysivaiheiden menetelmät. Kuten käy ilmi, uuden teknologian kehittyminen saa parhaassa tapauksessa aikaan lumivyörymäisen efektin, joka kulkee läpi elinkaarimallin luoden jokaiselle vaiheelle oman uuden systeemyömenetelmämallin. On syytä käydä läpi joitain systeemyömenetelmien olennaisia seikkoja, joista seuraavissa kahdessa luvussa enemmän.

3.1 Kuvaustavat

Kuvaustapoja eli notaatioita on todella monia erilaisia. Näitä notaatioita käytetään läpi elinkaaren, kun tietojärjestelmä määritellään ja lopulta toteutetaan. Näitä eri kuvaustapoja on olemassa tuhansia erilaisia ja se, mitä niistä kukin käyttää, on kiinni yrityksen omista linjauksista. Kuvaustapoja on aina pelkistetyistä tekstipohjaisista aina puhtaasti graafisiin esitystapoihin. Ottaen huomioon sen seikan, että tietojärjestelmät ovat yleensä ottaen melko laajoja, on koko järjestelmän kuvaaminen yhdellä kuvaustavalla lähes mahdotonta, jonka johdosta yleisin tapa on käyttää useaa rinnakkaista notaatiota. Kun nämä eri notaatiot yhdistetään, saadaan tietojärjestelmästä selkeä kokonaiskuva. Esimerkkinä voidaan ottaa järjestelmä, josta mallinnetaan viisi eri kokonaisuutta: toiminnot, tietosisältö, rakenne, aikariippuvuus ja käyttäjäinteraktiot.

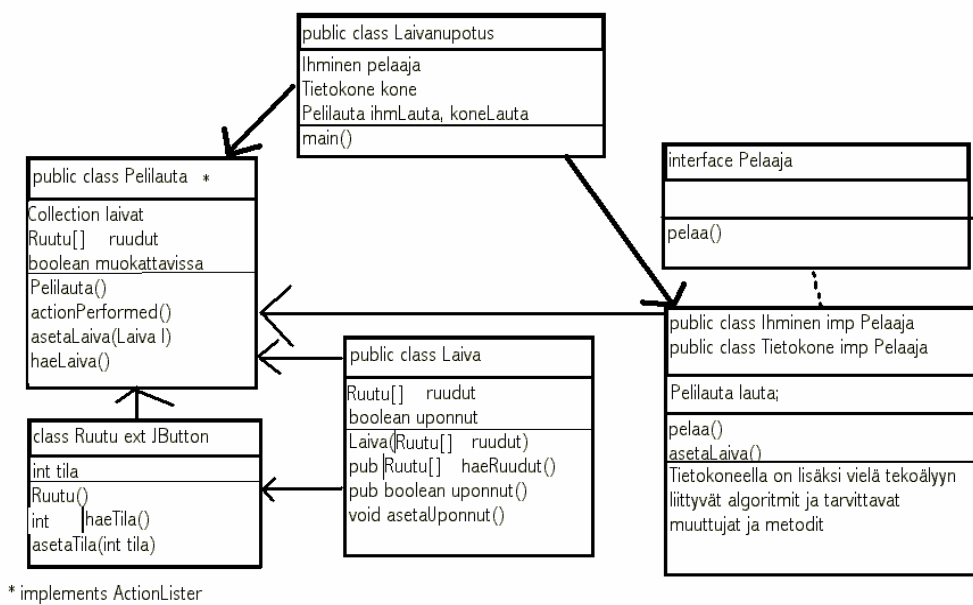
Toimintomallinnuksen tarkoitus on selvittää toteutettavan järjestelmän sisältämät toiminnot, sekä niiden sisältö ja niiden väliset yhteydet. Toimintomallinnusta kutsutaan usein myös prosessimallinnukseksi. Tietosisällön kuvaamisen tarkoituksena on tutkia järjestelmän käsittelemät tiedot, niiden sisältö, niiden yhteydet toisiinsa ja tietojen loogiset tallennusrakenteet. Rakenteen mallinnus on toisaalta turhaa, koska toiminto- ja tietomallit kuvaavat sen jo hyvin pitkälle, mutta se voidaan silti kuvata erikseen tavalla, joka on toteutusläheinen. Järjestelmän aikariippuvainen käyttäytyminen tulee kuvata, mikäli järjestelmässä olevien tehtävien ajoitus ja suoritusjärjestys on olennaista järjestelmän toiminnan kannalta. Käyttäjien ja järjestelmän välisten interaktioiden mallintaminen on käyttöliittymän suunnittelun kannalta olennaista.

Kaikkien eri kuvaustapojen yhteinen ajatus on kuitenkin antaa keinot yksinkertaistaa järjestelmän kuvausta siten, että voidaan rajata pois tarkastelun kannalta epäolennaiset yksityiskohdat. Esimerkiksi rakenteen mallintamiseen tarkoitettu menetelmä sisältää käsitteistön sekä esitystavat vain ja ainoastaan rakenteen mallintamiseen, mutta ei

esimerkiksi prosessien kuvaamiseen. Seuraavaksi tarkastellaan lähemmin muutamia eri malleja.

3.1.1 Oliomallit

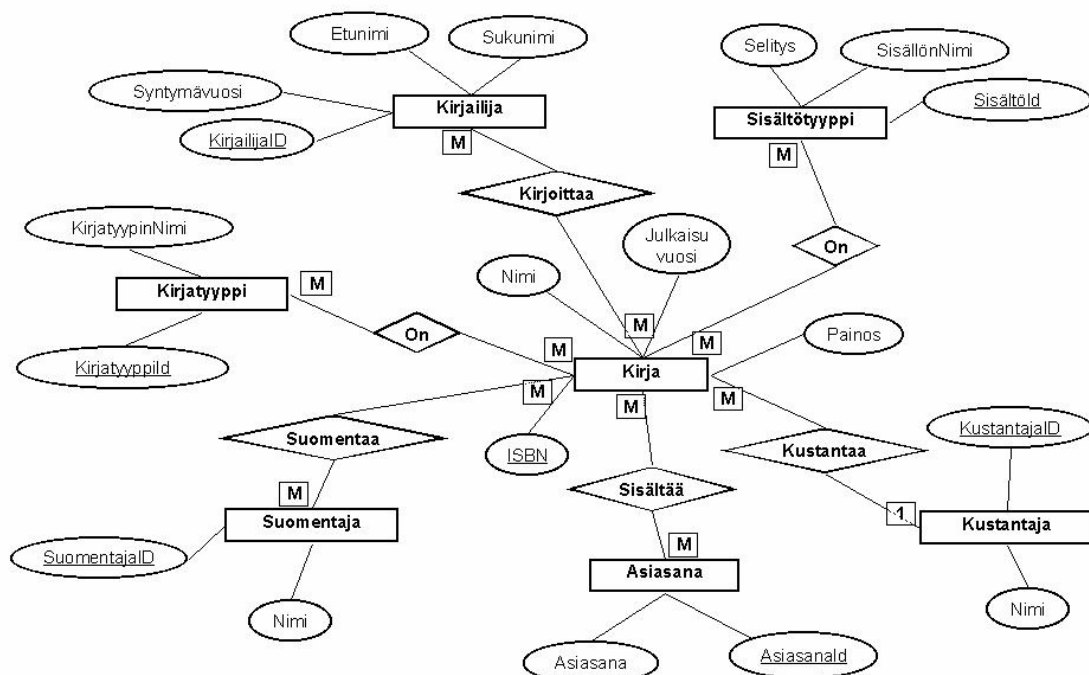
Oliomallit tarkastelevat järjestelmää rakennenäkökulmasta ja niillä voidaan kuvata koko järjestelmän rakenne. Malli tarkastelee järjestelmää oliojoukkona, jonka oliot ovat vuorovaikutuksessa keskenään. Jokainen olio sisältää omat käsittelyn kannalta olennaiset tiedot ja käyttäytymisen. Oliot sisältävät siis tietosisältöä ja operaatioita, joita se voi suorittaa. Nämä on mahdollista suojata siten, että toiset oliot eivät voi suoraan viitata niihin. Mikäli olio suojataan, on olion määriteltävä jonkinlainen palvelurajapinta, jonka kautta muut oliot voivat pyytää sitä suorittamaan sen omia operaatioita. Tämä on verrattavissa esimerkiksi kauppaan ja sen kalatiskiinkin. Ihminen ei voi mennä kauppaan ja ottaa tarjouksessa olevasta lohifileestä itselleen palaa omin päin, vaan hän menee tiskille eli palvelurajapinnalle ja pyytää myyjää leikkaamaan itselleen sopivan kokoisen kalapalan. Kalatiskin myyjä leikkaa kalasta palan ja antaa sen asiakkaalle. Järjestelmän rakenteen kuvaamisen kannalta olioiden perintäominaisuus on hyvin keskeinen. Kuvassa 3.1. on esitetty luokkakaavio, jolla oliomalleja usein esitetään. Kuvassa on laivanupotuspeliä kuvaava luokkakaavio, koska diplomityön osalta luokkakaavioita ei suunniteltu, sillä työn toteutuksessa ei käytetty olioita.



Kuva 3.1. Luokkakaavio /14/.

3.1.2 Tietomallit

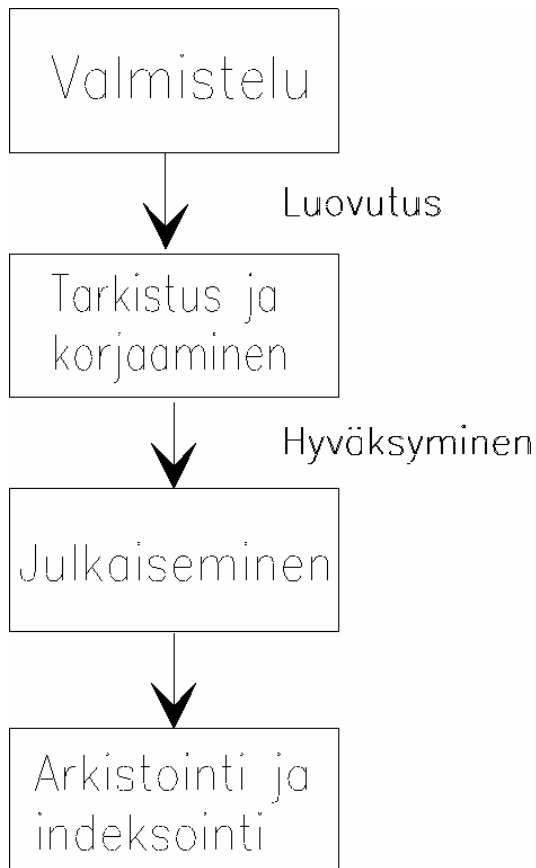
Tietomalleja käytetään myös rakenteen ja tiedon sisällön kuvaamiseen. Eli näkökulma on järjestelmän rakenteissa vain tiedon osalta, koska tietomallit eivät yleisesti ota kantaa siihen, millaisia rakenteita tarvitaan tiedon käsittelyyn. Tässä suhteessa oliomalli on siis laajempi kuin tietomalli. Tietomallissa tiedot pyritään jaottelemaan käsittelyn kannalta järkeviin käsitteisiin, kuten esimerkiksi asiakas ja tuote, sekä kuvaamaan näiden käsitteiden väliset suhteet. Myös ne tiedot, josta nämä käsitteet koostuvat, kuvataan tietomallissa. Varmasti yleisimmin käytetty kuvaustapa tietomallien esittämiseen on ER-kaavio, joka on myös kuvassa 3.2.



Kuva 3.2. ER-kaavio /15/.

3.1.3 Prosessimallit

Kuten nimestä jo voi päätellä, prosessimallia käytetään kuvaamaan järjestelmää prosessinäkökulmasta. Se esittää tehtäväkokonaisuuksien suorittamiseen tarvittavat toimenpiteet järjestyksineen yleisellä tasolla. Malleissa on mahdollista esittää toiminnan edetessä syntyvät haarautumis- tai toistotilanteet, mutta ei kuitenkaan yksittäisen toimenpiteen yksityiskohtia. Toisin sanoen prosessimalli kertoo karkeasti miten järjestelmä ja sen eri prosessit toimivat ja missä järjestyksessä ne suoritetaan. Kuvassa 3.3. on yksinkertainen prosessimalli joka kuvaa akateemisten julkaisujen prosessia.



Kuva 3.3. Prosessimalli /16/.

3.1.4 Tilamallit

Tilamallien tarkoitus on kuvata tietojärjestelmää sen prosessi- ja vuorovaikutusnäkökulmasta. Mallinnuksen peruseriaate on se, että järjestelmällä tai sen osalla on aina jokin tila, joka voi muuttua jonkin prosessin seurauksena. Myös ympäristön tilan muutokset voivat aiheuttaa jonkin osa-alueen tilan muutoksen. Järjestelmän toiminta koostuu siis eri osa-alueiden tilojen muutoksista. Järjestelmä nähdään siis suurena tilakoneena ja koska tilakoneajattelutapa on yksi tietojenkäsittelyn keskeisimpiä tarkastelutapoja, ei ole ihme että tilamallit ovat niinkin käyttökelpoisia ja suosittuja. Yleisin tilamalli on tilasiirtymäkaavio, jossa nähdään siirtymät eri tilojen välillä, ja syyt mitkä aiheuttavat siirtymän tilasta toiseen.

3.1.5 Tietovirtakaaviot

Tietovirtakaaviot ovat järjestelmän prosessien yksityiskohtaisempaa tarkastelua varten luotu tarkastelutapa. Niistä nähdään tietojärjestelmään saapuvat syötteet ympäristöstä, niiden käsittelyyn tarvittavat prosessit ja tiedot, prosessien ja tietojen väliset suhteet eli tietovirrat sekä järjestelmän ympäristölle palauttavat tuotokset. Tietovirtakaavio kuvaa siis järjestelmän syötteiden ja tulosteiden ketjuna. Tietovirtakaaviot ovat hyvinkin hyödyllisiä, jotta voidaan esimerkiksi testausvaiheessa tarkastella, miten järjestelmän tulisi reagoida tiettyihin syötteisiin ja mitä sen pitäisi antaa tuotoksena siitä.

3.2 Menetelmät

Tietojärjestelmäkehityksessä menetelmillä tarkoitetaan ohjeita siitä, miten kuvaustapaa tulee käyttää, jotta saadaan jokin tietty lopputulos. Ohjeiden avulla voidaan kuvata, miten eri kuvaustapoja voidaan yhdistellä. Kun kerätään tietty kokoelma eri menetelmiä ja sääntöjä yhteen niin, että ne määrittelevät kenen toimesta, missä järjestyksessä ja miten menetelmiä käytetään, saadaan menetelmistö eli metodologia. Se saattaa sisältää laajaltikin ohjeistusta myös muista kehityksen osista, kuten esimerkiksi projektinhallinnasta.

On tärkeää ymmärtää se fakta, että systeemyömenetelmiin liittyvä terminologia ei ole vielä vakiintunut, ja sitä myöten tässä työssä esitetty jako notaatioihin, menetelmiin ja metodologioihin on vain yksi esitystapa. Systeemyömenetelmät kehittyvät joka päivä, mutta kenties jonkinlaista harmonisoitumista on näköpiirissä. Tosin teknologinen kehitys jatkuu edelleen kohtuullisen voimakkaana, joka ajaa myös systeemyömenetelmien kehitystä eteenpäin jatkuvasti.

3.3 Systemityömenetelmien edut ja haitat

Milloin systemityömenetelmiä tarvitaan ja milloin ei? Aina niiden käyttö ei ole tarpeellista, kuten esimerkiksi silloin kun toteutettava tietojärjestelmä ja koko projekti on pieni, tai mikäli kehityshanke ei ole yritykselle kriittinen. On hyvä kuitenkin muistaa, että ohjelmiston monimutkaisuus ei kasva lineaarisesti suhteessa ohjelmiston kokoon, vaan pikemminkin eksponentiaalisesti. Toiseksi, kun arvioidaan kehityshankkeen kokoa, on määriteltävä kommunikaation määrän tarve, joka on toinen kriteeri kehityshankkeen kokoa määritettäessä. Mitä suurempi kehityshanke ja tietojärjestelmä, sen tärkeämpää on käyttää jotain systemityömenetelmää.

Mitä etuja menetelmät tuovat tietojärjestelmän kehitykseen. Seuraava lista on suora lainaus Risto Pohjosen Tietojärjestelmien kehitys kirjasta /8/:

- Menetelmät takaavat kehitystyön tuloksia ja parantavat laatua
- Menetelmät standardoivat systemityötä ja sen dokumentaatiota
- Menetelmät helpottavat ja nopeuttavat systemityötä
- Menetelmät tukevat kommunikaatiota
- Menetelmät helpottavat uudelleenkäyttöä ja sovellusten ylläpitoa
- Menetelmät vähentävät riippuvuutta avainhenkilöistä
- Menetelmät opettavat noviiseja
- Menetelmät helpottavat testausta

Systemityömenetelmien haitoista Pohjonen listaa seuraavia:

- Menetelmät vähentävät suunnittelijan vapautta pakottamalla käyttämään aina samaan tapaa (joka ei välttämättä sovi kaikkiin tilanteisiin)
- Menetelmät lisäävät työtä ohjelmiston elinkaaren alkuvaiheissa, jolloin menetelmien käytön hyödyt eivät ole välittömästi nähtävissä
- Menetelmät ovat byrokraattisia ja aikaa kuluttavia
- Menetelmillä laadittujen kuvausten ylläpito on työlästä

- Menetelmien oppiminen on vaikeaa
- Menetelmät eivät ole tarpeeksi kehittyneitä
- Oikean menetelmän löytäminen oikeaan ongelmaan on vaikeaa

Edut ja haitat ovat osittain ristiriidassa keskenään, joka johtunee pääosin siitä, että yleiskäyttöisiä menetelmiä ei ole vielä saatu kehitetyksi. Jostain syystä, vaikka menetelmillä on suuria etuja tietojärjestelmäkehityksessä, niiden käyttöönotto on ollut hidasta.

Samalla kun ohjelmistot ja vaatimukset kasvavat, kasvaa tarve käyttää systeemityömenetelmiä yhtälailla, koska kuten aiemmin kävi ilmi, on menetelmien edut sitä suuremmat mitä monimutkaisempi ja suurempi kehityshanke on kyseessä. Menetelmien käytön menestys on loppujen lopuksi kiinni siitä, kuinka hyvin ne valitaan ja integroidaan yrityksen omaan ohjelmistoprosessiin. Jotta menetelmien käyttöönotto ja niihin panostaminen kannattaa, on niiden tuotettava hyötyä jossain muussa ohjelmistoprosessin osassa, kuten kommunikaatiossa ja ylläpidossa. Ylläpitoon kohdistuneet hyödyt ovatkin suurimpia, koska se on elinkaaren pisin vaihe ja mahdollisesti sitä myötä myös kallein.

4 SÄHKÖMARKKINAT

4.1 Perusteet

Nord Pool on pohjoismainen sähköpörssi, joka perustettiin vuonna 1993. Sen tehtävänä on hallinnoida pohjoismaista sähkökauppaa eli sen kautta vaihdetaan eri maissa tuotettua sähköä niiden välillä. Nord Poolin tavoitteena on luoda tehokas markkinapaikka, jossa on hyvä likviditeetti ja korkea turvallisuustaso niille, jotka käyvät pörssissä kauppaa. Nord Poolin tavoitteena on myös luoda ns. referenssihintaa, jotta esimerkiksi Lappeenrannan

Energia Oy:n kaltaiset toimijat voivat miettiä kannattaako sähköä tuottaa itse vai ostaa pörssistä. Näin ollen toimijat voivat käyttää tuotantoresurssejaan mahdollisimman järkevästi.

Jokaisen maan toimijat tuottavat siis sähköä jonkin verran ja kuluttavat jonkin verran. Mikäli maassa tuotetaan sähköä vähemmän kuin kulutetaan, on sähköä tuotava toisesta maasta. Suomessa tilanne on yleensä juuri näin, kun taas esimerkiksi Norjassa tilanne on usein päinvastainen.

Pörssissä käydään kauppaa futuureilla, joita on vuosi-, kvartaali-, kuukausi-, viikko- ja päivätasolla. Pörssi noteeraa myös sähkön pörssihinnan jokaiselle tunnille vuorokaudessa. Hinta on kaikkien maiden hintojen keskihinta ja sitä kutsutaan systeemihinnaksi. Jokainen alue, esimerkiksi Suomi, saa sen lisäksi oman aluehinnan, joka ilmoitetaan joko lisäyksenä tai vähennyksenä koko markkina-alueen keskihintaan. Koska tämän työn osalta suurin painopiste on ensimmäisenä mainituilla futuurituotteilla, on olennaista selittää niistä tarvittavat seikat.

4.2 Pörssituotteet

Futuurutuotteita on siis vuosille, kvartaaleille, kuukausille, viikoille ja päiville. Eniten vaihdetuimpia tuotteita ovat seuraavan vuoden vuosituote, yksi tai kaksi seuraavaa kvartaalituotetta sekä mahdollisesti seuraavan kuukauden tuote, vaihdellen hieman sen hetken markkinatilanteesta. Tuotteet käyttäytyvät pörssissä samalla tavalla kuin osakkeet normaalissa pörssissä. Pörssipäivän päätteeksi näille futuurituotteille noteerataan päivän päätöskurssi, jota tämänkin työn osalta käytetään.

Pörssissä noteerataan aina seuraavat viisi kokonaista vuotta sen hetkisestä vuodesta. Seuraavan vuoden tuote on kaupankäynnin kohteena aina tämän vuoden viimeiseen pörssipäivään asti. Vuodenvaihteessa seuraavan vuoden tuote poistuu pörssistä ja vuosituotteiden loppuun ilmestyy uusi vuosituote.

Kvartaalituotteita on vuoden sisällä neljä, kuten nimestä voi päätellä, josta ensimmäinen kvartaali käsittää siis tammi-, helmi- ja maaliskuun. Kvartaalituotteita on kaupan sen vuoden jäljellä olevat kokonaiset kvartaalit ja kahden seuraavan vuoden kvartaalituotteet. Eli helmikuussa 2007 kaupan on vuoden 2007 kvartaalit kaksi, kolme ja neljä, sekä vuosien 2008 ja 2009 kaikki kvartaalit. Vuodenvaihteessa tulee kaupankäyntiin uudet neljä kvartaalituotetta eli esimerkin mukaan 2010 vuoden tuotteet.

Kuukausituotteista kaupankäynnin kohteena on aina kuusi seuraavaa kuukautta, aivan kuten viikkotuotteitakin. Päivätuotteita on 11 seuraavaa päivää sen hetkisestä päivästä.

4.3 SPOT-hinnat

Spot-markkinoilla määräytyy seuraavan päivän sähkön hinta kysynnän ja tarjonnan mukaan. SPOT-hinta tarkoittaa samaa kuin kohdassa 4.1 esitelty systeemihinta. Kuvassa 4.1 on kuva pörssialueen hinnoista eri pörssialueilla. SPOT-hinta noteerataan vuoden jokaiselle tunnille erikseen. Noteeraus tapahtuu siten, että noin kello 13 aikaan Spot-markkinoilla toimijat lähettävät tarjouksensa Nord Pooliin, jossa he kertovat, paljonko ja mihin hintaan he ovat valmiita ostamaan tai myymään sähköä kullakin tunnilla. Tarjousten perusteella lasketaan aggregoidut kysyntä- ja tarjontakäyrät, joiden leikkauspisteestä saadaan sähkön SPOT-hinta. Koska Spot-markkinoilla hinta määräytyy vain seuraavalle päivälle, on tämänkin työn kannalta olennaisimpia johdannaismarkkinoiden pörssituotteet eli futuurit, jotka esiteltiin kappaleessa 4.2. Näiden tuotteiden avulla yritykset kykenevät suunnittelemaan kassavirtansa paljon pidemmälle.



Kuva 4.1. Pörssialueen aluehinnat ja systeemihinta /4/

4.4 Dynaamiset sähkösovimukset

Suurille asiakkaille Lappeenrannan Energia Oy tarjoaa dynaamista sähkösovimusmallia. Siinä hintaa ei määritellä kiinteästi määräaikaista jaksoa eteenpäin, vaan hinta muodostuu dynaamisesti ajan kuluessa. Ideana on, että asiakas kiinnittää omaa sähköhintansa pörssituotteiden avulla. Hintaa on mahdollista kiinnittää minimissään kuukausitasolla, joka on se taso, millä hinta määräytyy. Toisin sanoen jokaiselle kuukaudelle tulee kiinnitysten myötä hinta, ja kuukausihintojen sekä niille annettujen energiankäytön ennusteiden mukaisesti lasketaan painotettu keskihinta, joka on sen vuoden sähkön hinta, millä asiakasta laskutetaan. Riippuen asiakkaan koosta, eli energiankäytön määrästä riippuen, asiakas voi jakaa tietyn kuukauden energiakiinnityksen yhdestä neljään eri osaan. Eli asiakas voi kuvitteellisessa tilanteessa kiinnittää 50% vuoden 2007 neljännen

kvartaalin energioista kvartaalituotteella ja myöhemmin 50% vuoden 2007 lokakuun energiasta lokakuun tuotteella. Tämän tilanteen jälkeen asiakkaalla on vuoden 2007 neljännessä kvartaalista kiinnittämättä marras- ja joulukuun energioistaan puolet. Ne on kiinnitettävä nyt kuukausituotteilla, koska kvartaalituotteella kiinnitettäisiin myös lokakuusta osa, joka on mahdotonta koska se on aiemmin jo kiinnitetty kokonaan.

Näin ollen asiakas pääsee kiinnittämään hintaa eteenpäin silloin, kun markkinahinta on alhainen, ja voi varmistaa itselleen halvemman hinnan etukäteen. Tämä auttaa suuria yrityksiä, kun ne suunnittelevat kassavirtaansa. Luonnollisesti myös energiayhtiö pystyy suojautumaan itse paremmin tappiolta, koska energia kiinnitetään etukäteen pörssi tuotteilla, joilla se itse ostaa energiaa, mahdollisen oman tuotannon lisäksi.

5 LRE OY:N TIETOJÄRJESTELMÄN SUUNNITTELU

5.1 Tietojenkäsittelyn tila ennen

Ennen uuden tietojärjestelmän kehitystä yrityksen myyntihenkilöstö, joka hoiti dynaamisia sähkösopimuksia, käytti Excel-taulukon luotua hinnanmääritystaulukkoa. Tämä taulukko toimi koko toiminnan pohjana ja koska sitä oli monimutkaista ylläpitää, haluttiin tilanteeseen muutos. Suuren Excel-taulukon haittana on myös sen hitaus koon kasvaessa. Tavoitteena oli saada myyntihenkilöstölle sellainen työkalu, jolla kaikki toiminnot voidaan hoitaa ilman, että tarvitsee huolehtia monimutkaisen taulukon ylläpidosta. Uuden järjestelmän toisena pääasiallisena tarkoituksena oli nopeuttaa ja selkeyttää työntekoa. Tavoitteena oli luoda järjestelmä, jota voisi käyttää sellainenkin henkilö, jolla ei olisi suurempaa kokemusta dynaamisista sähkösopimuksista. Järjestelmän tarkemmat vaatimukset esitellään seuraavassa kappaleessa.

5.2 Uuden tietojärjestelmän vaatimukset

Vaatimukset uudelle järjestelmälle kerättiin pääasiassa edellisessä kappaleessa mainitusta Excel-taulukosta, jonka lisäksi myyntihenkilöstön ideoita käytiin läpi lisäominaisuuksia mietittäessä. Ensimmäisten vaatimusten keräämisen jälkeen vaatimuksia lisättiin sitä mukaa suunnitelmaan, kun niitä keksittiin. Uusia lisäominaisuuksia lisättiin käytännössä vielä toteutusvaiheen lopullakin. Ensimmäisinä vaatimuksina listassa on järjestelmän vaatimat tiedot, joita se tarvitsee suorittaakseen perustoimintoja:

- Johdannaismarkkinoiden edellisen päivän päätöskurssinoteeraukset vuosi-, kvartaali- ja kuukausituotteille.
- Tieto kaupankäynnin kohteena olevista pörssituotteista
- Edellisenä päivänä noteeratut SPOT-hinnat

Sähköpörssistä saatavien tietojen lisäksi tarvitaan asiakkaalta tieto energiankäytöstä kuukausitasolla, jotta muodostettava hinta on juuri kyseisen asiakkaan käyttöprofiilin mukainen ja näin ollen myös yksilöllinen.

Tietojärjestelmältä vaadittavat prosessit on listattu seuraavassa:

- Pörssitiedoista muodostetaan hintataulukko, jota järjestelmä käyttää
- Asiakkaalle annetusta energiankäytöstä muodostetaan asiakkaalle ennuste, sen energian käytöstä kuukausitasolla
- Asiakkaalle tulee voida tehdä kiinnityksiä pörssituotteiden mukaisesti
- Järjestelmän pitää osata kiinnittää kuukausienergiat osissa ja laskea kuukaudelle keskihinta
- Järjestelmän pitää osata laskea vuodelle painotettu keskihinta
- Järjestelmään pitää pystyä syöttämään toteutunut energiankäyttö

Haluttiin myös, että järjestelmän kautta pystyisi saamaan jonkinlaisia raportteja asiakkaan tilanteesta, ja että ne olisi automatisoitu siten, että järjestelmä luo niistä automaattisesti Microsoft PowerPoint esityksen. Järjestelmän tulisi pystyä luomaan seuraavat raportit:

- **Tilannekatsaus:** Raportti, joka sisältää asiakkaan sen hetkisen tilanteen sinne asti, minne asiakkaalle on syötetty energiaennustetta. Raporttiin voidaan lisätä myös energiayhtiön mahdollisia kiinnityssuosituksia, ja siitä tulee nähdä näiden suositusten jälkeinen kiinnitystilanne. Liite 1 on malliraportti, jossa näkyy mainitut seikat.
- **Markkinatilanne:** Raportti laskee yhteen kaikkien asiakkaiden kuukausittaiset kokonaisenergiat, kiinnitetyt energiamäärät, niin futuurien kuin aluehintatuotteiden osalta, sekä kiinnitettyjen energioiden painotetun keskihinnan. Samalla se laskee, paljonko on prosentuaalisesti kiinnitetty jokaisen kuukauden kokonaisenergiamäärästä. Liite 2 on kyseisen raportin malliversio.
- **Hinnankiinnitykset:** Raporttiin tulostuu avoimna olevan asiakkaan kaikki tehdyt hinnankiinnitykset.

Käyttöliittymälle ei asetettu mitään erityisiä vaatimuksia yrityksen puolelta, joten se jäi täysin toteuttajan päätettäväksi. Loppujen lopuksi päädyttiin yksinkertaiseen valikkorakenteeseen, joka muistuttaa mitä tahansa Microsoft Windows-ohjelmaa. Tietoturvaan ja tietoliikenneyhteyksiin ei tarvinnut kohdistaa erityisiä vaatimuksia, koska jo olemassa olleet järjestelmät niiden osalta riittivät erittäin mainiosti uudelle tietojärjestelmälle. Tämän projektin osalta vaatimukset kohdistuivat täysin tietojärjestelmän toiminnallisiin osioihin, kuten jo edellä on huomattu.

5.3 Tietojärjestelmäratkaisun esittely

Projektin päätavoitteena oli luoda yrityksen myyntihenkilöstön käyttöön tietojärjestelmä, joka on nopea, vaivaton käyttää sekä luotettava. Käsittelyn nopeuden takaamiseksi päädyttiin tietokantaratkaisussa tekstitiedostoihin, eli projektissa ei käytetä mitään olemassa olevaa tietokantajärjestelmää, vaan kaikki tieto tallennetaan tekstitiedostoihin.

Tietojärjestelmä osaa lukea näitä tekstitiedostoja oikealla tavalla löytääkseen tarvitsemansa tiedon niiden sisältämästä informaatiosta. Tietojärjestelmälle oli luonnollisinta käyttää samaa tiedostojärjestelmää, kuin aiemmin raportissa mainittu määräaikaisten tarjousten tietojärjestelmä käyttää, koska ajatus järjestelmien mahdollisesta yhdistämisestä yhdeksi laajemmaksi järjestelmäksi oli mielessä. Myös se fakta, että järjestelmän koon jäädessä pienehköksi, edistyneemmän tietokannan luonti olisi tuonut lisää työtä kehitykseen, ja aikataulu olisi venynyt. Mahdollisesti se olisi myös lisännyt ohjelmiston rakenteen monimutkaisuutta sekä hidastanut ohjelmiston toimintaa, tosin näitä seikkoja ei testattu millään tavoin, koska sekin olisi vienyt ylimääräistä aikaa.

Tarkoitus on siis rakentaa tietojärjestelmä, joka toimii nopeasti ja vaivattomasti myyntihenkilöstön päivittäisessä työnteossa. Jotta tiedonkäsittely sujuu nopeasti, päätettiin kaikki tieto tallentaa tekstitiedostoihin, jotka nimettiin päätteellä .pgf, joka on lyhenne sanoista PriceGen File. PriceGen on aiemmin luodun määräaikaisten tarjousten laskentaan erikoistuneen järjestelmän nimi.

Kaikki tieto, jota itse laskentasovellus tarvitsee, tullaan tallentamaan sovelluksen alikansioon *data*. Sovellus sijoitetaan yrityksen tiedostopalvelimelle, jotta siihen on kaikilla helppo pääsy sekä säästetään vaivaa kun ei tarvitse asentaa ohjelmaa paikallisesti jokaiselle työasemalle. Sovelluksen ohjelmoinnissa huomioidaan sen mahdollinen yhtäaikaisten käyttö monelta työasemalta. Käytännössä tämä tarkoittaa sitä, että ohjelma käynnistyy jokaiselle työasemalle verkon läpi paikallisesti ja käyttää verkossa olevaa dataa vain lukumuodossa. Ainoana poikkeuksena ovat asiakkaiden tarjouslaskelmat, jotka sijaitsevat asiakkaiden omassa tiedostossa ja kun käyttäjä avaa kyseisen laskelman, varataan asiakastiedosto hänen käyttöönsä. Toisen käyttäjän yrittäessä avata laskelmaa, asiasta ilmoitetaan hänelle viestillä. Tiedostojärjestelmä ja sovelluksen muu rakenne käydään läpi tarkemmin kappaleessa 6.

Ajantasaisten pörssitietojen haku tullaan toteuttamaan pienellä FTP-moduulilla sovelluksen sisällä. Haun aikana kerätään myös puuttuvat toteutuneet hinnat historiatietoihin.

6 TIETOJÄRJESTELMÄN TOTEUTUS

6.1 Valmistelu

Työn esivalmisteluihin kului huomattava määrä aikaa ja työtä. Ensimmäinen työ oli selvittää yrityksen myyntihenkilöstön kanssa yhteistyössä, mitkä ovat olennaisimpia ominaisuuksia kehitettävässä järjestelmässä. Kun tämä työ oli saatu päätökseen, alkoi suunnittelutyö.

Tämän tietojärjestelmäprojektin osalta suunnittelun apuna käytettiin pääosin kappaleessa 3 esiteltyä prosessimallia ja käyttöliittymän osalta puhtaasti visuaalista hahmottelua. Prosessimallit kuvasivat puhtaasti tietojärjestelmän eri toimintoja ja niiden vuorovaikutusta sekä suoritusjärjestystä. Kyseiset prosessimallit olivat melko yleiskattavia, toisin sanoen ei kovin yksityiskohtaisia. Näitä prosessimalleja ei dokumentoitu myöhempää tarkastelua varten, koska silloin sille ei nähty tarvetta. Suunnittelun aikana luotiin myös muutamia vuokaavioita, jotka osoittautuivat hyvin luontevaksi järjestelmän kuvaamistavaksi.

Virallisia ja standardin mukaisia vaatimusmäärittely- ja suunnitteludokumentteja ei laadittu, osittain ajan puutteen ja osin sen takia, että niille ei nähty suurempaa tarvetta. Näiden puuttuminen osoittautui myöhemmin virheeksi, koska järjestelmän valmistuttua huomattiin, kuinka tärkeitä ne yritykselle olisivat ja jatkokehityksen kannalta kyseiset dokumentit olisivat elintärkeitä. Kuten myöhemmin tullaan kertomaan, nämä dokumentit toteutetaan myöhemmin jatkokehittelyvaiheen yhteydessä.

6.2 Toteutus

Rajallisen ajan puitteissa työssä ei tehty prototyyppejä ollenkaan, vaan pyrittiin tuottamaan yhdellä kerralla mahdollisimman valmis tietojärjestelmäkokonaisuus. Tämän mahdollisti se fakta, että loppujen lopuksi järjestelmä on melko yksinkertainen, ainakin verrattaessa suurempiin tietojärjestelmiin. Olkoonkin, että toteuttajana on vain yksi

henkilö, katsottiin että on parempi pyrkiä kerralla lopputulokseen, joka sisältää kaikki vaatimukset. Siispä järjestelmää päästiin testaamaan kokonaisuudessaan nopeammin.

6.2.1 Ohjelmointikieli

Ohjelmointi päätettiin tehdä käyttäen Visual Basic 6.0 -kieltä, koska sitä varten yrityksellä oli jo ohjelmistot, ja sillä oli myös toteutettu yritykselle aiemmin toinen tietojärjestelmä, jolla voidaan laskea määräaikaiset sähkö sopimukset. Kielivalinta ei sinänsä tuottanut vaikeuksia, koska toteuttaja oli tutustunut jo melko hyvin kyseiseen ohjelmointikielen edellä mainitun tietojärjestelmän toteutuksessa. Korvaamattomana apuna työn etenemisessä olivat kuitenkin Visual Basic 6 trainer ja Tehokäyttäjän opas: Visual Basic kirjat /6/ /7/. Kirjat auttoivat vaikeimpien ratkaisujen toteutuksessa hyvin paljon ja tekivät työstä näin nopeampaa ja sujuvampaa.

6.2.2 Tiedostojärjestelmä

Tiedostot, joita järjestelmä käyttää nimitään päätteellä .pgf, kuten aiemmassa järjestelmässä. Itse hallintasovellus ja sen vaatimat tiedostot sijoitettiin yrityksen verkkoon tiedostopalvelimelle omaan alihakemistoonsa. Hallintasovelluksen käynnistystiedosto sijaitsee myös kyseisessä alihakemistossa. Hakemiston juureen luotiin alihakemistot Data, doc ja src, joista Data-hakemistoon järjestelmä tallentaa kaiken tarvittavan datan omiin paikkoihinsa. Doc-kansio sisältää kaikki mahdolliset järjestelmää varten luodut dokumentit. Src-kansio sisältää sovelluksen lähdekoodin. Data-kansion alla on omat alikansionsa asiakastiedostoille sekä raporteille. Näiden kansioiden nimet ovat CFiles (Customer Files eli asiakastiedostot), Tilannekatsaukset ja ”Dynaamisten kokonaistilanne raportit”. Järjestelmä tallentaa kaikki mahdolliset väliaikaistiedostonsa Data-hakemistoon. Raportteja varten Data-hakemistoon on luotu kolme mallitiedostoa, joita käytetään raporttien luonnin pohjana. Tilannekatsauksia varten on Microsoft PowerPoint-esitys ja kokonaistilanneraportteja varten on luotu Microsoft Excel-taulukko,

johon raportti luodaan. Asiakkaan hinnankiinnityksistä kertovaa raporttia varten on luotu myös Microsoft Excel-taulukko.

Data-kansiossa sijaitsee muutamia pysyviä tiedostoja, jotka ovat mainitsemisen arvoisia. Kiinnitykset listataan lock.pgf tiedostoon, jossa ne pidetään niin kauan kunnes järjestelmä poistaa ne. Palaamme tähän asiaan myöhemmissä luvuissa. Tietojärjestelmän vaatimia perusmuuttujia varten on set.pgf. Se sisältää NordPoolin FTP-palvelinta varten tarvittavia muuttujia, kuten IP-osoite, käyttäjätunnus, salasana, alikansiot ja tiedostonimet jotka täytyy ladata, sekä kiinnitystiedoston ja kuukausi spot-hintojen tiedostonimet. Edellä mainitut kuluva vuoden kuukausien spot-hinnat tallennetaan spot.pgf tiedostoon. Siitä mihin kyseisiä tiedostoja tarvitaan, palataan myöhemmin.

6.2.3 Tietokantaratkaisu

Kuten edellisistä kappaleista käy ilmi, ei tähän tietojärjestelmään luotu tietokantaa. Tietojärjestelmälle luotiin siis verkkopalvelimelle kansiot, joita se käyttää tallentamiseen ja tiedon hakuun. Tietysti jonkinlaisena tietokantana voidaan pitää tietojärjestelmän omaa tiedosto- ja hakemistorakennetta. Sitä ei vain osaa lukea muu kuin tämä kyseinen tietojärjestelmä, joten järjestelmästä ei voida hakea tietoja muihin järjestelmiin, muutoin kuin luomalla jonkinlainen oma hakutyökalu, joka osaa lukea tiedostoja oikein ja muuntaa ne hakevalle järjestelmälle sopivaksi. Koska tämänkaltaista skenaariota ei ollut näköpiirissä, ei siihen senkään takia haluttu sen enempää varautua.

6.2.4 Markkinatietojen haku

Ajantasaisten markkinahintojen hakua varten sovellukseen kopioitiin oma FTP-moduuli olemassa olleesta tietojärjestelmästä. Sen tarkoituksena on hakea tarvittavat markkina-tiedot NordPoolin FTP-palvelimelta. Alisovellus toimii täysin normaalien FTP-asiakassovelluksien tapaan. Ainoa poikkeus on se, että käyttäjä ei voi puuttua

hakuprosessiin millään tavoin. Kaikki toiminnot on ohjelmoitu sovellukseen sisälle, ainoastaan yhteyden muodostusta varten tarvittavat tiedot löytyvät kappaleessa 7.2.1 mainitusta asetustiedostosta. Näin niitä päästään sujuvasti muuttamaan, mikäli niihin tulee muutoksia Nord Poolin toimesta. Koska käyttäjällä ei ole mahdollisuutta kajota FTP-prosessiin, saatiin tietojärjestelmään selkeyttä ja toiminnasta tuli läpinäkyvämpää. Näin ollen se ei häiritse käyttäjää, eikä hänellä tarvitse olla tietoutta FTP-sovelluksien toiminnasta ja käyttötavoista.

Sovellus suorittaa markkinatietojen haun käynnistyessään ensimmäisenä. Koska tietojen haun todettiin olevan niin nopeaa, päädyttiin ratkaisuun, jossa sovellus hakee markkinatiedot aina kun se käynnistetään, eikä vain yhden kerran työpäivän aikana. Näin ollen järjestelmällä on myös varmasti ajantasaiset tiedot käytössään. Haun aikana käyttäjälle näytetään aloitusikkuna, jossa myös kerrotaan informaatiota siitä, miten haku etenee. Kuvassa 6.1 on kyseinen aloitusikkuna.



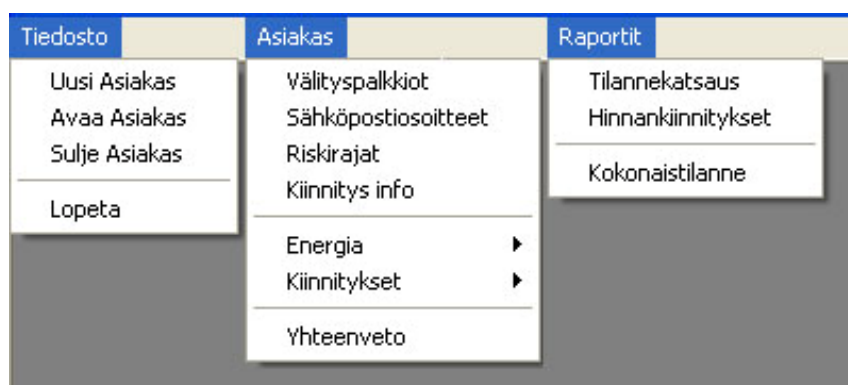
Kuva 6.1. Sovelluksen aloitusikkuna.

Kun tiedot on saatu onnistuneesti haettua yrityksen palvelimelle, niistä muodostetaan hinta- sekä tuotetaulukot järjestelmän muistiin. Kyseisiä taulukoita ei tallenneta siis

pysyvästi mihinkään. Tämä siksi, että kyseessä on jälleen sen verran nopea toimenpide, että se ei häiritse käyttäjää, ja näin vähennetään tallennettavan tiedon määrää. Haetuista tiedoista tallennetaan kappaleessa 6.2.1 mainittuun spot.pgf tiedostoon ainoastaan ne kuukausi spot-hinnat, joita järjestelmässä ei vielä ole.

6.2.5 Valikkorakenne

Tulevien kappaleiden asioiden selkeyttämiseksi on syytä käydä ensin sovelluksen valikkorakenne läpi. Alla olevassa kuvassa 6.2 näkyy valikon rakenne selkeästi, mutta käymme sen vielä sanallisesti läpi



Kuva 6.2. Sovelluksen valikkorakenne

Valikon perusotsikoihin kuuluu Tiedosto, Asiakas ja Raportit. Tiedosto-valikon alta löytyy sovelluksen perustoiminnot eli asiakastiedoston avaus, sulkeminen ja uuden asiakkaan luominen. Myös sovelluksen lopetus-valinta löytyy Tiedosto-valikon alta, kuten lähes kaikissa Windows-sovelluksissa. Kun sovellus avataan, on perusotsikoista valittavana muut, paitsi Asiakas-valikko. Sen jälkeen, kun on joko avattu olemassa oleva asiakas tai luotu uusi asiakas, tulee Asiakas-valikko aktiiviseksi.

Asiakas-valikon alta löytyy kaikki asiakkaan tietoihin liittyvät toimet, sekä niiden prosessien valinnat, jotka kohdistuvat asiakkaan tietojen muokkaamiseen. Kuten kuvasta 6.2 näkee, on Asiakas-valikko jaoteltu niin, että ensin on asiakkaan perustietoja koskevat

valinnat, jonka jälkeen on erotinviiva ja sen alla toiminnot, jotka koskevat asiakkaan energia- ja kiinnitystietoja. Alimpana on asiakkaan tietojen katseluun liittyvä Yhteenvedo- valinta.

Raportit-valikon alla on jaoteltu raportit niin, että erotinviivan yläpuolella on asiakasta koskevat raportit eli Tilannekatsaus- ja Hinnankiinnitykset-raportit. Alimpana on Kokonaistilanne-raportti, joka on aina aktiivinen, eli siis vaikka asiakasta ei olisi järjestelmään avattukaan vielä.

6.2.6 Perustietojen kerääminen

Kappaleessa 6.2.4 esitetyt markkinatiedot kuuluvat oleellisena osana järjestelmän perusominaisuuksiin, mutta lisäksi on tietysti muita perusominaisuuksia, jotka käydään nyt läpi samassa järjestyksessä, kuin prosessi joka käydään läpi kun uusi asiakas luodaan järjestelmään.

Kun järjestelmään luodaan uusi asiakas, järjestelmä kysyy käyttäjältä asiakkaasta tietoja. Kuva 6.3 on kuvasarja kysytyistä tiedoista, jossa kysymykset ovat numeroitu yhdestä viiteen kyselyjärjestyksen kuvaamiseksi. Ensimmäisenä asiakkaalle tulee tietenkin antaa nimi. Annetun nimen mukaan nimetään asiakkaan oma tiedosto CFiles- alihakemistoon. Asiakkaalle annetaan järjestyksessä oma numero. Numerointi alkaa numerosta 10 ja etenee aina kymmenen hyppäyksillä. Numerointi toteutettiin siksi, että järjestelmässä on hakuja, joissa on helpompi käyttää hakukriteerinä pelkkää asiakkaan numeroa, kuin koko nimeä. Nimeämisen jälkeen kysytään tietoa siitä, mistä sopimus asiakkaan kanssa alkaa. Tieto annetaan muodossa kk.vvvv eli kuukausi kahdella numerolla ja vuosi neljällä numerolla.

Näitä edellä mainittuja tietoja ei pääse tämän jälkeen enää muokkaaman. Seuraavat kysyttävät tiedot ovat sellaisia, joita pääsee valikkorakenteen kautta myöhemminkin muokkaamaan. Ne kuitenkin kysytään tässä vaiheessa, jotta voitaisiin minimoida se riski, että käyttäjä unohtaisi syöttää kyseiset tiedot asiakkaan luomisen jälkeen. Mikäli asiat

unohdettaisiin syöttää, ei järjestelmä välttämättä toimisi halutulla tavalla. Käyttäjältä kysytään sähköpostiosoitteet, johon ilmoitukset hinnankiinnityksistä lähetetään. Tämän jälkeen syötetään kiinnitysten infoteksti, joka on täysin vapaaehtoinen eikä vaikuta sinänsä järjestelmän toimintaan, vaikka se jätettäisiinkin tyhjäksi. Infoteksti näytetään hinnankiinnityksen teon yhteydessä.

The image shows five overlapping dialog boxes from a software application:

- Uusi asiakas** (1.): "Anna uuden asiakkaan nimi" (Enter the name of the new customer). Includes OK and Cancel buttons.
- Sähköpostiosoitteet** (3.): "Anna sähköpostiosoitteet johon kiinnitysilmoitukset lähetetään?" (Enter email addresses to which contract notices are sent). Includes OK and Cancel buttons.
- Sopimuksen alkamis PVM** (2.): "Anna sopimuksen aloitus kuukausi ja vuosi (mm.yyyy)" (Enter the start month and year of the contract). The input field contains "06.2007". Includes OK and Cancel buttons.
- Kiinnitykset** (4.): "Anna kiinnitysten info teksti" (Enter the contract info text). Includes OK and Cancel buttons.
- Välityspalkkiot** (5.): "Uusien palkkioiden alkamis PVM:" (New agency fees start date). It has input fields for SPOT, Kuukausi, Kvarttaali, and Vuosi. Includes Tallenna (Save) and Peruuta (Cancel) buttons.

Kuva 6.3. Kuvasarja perustiedot keräävistä kysymyksistä.

Viimeisenä kysytään asiakkaan välityspalkkiot, jotka lasketaan jo mainitulla yritykselle aiemmin toteutetulla tietojärjestelmällä. Koska tämä toimenpide oli jo toteutettu kyseiseen järjestelmään, ei sitä lähdetty toteuttamaan uudestaan tähän tietojärjestelmään, vaikka se ehkä käytön loogisuuden kannalta olisi ollut järkeväkin. Toiminnon siirtäminen tähän järjestelmään oli kyllä toteuttajan mielessä, mutta ajan loppuessa piti karsia tämänkaltaisen toissijainen toimenpide. Välityspalkkioista syötetään SPOT-, kuukausi-, kvarttaali- ja vuosituotteiden palkkiot. Kolme viimeistä välityspalkkiota käytetään aina sen mukaan, millä tuotteella suoritetaan hinnankiinnitys. SPOT-välityspalkkio on erikoistapaus, jota käytetään silloin kun jostain kuukaudesta on jäänyt

kiinnittämättä osa, ja siihen ei voida enää kiinnityksiä tehdä. Tällöin loppuosa kiinnitetään jälkikäteen sen kuukauden toteutuneella Spot-hinnalla. Välityspalkkioiden antaminen on tärkeää, koska jos kiinnitettäessä ei ole välityspalkkiota, se on silloin nolla. Tämän takia järjestelmään tehtiin lopussa vielä ominaisuus, joka varmistaa sen, että välityspalkkio tulee aina kun kiinnitys tehdään. Joten mikäli hintaa kiinnitettäessä ei ole annettua välityspalkkiota, järjestelmä kysyy sitä siinä kohtaa, ja muistuttaa että välityspalkkioita ei ole annettu.

Ainoa perustieto, mitä ei kysytä asiakkaan luomisen yhteydessä on riskirajat. Riskirajoja käytetään Tilannekatsaus-raporttia luodessa. Raporttiin tulostetaan annetut riskirajat, eli prosenttirajat, jota minimissään ja maksimissaan pitää liukuvasti seuraavalle vuodelle olla kiinnitetty. Rajat ovat siis pääasiallisesti informaationa asiakkaalle siitä, mitä heidän pitäisi tehdä lähitulevaisuudessa. Aiheeseen palataan lähemmin kappaleessa 6.2.10.

6.2.7 Energiatiedot

Jotta järjestelmä pystyy laskemaan hinnan yksilöllisesti jokaiselle asiakkaalle, on sillä oltava tiedossa asiakkaan energiankäyttöprofiili. Asiakkaan energiankäyttöprofiili tarvitaan ainoastaan kuukausitasolla, koska hinta lasketaan kuukausitasolla myös. Tämä johtuu siitä, että hinnankiinnityksessä käytetty lyhyimmän aikavälin hinta on kuukausihinta. Energiaprofiili saadaan joko asiakkaalta tarjouspyynnön yhteydessä tai mahdollisesti omasta tietokannasta, mikäli asiakas on Lappeenrannan Energia Oy:n jakeluverkon alueella. Kyseinen profiili toimii tästä eteenpäin ennusteena siitä, miten asiakas energiaansa käyttää.

Koska energiatiedot ovat tarjouspyynnöissä pääsääntöisesti tekstitiedostoissa, jokainen kuukausi omalla rivillään ja tietokannasta haettu vuoden energiajakso voidaan tuoda samanlaisessa muodossa, päädyttiin ratkaisuun jossa energiaprofiili kopioidaan leikepöydälle, josta se tämän jälkeen tuodaan järjestelmään. Ennen tuomista on valittava kuukausi ja vuosi, josta energiaprofiili alkaa. Tuotavan profiilin pituudelle ei ole asetettu

rajoitteita, jotta energiaprofiilien syöttö ei vaikeutuisi. Nyt järjestelmään voidaan tuoda vaikka kolmen kuukauden tai kolmen vuoden jakso kerralla. Useimmiten kuitenkin yhdellä kerralla tuotavan jakson pituus on yksi vuosi. Kuvassa 6.4 on kyseisen tapahtuman lomake.

Ennusteena käytetyn profiilin lisäksi järjestelmään voidaan tuoda jokaiselle kuukaudelle toteutuneet energiat. Tämä ominaisuus toteutettiin, koska niin myyjää kuin asiakasta kiinnostaa se, miten hyvin energiaennusteet ovat menneet kohdalleen. Lopullinen laskutuskin tietenkin hoidetaan toteutuneiden energioiden mukaan, eikä ennusteen mukaan. Toteutuneet energiat tuodaan samalla kopioi ja liitä-metodilla kuin energiaennustekin.



Kuva 6.4. Energiaennuste profiilin tuontiin luotu lomake

6.2.8 Hinnankiinnitykset

Hinnankiinnitykset on se syy, minkä takia tätä kyseistä järjestelmää oikeastaan tarvittiin. Kiinnitysten mukaan muodostuu sekä kuukausittaiset hinnat että koko vuoden hinta. Järjestelmän avulla voidaan siis syöttää kiinnitykset nopeasti, vaikka samantien kun ollaan asiakkaan kanssa puhelinyhteydessä. Järjestelmän tehtävänä on laskea kiinnitettyjen energioiden prosenttimäärät, päivittää ne laskelmaan ja sitä myötä päivittää hinta. Kuvassa 6.5 on hinnankiinnityksen lomake, jossa näkyy myös hinnankiinnitysten infoteksti, jota kysyttiin perustietojen keräämisen yhteydessä.

Kuva 6.5 Hinnankiinnityslomake

Kuten kuva selkeästi havainnollistaa, ensin valitaan alaspudotus-valikosta tuote jolla hintaa kiinnitetään. Sen jälkeen käyttäjä syöttää tuotteen reaaliaikaisen pörssihinnan. Käyttäjän tehtäväksi jää edellisen lisäksi kiinnityksen prosenttiosuuden syöttäminen lomakkeelle. Järjestelmä valitsee itse välityspalkkion valitun tuotteen mukaan. Kuten aiemmin on jo esitelty, kuukausi-, kvartaali- ja vuosituotteille on omat välityspalkkionsa. Jotta usean yhtä peräkkäisen kiinnityksen antaminen helpottuisi, toteutettiin lomakkeelle kaksi painiketta edellisten valintojen tallentamiseen. Toinen tallentaa lomakkeen tiedot ja poistuu lomakkeesta, kun toinen tallentaa tiedot ja tyhjentää tämän jälkeen lomakkeen uutta kiinnitystä varten. Tällainen ratkaisu toteutettiin, jotta käyttäjien ei tarvitse kiinnitysten välissä ladata lomaketta aina uudestaan. Estääkseen käyttäjää syöttämästä liian suurta prosenttiosuutta lomakkeelle, järjestelmä laskee valitun tuotteen perusteella suurimman mahdollisen prosenttimäärän, jolla tuotetta voidaan kiinnittää. Järjestelmään on silti mahdollisten virhelyöntien varalta ohjelmoitu tarkistus, joka tarkistaa voidaanko valittua tuotetta kiinnittää annetulla prosenttiosuudella.

Koska kaikki kiinnitykset halutaan varmistaa, luotiin järjestelmään automaattinen toiminto, joka luo käyttäjälle sähköpostiviestin, sen jälkeen kun hän painaa ”Lopeta ja sulje”-painiketta. Tietojärjestelmään ei luotu omaa sähköpostiohjelmaa, vaan päädyttiin käyttämään yrityksellä olemassa olevaa Microsoft Outlook-sovellusta. Päätös oli helppo,

koska näin säästettiin aikaa ja koska kehitystyökaluna toimiva Visual Basic on myös Microsoftin ohjelma, löytyy siitä suoraan referenssit Outlookille. Siten järjestelmään oli helppo toteuttaa ominaisuus, jossa järjestelmä aukaisee Outlookin, jos se ei ole jo auki. Sitten se luo uuden sähköpostiviestin, jonka otsikkoon tulostuu asiakkaan nimi ja päivämäärä. Vastaanottajiksi tuodaan automaattisesti ne osoitteet, jotka on syötetty perustietoja kysyttäessä. Viesti-kenttään tulee kaikki ne kiinnitykset, jotka syötettiin kyseiselle asiakkaalle edellisessä vaiheessa. Kiinnityksistä ilmoitetaan päivämäärä, pörssituote, hinta sekä prosenttiosuus. Järjestelmä ei kuitenkaan lähetä sähköpostiviestiä vielä, vaan jättää sen käyttäjän ruudulle, jotta hän voi lisätä viestiin mahdollisia omia asioitaan asiakkaalle. Tämän jälkeen järjestelmä päivittää asiakkaan kiinnitykset ja laskee uudet hinnat.

6.2.9 Tietojen katselu

Tietoja voi katsella asiakasvalikosta löytyvän Yhteenveto-kohdan kautta. Se on lomake, josta käyttäjä voi katsella kyseisen asiakkaan yhteenvetoa kuukausitasolla. Lomakkeella näkyy aina yksi kokonainen kalenterivuosi tammikuusta joulukuuhun. Tällainen kiinteä vuoden mittainen lomake toteutettiin sen yksinkertaisuuden vuoksi, ja jotta järjestelmän ei tarvitse dynaamisesti muokata lomakkeen tekstikenttiä ajon aikana. Näin vältytään myös mahdollisilta virhetilanteilta tässä kohdassa.

Lomakkeella näkyy siis annettu energiaennuste sekä toteutunut energia, mikäli se on annettu. Sähkö- ja Aluehinta- otsikon alla olevat aliotsikot tarkoittavat seuraavaa:

- Tot = Kiinnitetyn energiaosuuden hinta euroina
- Avoin = Avoimena olevan energiaosuuden eilisen päätöskurssin hinta euroina
- % = Kiinnitetty energiaosuus prosentteina

Välitys-otsikon alla on kyseiselle kuukaudelle muodostunut välityspalkkion hinta euroina ja Myyntihinta- otsikon alla on kuukauden sen hetkinen myyntihinta euroina. Spot- ja Vertailu-otsikoiden alle tulee ajan kuluessa lähinnä vertailevaa informaatiota hinnasta. Spot-otsikon alle järjestelmä hakee toteutuneet kuukausien SPOT-hinnat. Vertailu-otsikon alle on käyttäjän mahdollista antaa oma vertailuhinta. Näin ollen käyttäjä voi verrata lopullista myyntihintaa esimerkiksi toteutuneeseen markkinahintaan eli SPOT-hintaan. Näin nähdään kuinka hyvin kiinnitykset onnistuivat ja saadaan arvokasta tietoa tulevaisuutta varten.

	Energia		Sähkö		Aluehinta			Välitys	Myyntihinta	Spot	Vertailu	
	Ennuste	Tot	Tot	Avoin	%	Tot	Avoin					%
Tammikuu											1	
Helmikuu												
Maaliskuu												
Huhtikuu												
Toukokuu												
Kesäkuu	123	0	0,00 €	19,78 €	0	0,00 €	1,00 €	0	0,00 €	20,78 €	0	0
Heinäkuu	123	0	0,00 €	23,31 €	0	0,00 €	0,73 €	0	0,00 €	24,04 €	0	0
Elokuu	234	0	0,00 €	23,31 €	0	0,00 €	0,73 €	0	0,00 €	24,04 €	0	0
Syyskuu	234	0	0,00 €	23,31 €	0	0,00 €	0,73 €	0	0,00 €	24,04 €	0	0
Lokakuu	121	0	0,00 €	36,50 €	0	0,00 €	0,55 €	0	0,00 €	37,05 €	0	0
Marraskuu	231	0	0,00 €	36,50 €	0	0,00 €	0,55 €	0	0,00 €	37,05 €	0	0
Joulukuu	121	0	0,00 €	36,50 €	0	0,00 €	0,55 €	0	0,00 €	37,05 €	0	0
Yhteensä:	1187	0								28,89 €		0,00 €

Kuva 6.6. Asiakkaan yhteenveto-taulukko.

Kuukausi-informaation alapuolella on vielä kokoava rivi, jolla näkyy asiakkaan kokovuoden energiankäyttö ennusteena ja toteutuneena, sekä koko vuoden myyntihinta, joka lasketaan painotettuna keskihintana kuukausihinnoista. Myös SPOT- ja Vertailu-sarakkeesta lasketaan painotettu keskihinta koko vuodelle. Yhteenveto-taulukossa voi vaihtaa näytettävää vuotta vasemman yläkulman navigointinäppäimistä. Järjestelmä ei kuitenkaan anna siirtyä taaksepäin tilanteessa, jossa näkyvä vuosi on se, josta asiakkaan sopimus alkaa, kuten kuvassa 6.6. Myöskään järjestelmä ei anna siirtyä eteenpäin tilanteessa, jossa seuraavalla vuodelle ei ole energiaennustetta annettuna.

6.2.10 Raportointi

Työssä toteutettiin järjestelmän sisään kolme erilaista raporttia. Raporteista kaksi on asiakaskohtaisia ja yksi koskee koko yrityksen asiakaskantaa mikä järjestelemässä on. Raportoinnin tarkoituksena on antaa lisäinformaatiota yritykselle, sekä tuottaa valmis havainnollinen raportti, jota voidaan käyttää hyväksi asiakkaan kanssa käytävissä säännöllisissä neuvotteluissa.

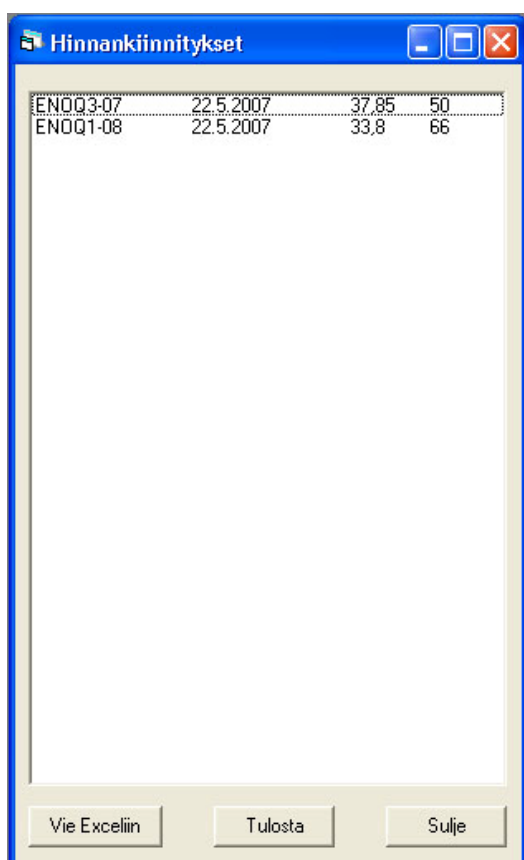
Lisätyt suositukset	
ENOQ3-07	33%
ENOQ1-08	50%

Kuva 6.7. Kiinnityssuosituksien syöttölomake Tilannekatsaus-raporttiin,

Tilannekatsaus-raportti on tarkoitettu käytettäväksi yhdessä asiakkaan kanssa. Raporttiin on mahdollista lisätä tämän hetken tilanteessa suositeltavia hinnankiinnityksiä. Kuvassa 6.7 on lomake, josta valitaan mahdolliset hinnankiinnityssuosituksia. Sitä ennen järjestelmä kysyy käyttäjältä, kuinka monta vuotta kuluva vuosi mukaan lukien raportille tulostetaan? Tämä valinta toteutettiin, koska aina ei ole mitään tarvetta tulostaa raportille

yli kahden vuoden päähän sijoittuvaa aikaa, etenkin jos sinne ei ole tehty eikä aiota tehdä kiinnityksiä. Kuten liitteestä 1 näkee, raporttiin tulostuu myös riskirajat. Riskirajat kertovat minimi- ja maksimimäärän, mitä asiakkaan energiasta pitää olla kiinnitettynä seuraavan tai sitä seuraavan liukuvan vuoden aikana. Riskirajat toimivat ohjenuorana asiakkaan ja yrityksen toiminnalle. Mikäli nähdään, että esimerkiksi ensimmäisen vuoden minimi kiinnitysmäärä on 50 prosenttia, mutta asiakas on kiinnittänyt vasta 40 prosenttia, voi energiayhtiö tehdä tilannekatsausraporttiin sellaiset suositukset, joilla päästään riskirajojen sisään. Näin asiakkaan on neuvotteluissa helppo todeta tilanne ja he saavat samalla ehdotuksen siitä, kuinka heidän tulisi jatkossa menetellä.

Toinen raporteista, joka koskettaa aina avoinna olevaa asiakasta, on raportti joka listaa kaikki kyseiselle asiakkaalle suoritettut hinnankiinnitykset. Kyseinen raportti toteutettiin, jotta käyttäjä voi tarvittaessa etsiä minkä kiinnityksen tahansa jälkikäteen. Tarvetta ei varsinaisesti kyseiselle toimenpiteelle usein ole, mutta raportti luotiin varmuuden vuoksi mahdollista työtä helpottamaan. Järjestelmä listaa omaan ikkunaansa kaikki hinnankiinnitykset. Mahdollista jatkokäsittelyä varten toteutettiin järjestelmään mahdollisuus viedä kyseiset tiedot Exceliin tai tulostaa ne paperille. Näin ollen ne on mahdollista välittää myös asiakkaalle, mikäli heillä on tarvetta kyseisille tiedoille.



ID	Päivä	Hinta	Määrä
ENOQ3-07	22.5.2007	37,85	50
ENOQ1-08	22.5.2007	33,8	66

Kuva 6.8. Hinnankiinnitys-raportti.

Kolmantena raporttina järjestelmään luotiin Kokonaistilanne-raportti, jonka tarkoitus on antaa käyttäjälle kokonaisvaltainen näkemys koko asiakaskannan tilanteesta. Raportti kertoo kuukausitasolla kokonaisenergiamäärän, kiinnitetyn energian määrän ja prosenttiosuuden. Kiinnitetty energiamäärä ja prosenttiosuus kerrotaan myös aluehintatuotteista erikseen, kuten jokaisen asiakkaan kohdalla tilannekatsaus-raportissa. Lopuksi kuukausitasolla lasketaan sen kuukauden kiinnitettyjen energioiden keskihinta euroina. Tämän kokonaistilannekatsauksen avulla energiayhtiö pystyy suorittamaan omat tarpeelliset ostot ja suojaukset sähköpörssissä. Kyseinen raportti luotiin juuri energiayhtiön omaa osto- ja suojatarpeen analysointia varten. Liite 2 on malli kyseisestä kokonaistilanne-raportista. Järjestelmä luo raportin Excel-tiedostoon, jotta mahdollinen jatkokäsittely olisi helpompaa. Kuitenkaan sen kaltaisille toimenpiteille ei nähty tarvetta, mutta näin päädyttiin tekemään tulevaisuuden varalta.

6.3 Testaus

Järjestelmää päästiin testaamaan vuoden 2006 alkusyksystä. Varsinaista virallista testaussuunnitelmadokumenttia ei luotu, koska pääasiallisesti testaus hoitui siten, että järjestelmä otettiin käyttöön ja sitä mukaan kun ongelmia havaittiin, niitä korjattiin ja käyttöä jatkettiin. Toteutuksen aikana suoritettiin jo moduulien pääasiallinen testaus toimivuuden osalta. Moduulit testattiin järjestelmällisesti siten, että edettiin tietojärjestelmän loogista etenemispolkua pitkin, ja katsottiin siirtykö tieto oikein moduulista toiseen. Moduulien toteutusvaiheen testauksen johdosta myöskään käytön aikana ei esiintynyt suurempia ongelmia, jotka olisivat johtaneet pitempiin viivästyksiin. Testaus noudatteli myöskin omalta osaltaan kappaleessa 2.2.6 mainittua palautumistestausta, jolla nähtiin kuinka järjestelmä kykenee hallitsemaan esimerkiksi käyttäjän virheelliset syötteet.

7 JOHTOPÄÄTÖKSET JA YHTEENVETO

Lopputuloksena saatiin sitä, mitä työalussa esiteltiin työn lähtökohdiksi. Yritys sai käyttöönsä tietojärjestelmän, jonka avulla myyntihenkilöstön työtaakkaa saatiin kevennettyä dynaamisten asiakkaiden kohdalla. Nyt yrityksen myyntihenkilöstö pystyy vastaamaan asiakkaiden pyyntöihin nopeasti, joka parantaa asiakaspalvelua ja tuo yritykselle näin kilpailuedun muihin energiayhtiöihin nähden. Uuden tietojärjestelmän avulla myyntihenkilöstö hallitsee kaikki dynaamisen sähkösopimuksen omaavat asiakkaat yhden helppokäyttöisen järjestelmän avulla, joka oli tämän työn pääasiallinen tarkoitus.

Se, miten paljon uusi tietojärjestelmä nopeuttaa myyntihenkilöstön työtä, on mahdotonta sanoa, koska mitään mittauksia ei ollut suoritettu ennen uutta tietojärjestelmää. Näin ollen myöskään uuden järjestelmän mukaisia suoritusajoja ei lähdetty testaamaan. Käyttäjäkokemuksiin pohjautuen voidaan todeta järjestelmän tuoneen yksinkertaisuutta ja selkeyttä toimintaan, sekä parempaa hallittavuutta.

Suunnittelun tärkeyttä pienissäkään projekteissa ei voi liikaa korostaa ja tämä seikka tuli hyvin esille myös tämän projektin aikana. Olkoonkin, että vähäisten tai jopa minimaalisten vaatimusten listaus ja niiden toteutuksen suunnittelu tuntuu turhalta, on se toteutusvaihetta ajatellen elintärkeää. Mikäli mitään suunnitelmia ei tehdä, ei myöskään toteutusvaiheessa saada aikaan juuri mitään järkevää. Jokainen lappu ja paperi, joka suunnitteluvaiheessa oli kirjoitettu, tuli otetuksi esiin toteutusvaiheessa, mikä kuvastaa hyvin suunnittelun tärkeyttä.

Myöskään testaamisen merkitystä ei pidä väheksyä. Tämän projektin kohdalla toteutusvaiheessa suoritettu testaus toi esille monia ongelmia ja mahdollisia ongelmia, joihin suunnitteluvaiheessa ei tullut kiinnitetyksi huomiota. Käytön aikaisessa testauksessa ilmenneet pienet korjaustarpeet eivät olleet järjestelmän toimintaa rajoittavia, mutta häiritseviä. Nämä ongelmat olisivat ilmenneet kunnollisessa testauksessa, mutta aikataulullisista syistä kyseinen riski hyväksyttiin ja järjestelmä otettiin käyttöön. Testaus suoritettiin siis käytön aikana.

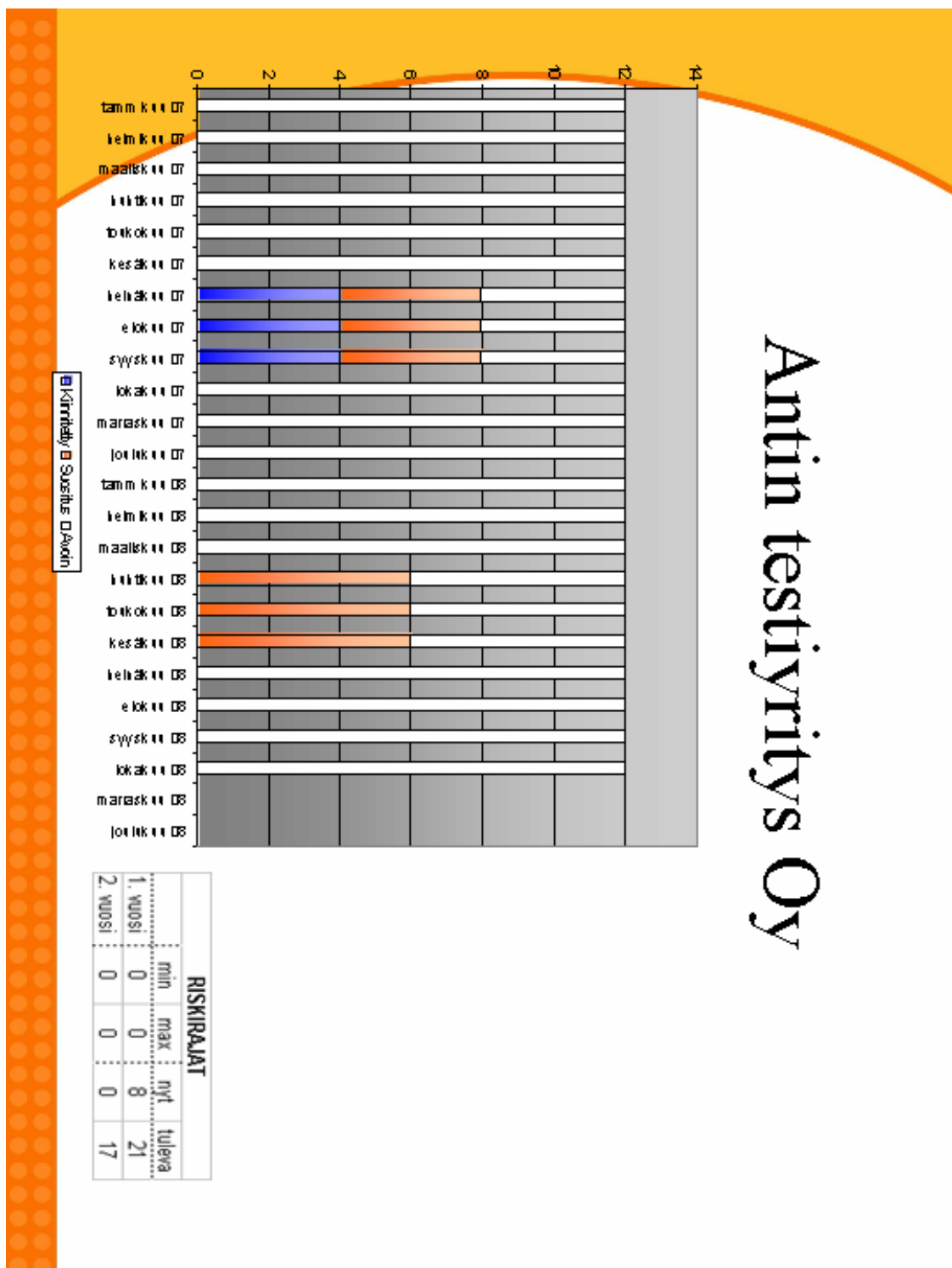
Loppujen lopuksi Lappeenrannan Energia Oy sai sellaisen tietojärjestelmän, jolle oli tarvetta ja sen myyntihenkilöstö pystyy tehostamaan työtään, parantaen näin yrityksen imagoa. Kuten jo aiemmin on todettu, huomattiin järjestelmän olevan yritykselle tärkeä ja sen jatkokehityksestä päätettiin hyvin pian tämän työn käytännön osuuden valmistuttua. Tietojärjestelmä toteutetaan ulkoisen ohjelmistoyrityksen toimesta web-pohjaiseksi. Siihen yhdistetään tässäkin työssä aiemmin mainittu määräaikaista asiakkaita varten toteutettu tietojärjestelmä. Web-pohjaisella käyttöliittymällä päästään myös eroon käyttöjärjestelmäriippuvuudesta. Yrityksessä mahdollisesti käyttöönotettavaan ekstranettiin saadaan asiakkaille web-pohjaisesta järjestelmästä huomattavasti helpommin informaatiota. Samassa yhteydessä luotiin standardinmukaiset dokumentaatiot tietojärjestelmästä, jotta yrityksen on tulevaisuudessa helpompi kehittää järjestelmää eteenpäin.

LÄHDELUETTELO

- /1/ Wikipedia, vapaa tietosanakirja. <http://fi.wikipedia.org/wiki/Tietojärjestelmä>.
21.2.2007
- /2/ Roukala Veikko, Soini Tapani. Tietojärjestelmien rakentaminen, toimintaa kehittävä systeemityö. Espoo: Weiling + Göös, 1989. ISBN 951-35-4631-4
- /3/ Kuivalahti Markku, Peltoniemi Jarmo. Tietojärjestelmän sisällön suunnittelu. Helsinki: Valtion painatuskeskus, 1988. ISBN 951-860-764-8
- /4/ Nord Poolin WWW-sivusto. <http://www.nordpool.com>. 19.12.2005
- /5/ Suomalainen ohjelmointisivusto. <http://www.ohjelmointiputka.net/>. 27.11.2005
- /6/ McKinney Bruce, Ketola Veli-Pekka. Visual Basic. Espoo: Suomen atk-kustannus, 1996. ISBN 951-762-435-2
- /7/ Halvorson Michael, Kuvaja Arto. Visual Basic 6 trainer. Helsinki: IT Press, 1999. ISBN 951-826-015-X
- /8/ Pohjonen Risto. Tietojärjestelmien kehittäminen. Jyväskylä: Docendo, 2002. ISBN 951-846-146-5
- /9/ Tampereen yliopisto. Tietojärjestelmien suunnittelumenetelmät kurssimateriaali. http://www.cs.uta.fi/tjsum/TJSUM_Luennot_viikolla37_2005.pdf. 1.3.2007
- /10/ Haikala Ilkka, Märijärvi Jukka. Ohjelmistotuotanto. Satku, Helsinki, 2002. ISBN 952-14-0486-8

- /11/ Oulun seudun ammattiopiston verkkokurssi materiaali.
http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elinkaarimallit_asia.htm. 30.3.2007
- /12/ Helsingin yliopiston luentomateriaali ohjelmistotuotannon prosessimalleista.
<http://www.cs.helsinki.fi/u/taina/ohtu/k-2001/luennot/prosessi/kaikki.html>. 30.3.2007
- /13/ Joensuun yliopiston opusteknologiakeskuksen oppimateriaalit.
<http://www.joensuu.fi/opusteknologiakeskus/palvelut/materiaalit/multimedia/esimerkit/spiraali.gif>. 30.3.2007
- /14/ Teknillisen korkeakoulun harjoitustyö essee.
<http://users.tkk.fi/~jvilppon/tekstit/essee5.htm> . 1.4.2007
- /15/ Jyväskylän yliopiston tietokannat-kurssin harjoitustyö. Tiina Pöyhönen.
<http://www.cc.jyu.fi/~tmpoyhon/tietokannat/index.html>. 1.4.2007
- /16/ Jyväskylän yliopisto. Tietojärjestelmätieteen pro gradu –tutkielma.
<http://www.cs.jyu.fi/gelpo/sgmlspl/node1.html>. 1.4.2007
- /17/ Wikipedia, vapaa tietosanakirja, englanninkielinen versio.
http://en.wikipedia.org/wiki/Information_system. 1.8.2007
- /18/ Alain Abran, James W. Moore. Guide to the software Engineering Body of Knowledge. Los Alamitos, CA: IEEE Computer Society Press, 2004. ISBN 0-7695-2330-7
- /19/ Wiegers, Karl E. Software Requirements 2: Practical techniques for gathering and managing requirements throughout the product development cycle. Redmond: Microsoft Press, 2003. ISBN 0-7356-1879-8

Liite 1. Tilannekatsaus-raportti



Antin testiyritys Oy

	energia	SAHKO			ALUE			Valitys	Myyntihinta	Spot	Vertailu
		tot	avoin	%	tot	avoin	%				
2007											
tammikuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
helmikuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
maaliskuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
huhtikuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
toukokuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
Kesäkuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
heinäkuu	12	45,0 €	23,3 €	33,3	0,0 €	0,0 €	0,0	2,15 €	32,7 €	0,0 €	0,0 €
elokuu	12	45,0 €	23,3 €	33,3	0,0 €	0,0 €	0,0	2,15 €	32,7 €	0,0 €	0,0 €
syyskuu	12	45,0 €	23,3 €	33,3	0,0 €	0,0 €	0,0	2,15 €	32,7 €	0,0 €	0,0 €
lokakuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
marraskuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
joulukuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
Yhteensä	144							8,2 €			
2008											
tammikuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
helmikuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
maaliskuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
huhtikuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
toukokuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
Kesäkuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
heinäkuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
elokuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
syyskuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
lokakuu	12	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
marraskuu	0	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
joulukuu	0	0,0 €	0,0 €	0,0	0,0 €	0,0 €	0,0	0,00 €	0,0 €	0,0 €	0,0 €
Yhteensä	120							0,0 €			

Liite 2. Markkinatilanne-raportti

	Tiedosto	Muokkaa	Näytä	Lisää	Muotoile	Työkalut	Tiedot	Ikuna	Ohje
	A	B	C	D	E	F	G		
1		Energia, TOT	Energia, sähkö	%	Energia, alue	%	Keskihinta		
2	06/tammikuu	2875	0	0	0	0	0,00		
3	06/helmikuu	2658	0	0	0	0	0,00		
4	06/maaliskuu	2932	0	0	0	0	0,00		
5	06/huhtikuu	2537	0	0	0	0	0,00		
6	06/toukokuu	2451	0	0	0	0	0,00		
7	06/kesäkuu	2233	0	0	0	0	0,00		
8	06/heinäkuu	2070	0	0	0	0	0,00		
9	06/elokuu	2537	0	0	0	0	0,00		
10	06/syyskuu	2459	0	0	0	0	0,00		
11	06/lokakuu	2837	0	0	0	0	0,00		
12	06/marraskuu	2858	0	0	0	0	0,00		
13	06/joulukuu	2850	0	0	0	0	0,00		
14	Yhteensä:	31'77	0	0	0	0	0,00		
15									
16	07/tammikuu	2952	50	1,69	0	0	59,56		
17	07/helmikuu	2734	70	2,56	0	0	59,56		
18	07/maaliskuu	2945	63,5	2,16	0	0	59,56		
19	07/huhtikuu	2545	55,5	2,18	0	0	59,56		
20	07/toukokuu	2524	50,5	2	0	0	59,56		
21	07/kesäkuu	2240	54	2,4'	0	0	59,56		
22	07/heinäkuu	2138	51,5	2,38	0	0	59,56		
23	07/elokuu	2633	57,5	2,2'	0	0	59,56		
24	07/syyskuu	2531	70	2,77	0	0	59,56		
25	07/lokakuu	2817	74,5	2,64	0	0	59,56		
26	07/marraskuu	2775	55,5	2	0	0	59,56		
27	07/joulukuu	2734	955,108	34,13	0	0	32,42		
28	Yhteensä:	31628	1607,608	5,083	0	0	57,30		
29									
30	08/tammikuu	121	0	0	0	0	0,00		
31	08/helmikuu	105	0	0	0	0	0,00		
32	08/maaliskuu	101	0	0	0	0	0,00		
33	08/huhtikuu	150	0	0	0	0	0,00		
34	08/toukokuu	143	0	0	0	0	0,00		
35	08/kesäkuu	123	0	0	0	0	0,00		
36	08/heinäkuu	111	0	0	0	0	0,00		
37	08/elokuu	149	0	0	0	0	0,00		
38	08/syyskuu	149	0	0	0	0	0,00		
39	08/lokakuu	0	0	0	0	0	0,00		
40	08/marraskuu	0	0	0	0	0	0,00		
41	08/joulukuu	0	0	0	0	0	0,00		