



TAMPEREEN
AMMATTIKORKEAKOULU

LIIKETALOUS

TUTKINTOTYÖRAPORTTI

JASPERREPORTS & CRYSTAL REPORTS
Raportointiohjelmistojen vertailu

Piia Mikkola ja Johanna Soinio

Tietojenkäsittelyn koulutusohjelma
kesäkuu 2005
Työn ohjaaja: Paula Hietala

TAMPERE 2005



Tekijä(t):	Mikkola Pii ja Soinio Johanna	
Koulutusohjelma(t):	Tietojenkäsittely	
Tutkintotyön nimi:	JasperReports & Crystal Reports – Raportointiohjelmistojen vertailu	
Title in English:	JasperReports & Crystal Reports – Comparison of Reporting tools	
Työn valmistumis- kuukausi ja -vuosi:	Kesäkuu 2005	
Työn ohjaaja:	Hietala Paula	Sivumäärä: 56

TIIVISTELMÄ

Yritykset tarvitsevat raportteja tietokannassa olevan tiedon järkevään esittämiseen ja tulostamiseen. Raportteja luodaan raportointiohjelmistoilla, joita on markkinoilla useita erilaisia. Tutkintotyön toimeksiantaja Datium Oy on Crystal Reports -ohjelmiston sertifioitu jälleenmyyjä, kouluttaja ja konsultoiija. Crystal Reportsia on myös käytetty yrityksessä menestyksekkäästi erilaisiin ohjelmisto- ja raportointiprojekteihin jo monia vuosia. Nyt haluttiin kartoittaa markkinoilla olevien open source -tuotteiden ominaisuuksia.

Tämän tutkintotyön tarkoituksena oli verrata open source -lisenssin alaista JasperReportsia Crystal Reportsiin. Työssä on keskitytty arvioimaan ja vertailemaan ohjelmistojen olennaisimpia toiminnallisia osia. Työssä on myös selvitetty JasperReportsin käyttöliittymän käytettävyyttä järjestämällä käytettävyydesti kahdelle ohjelman potentiaaliselle käyttäjälle.

Työssä on teoretietoa tietovarastoista, raportoinnista sekä käytettävyydestä. Crystal Reportsista ja JasperReportsista on selvitetty keskeiset toiminnalliset ominaisuudet, ja JasperReportsin käyttöliittymän JasperAssistantin käytettävyydestin vaiheet ja tulokset on selostettu. Järjestelmien toiminnallisia ominaisuuksia on verrattu keskenään ja arvioitu JasperAssistantin käytettävyyttä. Lopuksi on esitetty ehdotus ohjelmistojen käytöstä tulevaisuudessa ja arvioitu työn onnistumista.

Vertailun perusteella pystyttiin arvioimaan raportointiohjelmistojen käyttömahdollisuuksia tulevaisuudessa ja tekemään ehdotus ohjelmistojen käytöstä erilaisissa projekteissa. Raportointiohjelmiston valintaan vaikuttavat kuitenkin myös asiakkaan toiveet ja käytettävän raportointiohjelmiston kustannukset.

Sisällysluettelo

1 Johdanto.....	5
2 Työn taustaa.....	6
2.1 Toimeksiantaja	6
2.2 Työn tavoite ja sisältö lyhyesti	6
2.3 Crystal Reportsista lyhyesti	7
2.4 JasperReportsista lyhyesti	9
3 Tietovarastot ja raportointi	11
3.1 Tietokanta ja sen ominaisuudet	11
3.2 Operatiivinen kanta.....	11
3.3 Taustaa tietovarastoinnista	12
3.4 Tietovarastojen hyödyntäminen	13
3.5 Hyvän raportin ominaisuudet	14
3.6 Raportin suunnittelu	15
4 Käytettävyys	17
4.1 Käytettävyyden osatekijät.....	18
4.2 Käytettävyyden merkitys ohjelmiston valinnassa	19
4.3 Käytettävyyden arviointi	22
4.3.1 Jacob Nielsenin heuristiikat.....	22
4.3.2 Käytettävyystestit	26
5 Ohjelmistojen keskeinen toiminnallisuus	29
5.1 Crystal Reports	29
5.1.1 Datayhteydet.....	29
5.1.2 Integrointi omiin ohjelmistoihin.....	32
5.1.3 Kaavojen käyttö raporttien muodostamisessa	32
5.1.4 Kielten ja merkistöjen tuki	32
5.1.5 Raporttityypit.....	33
5.1.6 Interaktiiviset ominaisuudet	34
5.1.7 Raporttien muotoilu.....	34
5.1.8 Kuvien ja kaavioiden käyttö raportissa	35
5.1.9 Raporttien jakelu	36
5.2 JasperReports.....	37
5.2.1 Datayhteydet.....	38
5.2.2 Integrointi omiin ohjelmistoihin.....	38
5.2.3 Kaavojen käyttö raporttien muodostamisessa	39
5.2.4 Eri kielten ja merkistöjen tuki	39
5.2.5 Raporttityypit.....	39
5.2.6 Interaktiiviset ominaisuudet	40
5.2.7 Raporttien muotoilu.....	40
5.2.8 Kuvien ja kaavioiden käyttö raportissa	40
5.2.9 Raporttien jakelu	41
6 Käytettävyystesti: JasperReports – JasperAssistant	42
7 Ohjelmistojen vertailu	46
7.1 Toiminnallisuus	46
7.2 Käytettävyys	48

8 Johtopäätökset	50
8.1 Ehdotus ohjelmistojen käytöstä tulevaisuudessa.....	50
8.2 Yhteenveto työn onnistumisesta.....	51
Lähteet	52
Liitteet.....	54
Liite 1: JasperAssistant - käytettävyydesti.....	54
Liite 2: Vertailutaulukko	56

1 Johdanto

Raportointiohjelmistojen avulla voidaan analysoida tietokannoissa ja suurissa tietovarastoissa säilytettäviä tietoja ja luoda niistä raporteja. Toimeksiantajamme Datium Oy käyttää asiakasprojekteissaan tällä hetkellä raportointiin muun muassa Crystal Reports -ohjelmistoa, joka on laaja graafinen kysely-, raportointi- ja analysointiohjelmisto.

Crystal Reportsin lisensointia säätelee tarkoin sen omistaja Business Objects. Lisenssit ovat joko käyttäjäkohtaisia tai samanaikaisen käyttäjän lisensointia. Ohjelmiston hinta kasvaa siten käyttäjämäärän mukaan.

Open source -ohjelmistot ovat avoimen lähdekoodin ohjelmistoja, jotka ovat kaikkien saatavilla ilmaiseksi. Avoimen lähdekoodin lisenssi antaa oikeuden kopioida, muokata ja jakaa ohjelmiston alkuperäistä lähdekoodia. Mahdollisia open source -vaihtoehtoja raportointiohjelmistoksi on markkinoilla useita. Näistä vaihtoehdoista valittiin lähempään tarkasteluun toimeksiantajan alustavien selvitysten perusteella JasperReports, joka on täysin Java-pohjainen open source -raportointiväline.

JasperReports on yhden ihmisen toteuttama luokkakirjasto raporttien tekoon. Ohjelman käyttö laajentuu ja käyttäjäkunta kasvaa ympäri maailmaa, koska monet yksittäiset kehittäjät arvostavat open source -periaatteella leviäviä ohjelmia. Myös monissa yrityksissä on jo havaittu open source -kehitysvälineiden edut verrattuna suurten yhtiöiden kalliisiin lisenssiin alaisiin tuotteisiin. Open source -tuotteilla on myös yleensä laaja käyttäjäyhteisön tuki takanaan. Tuotteille on perustettu monia keskustelu- ja ohjesivustoja, joihin kehittäjät lähettävät kommentteja ja kysymyksiä sekä saavat vastavuoroisesti vastauksia ja mielipiteitä.

2 Työn taustaa

2.1 Toimeksiantaja

Datium Oy on yksityisomisteinen, Tampereella ja Helsingissä toimiva 15 henkilön järjestelmäintegraattoritalo, joka on perustettu vuonna 1999. Yhtiön vuoden 2004 tilikauden liikevaihto oli noin 1,4 miljoonaa euroa. (Datium Oy – Yritys)

Datium Oy tuottaa toimialasta riippumattomia teknisiä ja sovellusratkaisuja suurille ja keskiuurille yrityksille Suomessa. Yrityksen toiminnan painopistealueena on erikoisosaaminen Business Intelligence-, tietovarastointi- ja asiakaskohtaisissa ohjelmistokehitysratkaisuissa. (Datium Oy – Yritys) Business Intelligence tarkoittaa kokonaisuutta, jossa käsitellään eri puolilla organisaatiota sijaitsevaa dataa. Tavoitteena on päätöksenteon tukeminen niin operatiivisissa kuin strategisissa asioissa. (Sikanen 2004)

Datium Oy toimii myös alihankkijana ohjelmistotaloille ja järjestelmätoimittajille. Yritys kartoittaa lisäksi markkinoille tulevia uusia ohjelmistoratkaisuja, jotka puolestaan auttavat asiakkaiden oman liiketoiminnan kehittämistä sekä hyödyttävät vanhoja hankittuja tietojärjestelmiä. (Datium Oy – Yritys)

Datium Oy:n toimialueita ovat Business Intelligence -liiketoiminnan lisäksi Data Warehouse -konsultointi ja -toteutukset, raportointi- ja analysointiratkaisujen suunnittelu ja tuottaminen, OLAP- eli On-line Analytical Processing -toteutukset, eri tyyppiset raportointipalvelut sekä koulutuspalvelut. Yritys toimittaa myös asiakasorganisaatioiden tarvitsemat Internet-, extranet- ja intranet-ratkaisut sekä niihin liittyvät palvelut sekä strategisen, visuaalisen ja sisällöllisen suunnittelun. (Datium Oy – Yritys)

2.2 Työn tavoite ja sisältö lyhyesti

Tämän opinnäytetyön yhtenä tavoitteena on selvittää toimeksiantajalle Crystal Reportsin keskeiset toiminnalliset ominaisuudet. Ominaisuudet valitaan niiden tärkeyden ja käytön yleisyyden perusteella. Nämä samat ominaisuudet selvitetään myös JasperReportsin osalta. Lopuksi ohjelmistoja verrataan toiminnallisten ominaisuuksien suhteen niin, että Datiumille ja asiakasprojekteille keskeisimmät ominaisuudet otetaan erityisesti huomioon.

Koska käytettävyys on yksi ohjelmiston valintaan vaikuttava tekijä, työssä arvioidaan JasperReportsin käyttöliittymän käytettävyyttä. JasperReportsin JasperAssistant-käyttöliittymän käytöstä järjestetään kahdelle koehenkilölle käytettävyystesti, jonka tavoitteena on arvioida käytettävyyttä potentiaalisten käyttäjien näkökulmasta.

Vertailun avulla selvitetään, onko JasperReportsilla riittävästi ominaisuuksia Datiumin raportointitarpeisiin. Mikäli JasperReportsin todetaan täyttävän Datiumin vaatimukset, yrityksen pitäisi pystyä selvityksen perusteella tekemään päätös, voidaanko JasperReports ottaa käyttöön joissakin asiakasprojekteissa.

Työn alussa käydään läpi teoriaa tietokannoista, tietovarastoinnista ja käytettävyydestä, jotta lukijalle muodostuisi kuva työn aihealueesta. Teoriaosuuden jälkeen luvussa viisi selvitetään tutkittavien ohjelmistojen toiminnalliset ominaisuudet. Ominaisuudet selvitetään ensin toimeksiantajalla tällä hetkellä käytössä olevan Crystal Reportsin osalta, koska sen olemassa olevat ja toimeksiantajan tärkeiksi kokemat ominaisuudet toimivat koko vertailun pohjana. Tässä yhteydessä käsitteet selitetään ymmärrettävästi työn lukijalle.

Luvussa kuusi on selvitys JasperReportsilla tehdystä käytettävyystestistä ja sen tuloksista. Sen jälkeen ohjelmistoja vertaillaan ensin toiminnallisuuden ja sitten käytettävyyden suhteen. Käytettävyyden vertailu perustuu keskusteluihin testihenkilöiden kanssa. Lopuksi esitämme oman ehdotuksemme kahden tutkitun ohjelmiston käytöstä tulevaisuudessa sekä arvioimme työmme onnistumista.

2.3 Crystal Reportsista lyhyesti

Crystal Reports on raportointiohjelmisto, jonka avulla tietokannassa olevia tietoja voidaan hakea ja muokata raportointia ja analysointia varten. Gartner Groupin analyysien mukaan Crystal Reports on käytetyin web-pohjainen kysely-, raportointi- ja analysointiohjelmisto maailmassa (Datium Oy – Tuotteet).

Crystal Reportsin perustarkoitus on hakea tietoa tietokannasta ja tuottaa tiedoista raportteja. Tavoite on yksi vanhimmista tietokoneiden käyttötarkoituksista ja se on edelleen yksi yleisimmistä ja tärkeimmistä tietokoneen tehtävistä. Crystal Reports on suunniteltu tekemään tästä tehtävästä helpomman ja nopeamman tarjoamalla raporttien luomiseen graafisen ympäristön. Siinä on myös tuki useille tietokantayhteyksille, tietokantaohjelmistoille ja muille datalähteille sekä tiedostomuodoille raporttien jakamiseksi. (Crystal Reports 8.5 2001: 574.)

Crystal Reportsin edelläkävijä Quick Reports kehitettiin alkujaan 90-luvun alkupuolella Englannissa. Tuote oli tarkoitettu peruseräraportointiin tiedostopohjaisista tietokannoista. Seuraava versio sai uusien ominaisuuksien lisäksi uuden nimen: Crystal Reports 2.0. Tässä vaiheessa ohjelma sisällytettiin Microsoftin Visual Basic 3.0 Professional -versioon raportointityökaluksi. Käännetyt katseluohjelman sisältävät raportit, joita voidaan hyödyntää ilman Crystal Reports -ohjelmaa, tulivat jo versioon 3.0. (Crystal Reports 10 Jatkokurssi 2004: 98)

Vuoden 1995 alussa markkinoille tuli versio 4.0, jonka merkittävin uusi ominaisuus oli graafisten kaavioiden käyttö raporteissa. Samana vuonna julkaistiin myös Crystal Info -raporttipalvelinohjelmisto. Ohjelmiston avulla voidaan hallinnoida raporttien jakelua yrityksen tietoverkoissa. (Crystal Reports 10 Jatkokurssi 2004: 98)

Crystal Reports 5.0 Pro ilmestyi vuonna 1996, ja vuotta myöhemmin ilmestyi 6.0 Pro. Näissä versioissa ohjelman perusidea säilyi, mutta mukaan tuli uusia toimintoja, kuten raporttien uudet jakelumuodot ja raporttityylit. Lisäksi ohjelmoijille tarjottavia palveluita on jatkuvasti kehitetty lisäämällä ohjelmaan oliopohjaisia ominaisuuksia ja omiin sovelluksiin liitettävä komponentti. (Crystal Reports 10 Jatkokurssi 2004: 98)

Vuonna 1998 ilmestyi versio 7.0. Version uusia ominaisuuksia olivat muun muassa maantieteellisen kartan lisääminen ja oma ohjelma SQL-kyselyiden luomiseksi. Vanhoista toiminnoista ristiintaulukointi-, kaavio- ja parametriominaisuuksia oli kehitetty tässä versiossa huomattavasti. (Crystal Reports 10 Jatkokurssi 2004: 98)

Crystal Reports 8.0 ilmestyi keväällä 2000 ja versio 8.5 vuotta myöhemmin. Näihin versioihin oli lisätty funktioita ja exportausmuotoja ja raporttien jakelumahdollisuuksia oli parannettu. Myös yhteyksiä Microsoft Office -ohjelmiin oli parannettu. Tärkeimpiä uusia ominaisuuksia olivat hälytykset ja raporttien kääntäminen pdf-muotoon. (Crystal Reports 10 Jatkokurssi 2004: 98)

Versio 9.0 ilmestyi syksyllä 2002 ja sen uusia ominaisuuksia olivat mm. Crystal Repository eli objektien yhteinen tallennuspaikka, muokatut funktiot, rajoittamattomat SQL-komennot sekä muokattavat raporttipohjat. Vuoden 2003 ensimmäisellä neljänneksellä ilmestyneessä Crystal Reportsin versiossa 10.0 on lähinnä paranneltu 9.0:ssa ilmestyneitä uusia ominaisuuksia käyttäjäpalautteen pohjalta. Sen mukana toimitetaan myös rajattu versio Crystal Enterprise -ohjelmistosta, joka on tarkoitettu raporttien jakamiseen internetin tai intranetin välityksellä. (Crystal Reports 10 Jatkokurssi 2004: 98)

Crystal Reportsin uusi versio XI ilmestyi vuoden 2005 alussa, mutta keskityimme työssämme versioon 10.0, koska se on toimeksiantajalla tällä hetkellä käytössä.

Crystal Reports on upotettu lukuisiin sovelluskehitysvälineisiin ja ohjelmistoihin. Näistä tunnetuimpia esimerkkejä ovat Microsoft Visual Basic 5.0 ja 6.0, Microsoft Visual Studio .NET ja Borland dBase for Windows. Suomalaisia Crystal Reportsia käyttäviä ohjelmistoja ovat muun muassa Visma Softwaren Visio 3, Softlanen Nova-ohjelmat, Samstock Oy:n Samstock-salkunhallintaohjelmisto, QPR:n Balanced Scorecard ja useat TJ Groupin ohjelmat. Myös Datium käyttää Crystal Reportsia omissa sovelluksissaan. (Crystal Reports 10 Jatkokurssi 2004: 98)

2.4 JasperReportsista lyhyesti

JasperReports on tehokas open source -raportointiohjelmisto, joka on luokkakirjasto. Työkalun avulla voidaan luoda raportteja tulostettavaksi tai näytölle sekä talletettavaksi pdf-, html-, xls-, csv- tai xml-tiedostoiksi. JasperReports on kirjoitettu kokonaan Javalla, ja sitä voidaan käyttää monissa eri Java-sovelluksissa, mukaan lukien J2EE- ja Web-sovellukset. JasperReports-kirjaston päätarkoituksena on auttaa tekemään tulostusvalmiita dokumentteja helposti ja joustavasti hakemalla tietoa eri tietolähteistä. (JasperReports... 2005.)

Alun perin JasperReports-kirjastossa ei ollut mukana käyttöliittymätyökaluja. Käyttäjän piti manuaalisesti editoida XML-tiedostoa käyttäen yksinkertaista editoria ja esikatsella raportti jollakin esikatseluohjelmalla, jolla voidaan suunnitella raportteja. Nykyään kirjastolle on kehitetty monia käyttöliittymäratkaisuja, jotka auttavat käyttäjää suunnittelemaan ja tuottamaan raportteja helpommin. Kiitos tästä kuuluu nopeasti laajenevalle ”raporttientekoyhteisölle”.

JasperReportsin luoja Teodor Danciu on omaksunut Open Source strategian: "release early, release often". Danciu alkoi kehittää Java-pohjaista raportointivälinettä kesällä 2001. Sen jälkeen hän on julkaissut uusia versioita JasperReportsista nopeassa tahdissa. Teodor Danciu on kehitystyön alusta saakka ottanut huomioon mahdollisten käyttäjien ja muiden kehittäjien mielipiteet kehittäessään JasperReportsin uusia versioita. (Talk 2005.)

JasperReportsin viimeisin versio on julkaistu helmikuussa 2005. Siihen on jälleen lisätty uusia piirteitä ja vanhoja virheitä on korjattu. Esimerkiksi ennen uusinta versiota JasperReports oli kahden eri lisenssin alaisuudessa. Uudesta versiosta (0.6.5) alkaen tuote on kuulunut vain

yhden lisenssin alaisuuteen. Tämä lisenssi on GNU Lesser Public License eli LGPL. (Talk 2005.)

JasperReports-luokkakirjaston käyttämiseen tarvitaan siis jokin käyttöliittymä. Erilaisia mahdollisia käyttöliittymiä on markkinoilla useita. Tässä työssä keskitymme niistä kahteen, iReportiin ja Eclipsen päällä toimivaan JasperAssistiiniin, joita Datiumissa on jo jonkin verran kokeiltu käytännössä.

iReport on JasperReports-kirjaston käyttöön kehitetty käyttöliittymä. Sillä voidaan käsitellä muun muassa taulukoita, kuvia ja aliraportteja. Dataa voi hakea tietokannasta erilaisten datalähteiden avulla ja raportteja voi muuntaa moniin eri tiedostomuotoihin.

Myös JasperAssistant on käyttöliittymä JasperReportsin kirjastoa varten. JasperAssistant on rakennettu Eclipsen arkkitehtuurin päälle plug-inin avulla, ja sen päätavoite on auttaa raporttien luomisessa. JasperAssistant on ainoa kehitysväline, joka tällä hetkellä tukee kaikkia JasperReportsin piirteitä versioon 0.6.4 saakka (JasperAssistant... 2005).

3 Tietovarastot ja raportointi

3.1 Tietokanta ja sen ominaisuudet

Yritysten toiminnassaan tarvitsema tieto säilytetään tietokannoissa. Tietokantoja on monia erityyppisiä eri tarkoituksia varten. Yleisen määritelmän mukaan tietokanta on järjestetty ja loogisesti yhtenäinen kokoelma toisiinsa liitettyjä tietoja, joilla on jokin merkitys ja joita on helppo käyttää tehokkaasti. Lahtosen mukaan hyvä tietokanta on suunniteltu, rakennettu ja täytetty tiedolla jotain tiettyä tarkoitusta ja käyttäjäryhmää varten. Hän mainitsee, että hyvällä tietokannalla voi olla olemassa myös joitakin ennalta laadittuja ohjelmia, joita käyttäjät voivat käyttää. (Lahtonen: 2003.)

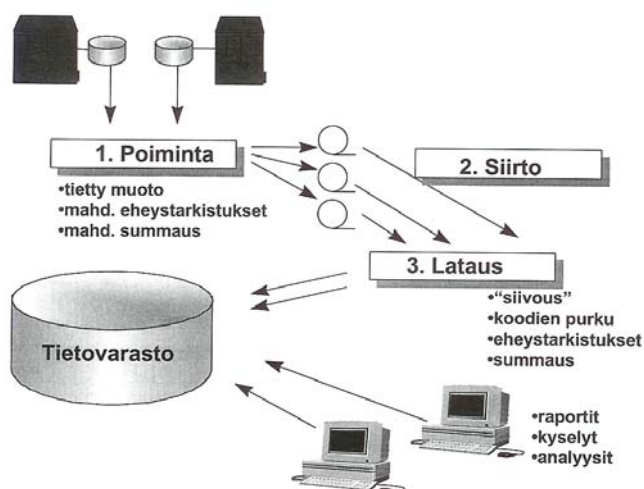
Jokainen tieto tallennetaan tietokannassa vain yhteen paikkaan, eli siinä ei esiinny turhaa toistuvuutta tai päällekkäisyyttä. Tietoja pitää pystyä hakemaan joustavasti erilaisin perustein, myös sellaisin, joita ei tietokantaa suunniteltaessa ole pystytty ennakoimaan. Tietokannan rakenteellisen muuttamisen olisi hyvä olla joustavaa. Myös tietoriippumattomuus on oleellinen tietokannan ominaisuus. Kannan käyttö ja sovellusohjelmat eivät saa olla riippuvaisia tietojen fyysisestä talletusrakenteesta. (Lahtonen: 2003)

3.2 Operatiivinen kanta

Operatiivinen tietokanta on tavallisten perussovellusten käyttöä varten suunniteltu kanta. Operatiiviset tietokannat palvelevat jokapäiväistä toimintaa ja niihin kerätään koko ajan ajankohtaista dataa. Ne ovat sovelluskohtaisia, jatkuvasti päivitettäviä ja niihin voidaan tehdä tarkkoja kyselyjä. (Tervonen: 2001)

Tiedot tulevat operatiiviseen kantaan käyttäjien syöttämänä päätteiden tai työasemien kautta, ei niinkään eräajoina. Esimerkkejä tavallisista operatiivisista järjestelmistä ovat esimerkiksi laskutusjärjestelmä, tilausten käsittely, pankin tilijärjestelmä, kiinteistörekisteri, myynti- ja ostoreskontra. Perusjärjestelmien tietokantoina käytetään sovelluksissa nykyään lähes yksinomaan SQL-pohjaisia relaatiokantoja. Kuvassa 1 selvitetään tiedon siirtymisen peruseriaate operatiivisista kannoista tietovarastoon. (Hovi 1997: 19.)

Perusperiaate



Kuva 1. Tiedon siirtyminen operatiivisista järjestelmistä tietovarastoon. (Hovi 1997: 51)

3.3 Taustaa tietovarastoinnista

Data Warehousing eli tietovarastointi on yksi nopeimmin kehittyviä tietotekniikan alueita. Useat yritykset ovat huomanneet perusjärjestelmien puutteet, kun on kysymys vaativista tietojen analysointi- ja raportointitarpeista. Hovi, Ylinen ja Koistinen (2001: 19) muistuttavat, että tiedot olisi hyvä koota omaan, erilliseen tietovarastotietokantaan. Koko tietovarastointi yleensäkin lähtee heidän mukaansa liiketoiminnan tarpeista, ja näin ollen tietovarastointiin liittyviä tuotteita ja välineitä on tullut markkinoille paljon.

Data Warehouse eli tietovarasto tarkoittaa suurta yrityksen toiminnon tai jopa koko yritystason kantaa, johon voidaan koota useiden sovellusalueiden perus- eli operatiivisten järjestelmien tietoja. Tietovarastoon voi siirtää myös ulkopuolisista lähteistä saatua tietoa. Tietovaraston päätarkoituksena on tarjota käyttäjille nopeasti oikeita vastauksia kyselyihin ja raportteihin. (Hovi ym. 2001: 50)

Tietovarasto on suunniteltu erityisesti tehokkaita hakuja varten. Siellä voidaan yhdistellä eri sovellusalueiden tietoja, ja siellä säilytetään myös historiaa. Kyselyt ja raportit muodostetaan tämän kannan tiedoista, jotka taas on talletettu keskitetysti tietovarastoon, ja loppukäyttäjillä onkin vain lukuoikeus näihin tietoihin. Operatiivisista järjestelmistä saa tietoa ulos yleensä vain järjestelmien omilla välineillä, kun taas tietovarastokanta on tämän suhteen avoimempi. (Suoranta)

Hovi ym. (2001: 24) ovat kehittäneet testin, jolla yrityksen on mahdollista tarkistaa nykyjärjestelmiensä mahdollisuudet tietojen analysoinnissa ja raportoinnissa.

Kuusi tärkeää kysymystä:

1. *Saatko nykytilanteessa nopeasti aikaan raportin, jossa on yhdisteltynä kahden (tai useamman) eri järjestelmän tietoja samaan raporttiin?*
2. *Voitko nopeasti tehdä raportin, jossa verrataan viime kuun tilannetta vuoden alun tilanteeseen tai viime vuoden takaiseen tilanteeseen?*
3. *Voiko operatiivisiin tietokantoihin kohdistaa päiväsaikaan runsaasti kyselyjä ja raportteja operatiivisten järjestelmien suorituskyvyn kärsimättä?*
4. *Ovatko tiedot järjestelmissä sopivassa muodossa raportointiin ja analyysihin (ei hankalia koodeja, kaikki tiedot selväkielisiä, selkeät tietorakenteet)?*
5. *Voiko raporttien tietoihin luottaa? Onko raporteissa eri tietoja samasta asiasta?*
6. *Pystyykö tiedon tarvitsija itse hakemaan haluamansa tiedon (ilman tietotekniikan ammattilaisen apua)?*

(Hovi, Ylinen & Koistinen 2001: 24)

3.4 Tietovarastojen hyödyntäminen

Integroitu, yhtenäinen ja keskitetty tietovarasto kokoaa parhaiten yrityksen arvokkaan tietoresurssin. Ennen kuin tieto laitetaan tietovarastoon, se pitää kuitenkin tarkistaa. Kun tällainen ”siivottu” tieto on saatu tietovarastoon, se voidaan samalla julkaista. Tiedot ovat silloin vapaasti käytettävissä, mutta rajoituksina ovat kuitenkin käyttäjien sallitut valtuudet. Tietojen vapaan saatavuuden vuoksi niiden tulisikin Hovin mukaan olla ehdottomasti oikeellisia ja ehyitä. (Hovi 1997: 16–17)

Tietovarastojen tietoja voidaan tutkia raporteina, taulukoina, ja myös esimerkiksi graafisina käyriä. Tietojen analysointiin ja liiketoiminnan kehityksen tarkasteluun on siis monta eri näkökulmaa. Jotta tiedoista saataisiin mahdollisimman paljon erilaisia asioita irti, täytyy tietoihin porautua syvemmälle. Osa tietojen saannin tarpeista on pysyviä, ja tällöin tarvitaan vakioraportteja. Joskus taas saattaa tulla nopeasti tarve uusien tietojen saantiin, jolloin pitää olla helppokäyttöinen väline, jotta voidaan muotoilla ad hoc -kyselyjä. (Hovi 1997: 90)

Porautuminen eli Drilling Down mahdollistaa Hovin mukaan moniulotteisen tiedon tarkastelun. Sen avulla käyttäjä voi tarkastella esimerkiksi myyntilukuja karkealla tasolla. Jos käyttäjä huomaa tietyllä alueella myynnin olevan poikkeuksellisen pientä, hän voi porautua syvemmälle ja katsoa tuon alueen sisällä tietoja esimerkiksi piireittäin. Kun poraudutaan alaspäin, mennään siis tarkemmalle tasolle ja saadaan mukaan uusia ulottuvuuksia. (Hovi 1997: 60)

Termillä ad hoc tarkoitetaan spontaaneja, ennalta arvaamattomia kyselyjä ja raportteja, joita ei siis ole ohjelmoitu valmiiksi. Ad hoc -kyselyjä voivat parhaimmassa tapauksessa tehdä tietovarastoon tai paikallisvarastoon liiketoiminnan edustajat itse. (Hovi ym. 2001: 264)

Osaa raporteista tarvitaan vakio- ja säännöllisesti. Tällaiset raportit voidaan ajaa automaattisesti aina, kun on ladattu tai summattu uusia asioita. Vakioraportteihin kuuluvat myös raportit, joita viranomaiset vaativat yrityksiltä. Käyttäjille kannattaa rakentaa vakio- ja säännöllisiä raportteja, joissa on tarpeeksi paljon parametreja. Tämä tarkoittaa sitä, että kyseisen raportin perusrakenne on vakio, mutta käyttäjä voi antaa raporttiin parametrisoituja arvoja. Näitä ovat esimerkiksi raportin tarkastelu-aika tai aikaväli, tuoteryhmä tai tuote, asiakasryhmä, alue tai piiri. (Hovi 1997: 90–91)

3.5 Hyvän raportin ominaisuudet

On tärkeää muistaa, että raporttien luontiin kuuluu monta erilaista vaihetta. Tärkein niistä on raportin suunnittelu. Jo tietokannan suunnitteluvaiheessa olisi hyvä ottaa huomioon tulevaisuuden raportointitarpeet. Näin vältettäisiin monet vaikeudet raporttien luomisessa ja toiminnassa. Raportin määrittelystä tulisi selvittää raporttiin halutut tiedot ja miten ne pitäisi ryhmitellä. (Crystal Reports 10 Peruskurssi 2004: 1–2)

Olisi ihanteellista, jos raporttien määrittely-, suunnittelu- ja toteutusvaiheissa otettaisiin huomioon sekä raportin käyttäjän että raportin suunnittelijan mielipiteet. Raportin tarvitsijan pitäisi pystyä tekemään järkeviä määritelmiä. Olisi hyvä, jos hän ymmärtäisi myös raportointityökalun ominaisuuksia. (Crystal Reports 10 Peruskurssi 2004: 1–2)

Raportin ajo sitoo raportointipalvelun ja työaseman resursseja melko paljon. Suurilla raporteilla nopeus onkin tärkeä tekijä. Nopeuteen vaikuttaa laitteisto, mutta merkittävä rooli on myös verkkoyhteyksillä ja tietokannan nopeudella. Itse raportti on myös tärkeä nopeuteen vaikuttava tekijä, sillä hyvin suunniteltu raportti toimii paremmin ja

nopeammin kuin huonosti suunniteltu. (Crystal Reports 10 Peruskurssi 2004: 1–2)

Raporttiin kannattaa ottaa mukaan vain ne taulut tietokannasta, jotka ovat välttämättömiä. Taulut liitetään toisiinsa ja nämä liitokset ovat aina suuri operaatio tietokannassa. Kun linkkikentiksi valitaan aina poikkeuksetta indeksoidut kentät, ei tauluja tarvitse lukea montaa kertaa läpi turhaan. Jos taas ei käytetä indeksoituja kenttiä, luetaan toinen linkitetty taulu kokonaan läpi niin monta kertaa kuin toisesta taulusta löytyy linkitettäviä tietoja. (Crystal Reports 10 Peruskurssi 2004: 1–2)

Kaavoja käytetään raportissa erilaisten tietojen esittämiseen eri tarkoituksissa. Jos kaavoja on paljon, raportin ajo hidastuu. Tieto tulisi myös rajata pieneksi joukoksi mahdollisimman hyvillä hakuehdoilla, ja hakuehtoina olevien kenttien olisi hyvä olla mukana indekseissä. (Crystal Reports 10 Peruskurssi 2004: 1–2)

Käytettäessä graafeja, kannattaa raportin suunnittelu aloittaa graafin näkökulmasta. Ensin mietitään, mitä tietoa graafissa pitää esittää ja minkälaista tiedon ryhmittelyä se raportilta vaatii. Kun ymmärtää, että graafikin esittää tiedon vain tietyllä tavalla ryhmiteltynä, on helppo tehdä vastaava raportti. (Crystal Reports 10 Peruskurssi 2004: 1–2)

3.6 Raportin suunnittelu

Raportin luomisprosessi on aina aloitettava raportin suunnittelulla ja tämä vaihe tulee tehdä mahdollisimman hyvin. Suunnitteluvaiheessa on hyvä muistaa, että raporttia voidaan muokata myöhemmin, mutta esimerkiksi tietokantakentän vaihtaminen saattaa aiheuttaa paljon hankaliakin muutoksia raportin muihin toimintoihin. (Crystal Reports 10 Jatkokurssi 2004: 102)

Raportin suunnittelun voi aloittaa esimerkiksi määrittelemällä sen tarkoituksen sekä tiedot, joita raporttiin halutaan tulostaa näkyviin. Esimerkkinä määrittelystä voi olla ”raportissa esitetään myynnit jokaiselle varaston tuotteelle, ja arvioidaan lisätilausten tarve aikaisempien myyntien perusteella”. Käyttäjäkunnan määrittelyn avulla tiedetään jo etukäteen minkälaista tietoa raporttiin voidaan tuoda, eikä kuka tahansa pääse esimerkiksi katselemaan yksittäisen työntekijän palkkatietoja. (Crystal Reports 10 Jatkokurssi 2004: 102)

Suunnittelun seuraavana vaiheena on raportin ulkoasu. Täytyy miettiä esimerkiksi raportin otsikko, lisätäänkö raporttiin yrityksen logo tai esitetäänkö tiedot mahdollisesti kaaviossa. Ad hoc -raportin ulkoasu ei ole niin tärkeä, mutta toistuvasti ajettavat sekä asiakkaille toimitettavat raportit vaativat tarkempaa suunnittelua. Layout kannattaa suunnitella paperilla mahdollisimman tarkasti vastaamaan haluttua ulkonäköä. Tässä

vaiheessa mietitään myös raportin tunnistetietojen, kuten päivämäärien, lisäämistä raporttiin. (Crystal Reports 10 Jatkokurssi 2004: 102)

Kun on saatu selville kaikki tiedot, joita raportissa halutaan näkyväksi (selkeä otsikko, sivunumerointi, mahdolliset parametrit, ajopäivä ja kellonaika, jakelutieto), täytyy selvittää käytettävä tietokanta ja kentät, joista tiedot haetaan tai lasketaan. Tässä vaiheessa otetaan huomioon myös kenttien tietotyypit, eli onko kenttä tekstiä vai numerotietoa. Tällä on merkitystä kun tehdään laskukaavoja. Nämä mahdolliset tarpeelliset laskukaavat olisi hyvä myös suunnitella valmiiksi. (Crystal Reports 10 Jatkokurssi 2004: 102)

Seuraaviin kysymyksiin pitäisi myös löytää vastaus raportin suunnitteluvaiheessa:

- halutaanko raportin tiedot ryhmitellä
- halutaanko tietueet lajitella ryhmien sisällä
- halutaanko raportista rajata joitakin tietueita tai jopa ryhmiä kokonaan pois
- halutaanko rajauksista pysyviä vai käyttäjän määrittelemiä parametreja
- halutaanko raportin tiedoista laskea välisummia
- halutaanko raporttiin lisätä hälytysrajoja

Kun kaikki edellä mainitut tiedot ovat selvillä, suunnittelijan täytyy vielä päättää raportin lohkoihin tulostettavien tietojen sijoittelu. (Crystal Reports 10 Jatkokurssi 2004: 102)

4 Käytettävyys

Yksinkertaisimmillaan käytettävyys tarkoittaa sitä, että tuote vastaa käyttäjän tarpeita. Käytettävyys on siis muutakin kuin käyttöliittymän visuaalinen ilme. Käytettävyyteen vaikuttavat käyttöliittymän ulkoasun lisäksi sen johdonmukaisuus ja toimintojen monipuolisuus sekä käyttäjän tiedot ja taidot. Käytettävyyden arvioiminen on siis subjektiivista.

Käsitteen laajuuden vuoksi käytettävyys on usein jaettu osiin. Antti Wiio (2004: 29–31) on kiteyttänyt käytettävyyden osatekijät selkeästi ja ytimekkäästi. Hän jakaa käytettävyyden neljään osa-alueeseen: ymmärrettävyyteen, vaivattomuuteen, kattavuuteen ja esteettiseen miellyttävyyteen.

Wiion (2004: 29–31) mielestä ymmärrettävyys, joka kattaa myös opittavuuden, on keskeinen käytettävyyden tekijä. Toinen tärkeä käytettävyyden tekijä on Wiion mukaan vaivattomuus. Hän määrittelee vaivattomuuden siten, että sovellus antaa käyttäjän suoriutua tehtävistään mahdollisimman yksinkertaisella tavalla. Ymmärrettävä sovellus ei kuitenkaan välttämättä ole Wiion mukaan vaivaton, sillä ymmärrettävyys on erittäin subjektiivinen asia. Vaikka käyttäjä ymmärtäisikin, miten tulee toimia, voi toiminnon suorittaminen silti olla vaivalloista.

Kattavuus ja esteettinen miellyttävyys kuuluvat Wiion (2004: 29–31) hyvän käytettävyyden määreisiin. Kattava sovellus tarjoaa kaikki ne toiminnot ja tiedot, joita käyttäjä tarvitsee hoitaakseen sen tilanteen tai tarpeen, johon sovellus on tarkoitettu. Selkeästä käytettävyyden osatekijöiden määrittelystään huolimatta Wiio myös tunnistaa sen tosiasian, että käytettävyyttä on erittäin vaikea määritellä aukottomasti.

Kuoppala ja Nieminen (1998) ovat artikkelissaan tiivistäneet ISO 9241-11 -standardin käytettävyyden määritelmän lyhyesti ja ymmärrettävästi. Heidän mukaansa käytettävyys esitetään standardissa kokonaisuudeksi, joka määrittää, miten hyvin tietyt käyttäjät kykenevät käyttämään käytössään olevaa työvälinettä tiettyjen tehtäviensä suorittamiseen tietyssä ympäristössä tavoitteidensa saavuttamiseen. Heidän mukaansa standardi nostaa tärkeiksi tarkasteltaviksi sisällöllisiksi kohteiksi käyttäjän, hänen tehtävänsä, käytetyt työvälineet ja toimintaympäristön.

Sinkkosen, Kuoppalan, Parkkisen ja Vastamäen (2002: 19) mukaan käytettävyys on teoriakenttä, jonka avulla käyttäjän ja laitteen yhteistoimintaa pyritään saamaan tehokkaammaksi ja miellyttävämmäksi. Käytettävyys käyttää hyväkseen kognitiivisen psykologian sekä ihmisen ja tietokoneen vuorovaikutuksen tutkimusta.

4.1 Käytettävyyden osatekijät

Luultavasti siteeratuin ja käytetyin käytettävyyden määritelmä on 1990-luvun alkupuolella ilmestynyt Jacob Nielsenin määritelmä. Seuraavassa on tarkasteltu käytettävyyden osatekijöitä hänen määritelmänsä pohjalta.

Nielsen (1993: 26–29) painottaa, että käytettävyys ei ole vain käyttöliittymän ominaisuus, vaan käytettävyys koostuu useista osista. Hän jakaa käytettävyyden viiteen osa-alueeseen, joita tarkastelemalla saadaan parempi käsitys käytettävyyden merkityksestä:

- opittavuus
- tehokkuus
- muistettavuus
- virheettömyys
- tyytyväisyys

Opittavuus

Järjestelmän tulisi Nielsenin (1993: 27–29) mukaan olla niin helposti opittava, että käyttäjä pystyy nopeasti oppimaan järjestelmän käytön perusteet pystyäkseen työskentelemään sen avulla. Opittavuus onkin yksi käytettävyyden keskeisimmistä osatekijöistä ja se on mukana useimmissa käytettävyyden määritelmissä. Se on Nielsenin mukaan myös käytettävyyden perustavanlaatuisin osatekijä, koska useimpien järjestelmien on oltava helppoja käyttää.

Opittavuus on erityisen tärkeää siksi, että käyttäjä harvoin ensin opettelee käyttämään järjestelmää ja vasta sitten käyttää sitä. Useimmiten käyttäjä aloittaa varsinaisen käytön heti perusteet opittuaan ja opettelee uusia ominaisuuksia matkan varrella.

Sinkkonen ym. (2002: 265) ovat todenneet, että opittavuus on myös ISO-9241-10 -standardin mukaan yksi käytettävyyden osatekijä. Heidän mukaansa standardi määrittelee vuorovaikutuksen tuotteen ja ihmisen välillä olevan sopivaa oppimiseen silloin, kun se tukee ja opastaa käyttäjää järjestelmän oppimisessa. Heidän mukaansa ISO 9241-11 standardi kuvaa opittavuutta tuotteen tuoton, tehokkuuden ja tuotteen miellyttävyyden kautta.

Tehokkuus

Järjestelmän käytön tulisi Nielsenin (1993: 30) mukaan olla myös tehokasta, jotta käyttäjä voisi käytön perusteet opittuaan saavuttaa tuotteen käytössä korkean tehokkuusasteen. Tehokkuutta voidaan mitata ajassa ja siten työelämässä myös rahassa. Tehokas ohjelmisto toimii nopeasti ja sen eri toiminnot ovat helposti löydettävissä ja käytettävissä. Tehokkuutta voidaan Nielsenin mukaan lisätä muun muassa pikapainikkeilla, eli niin sanotuilla oikopoluilla, jotka voidaan järjestää toiminnon käyttöasteen tai yleisyyden perusteella.

Muistettavuus	Nielsenin (1993: 31–32) mukaan järjestelmän pitäisi olla myös helposti muistettava, jotta satunnainen käyttäjä pystyy sovelluksen pariin palatessaan käyttämään sitä edelleen tehokkaasti. Muistettavuutta lisäävät selkeä nimeäminen ja selkeät kuvakkeet sekä standardien noudattaminen.
Virheettömyys	Käytettävyydeltään hyvässä ohjelmistossa tulisi Nielsenin (1993: 32) mukaan minimoida käyttäjän tekemien virheiden määrä ja järjestelmän pitäisi myös pystyä toipumaan tehdyistä virheistä helposti. Suuria virheitä ei saisi tapahtua. Virheitä voidaan parhaiten minimoida erilaisilla varmistuskysymyksillä ja syötetarkistuksilla.
Tyytyväisyys	Käyttäjän tulisi kokea tuotteen käyttö miellyttäväksi. Käyttäjän tyytyväisyys koostuu kaikista edellä mainituista käytettävyyden osatekijöistä sekä tuotteen visuaalisesta miellyttävyydestä. Nielsenin (1993: 33–37) mukaan kaikkien käytettävyyden osatekijöiden täydellinen toteutuminen järjestelmässä ei aina ole mahdollista. Esimerkiksi jotta vältettäisiin virhetilanteet, pitää järjestelmään ehkä lisätä varmistusdialogeja, jotka saattavat hidastaa kokeneen käyttäjän työskentelyä, mutta pelastaa kokemattoman käyttäjän monelta harmilta.

4.2 Käytettävyyden merkitys ohjelmiston valinnassa

Ohjelmistojen huonon käytettävyyden on todettu aiheuttavan yrityksille lisäkustannuksia. Sen vuoksi käytettävyyteen tulisi kiinnittää huomiota entistä enemmän, kun valitaan ohjelmistoa yrityksen käyttöön. Erityisesti ohjelman ymmärrettävyys ja opittavuus ovat tärkeitä käytettävyystekijöitä uutta ohjelmistoa valittaessa.

Jos käytettävyyteen kiinnitetään huomiota ohjelmiston valintavaiheessa, voidaan käyttöönottovaiheessa säästää työntekijöiden aikaa huomattavasti. Käytettävyydeltään hyvä ohjelmisto voi tuoda säästöjä myös käytön myöhemmässä vaiheessa. Sen kaikki ominaisuudet tulevat hyödynnetyiksi, kun ne ovat selkeästi löydettävissä.

Myös Wiio (2004: 30) on todennut vaivalloisen toimenpiteen vievän yleensä aikaa huomattavasti enemmän kuin vaivattoman ja näin aiheuttavan taloudellisia kustannuksia. Hänen mukaansa käytettävyyden ongelmilla voi olla useita välittömiä vaikutuksia:

- käyttäjä ei omaksu tuotetta käyttöönsä
- käyttäjältä jää sovelluksen ominaisuuksia hyödyntämättä
- käyttäjä tekee virheitä
- käyttäjältä kuluu tehtäviinsä tarpeettoman paljon aikaa

Näillä ilmiöillä on Wiion (2004: 33–34) mukaan myös seurauksia. Yrityksen sisäisessä käytössä ne voivat hänen mukaansa

- lisätä tuen ja koulutuksen tarvetta
- alentaa työn tuottavuutta
- alentaa yrityksen asiakaspalvelun laatua
- alentaa suunnittelun ja päätöksenteon laatua

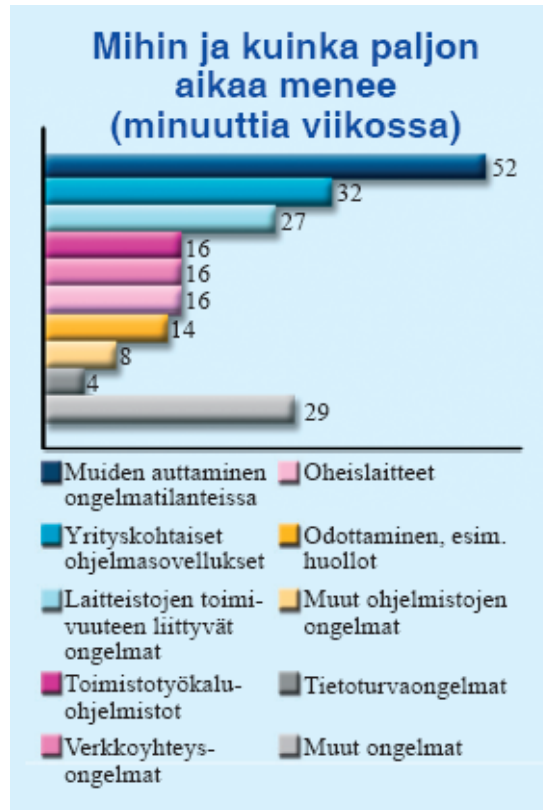
Kaikki edellä mainitut ongelmat voivat Wiion (2004: 34) mukaan alentaa yrityksen kilpailukykyä ja kannattavuutta. Ohjelmiston käytettävyyteen kannattaa siis todella kiinnittää huomiota ohjelmiston hankintavaiheessa, silloinkin, kun ohjelmisto on tarkoitettu yrityksen sisäiseen käyttöön.

Säästöt eivät aina ole täysin ilmeisiä ja suoraan mitattavissa olevia. Myös Tietotekniikan liiton ja Cap Gemini Ernst & Youngin vuonna 2003 tekemä tutkimus tietotekniikan käytön ongelmista suomalaisilla työpaikoilla osoittaa, että ongelmia on havaittu ja että niillä on myös taloudellisia vaikutuksia. Tutkimuksen mukaan tietotekniikasta aiheutuvat ongelmat vaarantavat tehokkuuden ja muodostavat merkittäviä suoria ja epäsuoria kustannuksia. (Tietotekniikan käytön... 2003)

Tietokoneiden käyttäminen on edellytys työn suorittamiselle nykyisissä työorganisaatioissa, joten käytettävyydellä on merkitystä. Tutkimuksen keskeisenä löydöksenä oli, että suomalaiset keskikokoiset ja suuryritykset menettävät vuositasolla lähes 30500 henkilötyövuotta tietotekniikasta johtuvien ongelmien vuoksi. Tämän lisäksi tutkimuksen mukaan yrityksen resursseja kuluttaa ongelmiin käytetty työpanos sekä keskeytykset työprosesseissa. (Tietotekniikan käytön... 2003.)

Lisäkustannuksia yrityksille aiheuttavat tutkimuksen mukaan muun muassa mikrotuen ajankäyttö, laite- ja sovellustoimittajien tuki sekä ulkopuolisten palveluiden käyttö. Tutkimus osoittaa myös, että yrityksille muodostuu lisäksi muita epäsuoria kustannuksia, jotka johtuvat tietojärjestelmien toimimattomuudesta. Esimerkiksi tilausten käsittely ja tekeminen voivat viivästyä ja samalla asiakastytyttömyys lisääntyä.

Tutkimuksessa tutkittiin kaikkia tietotekniikan aiheuttamia ongelmia, ja tulosten mukaan ohjelmistojen käytettävyyden osuus kaikista ongelmista on huomattava. Kuva 2 havainnollistaa tietotekniikan ongelmiin kuluvan työajan minuutteina viikossa. Käyttäjien muiden ongelmien, joiden voidaan katsoa koostuvan suurelta osin juuri käytettävyysongelmista, selvittämiseen kuluu siis viikossa keskimäärin 52 minuuttia. Myös yrityskohtaisten ja toimistosovellusten aiheuttamien ongelmien voidaan ymmärtää olevan pääasiassa käytettävyyteen liittyviä. Tutkimus siis vahvistaa, että ohjelmistojen käytettävyysongelmat tekevät ison loven tehokkaaseen työskentelyaikaan. (Tietotekniikan käytön... 2003.)



Kuva 2. Tietotekniikan ongelmien viemä työaika viikossa (Tietotekniikan käytön... 2003)

Tutkimus tähdentää, että paitsi yritystasolla, tietotekniikan ongelmat aiheuttavat kustannuksia myös kansantaloudellisesti, sillä ongelmat muodostavat lähes 3 miljardin euron kustannuserän. Kustannuserä muodostuu tietotekniikan ongelmien aiheuttamista “menetetyistä” palkkakustannuksista. Tutkimuksessa korostetaan, että kansantaloudellekin merkittävien rahallisten vaikutusten takia, on erittäin tärkeää varmistaa tietotekniikan käytettävyys niin yksilö- kuin yritystasollakin. Tutkimuksen mukaan käytettävyyttä parantavat esimerkiksi jatkuva ja riittävä perus- ja jatkokoulutus sekä pitkäjänteinen käyttäjälähtöinen tutkimus- ja kehitystoiminta. (Tietotekniikan käytön... 2003)

Koska kustannukset voivat huonon käytettävyyden vuoksi lisääntyä huomattavasti, on luonnollista, että käytettävyys otetaan huomioon uutta ohjelmistotyökalua valittaessa. Ohjelmiston hyvä käytettävyys vähentää kustannuksia, koska arvokasta työaika kuluu vähemmän ohjelmiston käytettävyysongelmiin ja enemmän varsinaisen työn tekemiseen. Käytettävyydeltään hyvän ohjelmiston käyttö voi jopa tuottaa alusta asti sen sijaan, että etenkin käytön alkumetreillä se aiheuttaisi pelkästään kustannuksia.

4.3 Käytettävyyden arviointi

Tuotteen käytettävyyden varmistamiseksi on olemassa useita eri menetelmiä. Valmiin tuotteen käytettävyyttä voidaan arvioida erilaisten perusteiden, kuten monesti käytettyjen heurististen sääntöjen pohjalta ja sen käytettävyyttä voidaan testata. Sinkkosen ym. (2002: 295) mukaan käytettävyyden testaaminen on joustava menetelmä, jota muuntelemalla saadaan tuotteen käytöstä monenlaista tietoa. Menetelmää voidaan heidän mukaansa käyttää hyvin monenlaisten tuotteiden testaamiseen.

Käytettävyyttä arviotaessa tulee tarkasti määritellä kenelle ja mihin tarkoitukseen ohjelmisto on tarkoitettu. Ammattikäyttöön tarkoitettun ohjelmiston käytettävyyttä arvioidaan eri kannalta kuin ohjelmiston, jonka kohderyhmänä ovat lähinnä harrastelijat. Alan asiantuntemusta omaava henkilö oppii käyttämään alan määreiden mukaan käytettävyydeltään hyvää ohjelmistoa helposti, vaikka harrastelijan näkökulmasta ohjelmiston käytettävyys olisi surkea. Näin ollen kohderyhmän määrittely vaikuttaa oleellisesti ohjelmiston käytettävyyteen.

Vaikka itse käyttöliittymässä kaikki olisi kunnossa, ohjelman käytettävyys on silti huono, jos siitä ei löydy niitä ominaisuuksia, joita käyttäjä tarvitsee tehtävistään suoriutumiseen. Sen vuoksi ohjelmiston toimintojen määrittely tulisi aina aloittaa kartoittamalla vaaditut ominaisuudet – oli kyse sitten uuden ohjelmiston suunnittelusta tai hankittavan ohjelmiston käytettävyyden arvioinnista. Ei ole mitään järkeä valita ohjelmistoa kauniin ja toimivan käyttöliittymän tai käytön yksinkertaisuuden vuoksi, jos sen avulla ei voida toteuttaa kaikkia niitä tehtäviä, joita sillä pitäisi pystyä suorittamaan.

4.3.1 Jacob Nielsenin heuristiikat

Kuutti (2003: 47) määrittelee heuristiikat listoiksi säännöistä ja ohjeista, joita käytettävyydeltään hyvän käyttöliittymän tulisi noudattaa. Käytettävyysheuristiikat ovat siis eräänlaisia ohjenuoria, joita noudattamalla tuotteen käytettävyyttä voidaan parantaa tai valmista ohjelmistoa voidaan arvioida. Kuutti (2003: 49) on myös todennut, että paras käytettävyyden arvioija on käytettävyysasiantuntija, joka tuntee hyvin testattavan sovelluksen sovellusalueen.

Heuristiikat voivat olla yleisiä tai hyvinkin sovelluslakohtaisia, jolloin ne eivät välttämättä ole sovelluskelpoisia muun tyyppisten sovellusten kohdalla. Erilaisia yleisiä käytettävyysohjeita ja heuristiikkoja löytyy alan kirjallisuudesta sadoittain, ja jotkut listat sisältävät yksinäänkin jopa tuhat läpikäytävää ohjetta. Tällaisten kokoelmien läpikäynti on hidasta, ja tästä syystä on esitetty tiivistettyjä ja yleisluontoisempia kokoelmia,

joista yksi on seuraavassa esitetty Jacob Nielsenin heurististen sääntöjen kokoelma.

Nielsenin (1993: 115–163) kokoelma koostuu kymmenestä säännöstä, joiden avulla tuotteen käyttöliittymää voidaan arvioida tai sen toimintaa voidaan parantaa:

- yksinkertainen ja luonnollinen dialogi
- puhu käyttäjän kieltä
- minimoï käyttäjän muistintarve
- yhdenmukaisuus
- palaute
- selkeästi merkityt poistumistiet
- oikopolut
- virheilmoitukset
- virheiden ehkäiseminen
- help-tiedostot ja dokumentaatio

Seuraavassa kutakin kymmenestä säännöstä on tarkasteltu hieman tarkemmin hankittavan sovelluksen arvioinnin näkökulmasta.

Yksinkertainen ja luonnollinen dialogi

Tuotteen käyttöliittymän tulisi olla mahdollisimman yksinkertainen, sillä ylimääräiset vaihtoehdot hidastavat oppimista ja lisäävät väärinkäsityksien mahdollisuutta. Sovelluksessa kullakin hetkellä esitetyn tiedon määrä tulisi olla mahdollisimman pieni selkeyden säilyttämiseksi. Sovelluksessa pitäisi siis esittää ainoastaan välttämätön määrä tietoa, jonka käyttäjä tarvitsee tietyn tehtävän suorittamiseen. (Nielsen 1993: 115–122)

Sovelluksen visuaalisen ilmeen olisi hyvä olla mahdollisimman yksinkertainen, mutta kuitenkin miellyttävä. Sovelluksen tärkeimpien elementtien tulisi löytyä helposti, mutta ne eivät saisi olla liiallisen korostuneita. On myös syytä tarkastaa, että sovellus toimii myös ilman värejä, sillä monet käyttäjät ovat värisokeita tai heillä on puutteellinen värinäkö. (Nielsen 1993: 115–122)

Puhu käyttäjän kieltä

Sovelluksessa käytettävien termien olisi hyvä vastata käyttäjien omaa kieltä. Liian teknisiä termejä ei saisi esiintyä, mutta tarvittaessa tulisi käyttää myös sovellusalakohtaista erikoistermistöä. Paras tapa varmistaa termistön toimivuus yrityksen omassa käytössä, on antaa varsinaisten käyttäjien arvioida termien ymmärrettävyyttä. Jos ohjelmistoa käytetään kansainvälisissä projekteissa, olisi syytä tarkistaa, että ainakin päivämäärät ja valuutat ovat oikeassa muodossa. (Nielsen 1993: 123–128)

Myös Sinkkonen ym. (2002: 77) ovat todenneet, että pystyäkseen käyttämään tuotetta, käyttäjän pitää pystyä havaitsemaan tuotteessa kaikki tehtävän suorittamisen kannalta oleelliset asiat. Käyttäjän vuorovaikutus käyttöliittymän kanssa perustuu Sinkkosen ym. (2002: 125) mukaan siihen, että käyttäjä osaa lukea suunnittelijan merkkikieltä. Vuorovaikutuksen rakentamisessa on käyttöliittymän suunnittelulla suuri merkitys, vaikka se ei yksinään käytettävyyttä määrittelekään. Ohjelmiston symboliikan ja nimeämisen tulisi siis olla mahdollisimman lähellä todellista maailmaa, jotta käyttö sujuu käyttäjältä.

Minimoi käyttäjän muistintarve

Ohjelmiston tulisi säilyttää ja tarvittaessa esittää käyttäjän sinne tallettamien asiain ilman, että käyttäjän tarvitsee muistella aiempia tekemisiään. Samoin esimerkiksi päivämäärän syöttömuoto pitäisi esittää suluissa syötekentän yhteydessä, jotta käyttäjä voi helposti todeta vaadittavan muodon. Yksi tehokas tapa minimoida muistintarvetta, on käyttää yleisiä komentoja kuten leikkaa, kopioi ja liitä. Käyttäjän ei myöskään pitäisi joutua muistamaan syötteiden pituusrajoituksia, vaan niiden ylittymisestä tulisi erikseen informoida käyttäjää. (Nielsen 1993: 129–131.)

Yhdenmukaisuus

Ohjelmiston opittavuutta ja ymmärrettävyyttä parantaa huomattavasti sen yhdenmukaisuus. Yhdenmukaisuuteen liittyvät ulkoasun lisäksi erot toimintojen toteuttamisessa. Tämän vuoksi tulisi tarkistaa, että samat käskyt toimivat samalla tavalla ja sama tieto löytyy samalta kohdalta eri ikkunoissa samalla tavalla esitettynä. (Nielsen 1993: 132–133.)

Palaute

Systemin tilan pitäisi olla koko ajan nähtävissä, ja ohjelmiston tulisi informoida käyttäjää hänen toimiansa vaikutuksista. Useampiosaisissa käskysarjoissa välivaiheiden tulisi olla selkeästi näkyvillä, ja peruuttamattomat toimet tulisi varmistaa kysymällä käyttäjältä ennen niiden suorittamista. Käyttäjälle olisi kuitenkin hyvä antaa mahdollisuus usein toistuvien turhien kysymysten poissulkemiseen. (Nielsen 1993: 134–138)

Ohjelmiston antama palaute voidaan jakaa kertaluonteiseen tai edistystä osoittavaan tyyppiin. Kertaluonteinen palaute vahvistaa jonkin toiminnon suorittamisen. Edistystä osoittavaa palautetta tulee puolestaan saada esimerkiksi edistyspalkin muodossa, jos vasteaika ylittää tietyn rajan, esimerkiksi 10 sekuntia. Näin käyttäjä tietää, että hänen toimintonsa on rekisteröity. Joissakin tilanteissa myös hidastukset ovat tärkeitä, jotta käyttäjä voisi pysyä perillä tapahtumista. (Nielsen 1993: 134–138)

Selkeästi merkityt poistumistiet

Käyttäjä tuntee olonsa usein ahdistuneeksi, jos hän ei pysty löytämään tietä pois ohjelmasta. Tästä syystä ohjelmistossa tulisi olla yhdenmukainen nimeämiskäytäntö poistumistielle ja sen pitäisi olla aina näkyvissä. Graafisissa käyttöliittymissä lopetusmahdollisuuden tulisi olla koko ajan näkyvissä samassa paikassa. (Nielsen 1993: 138–139.)

Oikopolut

Kokenutta käyttäjää varten tulisi järjestelmästä löytyä mahdollisuus automatisoida usein käytettyjä toimintoja. Oikopoluiksi voidaan laskea esimerkiksi makrot, nettisivujen kirjanmerkit, konsolien komentohistoriat, listat usein avatuista tiedostoista, järjestelmän antamat oletusarvot ja niin edelleen. Oikopolut pystyvät lisäämään sovelluksen käytön tehokkuutta huomattavasti. (Nielsen 1993: 139–142)

Virheilmoitukset

Käytettävyyden kannalta on olennaista ehkäistä virhetilanteita ja toisaalta auttaa käyttäjää toipumaan niistä. Koska ihminen tekee virheitä lähes poikkeuksetta, on tärkeää, että virheilmoitukset ovat mahdollisimman selkeitä. Tästä syystä pelkkiä virhekoodeja pitäisi välttää. Virheilmoituksen tulisi myös olla tarkka ja se voisi ehdottaa jatkotoimenpiteitä. Myös sovelluksen kyky toipua virheistä on syytä tarkistaa. (Nielsen 1993: 142–145)

Virheiden ehkäiseminen

Käyttäjän tekemiä virheitä voidaan ehkäistä monella tapaa. Käyttäjät tekevät usein virheitä kirjoittaessaan, joten saattaa olla parempi antaa heidän valita esimerkiksi haluttu tiedosto suoraan listasta. Erityisen virhealttiissa ja vakavissa tilanteissa tulisi olla varmistuskysymykset. Virheitä ehkäisee tehokkaasti myös ohjelmiston yhdenmukaisuus. (Nielsen 1993: 145–148)

Jos jokin ohjelmiston tila, eli niin sanottu moodi, edellyttää samojen käskyjen toimimisen eri tavalla, tulisi tilat erottaa selvästi toisistaan. Eri moodeille spesifisten näppäinkomentojen olisi hyvä mahdollisuuksien mukaan erota toisistaan mahdollisimman paljon. (Nielsen 1993: 145–148)

Ohjetiedostot ja -kirjat

Ihanteellista käyttöliittymää on niin helppo hallita, että ilman ohjetiedostojakin pääsee jo suhteellisen pitkälle. Ohjetiedostot ja muut dokumentaatiot ovat kuitenkin osa ohjelmistoa, joten myös niiden laatuun on hyvä kiinnittää huomiota ohjelmistoa hankittaessa. (Nielsen 1993: 148–155)

Useimmat käyttäjät eivät lue ohjekirjoja ennen ongelmatilanteen syntymistä, joten ohjekirjan tulisi rakenteellaan tukea nopeaa vastausten löytymistä. Ohjetiedostojen löytyminen suoraan ohjelmistosta on myös erittäin tärkeää, koska silloin ne ovat nopeasti käytettävissä ja vastauksen etsiminen sähköisestä tiedostosta on huomattavasti helpompaa ja nopeampaa. (Nielsen 1993: 148–155.)

4.3.2 Käytettävyysetit

Sinkkonen ym. (2002: 295) toteavat, että käytettävyysetin perusversiossa pyritään selvittämään käyttäjän ajatusmalleja niin, että käyttäjä suorittaessaan käytettävyysetin tehtäviä testattavalla tuotteella ajattelee samalla ääneen. Käyttäjälle muodostuvaa mallia tuotteesta voidaan heidän mukaansa selvittää muillakin keinoin, mutta ääneen ajattelun on tutkimusten perusteella havaittu antavan parasta tietoa ja ehkäpä siksi se onkin tunnetuin käytettävyysetimenetelmä.

Käytettävyysetien tarkoitus on yleensä tehdä tuotteesta käytettävämpi. Kun kehitetään uusia ohjelmistoja, voidaan samoja menetelmiä käyttää myös valmiin tuotteen analysointiin päätettäessä tuotteen hankinnasta. Ideaalista olisi, että jokainen markkinoilla oleva ohjelmisto olisi testattu ja käytettävyydeltään erinomaiseksi todettu, mutta valitettavasti asia ei todellisuudessa ole näin. Käytettävyys on subjektiivinen määre, joten yrityksen edes jonkinasteinen oma arviointi on tarpeen, kun hankitaan uutta ohjelmistoa.

Sinkkonen ym. (2002: 302) jakavat käytettävyysetin vaiheet karkeasti kolmeen osaan:

1. Testin järjestäminen ja testaussuunnitelman laatiminen.
2. Testin suorittaminen.
3. Testin analysointi ja testiraportin laatiminen.

Heidän mukaansa eri vaiheiden sisältö riippuu siitä, ollaanko suorittamassa kvantitatiivista vai kvalitatiivista testiä. Kvalitatiivisessa testissä mitataan käyttöliittymän laatua joko verrattuna annettuihin käytettävyysetavoitteisiin tai johonkin verrokkituotteeseen. Kvantitatiivisella testillä taas pyritään löytämään tuotteesta niin monta käytettävyydeltään ongelmallista kohtaa kuin mahdollista. (Sinkkonen ym. 2002: 302)

Käytettävyysetin järjestäminen ja testaussuunnitelman laadinta

Sinkkosen ym. (2002: 303–304) mukaan testaussuunnitelma käsittää ainakin seuraavat työvaiheet:

- testin tavoitteiden selvittäminen
- käytettävyysetvaatimusten selvittäminen

- tuotteeseen tutustuminen
- testattavien toimintojen valinta
- testikäyttäjien määrä ja valinta
- testitarinan ja -tehtävien laadinta
- testausmenetelmän valinta
- muita järjestelyasioita
- koulutustarve ennen testiä
- käsikirja mukaan vai ei
- testitilanteen määrittely
- pilottitestin järjestäminen

Kaikki nämä asiat kirjataan testaussuunnitelmaan, joka ensin testataan pilottitestin avulla. Lopulliset käytettävyydestit suoritetaan testaussuunnitelman mukaisesti ja jokaisesta laaditaan testiraportti.

Testin suorittaminen

Sinkkosen ym. (2002: 313) mukaan käytettävyydestin rakenne voi olla esimerkiksi seuraava

- testitilanteen selvittäminen käyttäjälle
- alkukysely tai -haastattelu
- testitehtävien tekeminen
- visuaalinen läpikäynti
- loppuhaastattelu

Kun ohjelmistoa testataan hankintavaiheessa, parhaita mahdollisia testihenkilöitä ovat hankittavan tuotteen tulevat käyttäjät. Heillä on luultavasti perustiedot ohjelmiston käyttöalalta ja ehkä myös kokemusta muista alan ohjelmistoista. Sen vuoksi heidän antamansa palaute on erityisen arvokasta.

Ennen varsinaista testiä käyttäjälle tulee selvittää testin kulku ja kaikki testiin liittyvät asiat. Käyttäjälle tulee myös korostaa, että testi ei testaa hänen osaamistaan vaan testattavaa tuotetta. Testin aluksi käyttäjältä voidaan kysellä taustatietoja, joiden avulla varmistetaan muun muassa se, että käyttäjä kuuluu kohderyhmään, ja joilla pyritään selvittämään hänen lähtötasonsa. (Sinkkonen ym. 2002: 314.)

Varsinaisen testin alkaessa testaajan rooli on sivustaseuraaja. Käyttäjän on tarkoitus paljastaa tuotteen käytettävyysoongelmia, joten neuvomista tulee välttää. Testitehtävät voidaan antaa testihenkilölle kaikki kerralla tai yksitellen. (Sinkkonen ym. 2002: 314–315.)

Mikäli testattava ohjelmisto halutaan käydä visuaalisesti läpi käyttäjän kanssa, se kannattaa tehdä Sinkkosen ym. (2002:315) mukaan vasta varsinaisen testin jälkeen, koska ennen testiä suoritettu katselmointi

saattaa vaikuttaa testin suorittamiseen. Viimeiseksi tehtävällä loppuhaastattelulla voidaan vielä selvittää käyttäjän tuntemuksia ja mielipiteitä tuotteesta.

Testin tulokset

Kun testataan ohjelmistoa sen hankintavaiheessa, testiraporttiin kerätään paitsi testien tulokset eli havaitut käytettävyysongelmat, myös testihenkilöiden kokemukset ja mielipiteet testattavasta tuotteesta ja sen käytöstä. Raporttien perusteella saadaan arvokasta tietoa ohjelman soveltuvuudesta oman yrityksen käyttöön. Näitä tietoja voidaan käyttää valittaessa parhaiten sopivaa ohjelmistoa. (Sinkkonen ym. 2002: 318.)

Ei tule kuitenkaan unohtaa, että käytettävyys on muutakin kuin miellyttävä ja toimiva käyttöliittymä tai ohjelmiston tehokasta toimintaa. Jos ohjelmistosta ei löydy kaikkia tarvittavia ominaisuuksia, on käytettävyydestejä turha edes tehdä. Ominaisuuksiltaan tasavertaisten ohjelmistojen vertailussa ne puolestaan antavat arvokasta lisätietoa.

5 Ohjelmistojen keskeinen toiminnallisuus

Valitsimme selvittävät toiminnalliset ominaisuudet sen perusteella, mitkä Crystal Reportsin ominaisuuksista on koettu toimeksiantajan aikaisemmissa asiakasprojekteissa tärkeiksi. Kartoitimme ja rajasimme selvittävät piirteet yhdessä toimeksiantajan raportointiasiantuntijoiden kanssa.

5.1 Crystal Reports

Crystal Reports on graafinen raportointiohjelmisto, jonka avulla tietokannassa olevia tietoja voidaan hakea ja muokata raportointia ja analysointia varten. Seuraavassa on selvitetty toimeksiantajan näkökulmasta Crystal Reportsin keskeiset ominaisuudet.

5.1.1 Datayhteydet

Crystal Reportsissa on useita datalähdeajureita eri datalähteiden käyttämiseen raportissa, ja sillä voidaankin luoda raportteja lähes minkä tahansa datalähteen sisältämästä datasta, esimerkiksi yksinkertaisista tekstitiedostoista, SQL-tietokannoista, XML-tiedostoista ja OLAP-tietokuutioista. OLAP on moniulotteisten tietorakenteiden analysointiprosessi, joka sisältää tietovarastointiin ja tiedon analysointiin liittyviä vaiheita (Hovi ym. 2001: 54). Crystal Reportsilla voidaan myös luoda kyselyjä ja näkymiä, eli niin sanottuja tiedon kokoomatiedostoja, joita itsessään voidaan käyttää datalähteinä.

Crystal Reports tukee kaikkia SQL- ja ODBC-tietokantoja. PC-tietokannoista Crystal Reports tukee muun muassa Microsoft Accessia, dBASEa, FoxProta, Clipperiä, Btrieveä ja Paradoxia. (Crystal Reports 8.5 2001: 77)

Data, jota Crystal Reports voi hyödyntää, voidaan jakaa viiteen eri kategoriaan:

- natiiviyhteys (suora yhteys tietokantatiedostoihin)
- ODBC-tietolähteet
- OLE DB -tietolähteet
- Crystal SQL Designer -tiedostot
- Crystal Dictionary -tiedostot

Jokaiseen datatyyppeihin otetaan yhteys erityisten dynaamisten linkkikirjastojen ja muiden tarvittavien tiedostojen avulla. (Crystal Reports 8.5 2001: 574–575)

Käytettävän yhteyden valitseminen riippuu monesta eri tekijästä. PC-tietokantoihin otetaan suora eli natiiviyhteys. Samoin, jos halutaan käyttää joitakin tietokantaohjelmiston erikoisominaisuuksista Crystal Reportsista käsin, pitää yhteys luoda natiiviyhteytenä. ODBC-yhteyden etuna taas on muun muassa mahdollisuus muuttaa raportti hakemaan tiedot myös toisesta rakenteeltaan vastaavasta tietokannasta.

Crystal SQL Designer -tiedostot

Crystal SQL Designerin avulla luodaan kyselyjä tietokantaan. Kun data on kerätty, se voidaan tallettaa Crystal SQL Designer -tiedostona ja tiedostoa voidaan sen jälkeen käyttää tietokantataulun tapaan raporttia luotaessa. Kerätystä datasta tulee uusi datakokoelma, joka voidaan lisätä raporttiin. (Crystal Reports 8.5 2001: 617)

Tämänkin yhteystavan etuna on SQL-lausekkeiden hallintamahdollisuus sekä prosessoinnin nopeus ja tehokkuus, koska suurin osa prosessoinnista tapahtuu tietokantapalvelimella. Käytettäessä Crystal SQL Designeria ei kuitenkaan voida samassa raportissa käyttää muita datalähteitä, kuten esimerkiksi tietokantataulua. Muodostettu SQL-lause myös rajaa raportissa käytettävissä olevan datan, joten sen ulkopuolelle jäävää dataa ei voida käyttää raportissa SQL-lausetta muokkaamatta, vaikka siihen myöhemmin tulisikin tarvetta. (Crystal Reports 8.5 2001: 617)

Crystal Dictionary -tiedostot (näkyvät)

Kokoomatiedosto (dictionary file) on rakenteinen ja yksinkertaistettu näkymä datasta. Näkymä voidaan luoda suodattamaan, yksinkertaistamaan ja selkeyttämään monimutkaista dataa henkilöille, joiden ei raporteja luodessaan tarvitse – tai jotka eivät saa – päästä käsiksi kaikkiin tietokannassa oleviin tietoihin. Käytettäessä näkymiä tietolähteinä ei samalla raportissa voida käyttää muita tietolähteitä. (Crystal Reports 8.5 2001: 618)

Useimmiten datan hallinnasta vastaava henkilö luo näkyvät, joten näkyvillä voidaan lisätä myös tietoturvallisuutta, kun tarpeettoman monet henkilöt eivät pääse käsiksi kaikkiin tietokannassa oleviin tietoihin. Samasta syystä näkyvät suojaavat myös datan eheyttä, kun vain asiantuntevat henkilöt pääsevät käsittelemään tietokannassa olevaa dataa. (Crystal Reports 8.5 2001: 618)

Koska näkyvät rajaavat ja järjestävät käytettävissä olevia tietoja, saadaan niitä käyttämällä aikaan myös säästöjä, kun tehokkuus kasvaa ja tukikustannukset pienenevät. Säästöjä seuraa myös siitä, että näkymiä käytettäessä kannan päivitysten vaikutus raporttiin on yksinkertaisesti hoidettavissa. Tällöin vain näkymätiedosto täytyy päivittää vastaamaan muuttunutta kantarakennetta tai uutta tietokantaa. (Crystal Reports 8.5 2001: 618)

Natiivi tietokantayhteys

Nopein tapa datan lukemiseksi kannasta on ottaa suora yhteys tietokantaan eli niin sanottu natiivi tietokantayhteys. Se on myös helppo tapa yhteyden muodostamiseen, koska käyttäjän tarvitsee vain valita raportissa käytettävät tietokantataulut. Tämän jälkeen Crystal Reports lukee tiedot suoraan kannasta. Suoran kantayhteyden huono puoli on se, että raportissa ei voida käyttää kuin yhdentyyppisiä tietokantoja. (Crystal Reports 8.5 2001: 575)

ODBC

Useat tietokannat tukevat ODBC-standardia ja sitä käytettäessä voidaan myös Crystal Reportsissa käyttää monia datalähteitä yhden rajapinnan kautta. Kun käytetään ODBC-rajapintaa yhteyden muodostamiseksi tietokantaan, Crystal Reports käyttää SQL-standardia datan hakemiseksi kannasta. Kun kehitysympäristössä luodaan raporttia, valmistaa Crystal Reports taustalla raporttia vastaavan SQL-kyselyn. Kun raportti ajetaan esikatseluun, lähettää Crystal kehittämänsä SQL-kyselyn ODBC-ajurille. Ajuri prosessoi kyselyn kantaan ja palauttaa vastauksena tulosjoukon, jota Crystal Reports vielä prosessoi. (Crystal Reports 8.5 2001: 575.)

Käytettäessä ODBC-yhteyttä Crystal Reportsilla tehdyn raportin voi kääntää käyttämään myös muuta tietokantaa (Crystal Reports 10 Jatkokurssi 2004: 39). Tämä edellyttää, että uuden tietokannan rakenne vastaa alkuperäisen kannan rakennetta. Ominaisuus on hyödyllinen, jos yrityksellä on esimerkiksi toimintaa useassa eri maassa ja maiden tietokannat ovat erilliset, mutta identtiset. Tällaisessa tapauksessa yhden maan kannasta tehty raportti voidaan kääntää käyttämään myös muiden maiden kantoja, jolloin koko raportin tekemiseen kuluu aikaa vain yhden raportin tekemisen verran.

Koska Crystal Reports käyttää SQL-kieltä kommunikoidessaan ODBC-rajapinnan kanssa, voidaan ODBC-rajapinnalle lähetettäviä SQL-lauseita muokata. SQL-teknologia takaa myös sen, että datan prosessointi on tehokkaampaa ja nopeampaa, koska suurin osa prosessoinnista tehdään jo palvelimella. Käytettäessä SQL-teknologiaa Crystal Reportsiin tuodaan siis vain ne tiedot, jotka vastaavat SQL-kyselyä. (Crystal Reports 8.5 2001: 602)

OLE DB

Crystal Reportsissa on tuki myös OLE DB -yhteydelle. Se on ODBC:n tapaan rajapinta, jonka avulla päästään käsiksi sellaisten datalähteiden tietoihin, jotka tukevat OLE DB:tä (Crystal Reports 8.5: 602). OLE DB eroaa ODBC:stä siten, että se sallii yhteydet useampiin erilaisiin tietolähteisiin kuten tekstitiedostoihin, sähköpostiin ja laskentataulukoihin (Crystal Reports 10 Jatkokurssi 2004: 39).

5.1.2 Integrointi omiin ohjelmistoihin

Crystal Reportsin raportointiominaisuuksia voidaan hyödyntää omissa sovelluksissa ja siinä on tuki yleisimmille ohjelmointikielille, kuten Javalle, C++:lle, Delphille ja Visual Basicille. Crystal Reportsin versiossa 10 on Java -raportointikomponentti, jonka avulla sovelluksessa voidaan luoda raportteja. Komponentti tukee Crystal Reportsin perustoimintoja, kuten lajittelua ja muotoilua sekä raporttien kääntämistä muun muassa pdf ja word -muotoihin. Komponentti mahdollistaa myös integroinnin kehitysympäristöihin, kuten Borlandin JBuilder 10:neen. Java-tageja käyttämällä voidaan raporttien katseluohjelmisto lisätä omiin sovelluksiin. (Crystal Reports 10 2003: 12)

Crystal Reportsilla luotuja raportteja voidaan hyödyntää myös omissa verkkosovelluksissa. Sovelluksesta voidaan kutsua Crystal Reportsin verkkopalvelinohjelmistoa, joka suorittaa toimenpiteet määrätyn raportin näyttämiseksi ohjelmiston käyttäjälle.

5.1.3 Kaavojen käyttö raporttien muodostamisessa

Raporttiin haettavan tiedon määrää halutaan usein rajata. Crystal Reportsissa on graafinen työkalu rajausten tekemiseen. Myös omien kaavojen, eli niin sanottujen formuloiden, luominen on mahdollista. (Crystal Reports 10 Jatkokurssi 2004: 51–55.)

Formula on Crystal Reports -termi, joka voidaan määritellä symboliseksi ilmaukseksi sellaisista tiedoista, jotka halutaan esittää tulostettavassa raportissa eri tavalla kuin ne ovat tallennettuna tietokantaan. Lisäksi formulalla voidaan luoda aivan uusia tietokenttiä tietokannan tietojen pohjalta. Formuloiden tekemisessä käyttäjä voi hyödyntää joko Crystal Reportsin omaa syntaksia tai Visual Basic -syntaksia. (Crystal Reports 10 Jatkokurssi 2004: 51–55)

Crystal Reports -formulakenttä on laskennallinen kenttä, jonka näyttämä tieto perustuu formulaan liitetyn kaavan lopputulokseen. Lopputulos voi olla tyypiltään jokin seitsemästä tietokantakentän tyypistä. Yksinkertainen esimerkki formulasta on etunimen ja sukunimen yhdistäminen samaan kenttään kaavan avulla. (Crystal Reports 10 Jatkokurssi 2004: 51–55)

5.1.4 Kielten ja merkistöjen tuki

Raportteja voidaan luoda lähes kaikilla kielillä. Jos halutun kielen tukea ei löydy valmiina, se voidaan asentaa erikseen. Crystal Reportsissa on myös Unicode-tuki, joka mahdollistaa raporttien tulostuksen kaikilla merkistöillä. (Crystal Reports 10 2003: 443)

5.1.5 Raporttityypit

Crystal Reportsilla voidaan luoda useita erityyppisiä raportteja. Sillä voidaan esimerkiksi tulostaa osoitetarroja ja luoda lomaketyylisiä raportteja. Seuraavassa on käyty tarkemmin läpi muutamia tärkeimpiä raporttityyppejä.

Ehdollinen raportointi

Ehdollisen raportoinnin avulla Crystal Reportsissa on mahdollista informoida käyttäjää merkittävistä seikoista muuttamalla kenttien väriä, reunaviivoja tai tekstin kirjasinta kentän arvon mukaan. Ehdollisen raportoinnin avulla voidaan esimerkiksi korostaa tietyn myyntimäärän ylittymistä muuttamalla kyseinen sarake punaiseksi ja näin informoida käyttäjää. Ehdollinen raportointi toteutetaan formuloilla. (Crystal Reports 10 Jatkokurssi 2004: 68–70)

Ristiintaulukointiraportit

Ristiintaulukointiraportti esitetään nimensä mukaisesti taulukon muodossa. Ristiintaulukoinnissa arvot on summattu kahdessa suunnassa. Crystal Reportsissa ristiintaulukointiraportit tehdään Cross-Tab -objektin avulla. (Crystal Reports 10 2003: 279.)

Drill down -raportit

Raportissa olevaan tietoon voidaan porautua syvemmälle (drill down), jos halutaan nähdä tiedot, joiden pohjalta yksittäinen ryhmä tai yhteenvetotieto (summary) on muodostettu. Porautuminen säästää tilaa raportissa ja tekee siitä selvemmän, kun kaikkia ryhmän tietoja tai yhteenvetotiedon perusteena käytettyjä tietoja, ei tarvitse tulostaa kerralla. Käyttäjä voi porautumisen avulla tutkia yksityiskohtaisemmin juuri niitä tietoja joita haluaa valitsemalla haluamansa ryhmäotsikon tai yhteenvetotiedon lähempään tarkasteluun. Myös raportissa käytettyjen graafisten kaavioiden ja karttojen pohjana oleviin tietoihin voidaan porautua syvemmälle. (Crystal Reports 10 2003: 86,105,235,251)

Aliraportit

Aliraportti on raportti toisen raportin sisällä. Pääraportit voivat sisältää aliraportteja joko niin, että ne on linkitetty pääraportin tietoihin tai siten, että ne ovat itsenäisiä raportteja. Aliraportti ei kuitenkaan voi esiintyä ilman pääraporttia. Koska aliraportti luodaan ensin itsenäisenä raporttina, se voidaan samalla tallettaa myös omana tiedostonaan ja näin käyttää myös erillään pääraportista. Aliraportti voidaan sijoittaa pääraporttiin myös siten, että se aukeaa tarvittaessa linkistä. Aliraportit tallennetaan raporttiobjekteina. (Crystal Reports 10 2003: 389.)

OLAP -raportit

Relaatiotietokannat ovat yleisin datalähde, mutta myös OLAP-tietovarastot (Online Analytical Processing) ovat yleistymässä. Crystal

Reportsissa on mahdollista käyttää myös OLAP-tietokuutioita datalähteinä. Silloin raporttityyppinä käytetään OLAP-raporttia. Se muistuttaa ristiintaulukointia, mutta siinä on huomioitu datalähteen erityisominaisuudet. (Crystal Reports 10 2003: 297.)

5.1.6 Interaktiiviset ominaisuudet

- Parametroitavuus** Myös raportin loppukäyttäjän on mahdollista rajata raporttiin tulostettavia tietoja antamalla yksilöiviä hakuehtoja niin sanottu parametrikenttään. Parametrille voidaan syöttää oletusarvoja ja se voidaan poimia valintalistalta. Parametrikentät antavat siis raportin ajajalle mahdollisuuden määrittellä hakuehtoja ja saada näin raportti haluamillaan ehdoilla. (Crystal Reports 10 2003: 373.)
- Hälytysrajat** Hälytysrajojen avulla voidaan muun muassa määrittellä raja-arvo, jota seurataan ja jonka ylitys aiheuttaa muokatun viestin ilmestymisen raporttiin. Hälytysraja on raporttikohtainen ja hälytysteksti muokattavissa. (Crystal Reports 10 User's 2003: 337)

5.1.7 Raporttien muotoilu

Crystal Reportsiin voidaan asettaa yleiset asetukset, joilla voidaan yhdenmukaistaa kaikkia ohjelmalla luotavat raportit. Lisäksi aktiivisena olevan raportin asetuksia voidaan muokata erikseen ja riippumatta yleisistä asetuksista.

Raportti koostuu lohkoista. Lohkoja on aina vähintään viisi: ensimmäisen sivun ylätunniste (otsikkotiedot), ylätunniste (kenttien otsikot), rivitiedot, raportin alatunniste (esim. loppusummat) ja sivun alatunniste (esim. sivunumerointi). Lohkoja voidaan muotoilla ja lisätä, mutta edellä mainittuja pakollisia lohkoja ei voida poistaa. Ne voidaan kuitenkin piilottaa. (Crystal Reports 10 2003: 173)

Raportin ulkoasua voidaan muotoilla monin tavoin. Yksittäisten kenttien muotoilun lisäksi voidaan muokata objektien reunaviivojen ja tekstin tyyliä sekä taustan väriä. Raporttiin voidaan tietyille objektille lisätä myös linkkiominaisuus, esimerkiksi internetsivuille tai haluttuun tiedostoon. (Crystal Reports 10 2003: 189)

- Tietokentät** Crystal Reports mahdollistaa seitsemän eri tietokantakenttätyyppin käytön: teksti-, numero-, valuutta-, joka on numerotyyppisen kentän erikoismuoto, päivämäärä- ja aika, päivämäärä-, aika- sekä totuusarvotyyppit.

Raporttiin voidaan lisätä tietokannasta haettavien tietokenttien lisäksi muun muassa tekstikenttiä ja kuvatiedostojen lisäämiseen tarkoitettuja

kenttiä. Tekstikenttiin voidaan lisätä tietokenttiä myös tekstin lomaan. Näin esimerkiksi kirjoitettaessa suoramarkkinointikirjeitä, tekstin sekaan voidaan lisätä kirjettä personoivia asioita, kuten nimi tai syntymäpäivä. Tekstikenttien ominaisuuksia voidaan muokata muun muassa muuttamalla sisennystä tai riviväliä. (Crystal Reports 10 2003: 79–84.)

Raporttiin voidaan lisätä myös niin sanottuja erikoiskenttiä, jotka eivät ole sidottuja tietokantakenttiin. Erikoiskenttiä voivat olla muun muassa sivunumeroinnit ja päiväykset. (Crystal Reports 10 2003: 79–84.)

Crystal Reportsilla tehdyn raportin kenttiä voidaan muotoilla monipuolisesti. Kenttiä voidaan piilottaa, niiden nimiä voidaan vaihtaa, käytetylle muotoilutyylille voidaan antaa nimi, kenttiä voidaan säätää ryhmittymään oikealle, vasemmalle tai keskelle, tekstin kallistusta voidaan säätää ja raportissa toistuvat peräkkäiset arvot voidaan piilottaa. (Crystal Reports 10 2003: 189)

Kenttää on mahdollista muokata myös sen tietotyypin perusteella. Numerokenttien kohdalla voidaan esimerkiksi säätää desimaalien määrää tai lisätä valuuttamerkki. Päiväyksen esitystapaa voidaan muuttaa ja boolean-tyyppisten kenttien kohdalla voidaan määritellä, kuinka totuusarvo ilmaistaan. (Crystal Reports 10 2003: 189)

Tietojen lajittelu ja ryhmittely

Tieto voidaan ryhmitellä merkitseviin ryhmiin ja lajitella. Crystal Reportsissa tietoja voidaan lajitella tai ryhmitellä eri tietokenttien perusteella. Kun raportti on ensin ryhmitelty tietyn kentän mukaiseen järjestykseen, itse ryhmittelyn sisällä voidaan tietoja vielä lajitella. Ryhmä on ensisijainen lajittelutekijä, mutta lajittelu voidaan siis suorittaa myös ryhmän sisällä. Raportti voi olla esimerkiksi ryhmitelty maiden nimien mukaan aakkosjärjestykseen, mutta ryhmien sisällä kaupungit voidaan vielä järjestää aakkosjärjestykseen. (Crystal Reports 10 2003: 47–49.)

Lajittelu voidaan tietyn tekijän perusteella tehdä laskevaan, nousevaan alkuperäiseen tai määriteltyyn järjestykseen. Alkuperäiseen järjestykseen järjestettäessä tiedot tulostetaan raporttiin siinä järjestyksessä, kun ne on tietokantaan tallennettu. Järjestys voidaan myös erikseen määritellä raportin laatijan haluamalla tavalla käyttämällä itse määriteltyjä ryhmiä. (Crystal Reports 10 Jatkokurssi 2004: 16)

5.1.8 Kuvien ja kaavioiden käyttö raportissa

Crystal Reports -ohjelmaan on rakennettu valmiita esimääriteltyjä graafisia kaavioita, kuten esimerkiksi pylväs-, käyrä- ja piiraskaavioita.

Kaavioita voidaan muokata monipuolisesti niin ulkoasun kuin mukaan otettavan datankin osalta. (Crystal Reports 10 2003: 235)

Kaavioiden lisäksi raporttiin voidaan lisätä karttoja, joilla voidaan havainnollistaa esimerkiksi kansainvälisen yrityksen myyntiä maittain siten, että maiden värit määräytyvät myynnin mukaan. (Crystal Reports 10 2003: 251.)

Raporttiin voidaan lisätä myös kuvia, kuten yrityslogoja tai muita symboleita. (Crystal Reports 10 User's Guide 2003: 83.) Crystal Reports tukee useimpia yleisesti käytössä olevia kuvatiedostotyyppisiä. (Crystal Reports 10 Peruskurssi 2004: 32)

5.1.9 Raporttien jakelu

Crystal Reportsilla luotuja raportteja voidaan jakaa muille käyttäjille usein eri tavoin. Raporttien käyttäjillä ei välttämättä tarvitse olla Crystal Reportsia asennettuna koneelleen voidakseen käyttää raporttia. Käyttö olisi kuitenkin helpointa, jos Crystal Reports on asennettu. Muita tapoja raporttien jakamiseen ovat Web-jakelu ja muuntaminen toisiin tiedostomuotoihin, eli exporttaus. (Crystal Reports 10 2003: 313–321.)

Crystal Reports -tuotteisiin kuuluu myös monipuoliset suojaus-, ajastus- ja jakeluominaisuudet sisältävä erillinen palvelinohjelmisto. Sen avulla raportteja voidaan jaella Internetin tai Intranetin välityksellä, eli niin sanottuna Web-jakeluna. Web-jakelua käytettäessä käyttäjä tarvitsee vain selainohjelmiston ja raportteja voidaan katsella useilla eri katseluohjelmilla (Viewer). (Crystal Reports 10 2003: 313–321.)

Kun käytetään Web-jakelua, raportissa voidaan näyttää joko reaaliaikaista, eli tietokannasta juuri raportin ajohetkellä haettua tietoa, tai raportin luomisen yhteydessä tallennettua tietoa. Jos raportissa olevat tiedot ovat sen kaltaisia, että niitä ei päivitetä kovin usein, kannattaa raportissa käytetyt tiedot tallentaa raporttiin, koska näin raportin latautuminen on huomattavasti nopeampaa. Jos taas raportoidaan sellaisia tietoja, jotka päivittyvät useasti, on hyödyllisempää käyttää reaaliaikaista tietoa. Tallennetut tiedotkin voidaan asettaa päivittämään tietokannasta automaattisesti halutuun väliajoin. (Crystal Reports 10 2003: 313–321.)

Exporttaus

Raportin sisältämä data voidaan muuntaa myös muihin tiedostomuotoihin tiedon myöhempiä käsittelyä varten. Crystal Reportsin tukemia tiedostomuotoja ovat:

- Adobe Acrobat (pdf)
- Comma-separated values (csv)
- Crystal Reports 7 (rpt)
- Data Interchange Format (dif)

- MS Excel (xls)
- HTML
- Rich Text Format (rtf)
- Teksti
- MS Word (doc)
- XML

(Crystal Reports 10 2003: 315–321.)

Datan siirtämisen yhteydessä menetetään suuri osa muotoiluista, sekä Crystal Reportsin ominaisuuksista. Menetettäviä ominaisuuksia ovat muun muassa porautuminen (drill down) ja hälytysrajojen asetukset. Menetettyt ominaisuudet ovat riippuvaisia tiedostomuodosta, johon exporttaus tapahtuu. Exportatun tiedoston tiedot eivät myöskään päivity. (Crystal Reports 10 2003: 315–321.)

Exporttaus voi tapahtua myös eri kohteisiin, kuten

- Application (datan siirto suoraan siihen sovellukseen, joka on kytketty valittuun tiedostomuotoon)
- Disk file (datan tallennus käyttäjän valitsemaan tiedostoon)
- Microsoft Exchange folder ja Lotus Domino (datan tallennus julkisiin kansioihin)
- Lotus cc:Mail (VIM) ja Microsoft Mail (MAPI) (datan lähetys halutussa muodossa sähköpostiliitteenä)

(Crystal Reports 10 2003: 315–321.)

5.2 JasperReports

JasperReports on kokonaan Javalla tehty luokkakirjasto raporttien luomiseen. JasperReports-kirjasto käyttää XML-raporttimallia raporttien generoimiseen. Kirjastoa voidaan käyttää monissa Java-sovelluksissa, kuten esimerkiksi J2EE- ja Web-sovelluksissa. JasperReports on open source -ohjelmisto, jota voi käyttää ilmaiseksi tarvitsematta maksaa kalliita lisenssejä kehittäjälle.

JasperReportsin käyttöön ja raporttien tekemiseen tarvitaan seuraavat ohjelmat koneelle asennettuna:

- JDK 1.2.2 tai uudempi
- JAXP 1.1 XML Parser (Jakarta Commons Digester Component (versio 1.1 tai uudempi)
- Jakarta Commons BeanUtils Component (versio 1.1 tai uudempi)
- Jakarta Commons Collections Component (versio 1.0 tai uudempi)
- Jakarta Commons Logging Component (versio 1.0 tai uudempi))
- JDBC 2.0 ajuri
- iText–Ilmainen Java-PDF -kirjasto (versio 1.01 tai uudempi)

- Jakarta POI (versio 2.0 tai uudempi XLS)
(Danciu, JasperReports – Requirements 2005.)

5.2.1 Datayhteydet

JasperReports tukee JDBC-yhteyksiä (Java Database Connectivity) ja neljää erityyppistä tietovarastoa riippuen käytettävästä käyttöliittymästä. JDBC on universaali ja toimittajariippumaton standardi, jonka avulla Java-sovellukset voivat toimia SQL-tietokantojen (Structured Query Language) kanssa. JasperReports tukee monia eri tyyppisiä datalähteitä käyttämällä erityistä rajapintaa. Rajapinta mahdollistaa minkä tahansa relaatio- eli RDMS -tietokannan (Relational Database Management System) käytön JDBC:n kautta. (Danciu, JasperReports – Tutorial 2005.)

JasperReportsin käyttöä varten suunniteltu iReport-käyttöliittymä voidaan konfiguroida myös ODBC-ajurille (Open Database Connectivity). Yhteyden muodostamiseksi tarvitaan kaksi tietokantayhteyttä: toinen Microsoft-yhteensopiva ja toinen Java Sun -yhteensopiva. Ensimmäistä käytetään iReportin sisällä ja toista iReportin ulkopuolella kääntämistä ja esikatselua varten. (iReport 2005)

iReport- sekä JasperAssistant-käyttöliittymät tukevat tietokantojen lisäksi kolmea eri datalähdettä. Tyhjällä datalähteellä voidaan tehdä nopea testi raporttiin. Testillä saadaan selville, ovatko asetukset kunnossa ja toimivia. Custom JRDataSource (JavaBeans) on mekanismi, joka antaa plug-in:n avulla mahdollisuuden käyttää tilaustyönä tehtyjä datalähteitä, joita käyttöliittymä ei muuten suoraan tue. Yleensä tällaiset datalähteet vaativat erityisen Java-alustetun koodin. XML-datalähde kopioi tarpeellisen tiedon XML-dokumentista. Mahdollisuutta käyttää XML-tiedostoja tai JavaBeans-komponentteja datalähteenä ei ole vielä implementoitu iReportiin. iReportiin mahdollisesti integroitavia tietokantoja ovat muun muassa MySQL, Microsoft SQLServer, DB2 ja Oracle. (JasperAssistant 2005, iReport 2005)

Tietokannasta haetaan tarpeellinen tieto käyttämällä SQL-kyselyä tai kutsumalla proseduuria. Tällainen tietokanta-datalähde vaatii tietokantaspesifisen JDBC-ajurin tietokantaan yhteydenottoa varten. (JasperAssistant 2005).

5.2.2 Integrointi omiin ohjelmistoihin

JasperReports kirjastoja voidaan käyttää omissa Java-sovelluksissa kuten mitä tahansa muita Java-kirjastoja. JasperAssistant-käyttöliittymän integrointi Eclipse-kehitysympäristöön helpottaa raporttien integrointia ohjelmistoprojekteihin.

5.2.3 Kaavojen käyttö raporttien muodostamisessa

Expressions eli kaavat ovat tärkeä ominaisuus JasperReportsissa. Niitä voi käyttää raportin useita laskutoimituksia suorittavien muuttujien esittelyyn, datan ryhmittelyyn raportissa, raportin tekstikenttien sisällön määrittelyyn tai objektien ulkoisten seikkojen uudelleenmuokkaamiseen. Pohjimmiltaan kaikki raportin kaavat ovat Java-ilmentymiä (expressions), jotka voivat viitata raportin kenttiin ja muuttujiin (Beumer 2005). Tyypillinen kaava on esimerkiksi etunimen ja sukunimen yhdistäminen. (Danciu, JasperReports – Tutorial 2005.)

5.2.4 Eri kielten ja merkistöjen tuki

JasperReportsissa on Unicode-merkistöstandardituki fonteille. Tämä mahdollistaa raporttien tekemisen millä merkistöllä tahansa. JasperReportsin uusin versio mahdollistaa myös Java- ja PDF-kirjasimien yhdistämisen. Raportteja voi tehdä eri kielillä JasperReportsin kielituen avulla. Ulkomaisten merkkien tuki on liitetty iText-kirjastoon. iText on ilmainen Java-PDF -kirjasto. (Danciu, JasperReports – Tutorial 2005.)

Viestien muokkaamiseen eri kielillä on myös oma metodinsa, ja kirjastossa säilyy tieto myös tekstin lukusuunnasta. Näin esimerkiksi arabiaa ja hepreaa voidaan tulkita oikein. (Danciu, JasperReports – Tutorial 2005.)

5.2.5 Raporttityypit

Ristiintaulukointiraportit

Tulevaisuudessa JasperReportsilla on mahdollista tehdä ristiintaulukointiraportteja, mutta ominaisuus ei ole vielä käytössä.

Aliraportit

JasperAssistantissa on mahdollista tehdä aliraportteja. Aliraportit mahdollistavat monimutkaisten raporttien luomisen ja samalla yksinkertaistavat suunnittelua. Ne antavat myös mahdollisuuden raportin osioiden uudelleenkäyttöön. Sisäkkäiset raportit vaativat yleensä monia parametreja. Nämä parametrit voidaan siirtää käyttämällä Parameters Map Expression -ominaisuutta. (JasperAssistant 2005.)

Drill down -raportit

JasperAssistantissa voi tehdä drill down -raportteja. Hyperlinkkejä voidaan käyttää linkkien laittamiseen raportin sisään. Lisäksi voidaan käyttää linkkejä, jotka liittävät raportin muihin ulkoisiin raportteihin. (Eclipse... 2005).

5.2.6 Interaktiiviset ominaisuudet

Hälytysrajat JasperReports mahdollistaa niin sanottujen hälytysrajojen tekemisen. Tekstikentän tietoa ei voida muuttaa dynaamisesti, joten ratkaisuna voidaan laittaa samaan paikkaan kaksi tekstikenttää, toinen toisen päälle. Molemmat näyttävät saman arvon, mutta toinen mustalla ja toinen punaisella värillä. Punainen kenttä näkyy ainoastaan, kun arvo on esimerkiksi yli 100 ja musta kenttä kaikissa muissa tapauksissa. (Danciu, JasperReports – Tutorial 2005.)

5.2.7 Raporttien muotoilu

Tietokentät Kenttiin haetaan tietoa datalähteestä luomalla rutiineja. Kun käytetään ResultSetiä (tulosjoukkoa) datalähteenä, pitää tällöin kaikilla kentillä olla samat nimet kuin sarakkeilla, joita ne vastaavat sekä yhteensopiva tietotyyppi. Jos esitellään kenttä, jolla ei ole vastaavan nimistä saraketta tulosjoukossa, ajon aikana tulee poikkeus. Tulosjoukossa olevat sarakkeet, joilla ei ole vastinkenttiä raportissa, eivät vaikuta raportin täyttöön, mutta niitä ei myöskään voida hyödyntää raportissa.

JasperReportsin kenttä voi olla peräisin luokasta Object, Boolean, Byte, Date, Timestamp, Time, Double, Float, Integer, InputStream, Long, Short, BigDecimal tai String. (Danciu, JasperReports – Tutorial 2005.)

Lajittelu Raportissa tiedot voidaan lajitella lohkoilla sekä ryhmillä. Raporttipohja koostuu lohkoista. Lohkot ovat raportin osia, joilla on tietty korkeus ja leveys. Ne voivat sisältää raportin olioita, kuten viivoja, suorakulmioita, kuvia tai tekstikenttiä. JasperReportsin terminologiassa lohkoja kutsutaan myös nimellä Bands.

Ryhmät (groups) ovat joustava tapa järjestää dataa raportissa. Raportti voi määritellä lukuisia ryhmittelyitä. Ryhmät määrittelevät raporttiin kuuluvan datan visuaalisen ryhmittelyn. Kun ryhmä on kerran määritelty, sitä voidaan käyttää monessa paikassa raportissa. (JasperAssistant 2005)

5.2.8 Kuvien ja kaavioiden käyttö raportissa

Koska JasperReports-kirjasto ei tuota grafiikkaa ja kaavioita, voidaan käyttää muita erityisiä kirjastoja, joilla voi helposti tuoda kyseisiä elementtejä raporttiin. Esimerkkinä JFreeChart, joka on integroitu esimerkiksi iReportiin. Kuvia voidaan tuoda raporttiin esimerkiksi tiedostosta, web-osoitteesta tai tietokantakentästä. (JasperAssistant 2005)

5.2.9 Raporttien jakelu

Raportteja voidaan kääntää moniin eri tiedostomuotoihin. JasperReports tukee seuraavia tiedostomuotoja: pdf, csv, xls, html ja xml.

JasperReportsissa voidaan määritellä raportin ulkoasu siten, että se pysyy muuttumattomana eri alustoilla. Pdf-formaatti tukee tätä ominaisuutta kaiken tekstin ja grafiikan osalta. Siksi pdf on niin suosittu ja paljon käytetty eri alustoissa. Muut formaatit, kuten esimerkiksi html tai xls eivät tue muuttumatonta layoutia. (Danciu, JasperReports-tips & tricks 2005.)

6 Käytettävyydesti: JasperReports – JasperAssistant

Käytettävyydesti tehtiin JasperReportsin JasperAssistant-käyttöliittymälle, koska työn etenemisen yhteydessä toimeksiantaja päätyi siihen, että JasperAssistant tulisi olemaan yrityksessä käytettävä JasperReportsin käyttöliittymä. Toimeksiantajan valinta perustuu siihen, että JasperAssistant soveltuu kahdesta tarkastelemastamme JasperReportsin käyttöliittymästä paremmin ohjelmistoprojektien raportointityökaluksi.

Crystal Reportsin käytettävyys on työn kannalta toissijaista, koska se on tällä hetkellä toimeksiantajan käytössä, eikä sitä ole tarkoitus kokonaan korvata uudella ohjelmistolla. Sen vuoksi sille ei tehty käytettävyydestiä.

Testin järjestäminen ja testaussuunnitelman laatiminen

Tutustuimme testattavaan käyttöliittymään tarkemmin laatimalla itse koeraportin. Sen jälkeen laadimme käytettävyydestistä testaussuunnitelman. Suunnitelmassa määriteltiin testin tavoitteet, hankittavalle tuotteelle asetetut käytettävyyksivaatimukset ja testattavat ominaisuudet sekä testihenkilöiden määrä. Lopuksi laadimme testitehtävät.

Testin tavoitteet Testin tavoitteena oli arvioida JasperReportsin JasperAssistant-käyttöliittymän yleistä käytettävyyttä sekä erityisesti sen opittavuutta tulevien käyttäjien näkökulmasta. Testeissä havaittuja ongelmia ei siis varsinaisesti luokiteltu niiden vakavuusasteen mukaan, koska tarkoituksena oli arvioida ohjelmiston yleistä soveltuvuutta toimeksiantajan käyttöön.

Käytettävyyksivaatimukset

Testattavan tuotteen käyttöalueita toimeksiantajalla tulevat olemaan ohjelmistoprojektien raportointiosuuksien toteuttaminen ja asiakkaiden tilaamien raporttien laadinta. Käyttöliittymän ja raportointikirjastojen käyttäjiä olisivat sekä ohjelmointitehtävissä pääsääntöisesti toimivat työntekijät että raportointiasiantuntijat. Näin ollen käyttöliittymän termien ja ohjelmiston toiminnan pitäisi olla niin ymmärrettäviä, että ohjelmisto soveltuisi molempien käyttäjäryhmien työvälineeksi.

Testattavat toiminnot

Suunnittelimme testitehtäväksi (Liite 1) pienen perusraportin luomisen. Raportti luotiin MySQL:llä toteutetusta testikannasta. Kantayhteyden muodostaminen ei kuitenkaan kuulunut testiin. Testissä luotiin uusi raportti, valittiin käytettävä tietokanta sekä siitä käytettävät taulut ja niistä käytettävät tietokentät. Raporttiin lisättiin lohkoja, tekstikenttiä ja

kaavoja sekä muotoiltiin raportin ulkoasua. Lisäksi testihenkilön piti tallentaa ja esikatsella raportti sekä määrittää muoto, jossa se haluttaisiin tulostaa.

Testikäyttäjien valinta

Valitsimme kaksi testihenkilöä suorittamaan testin. Valitut henkilöt ovat toimeksiantajan palveluksessa. Toinen heistä on ohjelmistoalan ja toinen raportointialan asiantuntija. Molemmat testihenkilöt ovat siis JasperAssistantin potentiaalisia tulevia käyttäjiä. He ovat myös luonteeltaan peruspuheliaita, ”mikä auttaa henkisen kuormituksen kasvaessa testin aikana” (Sinkkonen ym. 2002: 309). Kumpikaan testihenkilöistä ei ollut aikaisemmin käyttänyt JasperAssistantia.

Pilottitesti

Päätimme järjestää pilottitestin, koska testattava ohjelmisto oli myös meille suhteellisen uutuus. Pilottitestin avulla saimme vielä korjattua muutamia epäselvästi ilmaistuja testitehtäviä sekä parannettua testin ulkonäköä siten, että testitehtävissä eteneminen helpottui.

Testien suoritus

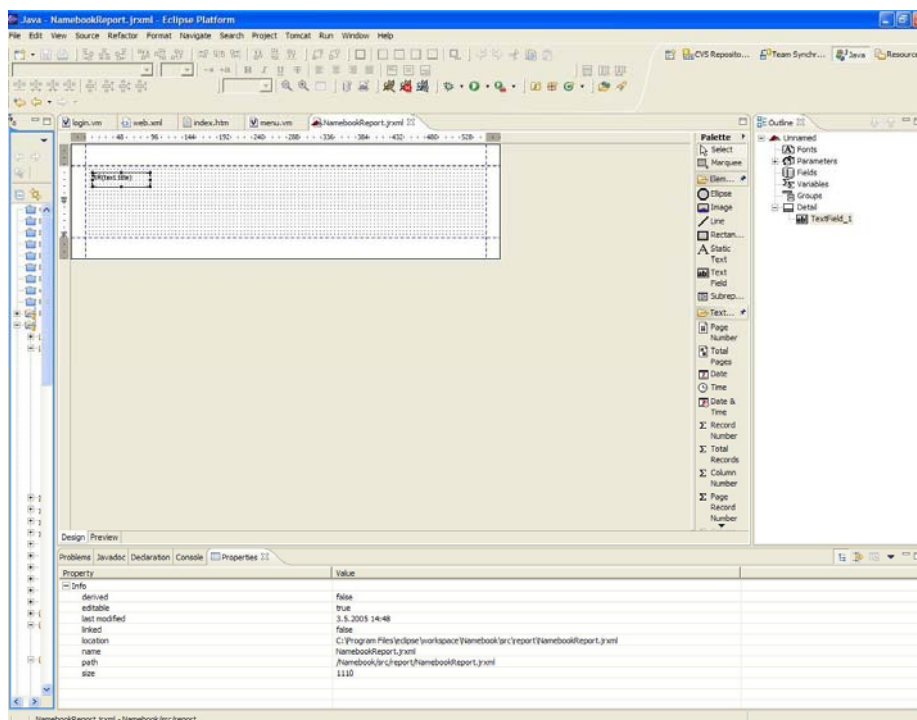
Pilottitestissä ilmenneiden puutteiden korjaamisen jälkeen oli vuorossa varsinaisten testien järjestäminen. Testit järjestettiin normaalissa toimistoympäristössä toimeksiantajan tiloissa työpäivän aikana. Testiolosuhteet olivat siis mahdollisimman lähellä oikeita työskentelyolosuhteita.

Testien tulokset

Testien tulokset vahvistivat omia näkemyksiämme testattavasta ohjelmistosta. JasperAssistantin peruseräraportointinäkö oli molemmille testihenkilöille uutuus, joten heiltä kului jonkin verran aikaa ennen kuin näkymän eri alueet selkenivät. Ohjelmistoalan asiantuntijalle perusnäkö ilman JasperAssistantin erikoistoimintoja oli kuitenkin tuttu Eclipsen käyttökokemuksen vuoksi.

JasperAssistantin käyttöliittymässä (Kuva 3) on monia eri toimintojen pikapainikkeita ja muuta sisältöä. Käyttöliittymä toimii Eclipsen päällä, joten siinä on samaan aikaan näkyvissä myös paljon raportointiin liittymättömiä komponentteja. Käyttöliittymän ulkoasu ei siis noudattele Nielsenin heuristiikoissa esitettyä yksinkertaisen dialogin vaatimusta.

JasperAssistant-näkö tuntui testikäyttäjistä sekavalta ja se hankaloitti selvästi etenkin raportointiasiantuntijan selviytymistä testitehtävästä. Molemmilla testihenkilöillä oli ongelmia esimerkiksi Ominaisuusnäkö löytämisessä, mutta kun näkö oli kerran löytynyt, sen käyttö sujui luontevasti. Siltä osin ohjelmiston toiminta olikin yhdenmukaista, kun lähes kaikki toiminnot pystyttiin suorittamaan samasta paikasta.



Kuva 3. Eclipsen JasperAssistant-näkymä.

Ohjelmisto noudatteli graafisten käyttöliittymien standardeja perustoimintojen suorittamisen osalta. Standardien noudattaminen minimoi käyttäjän muistintarvetta, kun perustoiminnot voidaan suorittaa samalla tavalla kuin monissa muissakin ohjelmistoissa. Standardeiksi muodostuneiden valikkojen ja pikatoimintojen käyttö lisää ohjelmiston opittavuutta ja yhdenmukaisuutta.

Perustoiminnoista muun muassa uuden raportin luominen ja tallentaminen löytyivät JasperAssistantissa samoilta tutuilta paikoilta kuin muissakin graafisissa käyttöliittymissä, joten näiden toimintojen löytäminen ei ollut testihenkilöiden mielestä hankalaa. Hiiren oikealla näppäimellä saa myös JasperAssistantissa pikavalikon auki monien toimintojen suorittamiseksi. Testihenkilöt kokeilivatkin sitä nopeasti ja löysivät sen kautta monet tarvittavat työkalut tehtävien suorittamiseksi.

Ohjelmistossa ei ollut kaikilta osin käytetty raportointialan perustermejä, joten käyttöliittymässä käytetty kieli ei joissakin kohdissa vastannut käyttäjien kieltä. Vaikka pilottitestin jälkeen lisäsimme tehtävänantoon JasperAssistantissa käytetyn englanninkielisen termin *Band* viereen suomennoksen *lohko* selventämään tehtävänantoa, jouduttiin molempia testihenkilöitä avustamaan lohkon lisäämisessä raporttiin. Osittain avun tarve johtui siitä, että tutussa Crystal Reportsissa käytetään lohkoista termiä *Section*.

Kenttätyyppien nimeäminen aiheutti testihenkilöissä hämmennystä, koska esimerkiksi kaavoja luodaan normaaleihin tekstikenttiin erityisten kaavakenttien sijaan. Myös tämä ongelma liittyy käyttäjän ymmärtämisen kielen käyttämiseen, koska sana *tekstikenttä* ei viestitä käyttäjälle, että kenttään voidaan syöttää myös kaavoja.

Virheiden ehkäisyä ei ollut kaikilta osin otettu ohjelmistossa huomioon riittävästi, ja raportointiasiantuntija kaipasikin esimerkiksi SQL-lauseen ja kaavan syöttöön syntaksin tarkistusta. Hän myös huomioi, että lohkojen nimien olisi hyvä olla jatkuvasti näkyvissä suunnittelunäkymässä, mikä puolestaan vähentäisi käyttäjän muistintarvetta.

Lisäksi raportointiasiantuntija huomioi, että esikatselunäkymästä puuttui pienten kenttien muotoilumahdollisuus. Tähän toimintoon tarvittava pieni, mutta tärkeä oikopolku helpottaisi ja tehostaisi huomattavasti raportin laatijan työskentelyä. Sen avulla esimerkiksi kenttien sijoittelua voisi helposti muuttaa tarvitsematta palata takaisin suunnittelunäkymään.

Oli mielenkiintoista todeta, että kaikki testihenkilöt, mukaan lukien pilottitestin suorittaja, joutuivat raportin luomisen aikana pohtimaan samoja ongelmakohtia. JasperAssitantista puuttuu vielä monia käyttäjän toimia helpottavia ja virheitä ehkäiseviä ominaisuuksia, joita pitemmälle kehitetyssä Crystal Reportsissa on. Samoin JasperAssistant-näkymän erillaisuus ja siinä esitetyn informaation määrä poikkeavat raportointiin keskittyvän Crystal Reportsin näkymästä. Nämä seikat hidastivat testihenkilöiden suoriutumista testistä. He saivat testin kuitenkin suoritettua loppuun ja arvioivat ohjelmistoa vaikkakin oudoksi niin kuitenkin opittavaksi.

7 Ohjelmistojen vertailu

7.1 Toiminnallisuus

Kartoitimme Crystal Reportsista keskeisimmät toiminnalliset ominaisuudet ja etsimme samoja ominaisuuksia myös JasperReportsista. Datiumissa on Crystal Reports -ohjelmasta paljon käyttökokemusta ja asiantuntemusta. Sen käytöstä järjestetään myös koulutusta asiakasyrityksille, joten tuotteesta löytyy paljon kirjallista materiaalia sekä asiantuntijoiden hiljaista tietoa.

Vastaavien ominaisuuksien kartoittaminen JasperReportsista oli haasteellista, koska siitä ei ole julkaistu varsinaista ohjekirjaa. Tieto piti koota monesta eri paikasta, jotka kaikki sijaitsivat Internetissä muun muassa käyttäjien keskustelu- ja kysymyspalstoilla. JasperReportsin toiminnallisten piirteiden arviointi perustui myös käyttöliittymään, jolla JasperReportsia käytetään.

Ohjelmistojen toiminnalliset ominaisuudet keräsimme taulukkoon (Liite 2), josta saa helposti muutamalla silmäyksellä tietää, löytyykö ohjelmistosta etsittyä ominaisuutta. Taulukkoon on merkitty rasti niiden toimintojen kohdalle, jotka löytyvät kyseessä olevasta raportointiohjelmasta. Jos ominaisuus ei ole yksiselitteinen, on viereen merkitty lisäselvitys.

Datayhteydet

Crystal Reports tukee useimpia tietokantoja ja yhteystapoja. CrystalReports ottaa yhteyttä ODBC:n, OLE DB:n tai natiiviyhteyden avulla, kun taas Java-pohjainen JasperReports käyttää JDBC:tä. JasperReportsiin saa konfiguroitua myös ODBC:n, mutta se tapahtuu hiukan monimutkaisemmin.

ODBC-yhteyden huonona puolena verrattuna natiiviyhteyteen on sen hitaus. Hitaus johtuu siitä, että kysely välitetään usean kerroksen kautta. Sen vuoksi natiiviyhteyden tuen puuttuminen on selvä puute JasperReportsin ominaisuuksissa.

Crystal Reportsissa on tietokantojen lisäksi tuki myös monille muille datalähteille. Tutkituista datalähteistä ainoastaan OLAP-kuutiot ovat sellaisia, joita JasperReports ei tue.

Raporttien integrointi omiin ohjelmistoihin

Oleellista toimeksiantajan kannalta on raporttien integrointimahdollisuus Java-ohjelmistoihin. Siihen JasperReports soveltuu Java-pohjaisuutensa ja kirjastomaisuutensa ansioista jopa paremmin kuin CrystalReports. Kehitystyötä helpottaa se, että JasperAssistant toimii Eclipsen päällä.

Kaavojen käyttö raporttien muodostamisessa

Kuvien ja kaavioiden lisääminen raporttiin on mahdollista molemmissa työkaluissa. Niiden käyttö on helppoa tehtäessä raportteja Crystal Reportsilla, mutta JasperReportsissa ei vielä ole tukea kuvien ja kaavioiden käyttämiseen raportissa. Niitä kuitenkin saadaan JasperReports-raporttiin käyttämällä erillistä luokkakirjastoa. Ainoastaan xls-tiedostoihin ei saada kuvia, koska exporttaus kyseiseen formaattiin käyttää Jakarta POI -kirjastoa, joka ei tue kuvia.

Kielien ja merkistöjen tuki

Crystal Reports tukee lähes kaikkia eri kieliä ja merkistöjä. JasperReportsissa on myös eri merkistöjen ja kielten tuki. Näin raportteja voidaan tehdä sujuvasti molemmilla ohjelmistoilla myös esimerkiksi arabiaksi tai kiinaksi.

Raporttityypit

Molemmissa raportointiohjelmissa on mahdollista tehdä viivakoodeja, aliraportteja sekä drill down -raportteja eli porautua tietoihin syvemmälle. Ainoastaan Crystal Reportsissa on mahdollisuus ristiintaulukointiin sekä osoitetarjosten tekemiseen. Ristiintaulukointi on JasperReportsissa kuitenkin jo suunnitteluasteella oleva ominaisuus.

Interaktiiviset ominaisuudet

Raportoinnissa tärkein interaktiivinen ominaisuus on parametroitavuus. Parametroinnin avulla käyttäjä voi halutessaan määrittellä raportin ajovaiheessa esimerkiksi päivämäärät, joiden väliin jäävien päivien tiedot halutaan raporttiin näkyviin. Crystal Reports tukee parametroitia, mutta käytettäessä JasperReportsilla luotuja raportteja, parametroiti ei ole mahdollista

Molempien ohjelmistojen raportteihin voi tehdä esimerkiksi hälytysrajoja. Käyttäjä voi niiden avulla määrittää tietyt rajat, joiden ulkopuolelle menevä määrä näkyy raportissa esimerkiksi punaisella tekstillä.

Raporttien muotoilu

Molemmat ohjelmat tukevat tietokantojen standarditietotyyppejä, vaikka tietotyyppien nimet hieman vaihtelevatkin ohjelman mukaan. Kummassakin ohjelmassa voidaan myös käyttää esimerkiksi tekstikenttiä, joiden lomaan lisätään tietokenttiä ja erikoiskenttiä kuten päivämääriä ja sivunumeroita. Lajittelu ja ryhmittely ovat molemmissa ohjelmissa mahdollista samoin periaattein.

Raporttien jakelu

Raportin kääntäminen toiseen tiedostomuotoon on mahdollista molemmissa ohjelmissa. Molemmat tukevat exporttausta xml-, pdf-, csv-, html- sekä xls-tiedostomuotoihin. Crystal Reportsista on mahdollista exportata raportteja lisäksi levykkeelle, sovellukseen, ODBC-tietokantaan, Lotus Notesiin sekä Microsoft Mailiin. Sähköpostiin exporttaaminen on oleellisen tärkeää, koska siten ajastetut raportit pystytään toimittamaan määriteltyjen henkilöiden sähköposteihin automaattisesti.

Crystal Reportsissa on exporttaamisen lisäksi mahdollisuus raporttien verkkojakeluun verkkopalvelinohjelmiston avulla. JasperReportsissa ei ole tätä mahdollisuutta.

7.2 Käytettävyys

JasperReportsin käyttöliittymiä ovat muun muassa iReport sekä JasperAssistant, joka on integroitu Eclipseen. Kyseisillä käyttöliittymillä on paljon eroja. iReport on helppokäyttöisempi, koska se on ulkoasultaan selkeämpi ja nopeammin opittavissa. JasperAssistantissa puolestaan on paljon toimintoja, jotka täytyy ottaa huomioon. Eclipseen on usein integroitu paljon muitakin ohjelmia, joten näytöllä on aina paljon raportoinnin kannalta ylimääräisiä ja usein myös raportin laatijaa häiritseviä toimintoja.

Toimeksiantajan päätöksen mukaan JasperReportsin kahdesta tutkitusta käyttöliittymästä testaukseen valittiin JasperAssistant. Sen termistö ja ulkoasu poikkeavat käytettävyydestin perusteella merkittävästi Crystal Reportsista. Edes raportointiasiantuntijalle ei käytettävyydestessään ollut paljonkaan apua aikaisemmasta vankasta raportointityökalujen käyttötaidostaan, vaan hän joutui opiskelemaan käyttöliittymän käyttöä testitehtäviä tehdessään.

Helpommin opittavan JasperReports-kirjastoa hyödyntävän raporttienluomisympäristön tarjoaisi iReport, mutta toistaiseksi toimeksiantaja ei ole ollut halukas ottamaan käyttöönsä kahta eri JasperReportsin käyttöliittymää.

Vaikka JasperAssistant onkin melko erilainen käyttöliittymä Crystal Reportsin käyttöliittymään verrattuna, se ei välttämättä ole käytettävyydeltään huono. Käyttöliittymien erot aiheuttavat kuitenkin sen, että uuden ohjelmiston käyttöönotosta tulee aiheutumaan jonkin verran kustannuksia koulutuksen vuoksi, koska ohjelmiston käytön opettelu ei suju varsinaista työtä tehdessä.

Crystal Reportsissa on paljon raporttien luomista helpottavia toimintoja, joita JasperAssistantista ei löydy. Voidaan siis olettaa, että varsinkin JasperReportsin käyttöönoton alkuvaiheessa raporttien luomiseen kuluu normaalia enemmän aikaa, mikä taas aiheuttaa lisäkustannuksia.

Käytettävyydestien suorittamisen, ohjelmistoon tutustumisen ja toimeksiantajan asiantuntijoiden kanssa keskustelun lopputuloksena arvioimme, että JasperReportsin käytön oppiminen JasperAssistant-käyttöliittymällä ei ole niin vaikeaa, että se estäisi käyttöliittymän ja JasperReports-kirjaston käyttöönoton.

8 Johtopäätökset

8.1 Ehdotus ohjelmistojen käytöstä tulevaisuudessa

Raportointiohjelmiston valinnassa täytyy ottaa huomioon paitsi käytettävissä olevat tietokantayhteydet myös toimintojen laajuus. Crystal Reports sisältää useita sellaisia toimintoja, joita JasperReports-kirjastossa ei vielä tällä hetkellä ole, kuten raportin loppukäyttäjän parametriensyöttömahdollisuus tai ristiintaulukointi.

Kahden tarkastelemamme JasperReportsin käyttöliittymän erot ovat selvät. iReport on JasperReports-kirjaston käyttöä helpottamaan suunniteltu visuaalinen raportinteko-ohjelma. Ohjelman tarkoituksena on tehdä raportteja, eikä siihen ole liitetty yhtään turhaa tai ylimääräistä toimintoa. Raportin teko sujuu ilman, että tarvitsee opetella monimutkaisia toimintoja, joten käyttöliittymän opittavuus on melko hyvä.

JasperAssistant sen sijaan toimii Eclipsen kehitysympäristössä siihen suoraan integroituna. Osittain siitä johtuu, että käyttöliittymän ulkoasu on monimutkainen. Erityisesti käyttäjän, joka ei tunne Eclipseä, voi aluksi olla hankala hallita ohjelmaa. JasperAssistantin etuina ovat muun muassa raportin yhdistäminen omaan koodiin, raportin muuttaminen kooditasolla sekä versionhallintamahdollisuus.

JasperReportsin merkittävimpiä puutteita ovat raportin ajajan parametrintimahdollisuuden ja ajastettujen raporttien automaattisen sähköpostijakelun puuttuminen. Nämä puutteet rajoittavat ohjelmiston käyttöä raportointiprojekteissa. JasperReportsin hyvänä ominaisuutena on kuitenkin integrointimahdollisuus omiin Java-ohjelmistoihin.

Koska omissa ohjelmistoissa voidaan edellä mainitut puutteet täydentää ohjelmallisesti, olemme päätyneet siihen, että JasperReportsia voitaisiin käyttää ohjelmistoprojektien ensisijaisena raportointityökaluna. Ohjelmistoprojektien raporttien luomisessa suositeltavin JasperReportsin käyttöliittymä kahdesta tutkitusta vaihtoehdosta on JasperAssistant, koska toimeksiantaja käyttää ohjelmistojen kehitysympäristönä pääsääntöisesti Eclipseä.

Sen sijaan töihin, joissa tehdään ainoastaan raportteja, ehdotamme projektin laajuudesta riippuen käytettäväksi joko Crystal Reportsia tai iReportia. Pienten raportointiprojektien toteuttamiseen soveltuu hyvin iReport, mutta laajemmissa projekteissa erityisesti JasperReportsin käyttöönoton alkuvaiheessa suosittelemme Crystal Reportsin käyttämistä. Käytettävän raportointiohjelmiston valintaan tulee vaikuttamaan kussakin projektissa myös asiakkaan toiveet ja käytettävästä raportointiohjelmistosta johtuvat kustannuserot.

8.2 Yhteenveto työn onnistumisesta

Työmme tarkoituksena oli selvittää sopisiko open source -ohjelmisto JasperReports vaihtoehdokseksi raportointiohjelmistoksi lisenssiltään kalliin Crystal Reportsin rinnalle. Vertailimme ohjelmistojen toiminnallisuutta toimeksiantajan kanssa määriteltyjen ominaisuuksien suhteen sekä testasimme JasperReportsin JasperAssistant-käyttöliittymän käytettävyyttä.

Mielestämme työ onnistui odotustemme mukaisesti. Saimme hyvin selvitettyä Crystal Reportsista toimeksiantajalle keskeiset ominaisuudet. Suurin työ tehtiin selvittäessä vastaavien ominaisuuksien löytymistä JasperReportsista, mutta sitkeän etsimisen ansiosta myös siinä onnistuttiin suhteellisen hyvin. Internetsivuja ja -artikkeleita lukemalla saimme vähitellen selville tietoa JasperReportsin ominaisuuksista. Näiden tietojen avulla harjoittelimme raporttien tekoa kahdella eri käyttöliittymällä ja järjestimme käytettävyytestin, jossa arvioitiin JasperAssistantin käytettävyyttä.

Selvitystyön perusteella pystyimme suosittelemaan Datiumille JasperReportsin käyttöä erityisesti ohjelmistoprojektien yhteydessä. Saimme myös selville JasperReportsin puutteet verrattuna Crystal Reportsiin sekä JasperReportsin käyttöliittymien eroja. Mielestämme nämä seikat ovat työn onnistumisen ”mittari”.

Lähteet

- Beumer, Gregory 2005. Starting with JasperReports technology. [online] [viitattu 19.3.2005].
amis.nl/blog/index.php?p=346
- Crystal Reports 8.5 User's Guide 2001. California, USA: Crystal Decisions, Inc.
- Crystal Reports 10 Jatkokurssi Kurssimateriaali 2004. Datium Oy.
- Crystal Reports 10 Peruskurssi Kurssimateriaali 2004. Datium Oy.
- Crystal Reports 10 User's Guide 2003. California, USA: Crystal Decisions, Inc.
- Danciu, Teodor 2005. JasperReports – Requirements. [online] [viitattu 27.3.2005].
jasperreports.sourceforge.net/requirements.html
- Danciu, Teodor 2005. JasperReports – tips & tricks. [online] [viitattu 25.3.2005].
jasperreports.sourceforge.net/tips.tricks.html
- Danciu, Teodor 2005. JasperReports – Tutorial. [online] [viitattu 23.2.2005].
jasperreports.sourceforge.net/tutorial/index.html
- Datium Oy – Tuotteet. [online] [viitattu 25.4.2005].
www.datium.fi/tuotteet.html
- Datium Oy – Yritys. [online] [viitattu 14.2.2005].
www.datium.com
- Eclipse Plugin Central 2005. [online] [viitattu 28.3.2005].
www.eclipseplugincentral.com/PNphpBB2+file-viewtopic-t-1315-sid-bd89f54bf29c2817565c5de7724a6fdd.html
- Hovi, Ari 1997. Data Warehousing – Tietovarastotekniikka. Espoo: Suomen Atk-kustannus Oy.
- Hovi, Ari, Ylinen, Jari & Koistinen, Heikki 2001. Tietovarastot liiketoiminnan tukena. Helsinki: Satku – Kauppakaari.
- iReport Manual 2005. [online] [viitattu 23.2.2005].
ireport.sourceforge.net/manual0.2.0.html
- Jasper Assistant User Guide 2005. [online] [viitattu 7.3.2005].
www.jasperassistant.com/docs/guide
- JasperReports – Home. [online] [viitattu 23.2.2005].
jasperreports.sourceforge.net

- Kuutti, Wille 2003. Käytettävyyden suunnittelu ja arviointi. Helsinki: Talentum.
- Lahtonen, Tommi 2003. Mikä on tietokanta? [online] [viitattu 15.3.2005].
appro.mit.jyu.fi/2003/syksy/tiedonhallinta/luennot/luento1/
- Nielsen, Jacob 1993. Usability Engineering. San Diego, USA: ACADEMIC PRESS.
- Nieminen, Marko ja Kuoppala, Hannu 1998. Standardinmukainen käytettävyys - tulevaisuuden vaatimus? [online] [viitattu 14.2.2005]. Sytyke ry - Systeemyö 4/98.
www.pcuf.fi/sytyke/lehti/kirj/st19984/19.pdf
- Sikanen, Samuli 2004. SAP Info 2004. [online] [viitattu 23.4.2005].
www50.sap.com/finland/sapinfo/4_4.asp
- Sinkkonen, Irmeli, Kuoppala, Hannu, Parkkinen, Jarmo ja Vastamäki, Raino 2002. Käytettävyyden psykologia. Helsinki: Edita.
- Suoranta, Veli-Matti. Tietovarastointi Data Warehouse. [online] [viitattu 15.3.2005].
www.cs.helsinki.fi/u/suoranta/dw_public/dokumentit/Taloudswohje4.htm
- Talk, Chuck 2005. An Interview with Teodor Danciu of JasperReports. [online] [viitattu 14.3.2005]. www.orangecrate.com/article.php?sid=968
- Tervonen, Ilkka 2001. Tietokannat ja tiedon hallintajärjestelmät. [online] [viitattu 15.3.2005].
www.tol oulu.fi/~tmannine/jott01/JOTT_2001_8.pdf
- Tietotekniikan liitto, Cap Gemini Ernst & Young 2003. Tietotekniikan käytön ongelmat suomalaisilla työpaikoilla. [online] [viitattu 22.2.2005]
www.tt-tori.fi/pls/ttl/docs/F1447099676/Tietotekniikangelmat%20final.pdf
- Wiio, Antti 2004. Käyttäjystävällisen sovelluksen suunnittelu. Helsinki: IT Press.

Liitteet

Liite 1: JasperAssistant - käytettävyydestä

Raportin luonti ja asetusten määrittely:

- Luo uusi raportti wizardilla jaspertesti-projektiin (kansioon) ja anna sen nimeksi **omanimi_testi**
- Valitse datalähteeksi ”testi”-kanta
- Lisää sql-kyselylause ”**SELECT * FROM tuote,tuoteryhma WHERE tuote.ryhmanumero_=tuoteryhma.ryhmanumero AND tuoteryhma.ryhmanumero=2 ORDER BY tuote.tuotenimi**” sille varattuun paikkaan
- Valitse käytettävät kentät tauluista:

TUOTERYHMA

ryhmanimi, ryhmakuvaus, ryhmanumero, kateprosentti

TUOTE

tuotenimi, ostohinta, tuotenumero, ryhmanumero_

- Valitse kyseisessä raportissa näkyvien komponenttien tiedot näkyviin vasempaan reunaan
- Muuta raportin nimeksi **Viherkasvit**

Raportin muotoilu:

- Lisää raportin otsikkoa varten otsikkolohko (lohko = band)
- Lisää otsikkolohkoon pääotsikoksi **Viherkasvit**
- Muuta otsikkotekstin fontin kooksi 28, lihavoit teksti sekä keskitä se
- Tallenna raportti
- Lisää sarakeotsikoita varten sarakeotsikkolohko
- Tee kolme samankokoista sarakeotsikkoa edellä luotuun lohkoon (Tuotenimi, Ostohinta, Myyntihinta)
- Muuta fontin kooksi 16, lihavoit tekstit

- Aseta Tuotenimi kentän vasempaan reunaan, Ostohinta ja Myyntihinta kentän oikeaan reunaan
- Tee kolme uutta kenttää (joihin haetaan kannasta tietoa) ja aseta kentät otsikkokenttien alle "Detail Band" -lohkoon
- Lisää Tuotenimen ja Ostohinnan alla oleviin kenttiin otsikkoa vastaava tietokenttä ja muuta sen arvo jos tarvitsee (esim. java.lang.string -> java.lang.integer)
- (Aseta Tuotenimen kaava kentän vasempaan reunaan, Ostohinnan ja Myyntihinnan kaava kentän oikeaan reunaan)
- Tee Myyntihinnan alla olevaan kenttään laskutoimitus, jossa lasketaan ostohinnasta ja kateprosentista myyntihinta **$(\text{kateprosentti}/100 + 1) * \text{ostohinta}$**
- Pienennä lohkon korkeutta sopivan korkuiseksi
- Haluat ajaa raportin PDF-muodossa, aseta siis PDF-viewer käyttöön
- Esikatsele raportti ja **ajaa se sitten pdf-muodossa**

Liite 2: Vertailutaulukko

	Crystal Reports	Jasper Reports
Tietokannat		
Oracle	x	x
MS SQL Server	x	x
My SQL	x	x
DB2	x	x
Muut datalähteet		
XML	x	x
OLAP	x	
Excel	x	x
Datayhteydet		
ODBC	x	x
JDBC		x
OLE DB	x	
Natiiviyhteys	x	
Raporttityypit		
Viivakoodituki	x	x
Ristiintaulukointi	x	
Aliraportit	x	x
Ehdollinen raportointi	x	
Drill-down	x	x
OLAP-raportit	x	
Interaktiiviset piirteet		
Parametroitavuus	x	
Hälytysrajat	x	x
Exporttaustuki		
XML	x	x
PDF	x	x
HTML	x	x
Excel	x	x
CSV	x	x
RTF	x	
Levykkeelle	x	
Sovellukseen	x	
ODBC -tietokantaan	x	
Lotus Notesiin	x	
MS Mailiin	x	
Integrointi omiin ohjelmistoihin		
Java	x	x
Graafiset elementit		
Kaaviot	x	x*
Kuvat	x	x*
Kielet ja fontit		
Unicode-tuki	x	x
Kielitiedostot	x	x

*)mahdollista erillisellä kirjastolla