

Satakunnan ammattikorkeakoulu

Mika Peuralahti

TUOTETIEDON JA SÄHKÖISEN KAUPANKÄYNNIN
WWW-JULKAISUYMPÄRISTÖN KEHITTÄMINEN

Automaatioteknologian koulutusohjelma

2008

TUOTETIEDON JA SÄHKÖISEN KAUPANKÄYNNIN WWW-JULKAISUYMPÄRISTÖN KEHITTÄMINEN

Peuralahti Mika
Satakunnan ammattikorkeakoulu
Tekniikka ja merenkulku
Automaatioteknologian koulutusohjelma
Tammikuu 2008
Aarinen Reino
UDK: 004.65, 004.738.5, 658.8
Sivumäärä: 67

Asiasanat: verkkokauppa, tietokannat, ASP.NET

Sähköinen kaupankäynti ja sisällönhallintajärjestelmät ovat ajankohtaisia internet-projekteissa. Tässä opinnäytetyössä tutkittiin tietotekniikan sovellusten tarjoamia teknisiä mahdollisuuksia erityisesti web-palveluiden näkökulmasta ja sovellettiin tuloksia käytäntöön sähköisen kaupankäynnin järjestelmän kehittämiseksi.

Teoreettinen osuus käsitti tietoteknisten laitteiden ja ohjelmistojen sekä verkkosovellusten kehitykseen perehtymisen. Tietokantoihin ja webin eri ohjelmointitekniikoihin kiinnitettiin erityishuomiota. Sen jälkeen perehdyttiin tarkemmin ohjelmistotuotantoon ja projektinhallintaan yleisellä tasolla.

Tutkimusmenetelmänä käytettiin konstruktivistista tutkimusotetta, ja teoriaa sovellettiin käytännössä ohjelmistotuotteen kehittämiseen. Ensin perehdyttiin opinnäytetyön tilaajan, Hypermedia Oy:n, ohjelmistotyön käytäntöihin ja yrityksen sähköisen kaupankäynnin järjestelmän, Captum2:n, aikaisempaan versioon ja sen uudistamistavoitteisiin.

Verkkokauppajärjestelmän uuden version tuottamiseksi käynnistettiin ohjelmistoprojekti syksyllä 2006. Aluksi tuotettiin toiminnallinen määrittely, jossa kuvataan asiakkaan ja ylläpitäjän näkymät. Ohjelmiston perusversion toteutus tapahtui vuoden 2007 loppuun mennessä ja jatkuu ensimmäisen tilaajakohtaisen version räätälöinnillä tämän opinnäytetyön valmistumisen jälkeen.

Captum2 versio 3.5 on Windows-palvelimella toimiva selainkäyttöinen ohjelmisto, joka toteutettiin Microsoftin ASP.NET -ohjelmointitekniikalla SQL Server -tietokantaa käyttäen. Se sisältää verkkokaupan lisäksi toiminnot laajan web-sivuston ylläpitoon. Captum2:n uuden version kehittäminen lisäsi Hypermedia Oy:n osaamista ja päivitti tuotteistettujen ohjelmistoratkaisujen valikoiman nykypäivän vaatimusten tasolle.

DEVELOPMENT OF PRODUCT INFORMATION AND E-COMMERCE WEB PUBLISHING SYSTEM

Peuralahti Mika
Satakunta University of Applied Sciences
Faculty of Technology and Maritime Management
Degree Programme in Automation Technology
January 2008
Aarinen Reino
UDC: 004.65, 004.738.5, 658.8
Pages: 67

Keywords: electronic commerce, e-commerce, databases, ASP.NET

Electronic commerce and content management systems are currently interesting subjects within internet projects. Technical possibilities provided by information technology applications especially from the web services point of view were studied in this thesis. Results were applied to practice in order to develop an e-commerce system.

The theoretical framework included studying development of computer hardware, software and web applications. A special attention was called to databases and different web programming technologies. Software engineering and general project management were studied in detail after that.

The research method used in this thesis was constructive, and the theory was applied into practice by developing a software product. Software engineering procedures of Hypermedia Ltd., commissioner of this development work, and previous version of company's e-commerce system, Captum2, and its modernizing plans were studied first.

A software project was started in autumn 2006 in order to produce a new version of e-commerce system. At first a functional specification was produced, which describes customer's and administrator's views. A base version of the software was implemented by the end of year 2007 and implementation continues with tailoring the first purchaser specific version after completion of this thesis.

The Captum2 version 3.5 is a software application that is used via web browser on Windows Server platform. It was implemented using Microsoft ASP.NET programming technology and SQL Server database. In addition to e-commerce it includes functions to manage a large web site. By developing a new version of Captum2 Hypermedia Ltd. increased its knowledge and updated company's software portfolio to current level.

SISÄLLYS

TIIVISTELMÄ	2
SUMMARY	3
SISÄLLYS	4
SYMBOLIT JA TERMIT	5
1 JOHDANTO	6
2 MENETELMIEN VALINTA	8
2.1 Tietotekniikan yleinen kehitys ja digitaalinen konvergenssi	8
2.2 Käyttöjärjestelmien ja sovellusohjelmien kehitys.....	10
2.2.1 Useamman käyttäjän tietojärjestelmät	12
2.2.2 Tietokannat ja relaatiomalli.....	13
2.3 Verkkojen ja web-sovellusten kehitys	17
2.3.1 Staattisista dynaamisiin web-sivuihin.....	18
2.3.2 Web-pohjaisten järjestelmien haasteet.....	21
2.4 Ohjelmistotuotannon kehitys	24
2.4.1 Konekielestä olio-ohjelmointiin.....	25
2.4.2 Ohjelmistotuotannon menetelmät	26
2.4.3 Projektinhallinta	32
2.5 Hypermedia Oy:n käytännöt ohjelmistotyössä	33
2.6 Uuden Captum2:n tavoitteet	38
2.7 Ohjelmistoprojektin käynnistys	40
3 OHJELMISTOPROJEKTIN TULOKSET	42
3.1 Asiakkaan näkymät	43
3.1.1 Tuoteryhmät ja hakutoiminto	44
3.1.2 Tuotekortti.....	45
3.1.3 Keräilylista (ostoskori).....	46
3.1.4 Tilauslomake ja tarjouspyyntölomake	47
3.2 Ylläpitäjän näkymät	48
3.2.1 Tuotteiden lisäys ja muokkaus.....	50
3.2.2 Tarjouspyynnöt ja tilaukset.....	52
3.2.3 Raportointi ja tilastot.....	53
3.3 Ohjelmiston toteutus	54
4 JOHTOPÄÄTÖKSET.....	55
LÄHTEET.....	58
LIITTEET	60

SYMBOLIT JA TERMIT

AJAX	Asynchronous JavaScript And XML
ASP	Active Server Pages
CGI	Common Gateway Interface
HTML	Hypertext Markup Language
IIS	Internet Information Services
JSP	Java Server Pages
PDF	Portable Document Format
PHP	Hypertext Preprocessor
RIA	Rich Internet Application
SQL	Structured Query Language
TCP/IP	Transmission Control Protocol / Internet Protocol
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language

1 JOHDANTO

Tietotekniikan käyttö yrityksissä ja kotitalouksissa on koko lyhyen historiansa aikana lisääntynyt kiihtyvällä vauhdilla. Samalla monet innovaatiot ovat vakiintuneet ja arkipäiväistyneet kuten esimerkiksi pankkiasioiden hoito tietoverkon kautta. Mikro-tietokoneiden aloittaessa maailmanvalloituksensa 1980-luvulla IBM PC:n myötä vain harvat rohkenivat ennustaa mikroille niin valtaisa menestystä kuin mitä viime vuosikymmenet ovat osoittaneet. Teollisuusstandardin mukaiset PC:t levisivät ensin yrityksiin ja 1990-luvulla vähitellen myös koteihin. Viimeisten reilun kymmenen vuoden aikana internet on osaltaan vauhdittanut kehitystä mahdollistaen näyttävän ja ajantasaisen tiedon tuomisen suoraan käyttäjien työpöydille. Osansa tietotekniikan yleistymisessä on ollut myös entistä helppokäyttöisemmällä ja vakaammilla käyttöjärjestelmillä ja riittävällä sovellustarjonnalla. Vaikka moni käyttääkin nykyään kannettavaa tietokonetta, tulevaisuuden laitteet ovat todella henkilökohtaisia vasta, kun ne kulkevat mukana kuten matkapuhelimet ja ovat jatkuvasti yhteydessä erilaisiin verkkoihin.

Tehokkaammat laitteet ja nopeammat yhteydet ovat mahdollistaneet uudenlaiset tietokoneiden käyttötavat ja -tarkoitukset. Yhä useammat sovellukset ovat yhteydessä verkkoon hakien tietoa taustalla. Toisaalta web-selaimesta on tullut monille tärkein sovellus, jolla käytetään erilaisia sivustoja ja järjestelmiä niin julkisessa internetissä kuin suljetuissa intraneteissakin. Web-palvelimilla ajettavat sivustot ja järjestelmät ovat myös laajentuneet ja monimutkaistuneet koko puolentoista vuosikymmenen olemassaolonsa ajan. Kun 1990-luvun puolivälissä staattiset HTML-sivut ja Perl-kielillä toteutetut alkeelliset CGI-skriptit olivat vielä arkipäivää, on niistä siirrytty kehittyneempien ohjelmointimenetelmien (ASP, PHP) kautta nykypäivän laajoihin tietokantapohjaisiin dynaamisiin sivustoihin, joilla käytetään mm. ASP.NET ja Java Servlet -tekniikoita. Web-sovellukset ovat korvaamassa perinteisiä työasemasovelluksia, jolloin käyttäjä tarvitsee vain nykyaikaisen selaimen ja internet-yhteyden.

Eräs internetin mahdollistama sovellus on verkkokauppa, josta voi tilata erilaisia hyödykkeitä ajasta ja paikasta riippumatta. Postimyyntiin verrattuna verkkokauppa

mahdollistaa laajemmat tuote-esittelyt, reaaliaikaisuuden ja helpommat tilaus- ja maksutavat. Vuonna 2006 suomalaiset kuluttajat ostivat verkosta 3,2 miljardilla eurolla ja kaupan arvo kasvoi 20 % edellisvuoteen verrattuna (Sähköisen kaupan palvelukeskus 2007). Myös yritysten välinen (B2B, Business to Business) verkkokauppa on lisääntynyt internetin arkipäiväistymisen myötä.

Tässä opinnäytetyössä tarkastellaan tietotekniikan tarjoamia mahdollisuuksia ja Hypermedia Oy:n sähköisen kaupankäynnin web-sovelluksen kehittämistä. Hypermedia Oy on vuonna 1989 perustettu yritys, joka on nykyään keskittynyt vaativien www-julkaisuratkaisujen ja ammatillisen koulutuksen toimintajärjestelmien toteuttamiseen. Hypermedia Oy:n keskeisimmät omat ohjelmistotuotteet ovat julkaisujärjestelmä Hyperscriptor XG sekä sitä täydentävä tuotetiedon esittelyn ja sähköisen kaupankäynnin järjestelmä Captum2. Hyperscriptor XG:tä voidaan laajentaa myös useilla pienemmillä valmiilla moduleilla tai tilaajakohtaisesti toteutettavilla ominaisuuksilla. Koulutuksen järjestelmiä toteutetaan asiakaskohtaisesti, joten niitä ei ole samalla tavalla tuotteistettu kuin pääosin yritysmaailmaan toimitettavia julkaisuratkaisuja.

Captum2 on ollut olemassa sähköisen kaupankäynnin sovelluksena jo vuosituhaten vaihteesta asti. Sen ensimmäinen versio kehitettiin palvelemaan puutarha.net-verkkopalvelun siemen- ja tarvikekauppaa. Tein järjestelmästä tuolloin opinnäytetyön insinööri (AMK) -tutkintoa varten. Sen jälkeen järjestelmää on jonkin verran päivitetty (mm. euron käyttöönotto ja ylläpidon integrointi Hyperscriptoriin), mutta Captum2-alustan perustekniikkaan on tehty melko vähän muutoksia, vaikka sitä käytetään muissakin sivustoissa. Tekniikka on kuitenkin kehittynyt seitsemässä vuodessa melkoisesti, ja tämän työn aikana selvitettiin Captum2:n tarvitsemat muutokset, jotta se vastaa Hypermedia Oy:n muiden sovellusten tasoa niin tekniikan kuin toimintojenkin osalta.

Alun perin Captum2:lle oli tarkoitus antaa samalla uusi nimi, Hyperscriptor Product, mutta tätä päädyttiin käyttämään vain apunimenä markkinoitaessa Captum2:a tuoteluettelosovellukseksi ilman kauppatoimintoja. Sen sijaan uuden Captum2:n sisäiseksi versionumeroksi päätettiin asettaa 3.5, joka on sama kuin Hyperscriptor XG:ssä. Nykyisillä välineillä toteutettuna molemmat ohjelmistot säilyvät helposti ylläpidettävänä ja mukautuvina myös lähitulevaisuuden tarpeisiin.

2 MENETELMIEN VALINTA

Tässä kappaleessa käsitellään ensin tietotekniikan (laitteistojen, ohjelmistojen ja verkkojen) sekä ohjelmistokehityksen yleistä kehitystä teknologisen viitekehyksen taustoittamiseksi ja siirrytään sitten Hypermedia Oy:n käytäntöjen esittelyn kautta Captum2:n tavoitteisiin ja toteutustapoihin. Kun kyseessä on kaupallinen tuote, sen ominaisuuksien täytyy vastata kysyntää ja olla tarkoituksenmukaisia. Kaikkia teknisesti mahdollisia ideoita ei voida toteuttaa, jotta tuotteen käytettävyys säilyy hyvänä ja kehitysprojektin resurssit riittävät. Siksi tekstissä käsitellään myös projektinhallinnan näkökulmaa halutun lopputuloksen tuottamiseksi.

Tämä opinnäytetyö tarkastelee Captum2:n kehitystä konstruktiivisella tutkimusotteella, joka on tapaustutkimuksen (case-tutkimuksen) eräs alametodi, joka soveltuu hyvin reaalimaailman ongelmien ratkaisemiseen. Konstruktiivisen tutkimusotteen tavoitteena on kehittää uusi innovatiivinen konstruktio, joka tässä tapauksessa on ohjelmistotuote. Konstruktiot eivät ole löydettyjä, vaan ne keksitään ja kehitetään. Konstruktiivinen tutkimusote yhdistää teorian ja käytännön siten, että se pyrkii hyödyttämään molempia. (Lukka 2001.) Tämän opinnäytetyön kohdalla tavoitteena on, että konstruktio on samalla sekä jatkumo tässä kappaleessa kuvatulle teoreettiselle viitekehykselle että käytännössä hyödynnettävä ohjelmistotuote Hypermedia Oy:lle.

2.1 Tietotekniikan yleinen kehitys ja digitaalinen konvergenssi

Ensimmäiset sähköisesti toimivat tietokoneet perustuivat releisiin, mutta 1940-luvulla siirryttiin käyttämään elektroniputkia ja 1950-luvulla transistoreita. Integroidut piirit mahdollistivat transistoriheyden kasvattamisen ja lopulta ohjelmoitavan mikroprosessorin kehittämisen. Sen jälkeen tietokoneiden prosessoritehot ovat kasvaneet Mooren lakia noudattaen aivan näihin päiviin saakka ja esimerkiksi yritysverkkojen työasemat ovat nykyään teholtaan samaa luokkaa kuin palvelimetkin. Keskusmuistia palvelimista saattaa vielä löytyä enemmän, mutta työasemissakin siirrytään kovaa vauhtia megatavuista gigatavuihin. PC:n keskusmuistin määrä on tu-

hatkertaistunut 15 vuoden aikana. Tällainen suorituskyky on mahdollistanut siirtymisen täysin grafiikkapohjaisiin käyttöliittymiin ja perinteisen tekstinkäsittelyn ohella nykymikrot käsittelevät sujuvasti kuvaa, ääntä ja jopa teräväpiirtoista videokuvaa. Tämän kehityksen rinnalla myös näytönohjainten suorittimien laskentateho ja muistien määrä on kasvanut, mikä mahdollistaa entistä näyttävämmän grafiikan käyttämisen niin peleissä kuin muissakin sovelluksissa ja jopa käyttöjärjestelmissä (esim. Windows Vistan Aero-käyttöliittymä). Palvelintenkään suorituskykyä tarvitsee enää harvemmin lisätä klusteroinnilla ja kuormantasauksella, kun päinvastoin erilaisilla virtualisointiratkaisuilla useampia loogisia palvelimia keskitetään samaan fyysiseen laitteistoon.

Kehitys on edellyttänyt melkoisia harppauksia myös massamuistien osalta, vaikka uusissa tietokoneissa näkyikin levykeasemia vielä pitkään 2000-luvulla. Nauhatekniikkaa on käytetty mikroissa ainoastaan varmistusmedianä. Kiintolevyjen kapasiteetit ovat kasvaneet tuhatkertaisiksi reilussa kymmenessä vuodessa, ja mukaan on tullut ensin lukevat ja myöhemmin kirjoittavat CD-asemat. Nykyään kirjoittava DVD-asema on uuden tietokoneen vakiovaruste, ja entistä suurikapasiteettisemmat optisten levyjen tekniikat tekevät tuloaan markkinoille. Yhteenvedona voidaankin todeta, että tietokoneiden käyttömahdollisuudet ja käytön helppous ovat lisääntyneet merkittävästi. Peruskäytössä laitteet eivät myöskään vanhene enää niin nopeasti kuin viime vuosikymmenellä, joten suorituskyvyltään pöytäkonetta hieman vaatimattomampi, mutta siirreltävyydeltään ylivoimainen kannettava mikro on monille riittävä työkalu.

Vauhdikkaalla kehityksellä on myös rajansa, vaikka niitä onkin säännöllisesti saatu siirrettyä kauemmas tulevaisuuteen. Prosessorien lämmöntuotto on merkittävästi vaikeuttanut kellotaajuuksien nostoa, ja tehonlisäystä täytyykin nykyään hakea useammalla suoritusnopeudella, mikä lisää prosessoriarkkitehtuurin haasteita. Vastaavasti kiintolevyissä on siirrytty pystysuuntaiseen tallennukseen, jotta yhden bitin säilömiseen on käytettävissä riittävästi atomeita. Flash-muistiratkaisut ovat yleistyneet muistitikujen ja -korttien myötä, mutta ne eivät pysty vielä vuosiin kilpailemaan magneettisen tallennuksen kanssa samassa hintaluokassa. Voidaan vain kysyä, kuinka kauan muutaman vuoden aikana tapahtuva kapasiteetin kymmenkertaistuminen voi jatkua. Toisaalta, uudet sovellukset eivät enää vaadi niin suuria tehon- ja kapasiteettilisäyksiä kuin aikaisemmin.

Tietokoneiden komponenttien ohella kehitystä on tapahtunut myös oheislaitteissa: tulostimet ja skannerit ovat entistä nopeampia, tarkempia ja halvempia, värien käyttö tulosteissa on lisääntynyt, näytöt ovat entistä suurempia ja tarkempia, mutta litteitä. Tietokoneisiin saa liitettyä myös mitä mielikuvituksellisimpia laitteita USB-liitäntäisestä mukinlämmittimestä kehittyneisiin digi-tv- ja GPS-vastaanottimiin. Laitteiston kustannusten lasku on mahdollistanut ”älykkyyden” lisääntymisen esimerkiksi verkkojen reitittimisessä, joihin saa vakiona palomuuritoiminnot ja VPN-verkkojen (Virtual Private Network) tuen. Monia oheislaitteita (esim. tulostimia) voi hallinnoida web-käyttöliittymän kautta, kun ennen asetusten muuttaminen vaati hankalien DIP-kytkinten asettelua tai vaikeasti muistettavia näppäilyjä laitteen hallintapaneelista.

Tietotekniikan nopea kehitys on digitalisoinut monia muitakin teknisiä asioita, ensin matkapuhelimet, sitten kamerat ja lopulta myös televisiolähetykset. Sähköisestä paperista on puhuttu pitkään, mutta se ei ole vielä lähtenyt yleistymään. Silti uusia digikeksintöjä on tulossa markkinoille jatkuvasti. Digitaalinen konvergenssi on pakottanut monet opiskelemaan uusia tapoja tehdä asioita. Myös opetus on muuttunut tietoteknisemmäksi, minkä on huomannut omakohtaisesti esimerkiksi Satakunnan ammattikorkeakoulussa vertaamalla PowerPoint-esitysten määrää luennoilla siihen tilanteeseen, joka oli kymmenen vuotta sitten AMK-perustutkintoa suorittaessani. Sähköiset oppimateriaalit ja oppimisympäristöt kuten myös kirjaston etäkäytettävät tietokannat helpottavat työskentelyä ja tekevät opiskelusta vähemmän aikaan tai paikkaan sidottua.

2.2 Käyttöjärjestelmien ja sovellusohjelmien kehitys

Tietokoneet kautta aikojen ovat tarvinneet ohjelmistoja toimiakseen. Laitteisto suorittaa käyttöjärjestelmän tai sen päällä ajettavan sovellusohjelman mukaisia käskyjä ja lopputuloksena on järjestelmä, joka esittää käyttäjälle käyttöliittymän, käsittelee käyttäjän syöttämän informaation ja tuottaa syötteistä tuloksen. Käyttöjärjestelmä tarjoaa sovellusohjelmille palveluita esim. tiedostojen käsittelyyn ja eri sovellusten moniajoon.

Alkuaikojen tietokoneet olivat lähinnä laskimia eli niitä käytettiin matemaattisten ongelmien ratkomiseen. Tällöin käyttöliittymällä ei ollut niin keskeinen merkitys ja fyysisesti isojen tietokoneiden ohjelmistot olivat hyvin pieniä ja tiiviitä. Erillistä käyttöjärjestelmää ei vielä ollut ja ohjelmat ladattiin reikäkorteilta tai -nauhoilta. Suorituskyvyn ja käyttäjäkunnan kasvaessa vaatimukset niin suoritettavien tehtävien laajuudelle kuin käyttöliittymällekkin kasvoivat. Yhden ohjelman eräkäytöstä siirryttiin useamman ohjelman tai käyttäjän osituskäyttöön, jolloin laitteiston ja sovellusohjelman väliin tarvittiin käyttöjärjestelmä. (Koikkalainen & Orponen 2002.)

1970-luvulla yleistynyt UNIX on näihin päiviin saakka säilyttänyt asemansa tiedeyhteisön ja korkeakoulumaailman käyttöjärjestelmänä. Siinä on monia kehittyneitä prosessien ja muistin hallintaan liittyviä ominaisuuksia, jotka mahdollistavat sujuvan monen käyttäjän yhteiskäytön. Mikrojen huomattavasti yksinkertaisempi MS-DOS (Microsoft Disk Operating System) oli 1980-luvulla kehittymättömyydestään huolimatta käyttöjärjestelmien teollisuusstandardi, jonka päälle mikrojen sovellukset ohjelmoitiin. (Koikkalainen & Orponen 2002.)

Aluksi hyötyohjelmistot olivat pääosin merkkipohjaisia, koska grafiikan näyttämiseen tarvittiin erikoisnäyttöjä ja graafiselle käyttöliittymälle ei ollut tarpeeksi laskentatehoa. Kuitenkin esimerkiksi koko ruudun tekstiilassa toimiva tekstinkäsittely edusti jo hyvin pitkälle kehittyneitä sovellusta, jossa käyttöliittymässä oli monia elementtejä kuten valikoita. Nykyisin monet sulautetut järjestelmät ovat juuri ohittaneet tämän vaiheen ja ovat matkalla kohti värikkäitä graafisia käyttöliittymiä.

1990-luvulla Microsoft Windows ja avoimen lähdekoodin Linux, joka toimii hyvin pitkälle kuten UNIX, toivat graafisuuden ja moniajon suuren yleisön mikroihin. Apple oli ollut omissa mikroissaan jo vuosia PC:n kehityksen edellä. Myös UNIX-järjestelmiä alettiin käyttää yhä useammin graafisen käyttöliittymän kautta. Graafisessa käyttöliittymässä on hiirellä siirreltäviä sovellusikkunoita ja esimerkiksi tekstinkäsittelyohjelma esittää käyttäjän kirjoittaman tekstin samannäköisenä kuin se tulostuu paperille (WYSIWYG). 2000-luvun ensimmäisellä vuosikymmenellä Windowsin eri versiot sekä kilpailevat Linux-toteutukset ovat tuoneet mikrojen käyttäjille sen helppokäyttöisyyden ja vakauden, joka niissä olisi pitänyt olla jo aikaisemmin. Vaikka Applen valmistamat tietokoneet, käyttöjärjestelmät ja sovellukset ovat olleet

helppokäyttöisiä pidempään, ne eivät ole siitä huolimatta saavuttaneet kuin muutama prosentti markkinaosuuden.

Tulevaisuudessa ohjelmistot kehittyvät entistä helppokäyttöisemmiksi ja käyttöliittymältään näyttävämmän näköisiksi. Sama kehitys näkyy viiveellä matkapuhelinten ja muiden sulautettujen järjestelmien sovelluksissa, jotka myös saavat entistä enemmän suorituskykyä, värejä ja näyttöpinta-alaa käyttöönsä. Muissakin kuin ns. älypuhelimissa yleistyy käyttöjärjestelmä, joka mahdollistaa mm. mobiilisovellusten moniajon.

2.2.1 Useamman käyttäjän tietojärjestelmät

Yrityskäytössä tietotekniikkaa käytetään usein jonkin järjestelmän, esim. laskutusjärjestelmän tai asiakkuudenhallintajärjestelmän, ajamiseen. Nämä järjestelmät ovat kehittyneet sovellusohjelmien yleisten trendien mukana. Vanhoissa keskuskonepohjaisissa (mainframe) tietojärjestelmissä koko järjestelmä oli sijoitettu yhdelle tietokoneelle (1-tier), jota käytettiin merkkipohjaisten päätteiden kautta. Tällöin isollakin organisaatiolla saattoi olla vain yksi varsinainen tietokone. Näin siis aikakaudella ennen mikroja, jolloin tietokoneet ja niiden ohjelmat olivat kalliita ja valmistajakoh-
taisia.

1980-luvulla PC-koneiden myötä yleistyivät lähiverkot sekä asiakas/palvelin-arkkitehtuuri (2-tier), jossa asiakasohjelmaa ajavat työasemat korvasivat päätteet. Usein mahdollisimman suuri osa tietojenkäsittelystä tehtiin asiakasohjelmassa, jolloin kaikkien työasemien yhteiseksi palvelimeksi riitti suhteellisen pienitehoinen kone, yleensä myös PC. Windowsin myötä tietojärjestelmien asiakasohjelmiin tuli graafinen käyttöliittymä.

1990-luvun aikana yleistyivät hajautetut tietojärjestelmät (n-tier), joissa arkkitehtuuri on jaettu vähintään kolmeen kerrokseen (3-tier). Ylimpänä käyttäjällä on kevyt asiakasovellus (thin client), esimerkiksi web-selain, jolle palvelin lähettää käyttöliittymän. Varsinainen sovelluslogiikka on koottu palvelimelle ja sen alapuolella kolmannella kerroksella (mahdollisesti erillisellä palvelimella) ovat tarvittavat taustajärjes-

telmät, esim. tietokannat. Palvelinpään suorituskykyvaatimukset saadaan täytettyä tasoja ja palvelimia lisäämällä.

Hajautetut tietojärjestelmät ovat lähes syrjäyttäneet perinteiset asiakas/palvelin-arkkitehtuurin mukaiset järjestelmät, koska niiden lähestymistapa tarjoaa mm. seuraavia etuja:

- Vakioitujen ja luonteviini kohtiin sijoitettujen rajapintojen ansiosta mikä tahansa tietojärjestelmän kerros voidaan vaihtaa eri tekniikalla toteutettuun.
- Asiakassovellus on niin kevyt, että tarvittaessa sitä voidaan ajaa myös mobiililaitteissa.
- Kun asiakassovelluksena käytetään web-selainta, verkkona voidaan käyttää kaikkialle ulottuvaa internetiä, jolloin sovelluksen käyttö ei ole sidottu aikaan, paikkaan tai päätelaitteeseen.

Monesti käyttäjälle yksinkertaisen käyttöliittymän takana on paljon monimutkaista logiikkaa ja laskentaa hajautetun tietojärjestelmän muilla kerroksilla. Tästä hyvänä esimerkkinä on Google-haku, jossa yhden tekstikentän ja painonapin kautta pääsee nopeasti käsiksi internetin valtavaan tietomassaan.

2.2.2 Tietokannat ja relaatiomalli

Jo yksittäinen sovellusohjelma saattaa tarvita käyttöjärjestelmän tarjoamia tiedostopalveluita monipuolisempia informaation tallennus- ja hakupalveluita. Useamman käyttäjän tietojärjestelmässä tarve on merkittävästi suurempi, koska kaikki käyttäjät käsittelevät samoja tietoja ja informaatiota on enemmän. Nämä tarpeet täyttävien tietokantojen kehitys on luku sinänsä. Tietokanta on toisiinsa liitetyistä tietoalkioista muodostuva tietovarasto, joka tallentaa jollain muulla sovelluksella tuotettua informaatiota tietomallin mukaisesti. Tietomalleja ovat hierarkkinen tietomalli (puurakenne), verkkomalli, relaatiomalli sekä oliomalli. Nykyään ylivoimaisesti yleisin tietomalli on relaatiomalli, jonka Edgar F. Codd esitteli jo vuonna 1970. Tosin oliomallia käyttäviä tietokantatoteutuksiakin on markkinoilla, mutta ne eivät ole vielä laajasti käytössä. (Koikkalainen & Orponen 2002.)

Markkinoilla olevat tietokannat ovat kehittyneet tietokoneiden laskentakapasiteetin kasvaessa. Ensimmäiset tietokannat eivät toimineet relaatiomallin mukaisesti, vaan olivat useina yksittäisinä tiedostoina toteutettuja tietokantoja, joissa käytettiin hierarkkista tietomallia. IBM kehitti ensimmäiset tällaiset tietokannat 1960-luvulla Apollo-projektin tarpeisiin. Niitä seurasivat verkkomallin mukaiset tietokannat sekä ns. flat-file -tietokannat, joissa tiedot on koottu yhteen tiedostoon. Nykyisin avoimen lähdekoodin tietokantana tunnettu Ingres puolestaan oli yksi ensimmäisistä 1970-luvulla relaatiomallia soveltaneista tietokannoista ja myöhemmin Oracle teki relaatiomallin mukaisesta tietokannasta kaupallisen menestyksen monilla eri laitteistoalustoilla.

Relaatiomallissa informaatio jaetaan relaatioihin (tauluihin), jotka muodostuvat monikoista (riveistä) ja attribuuteista (sarakkeista). Käytännön toteutuksissa monikkoja kutsutaan usein tietueiksi (record) ja attribuutteja kentiksi (field). Jokaisessa relaatiossa on yhdestä tai useammasta attribuutista muodostuva perusavain (primary key), joka yksilöi monikot. Perusavaimen attribuuteissa ei saa olla samoja arvoja kahdessa eri monikossa. Toisaalta useammasta attribuutista muodostuvasta perusavaimesta ei saa voida poistaa yhtäkään attribuuttia rikkomatta em. sääntöä eli perusavaimessa ei saa olla mukana ylimääräisiä attribuutteja. (Koikkalainen & Orponen 2002.)

Perusavainten lisäksi relaatioihin voi määrittellä viiteavaimia (foreign key), jotka viittaavat toisten relaatioiden perusavaimiin. Viiteavainten avulla tietokanta saadaan pysymään koossa, vaikka tiedot hajautetaan useisiin relaatioihin. Viite-eheys säilytetään määrittelemällä viiteavainkohtaisesti mitä tehdään, jos perusavaimen arvoa muutetaan tai sen mukainen monikko poistetaan toisesta relaatiosta. Vaihtoehtoja ovat muutoksen ja poiston estäminen (on update restrict, on delete restrict) tai muutoksen teko myös viiteavaimen (on update cascade) ja poistettavaan perusavaimen viittaavan viiteavaimen sisältävien monikkojen poistaminen (on delete cascade).

Keskeinen asia relaatiomallissa on myös tietokannan normalisointi eli tarkistaminen siten, että kanta täyttää vähintään kolme ensimmäistä ns. normaalimuotoa (normal form). Ensimmäinen normaalimuoto (1NF) määrittelee, että monikkoon tallennettavien arvojen on oltava atomisia eli esim. henkilöluettelon etu- ja sukunimi on tallennettava eri attribuutteihin.

Schneider (1997, 13-14) on esittänyt toisen esimerkin, jossa henkilön tietojen yhteyteen on varattu tila kahdelle puhelinnumerolle. Jos sitten rekisteröitävillä henkilöillä onkin kolme numeroa, pitäisi tauluun lisätä kolmas puhelinnumerokenttä ja kaikki taulua käsittelevät sovellukset tarkistaa ja korjata käyttämään kolmea kenttää. Entä sitten, jos tämän jälkeen pitäisikö lisätä vielä neljäs puhelinnumero? Parempi ratkaisu on normalisoida (1NF) tietokanta heti siten, että puhelinnumerot ovat omassa taulussaan ja jokaiseen numeroon liittyy viiteavain henkilötauluun sekä viittauslaskuri. Saman henkilön ensimmäiselle puhelinnumerolle viittauslaskurin arvo on 1, toiselle 2, kolmannelle 3 jne. Viiteavain ja viittauslaskuri yhdessä muodostavat puhelinnumerotaulun perusavaimen. (Schneider 1997, 13-14.)

Toinen normaalimuoto (2NF) laajentaa ensimmäistä normaalimuotoa vaatimuksella, että kaikkien perusavaimen sisältymättömien attribuuttien on oltava kokonaisriippuvia perusavaimesta. Tämä tarkoittaa, että jos relaation perusavain muodostuu esim. kahdesta attribuutista, niin samassa relaatiossa joku muu attribuutti ei saa riippua pelkästään toisesta perusavaimen kuuluvasta attribuutista. 2NF:n saavuttamiseksi relaatio joudutaan usein jakamaan useammaksi relaatioksi.

Schneider (1997, 14-15) on esittänyt esimerkin, jossa on kolme taulua: huoltokohteet, huoltotoimenpiteet ja huoltokutsut. Ensimmäisessä taulussa on huollettavien kohteiden katuosoitteet ja kaupunkien nimet, toisessa huoltotoimenpiteiden hinnat ja kolmannessa viittaukset kahteen edelliseen sekä katuosoite, työn kuvaus ja päivämäärä. Katuosoite on mukana kahteen kertaan ja huoltokutsut-taulussa se riippuu pelkästään huoltokohteet-tauluun viittaavan kentän arvosta, ts. tietokanta ei ole 2NF:ssä. Koska osoite on mukana huoltokohteet-taulussa, riittää kun se poistetaan huoltokutsut-taulusta. Mutta tietokanta ei silti ole 2NF:ssä, koska työn kuvaus riippuu pelkästään huoltotoimenpiteet-tauluun viittaavan kentän arvosta. Tämä voidaan korjata siirtämällä työn kuvaus huoltotoimenpiteet-tauluun, johon se loogisesti ajatellen kuuluukin. (Schneider 1997, 14-15.)

Kolmas normaalimuoto (3NF) laajentaa toista normaalimuotoa sillä edellytyksellä, että mikään perusavaimen kuulumattomista attribuuteista ei ole transitiivisesti riippuva relaation perusavaimesta. Transitiivinen riippuvuus on olemassa silloin, kun taulu sisältää sarakkeen, joka ei ole avain, mutta joka silti määrittelee muiden sarak-

keiden arvoja. 3NF:n saavuttaminen edellyttää, että taulun sisällä jonkin sarakkeen arvoa ei esimerkiksi voida laskea muiden sarakkeiden arvojen perusteella. (Schneider 1997, 16.)

Vaikka normalisoinnin tarkoituksena on varmistaa intuitiivisesti suunniteltujen taulujen tehokkuus, ei normalisointia aina voida toteuttaa käytännössä. Eräs tällainen tilanne on historiatietojen säilyttäminen. Esimerkiksi verkkokaupan asiakasrekisterissä on samoja tietoja kuin tilausrekisterissä. Tietoja ei kuitenkaan voi yhdistää viiteavaimella tilausrekisteristä asiakasrekisteriin, koska vanhojen tilaustietojen on säilyttävä asiakastietojen muuttuessa.

Relaatiomallin mukaista tietokantaa käsitellään SQL-kyselykielellä, joka on hyvin ilmaisuvoimainen kieli informaation hakuun, tallennukseen, päivitykseen ja poistamiseen tietokannasta. Yhteen kyselyyn voidaan yhdistää useita relaatioita ja hakutuloksen voi lajitella tai ryhmitellä monipuolisesti. Tiedon käsittelyn lisäksi SQL:llä voidaan myös määritellä relaatioiden rakenteet, avaimet ja hakuja nopeuttavat indeksit. SQL-standardiin sisältyy myös joukko funktioita esim. minimi- ja maksimiarvojen hakuun sekä merkkijonojen käsittelyyn. Lisäksi markkinoilla olevissa tietokannoissa on runsaasti valmistajakohtaisia funktioita, joiden syntaksi vaihtelee. Valmiin funktion käyttäminen on yleensä nopein tapa suorittaa jokin toiminto, koska tällöin tietokanta voi optimoida toimintansa funktiota varten. Jos vastaava toiminto suoritetaan hajautetun tietojärjestelmän sovelluslogiikassa, informaation käsittely ei yleensä ole yhtä nopeaa. On huomattava, että vaikka sovellusohjelmoija käsittelee tietokantaa SQL:llä ja suunnittelee tietorakenteet relaatiomallin mukaisesti, niin sovelluksen loppukäyttäjälle näkyy vain tallennettu informaatio tarkoituksenmukaisesti esitettyinä. (Koikkalainen & Orponen 2002.)

Hajautetun tietojärjestelmän periaatteisiin kuuluu, että myös tietokantakerros voidaan vaihtaa. SQL vakioi tietokannan kyselykielen, joskin siinäkin voi olla valmistajakohtaisia eroja. Lisäksi tarvitaan vakioitu rajapinta, jonka kautta tietokantaa käytetään. Tällaisia ovat esimerkiksi ODBC (Open Database Connectivity) sekä Java-maailman JDBC (Java Database Connectivity). Ne perustuvat ajuriin, jonka kautta tietokannalle lähetetään SQL-kyselyitä ja otetaan vastaan kyselyiden tuloksia. OLE DB on uudempi oliopohjainen tekniikka, joka perustuu Microsoft COM:iin (Component

Object Model). OLE DB:n avulla voidaan joissain tapauksissa yksinkertaistaa tietokannan käsittelyä käyttämällä sitä ODBC:n päällä. ADO (ActiveX Data Objects) puolestaan toimii OLE DB:n päällä ja se on yleistynyt monipuolisuutensa ja helppokäyttöisyytensä ansiosta. (Falck 1999, 107.) Uudempi versio ADO:sta on Microsoft .NET Frameworkin yhteydessä toimiva ADO.NET.

Microsoft julkaisi 1990-luvun alussa Access-tietokannan, josta tuli osa Office-pakettia. Access on helppokäyttöinen perustietokanta, joka tukee nykyään myös relaatiomallia. Siinä on monia ohjattuja toimintoja ja valmiita mallipohjia, mutta se on tarkoitettu vain yhden käyttäjän tietokannaksi. Käytännössä Access-tietokanta on melko epävakaata, jos sillä on useampia yhtäaikaista käyttäjiä. Accessia vastaava avoimen lähdekoodin tietokanta on OpenOffice Base. Microsoftin järeämpi tietokantaratkaisu on SQL Server, joka on mitoitettu raskaiden tietomassojen käsittelyyn monen käyttäjän palvelinympäristössä. Sitä vastaava, joskin huomattavasti kevyempi, avoimen lähdekoodin ratkaisu on MySQL.

Sovellusten koon ohella myös niiden käsittelemän tietosisällön määrä on kasvanut huomasti. Kalifornian yliopiston tutkimuksen mukaan vuoden 2003 aikana maailmassa tuotettiin 1,5 miljoonaa teratavua informaatiota eli keskimäärin 250 megatavua jokaista ihmistä kohti (<http://www2.sims.berkeley.edu/research/projects/how-much-info>). Informaation tallennuksessa on kuljettu pitkä matka nauhakeloista ja tiedostoista, joista tietoa voitiin hakea vain peräkkäismuodossa, nykyaikaisiin massamuistiteihin ja tietokantoihin, jotka sallivat tiedon haun nopeasti missä tahansa järjestyksessä. Monet sovellukset käyttävät nykyään rakenteellista tallennusmuotoa, kuten XML:ää, jota useat tietokannat tukevat omana tietotyypinään. Vastaavasti binäärisen informaation, esim. kuvien, pakkaus- ja tallennusmenetelmät ovat kehittyneet myös tietokannoissa.

2.3 Verkkojen ja web-sovellusten kehitys

Tietotekniikan yleistyessä laitteet ovat verkottuneet, kun useammat käyttäjät ovat tarvinneet samoja resursseja. Lähiverkkojen (LAN) lisäksi yhteyksiä on tarvittu myös ulospäin. Verkkojen alkuaikoina yhteyksiä tarvittiin keskuskoneen päätepalve-

luiden käyttöön esim. yrityksen haarakonttorista, jolloin hitaammatkin nopeudet riittivät. Kotimikrojen myötä kaksikymmentä vuotta sitten yleistyivät BBS:t (Bulletin Board System) eli ”purkit”, joita voidaan pitää nykyisten internet-yhteisöjen esikuvana. Purkkeihin soitettiin puhelinmodeemeilla ja niitä käytettiin tyypillisesti viestien välittämiseen ja pienten tiedostojen siirtoon. Yhteydet olivat piirikytkentäisiä ja aikaveloitteisia kuten äänipuhelut vielä nykyäänkin yleisesti. 1990-luvun puolivälissä pakettikytkentäisen internetin web-sivustot alkoivat yleistyä eri selainten myötä ja käyttäjät siirtyivät World Wide Webin ja internet-sähköpostin äärelle. Internet-sähköposti tarjosi myös yrityskäyttäjille uuden tehokkaan viestintätavan, vaikka sähköpostin muut toteutukset olivatkin olleet yritysten käytössä jo pitkään.

Tietoverkkoyhteyksien nopeudet ovat kasvaneet 1990-luvun alun muutamista kilobiteistä nykypäivän megabiteihin sekunnissa, vaikka runkoverkkoja lukuun ottamatta yhteydet perustuvat edelleen pääosin kuparijohtimiin valokuidun sijasta. Samalla lähiverkkojen nopeudet ovat kasvaneet gigabittiin sekunnissa. Internetin yleistymisen teki sen perusprotokollista eli TCP/IP:stä lingua francan, yhteiskielen, jota kaikki verkottuneet laitteet nykyään osaavat. Kuluttajien ja pienyritysten internet-yhteyksissä ADSL on korvannut modeemit ja väliinpuotoajaksi jääneet ISDN-yhteydet. Langattomat dataverkot ovat yleistyneet niin Wi-Fi-standardeihin perustuvina WLAN-verkkoina kuin matkapuhelinyhteyksinäkin (GPRS, EDGE, 3G, HSPA). Käyttöaikoihin perustuva hinnoittelu on jäänyt menneisyyteen ja useimmiten siirretyn informaation määrästä ei myöskään enää laskuteta edes mobiiliyhteyksissä. Aina auki olevat yhteydet mahdollistavat suurtenkin tietomäärien siirrot ja taustalla toimivat verkko-ohjelmat esim. ohjelmistopäivitysten jakeluun. Jatkuva yhteys verkkoon muodostaa kuitenkin myös uhkan, jota torjumaan tarvitaan palomuuuri ja virustorjuntaohjelma. Perustason nettiohjelmat tulevatkin nykyään käyttöjärjestelmän mukana.

2.3.1 Staattisista dynaamisiin web-sivuihin

Aluksi web-sivut olivat staattisia, pelkkää HTML-koodia sisältäviä tiedostoja ja kuvia, joten niiden jakamiseen tarvittavan palvelimenkaan ei tarvinnut olla erityisen tehokas. Tällainen sivusto on nykyäänkin käyttökelpoinen, mutta sen ylläpito vaatii paljon aikaa ja osaamista. Vähitellen erilaiset dynaamiset, käyttäjäkohtaisesti luodut

sivut mahdollistavat tekniikat yleistyivät. Ensin kyse oli vain yksinkertaisista CGI-skripteistä, joilla HTML-koodin joukkoon oli mahdollista upottaa esim. palvelimen kellonaika tai käsitellä yksinkertainen HTML-lomake. Microsoft tarjosi IDC/HTX -tekniikkaa (Internet Database Connector / HTML Extension), jolla oli pääsy tietokantoihin. Dynaamisuuden toteuttaminen vaati kuitenkin usein eri tekniikoiden sovitamista toimimaan yhdessä eikä ollut kovin tehokasta, mutta riitti aikansa tarpeisiin.

Microsoft esitteli vuonna 1996 ASP-tekniikan, jolla oli mahdollista toteuttaa web-sovelluksia, jotka hakevat monipuolisesti tietoja tietokannasta ja esittävät ne sivua pyydettyä käyttäjälle. ASP perustuu palvelimella ajettavaan, tulkittavaan VBScript-kieleen (vaihtoehtoisesti JScript), jolla voi käsitellä viittä ASP:n oliomallin mukaista objektia (Request, Response, Server, Application ja Session). ASP sisältää helpon tavan käsitellä tietokantoja ADO-rajapinnan kautta. Käyttäjän on mahdollista suorittaa erilaisia hakuja ja toisaalta sivuston ylläpito voidaan toteuttaa selainpohjaisesti: ylläpitäjä syöttää julkisella puolella näkyvät tiedot tietokantaan erityisten ylläpitosivujen kautta. ASP on hyvä esimerkki tekniikasta, joka mahdollistaa hajautetun tietojärjestelmän toteuttamisen. ASP:tä vastaava avoimen lähdekoodin tekniikka on PHP ja Java-maailmassa JSP.

Perinteiset dynaamisten sivujen ohjelmointitekniikat eivät skaalaudu kovin hyvin sivuston käyttömäärien kasvaessa. ASP:n tapauksessa VBScript koettiin rajoittuneeksi ohjelmointikieleksi ja ympäristö ei tarjoa valmiita työkaluja esim. käyttäjän autentikointiin. Microsoftin ratkaisu näihin ongelmiin on ASP.NET, joka on osa .NET Framework -nimistä ohjelmistokokonaisuutta. Frameworkin ensimmäinen versio julkaistiin vuonna 2002 ja vuoden 2007 lopussa uusin versio on .NET Framework 3.5. Framework on sitä varten kehitettyjen Windows- ja web-ohjelmien ajoympäristö, joka perustuu .NET luokkakirjastojen käyttöön ja tarjoaa mm. yhteydet tietokantoihin (ADO.NET). Framework on Microsoftin vastine Sun Microsystemsin Javalle, josta lisää hieman myöhemmin.

ASP.NET ei ole .NET Frameworkin periaatteiden mukaisesti sidottu vain yhteen ohjelmointikieleen, vaan sen yhteydessä on mahdollista käyttää monia kieliä. Microsoft tarjoaa sisäänrakennetun tuen kolmelle ohjelmointikielelle: C#, VB.NET ja JScript. Näistä C# (C Sharp) on aivan uusi kieli, joka on kehitetty kilpailijaksi Ja-

va-kielelle. VB.NET on Visual Basicin uudistettu versio ja JScript vanhaa Microsoftin selaimessaan tukemaa skriptikieltä vastaava kieli, jota voi käyttää myös ASP-ohjelmoinnissa. Millä tahansa ohjelmointikielellä tuotettu ASP.NET -koodi käännetään erityiselle välikielelle, joka on huomattavasti nopeampaa suorittaa kuin ASP:n tulkittava VBScript. Välimuistitekniikka parantaa ASP.NETin suorituskykyä vielä entisestään. Käännöksestä huolimatta ASP.NET -sovelluksia ei tarvitse erikseen asentaa, vaan riittää kun ne kopioidaan web-palvelimen hakemistoon. (Refsnes Data W3Schools.)

ASP.NET sisältää ASP:tä merkittävästi laajemman oliomallin, jossa lähes minkä tahansa HTML-elementin ominaisuuksia voi muuttaa ohjelmallisesti. Mukana on myös tehokkaita tiedon esittämiseen kehitettyjä kontrolleja kuten DataGrid, jolla tietokantakyselyn tulosjoukon saa esitettyä taulukkomuodossa muutamalla koodirivillä. .NET Framework puolestaan tarjoaa laajan valikoiman valmiita luokkia esim. päivämäärien ja hajautustaulujen käsittelyyn. ASP.NETin tapahtumankäsittelymalli on kehittyneempi kuin ASP:ssä, koska esim. painonappeihin voi suoraan liittää klikkaustapahtuman kuuntelijan ja ohjelmoijan ei tarvitse huolehtia HTTP:n (Hypertext Transfer Protocol) mukaisista GET- ja POST-sanomista. Myös poikkeuksien käsittelyyn sekä käyttäjän luotettavaan autentikointiin ja rooleihin perustuvaan käyttöoikeuksien hallintaan on useita valmiita käytäntöjä, jolloin ohjelmoijan ei välttämättä tarvitse toteuttaa sovellukseen omia tunnistusrutiinejaan. Monet sovelluksen asetuksiin liittyvät asiat on koottu XML-muotoiseen tekstitiedostoon web.config, joka on vastaava kuin ASP:n asetustiedosto global.asa, mutta sisältää monia sellaisia asetuksia, jotka aikaisemmin vaativat muutoksia web-palvelimen asetuksiin tai jopa käyttöjärjestelmän rekisteriin. (Refsnes Data W3Schools.)

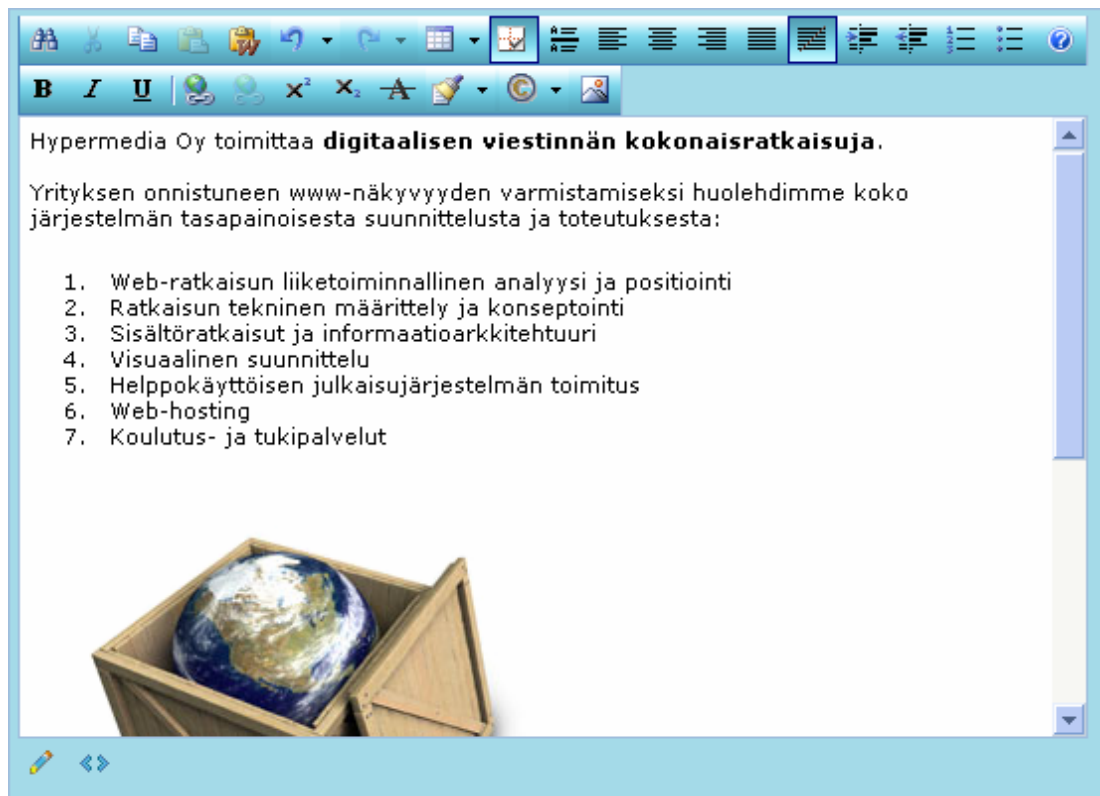
Yhteenvetona voidaan todeta, että ASP.NET on kehittynyt merkittävästi vanhasta ASP-tekniikasta, jota nykyään kutsutaan klassiseksi ASP:ksi, ja vastaa huomattavasti paremmin ammattimaisen web-sovelluskehityksen tarpeita. ASP-sovellukset eivät ole suoraan yhteensopivia ASP.NETin kanssa ja käytännössä uudemman tekniikan lähestymistapa eroaa niin paljon vanhasta, että täyden potentiaalinen hyödyntämiseksi sovellukset kannattaa automaattikonversion sijaan suunnitella kokonaan uudelleen. ASP.NETin uudemmat versiot 2.0 ja 3.5 sisältävät vielä kehittyneempiä ominaisuuksia mm. sivujen ulkoasun ja ohjelmakoodin erottamiseen.

Microsoftin ohella myös Sun Microsystems on kehittänyt internetissä yleisesti käytettyjä Java-kieleen perustuvia ohjelmointitekniikoita. Java kehitettiin alun perin sulautettujen laitteiden ohjelmointiin, mutta se alkoi yleistyä selaimissa ajettavien sovelmien eli applettien muodossa. Java-kieltä voi käyttää myös palvelimella ajettavien servlettien ja JSP-sivujen ohjelmointiin. J2EE-teknologiaa tukeva sovelluspalvelin on Java-maailman vastine Microsoftin .NET Frameworkille ja IIS web-palvelimelle. Se on hieman vanhempaa ja siten koetellumpaa tekniikkaa kuin Microsoftin web-palveluiden alusta, joten J2EE on ainakin toistaiseksi yleisempi isoissa, pitkään kehitetyissä järjestelmissä. (Ahonen, Hämeen-Anttila & Åstrand 2003, 29-30.)

2.3.2 Web-pohjaisten järjestelmien haasteet

Teknisestä alustasta riippumatta eräs tärkeä web-sovellusten käytettävyyteen vaikuttava asia ovat riittävän nopeat vasteajat. Web-järjestelmissä nopeus korostuu, koska sivun päivitys palvelimelta kestää aina jonkin aikaa. TCP/IP ei ole läheskään reaaliaikainen protokolla ja runsaan kuvituksen tai monimutkaisten taulukkorakenteiden näyttö vie selaimelta aikaa nopeallakin tietokoneella ja yhteydellä. AJAX-tekniikka parantaa käytettävyyttä mahdollistamalla vain tarvittavien osien siirtämisen verkon yli. Se yhdistää JavaScriptiä, XML:n mukaista HTML-koodia (XHTML) sekä CSS:ää (Cascading Style Sheets). AJAX on parhaimmillaan vuorovaikutteisten toimintojen yhteydessä, jolloin viiveettömyys on käyttäjälle erityisen tärkeää.

Esimerkkinä AJAX-tekniikan käytöstä on Hyperscriptor XG:n WYSIWYG-tekstinmuokkaustyökalu (kuva 1), jossa tekstin muokkaus ja muotoilut tehdään selaimessa ja vasta valmis teksti lähetetään palvelimelle HTML-koodina. Tekstinmuokkaustyökalu on hyvä esimerkki muistakin hyvän käytettävyyden piirteistä, koska se esittää tehdyt muutokset havainnollisesti ja samaa työkalua käytetään kaikissa tekstin muokkausta vaativissa toiminnoissa. Työkalu on toteutettu kaupallisella Telerik RadEditor-komponentilla, mutta esim. Microsoft tarjoaa ASP.NET -ympäristöön integroitua AJAX-laajennusta ja siihen liittyviä kontrolleja käyttöliittymien toteuttamiseen alusta alkaen.



Kuva 1. Hyperscriptort XG:ssä käytetty tekstinmuokkaustyökalu

Tiedon julkaisun lisäksi web-sivustoilla on muitakin käyttötarkoituksia, koska niiden kautta voidaan toteuttaa kokonaisia tietojärjestelmiä. Perinteisesti sovellusohjelmat on ostettu kaupasta pahvilaatikossa ja asennettu työasemiin rompuilta. Kappaleessa 2.2.1 kuvattujen hajautettujen tietojärjestelmien myötä koko sovelluksen ei enää tarvitse olla käyttäjäorganisaation tiloissa asennettuna. Itse asiassa web-selainta käytettäessä käyttäjien koneille ei tarvitse asentaa käyttöjärjestelmän lisäksi mitään muuta ja internetin ansiosta sovelluksia voidaan käyttää verkon läpi. Tästä seuraa mm. seuraavia etuja:

- Sovellusten asennus, ylläpito ja päivitykset sekä tietojen varmuuskopiointi jäävät sovelluspalveluntarjoajan vastuulle, käyttäjäorganisaatio ei välttämättä tarvitse omia asiantuntijoita näihin erikoistehtäviin.
- Vuokrausmallin avulla tietojärjestelmän hankkimiskustannukset (palvelinlaitteisto ja -lisenssit) voidaan tasata pidemmälle aikavälille.
- Järjestelmän etäkäyttö on helpompaa, koska käsiteltävät tiedot ovat ajan tasalla ilman synkronointia ja eräajoja.

- Web-käyttöliittymän ansiosta järjestelmää on helppo oppia käyttämään, työntekijöiden tuottavuus paranee (edellyttää hyvää käyttöliittymäsuunnittelua).

Internetin läpi käytettävillä sovelluksilla on perinteisiin sovelluksiin verrattuna myös haittapuolia, joista keskeisimpiä ovat:

- Järjestelmä on haavoittuvampi kuin perinteiset sovellukset, koska sen toiminta riippuu jatkuvasti useammista osapuolista (sovelluspalveluntarjoaja, internet-yhteyden tarjoaja).
- Internetin avoimen luonteen vuoksi järjestelmää on helpompi häiritä esim. palvelunestohyökkäyksellä.

Web-sovellukset ovat yleistyneet voimakkaasti tällä vuosikymmenellä ja korvanneet monia työasemaan asennettavia sovelluksia. Googlen ja Microsoftin visioissa jopa perinteiset toimistosovellukset korvautuisivat netin kautta toimivilla ohjelmilla. Tällaisia sovelluksia kutsutaan lyhenteellä RIA (Rich Internet Application), joka tarkoittaa työpöytäsovelluksen web-versiota. Perinteinen web-sovellus muodostuu joukosta sivuja, joiden välillä käyttäjä navigoi linkkien tai web-lomakkeiden toimintopainikkeiden kautta. AJAX-tekniikka lisää tähän mahdollisuuden suorittaa osa toiminnoista selainpäässä. RIA hyödyntää AJAX:ia tai vastaavaa tekniikkaa luodakseen vaikutelman selaimessa toimivasta työpöytäsovelluksesta, jossa ei enää navigoida lomakesivulta toiselle, vaan käyttöliittymä on vuorovaikutteisempi ja se päivittää vain tarvittavat tiedot palvelimelta. (Maurer 2006.)

RIA-sovellukset ovat muita web-sovelluksia vaikeampia suunnitella, koska tietojen esitystavan lisäksi täytyy suunnitella niiden välinen vuorovaikutus, esim. hiirellä suoritettavat vedä ja pudota -toiminnot. Lisäksi vuorovaikutteisten kontrollien on oltava tarpeeksi havainnollisia, jotta loppukäyttäjät osaavat käyttää niitä tehokkaasti. Erilaiset työkaluvihjeet ja korostukset ovatkin tyypillisiä RIA-sovelluksille, jotta käyttäjä erottaa päivitetetyt tiedot. RIA-sovelluksina on menestyksekkäästi toteutettu mm. sähköpostiohjelmaa ja karttapalveluita. (Maurer 2006.) Täytyy kuitenkin muistaa, että on myös sovellusalueita, joilla web-sovelluksilla ei ole juurikaan käyttömahdollisuuksia, vaikka käytössä olisi kuinka edistyksellisiä käyttöliittymiä. Tällai-

sia ovat esim. kuvan, äänen tai videon käsittelyyn liittyvät sovellukset, joiden käsittelemät datamäärät ovat suuria ja edellyttävät kovia reaaliaikavaatimuksia.

Parhaiten web-sovellukset sopivat vuorovaikutteiseen monen käyttäjän käyttöön. Nämä tarpeet ymmärättäviä sovelluksia kutsutaan nimellä Web 2.0. Se on yleisnimi kaikille seuraavan sukupolven nettisovelluksille, joissa on piirteitä verkkopäiväkirjoista (blogit) ja yhteisöllisyydestä (esim. verkkotietosanakirja Wikipedia). Web 2.0 ei sisällä mitään tiettyä tekniikkaa, joten se on ylemmän tason käsite kuin RIA tai AJAX. Monet tyypilliset Web 2.0 -konseptin mukaiset sovellukset ovat keränneet nopeasti suosiota, esim. videoiden jakopalvelu YouTube ohitti käyttömäärillään BBC:n verkkopalvelun vain reilu vuosi aloittamisensa jälkeen. Jacob Nielsenin tutkimuksen mukaan 90 % yhteisöllisten palveluiden käyttäjistä ei kuitenkaan tuota mitään sisältöä palveluihin ja suurin osa uudesta sisällöstä on yhden prosentin käyttäjäsouden varassa. Käyttäjien luoma sisältö on keskeisessä roolissa, mutta Web 2.0:aan liittyy myös uusia ansaintamalleja kuten käyttäjien ottaminen mukaan tuotteiden ideointiin ja kehittämiseen. (Hintikka 2007.) Koska termi Web 2.0 on hyvin löyhästi määritelty, täytyy vain toivoa ettei siitä muodostu samanlaista kuplaa kuin vuosituhaten vaihteen uudesta taloudesta.

Lopuksi todettakoon, että selaimella käytettävien web-sovellusten lisäksi TCP/IP-verkkojen päällä ajetaan monia muitakin sovelluksia, jotka hyödyntävät internetin maailmanlaajuista levinneisyyttä. Esimerkkinä näistä voisi mainita IP-puhelut (VOIP, Voice over Internet Protocol), joiden avulla tietokoneesta tai muusta soveltuvasta päätelaitteesta saa muodostettua puheyhteyden mihin tahansa maailman kolkkaan, jossa vain on nettiyhteys ja puhelun vastaanottaja.

2.4 Ohjelmistotuotannon kehitys

Edellä on jo useasti viitattu siihen, että tietoteknisten ohjelmistojen laajuus ja käyttökohteiden määrä ovat jatkuvassa kasvussa; ohjelmistot ovat yhä ”älykkäämpiä” ja monipuolisempia. Tämä aiheuttaa lisääntyviä haasteita paitsi ohjelmointityölle myös käyttöliittymille, jotta ohjelmistojen suorituskyky, luotettavuus, käytettävyys ja muut laatuun vaikuttavat seikat saadaan pidettyä riittävän korkealla tasolla ohjelmistopro-

jektien yhä kiristyvien aikataulujen puitteissa. Tässä kappaleessa käsitellään ensin ohjelmointikielten kehitystä ja siirrytään sitten ohjelmistotuotannon menetelmien tarkasteluun. Monia asioita on käsitelty jo edeltävissä kappaleissa hieman eri näkökulmasta. Lopuksi tarkastellaan yleisiä projektinhallintaan liittyviä asioita.

2.4.1 Konekielestä olio-ohjelmointiin

Ensimmäisiä mikroprosessoriin perustuvia tietokoneita ohjelmoitiin suoraan prosessorin konekielellä (assembler). Käskyt olivat yksinkertaisia ja niitä oli vähän, mutta pienenkin toiminnon toteuttamiseen tarvittiin valtava määrä käskyjä. Ohjelmat olivat pieniä ja laitteiden suorituskykyyn verrattuna tehokkaita, mutta koodiltaan vaikeasti luettavia. Kun tietokoneiden suorituskyky lisääntyi, kehitettiin ohjelmointikieliä, jotka kuvasivat ohjelman toimintaa yleisemmällä tasolla, esim. erilaisina ehtolauseina ja silmukoina. Nämä ohjelmat piti erikseen kääntää konekielelle, jolloin kääntäjä-ohjelma tuotti tarvittavat prosessorin käskyt. Esimerkkejä käännettävistä kielistä ovat C ja C++, jotka kehitettiin 1970- ja 80-luvuilla. Ohjelmointi oli yleensä proseduraalista eli ohjelma eteni alusta loppuun haarautuen tarvittavissa kohdissa, mutta C++ mahdollisti myös olio-ohjelmoinnin eli ohjelmien koostamisen itsenäisistä kokonaisuuksista. Kehitettiin myös tulkattavia ohjelmointikieliä, joita ei tarvinnut kääntää, mutta jotka ajettiin erityisen tulkin kautta. Tunnetuin tulkattava kieli on Basic, joka kehitettiin alun perin ohjelmoinnin opiskeluun. Tulkattavat kielet olivat helpompia oppia ja käyttää, mutta eivät niin suorituskykyisiä kuin käännettävät kielet.

Microsoft julkaisi Basicista kehitetyn Visual Basic -kielen vuonna 1991 Windows-ohjelmien tekemiseen. Graafisessa käyttöliittymässä ohjelmaa ei ajeta ”alusta loppuun”, vaan se reagoi erilaisiin tapahtumiin kuten hiiren klikkauksiin. Visual Basic oli käännettävä kieli, joka tarvitsi ohjelman ajon aikana kieleen kuuluvan dynaamisen linkkikirjaston (DLL, Dynamic Link Library). Ohjelmat eivät olleet erityisen tehokkaita ja siksi kaupallisten ohjelmien kielenä oli 1990-luvulla yleensä C++. Vuonna 1995 Sun Microsystems esitteli oliopohjaisen Java-kielen, josta tuli nopeasti suosittu sen tehokkuuden ja turvallisuuden takia. Javan syntaksi on hyvin lähellä C++:aa, mutta siinä ajoympäristö huolehtii asioista, joiden unohtaminen C++-ohjelmasta aiheuttaa helposti muistivuotoja tai ohjelmien kaatumista. Java yhdistää

käännettävien ja tulkattavien kielten parhaat puolet, kun ohjelma käännetään erityisesti Java-tavukoodiksi ja tulkataan ajon aikana nopealla Java-virtuaalikoneella. Eri käyttöjärjestelmille tehdyt virtuaalikoneet pystyvät ainakin periaatteessa ajamaan samaa tavukoodia. (Ahonen, Hämeen-Anttila & Åstrand 2003, 22-23.)

2000-luvulla Microsoft vastasi Javan haasteeseen esittelemällä .NET Frameworkin päälle ohjelmakoodia tuottavan Visual Studio .NET sovelluskehitysympäristön, joka käyttää tehokkaasti Frameworkin tarjoamia luokkia. Visual Studio oli tuotenimenä ollut olemassa jo 1990-luvulla, mutta silloin se edusti Microsoftin eri ohjelmointityökalujen (Visual Basic, Visual C++, Visual InterDev) kokoelmaa, jonka .NET uudisti täysin. C++ sai rinnalleen uuden ohjelmointikielen C#:n, joka on hyvin paljon Javan kaltainen korjaten suunnilleen samat C++:n puutteet kuin Java. Myös Visual Basic .NET alkoi tuottaa tehokasta käännettyä koodia, jota MSIL-tulkki (Microsoft Intermediate Language) suorittaa. Microsoftin visiossa ohjelmointikielenä voikin käyttää mitä tahansa .NET -kieltä ja lopputuloksena on aina yhtä tehokas ohjelma. Ohjelmat toimivat ainoastaan Windows-käyttöjärjestelmissä, joskin avoimen lähdekoodin yhteisö on kehittänyt ajoympäristöä myös Linuxille. Uudemmissa versioissa .NET on jätetty pois tuotenimistä (Visual Studio 2008, Visual Basic 2008 jne.)

2.4.2 Ohjelmistotuotannon menetelmät

Tietokonesovellusten käytössä olevat resurssit ovat aina jollain tavalla rajattuja, mutta kuten kappaleessa 2.1 todettiin, esim. muistitilaa on nykyään käytettävissä aivan eri luokkaa oleva määrä kuin pari vuosikymmentä sitten, vaikka sovellusten laajeneminen huomioitaisiin kehitystä tarkasteltaessa. Muistipaikkoja ei enää tarvitse säästää käyttämällä yhtä muuttujaa useaan eri tarkoitukseen ohjelmassa tai sekoittamalla suoritettavaa koodia ja dataa keskenään. Myöskään pelätyn vuosi 2000 -ongelman aiheuttaneeseen vuosituhannen ja vuosisadan poisjättämiseen tallennettavasta päivämäärästä ei ole enää ainakaan kapasiteettiongelma johtuvia syitä. Voidaan siis keskittyä ohjelmakoodin rakenteen selkiyttämiseen, jotta laajat kokonaisuudet pysyvät hallinnassa. Virheettömiä nykyisetkään ohjelmat eivät silti ole ja tietoturvan osalta virheiden merkitys vain kasvaa.

Ohjelmistotuotannolla (software engineering) tarkoitetaan hallittua prosessia, jossa ohjelmistokehityksen ammattilaiset suunnittelevat ja toteuttavat tietokoneohjelmia työasemiin, palvelimiin tai sulautettuihin järjestelmiin. Ohjelmien tulee vastata käyttäjien niille asettamia vaatimuksia, jotka voivat ajan mittaan muuttua. Ohjelmistotuotannon historia on lyhyt, mutta tekniikan nopea kehitys on tehnyt ohjelmistoista monimutkaisimpia insinööritöitä ihmiskunnan historiassa. Alan nuoruus huomioiden ei siis ole mikään ihme, että ohjelmia pitää usein korjata ja niiden kehittämiseen ei ole vakiintuneita systemaattisia menetelmiä. (Haikala & Märijärvi 2006, 25-26.)

Koko teollisen ohjelmistotuotannon olemassaolon ajan on pyritty kehittämään menetelmiä suurten kokonaisuuksien ja ruohonjuuritason ohjelmointityön välisen yhä suurenevan etäisyyden mallintamiseen. Toimiva mallinnusmenetelmä estää ohjelmistoprojektien henkilöitymisen niin, että vain projektiin syvällisesti perehtyneet henkilöt ymmärtävät kokonaisuuden ja sen yksityiskohtien väliset yhteydet. Näiden yhteyksien ymmärtäminen on ohjelmien jatkokehityksen kannalta välttämätöntä. Mallinnuksen täydellisyys ja ymmärrettävyys ovat kuitenkin keskenään ristiriitaisia ominaisuuksia: tarkkuuden kasvaessa ymmärrettävyys heikkenee ja päinvastoin.

Perinteisesti ohjelmistojen toimintaa on mallinnettu erilaisilla tila- ja vuokaavioilla. Joillekin sovellusalueille on myös kehitetty omia, esim. sulautetun järjestelmän erityistarpeet huomioivia menetelmiä mallinnukseen. Yksi esimerkki nykyisin käytössä olevasta yleiskäyttöisestä mallinnusmenetelmästä on UML (Unified Modeling Language). Se kehitettiin Object Management Groupin toimesta kymmenen vuotta sitten ja uusin versio 2 on vuodelta 2004. UML:n järjestelmämalli perustuu graafisiin kaavioihin, joilla mallinnetaan järjestelmän rakennetta, käyttäytymistä ja vuorovaikutusta tarvittaessa hyvinkin yksityiskohtaisesti. (Wikipedia.)

UML:n luokka- ja oliokaavioissa järjestelmän rakenteen mallinnuksen lähestymistapa on olio-ohjelmoinnin mukainen, koska niissä järjestelmä pyritään mallintamaan olioina, joita järjestelmään ohjelmoitavat luokat (class) vastaavat. Olion ilmentymällä on määritellyt metodit, joiden avulla sitä kutsutaan, mutta olion rakenne tai sisäinen tila ei näy ulospäin. Tällöin olioita voidaan kehittää itsenäisesti järjestelmän kehitystarpeiden mukaan. Järjestelmän käyttäytymistä mallinnetaan yleisellä tasolla UML:n aktiviteettikaavioilla ja yksityiskohtaisemmin käyttötapa-kaavioilla. Olioi-

den keskinäistä vuorovaikutusta kuvataan puolestaan mm. sekvenssi- ja kommunikaatiokaavioilla. (Wikipedia.)

Ohjelmistojen laatu muodostuu useista osatekijöistä, ja laadukkaan ohjelmiston tuottaminen edellyttää toimivan mallinnusmenetelmän lisäksi systemaattisia toimintatapoja, valvontaa ja testausta. Hallittu suunnitteluprosessi on entistä keskeisemmässä asemassa järjestelmien koon kasvaessa, koska perinteisillä testausmenetelmillä ei enää saavuteta riittävää kattavuutta. Erilaiset elinkaarimallit jäsentävät eri vaiheissa käytettävät menetelmät ja prosessit, joihin ohjelmistotuotannon työkalujen käyttö parhaimmillaan perustuu. Vesiputousmallissa ohjelmiston elinkaari muodostuu alla luetelluista toisiaan seuraavista vaiheista, joita toistetaan iteratiivisesti kunnes voidaan siirtyä ”vesiputouksen” seuraavalle tasolle (Haikala & Märijärvi 2006, 35-36):

- määrittely: vaatimusmäärittely, toiminnallinen määrittely
- suunnittelu: ohjelmistosuunnittelu, tietokannan suunnittelu, käyttöliittymäsuunnittelu
- toteutus: uusien toimintojen ohjelmointi, koodikirjastosta saatavien toimintojen räätälöinti, valmiiden komponenttien konfigurointi
- testaus: yksikkötestaus, integraatiotestaus, beta-testaus
- käyttöönotto: tietosisällön siirto, asennus ja integrointi muihin järjestelmiin, käyttäjien ja ylläpitäjien koulutus
- ylläpito: varmistukset, päivitykset, seuranta

Määrittelyvaiheen tarkoituksena on selvittää asiakkaan (ohjelmiston tilaajan, ohjelmistotuotteen kohderyhmän tms.) odotukset ohjelmistolta sekä laatia niitä vastaava toiminnallinen määrittely. Optimaalisessa tapauksessa asiakas tuottaa yksityiskohtaisen vaatimusmäärittelyn, jonka perusteella toiminnallinen määrittely on helppo laatia. Käytännössä tarvitaan usein ohjelmiston tuottajan näkemystä asiakasvaatimusten selvittämiseksi, jolloin puhutaan esiselvityksestä. Määrittelyvaihetta voidaan edelleen jakaa osiin esim. sulautetuissa järjestelmissä tarvittavan laitteiston sekä varsinaisen ohjelmiston määrittelyyn aina yksittäisten toimintojen tasolle asti. (Haikala & Märijärvi 2006, 37-39.) Toiminnallisen määrittelyn perusteella ohjelmiston tuottaja pystyy tekemään asiakkaalle tarjouksen tai hinnoittelemaan tuotteensa.

Suunnitteluvaihe käynnistyy, kun asiakas tilaa ohjelmiston tai ohjelmistotuotetta päätetään ryhtyä toteuttamaan. Määrittelyvaiheessa keskeisin kysymys oli ”Mitä?”, kun taas suunnitteluvaiheessa vastataan kysymykseen ”Miten?” (Haikala & Märijärvi 2006, 40). Suunnittelu etenee rinnakkain ohjelmalogiikan, tietorakenteiden ja käyttöliittymän osalta. Suunnittelu voidaan myös jakaa yleisempään arkkitehtuurisuunnitteluun sekä yksityiskohtaisempaan modulisuunnitteluun, jolloin testaussuunnitelma voidaan tehdä modulin tarkkuudella (Haikala & Märijärvi 2006, 81).

Kun ohjelmisto on enemmän tai vähemmän valmiiksi suunniteltu, sitä aletaan toteuttaa eli ohjelmoida valitulla ohjelmointikielellä sekä muuten valmistaa mm. graafisten elementtien osalta. Osa toteutuksesta voi syntyä automaattisesti esim. UML-mallista. Toteutuksessa voidaan käyttää apuna aikaisemmin kehitettyjä tai kolmansilta osapuolilta hankittuja ohjelman osasia (koodikirjastot ja valmiit ohjelmistokomponentit). Kolmannen osapuolen komponenttien hallintaan liittyy omia erityisvaatimuksia kuten yhteensopivuuden varmistaminen ja mahdollisten päivitysten hankkiminen. Joissain tapauksissa toteutus on lähes pelkkää valmiin sovelluksen konfigurointia.

Ohjelmakoodi voidaan kirjoittaa tekstieditorilla, joka esim. sisentää koodia automaattisesti ja värjää ohjelmointikielen varatut sanat ja funktioiden nimet. Tällöin koodin kääntäminen ja mahdollinen virheiden etsintä (debug) suoritetaan eri ohjelmilla. Markkinoiden toista ääripäätä edustavat laajat integroidut kehitysympäristöt (IDE, Integrated Development Environment, kuten kappaleessa 2.4.1 esitelty Visual Studio), joissa sovelluksia käsitellään projekteina ja ohjelma käännetään ja suoritetaan kehitysympäristön sisällä. Tällöin on mahdollista esim. merkitä tietty ohjelmakoodin rivi, jolle ohjelman suoritus pysähtyy (breakpoint), ja sen jälkeen tarkastella ohjelman sisäisten muuttujien arvoja ilman että niitä tarvitsisi ohjelmassa tulostaa näkyviin. Koodieditori osaa myös ehdottaa objektille sopivia ominaisuuksia ja metodeja. Tällainen työskentely nopeuttaa merkittävästi ohjelmointityötä ja testausta.

Testaus suoritetaan järjestelmän toteutuksen rinnalla ainakin kolmessa vaiheessa. Testauksen V-mallissa järjestelmän osia tai suurempia kokonaisuuksia verrataan vastaaviin määrittelydokumentteihin. Järjestelmän ohjelmoinnista vastaava henkilö testaa koodia tuottaessaan jokaisen toiminnon tai osatoiminnon niiden kirjoittamisen jälkeen (yksikkö- eli modulitestaus). Suuremman kokonaisuuden valmistumisen jäl-

keen suoritetaan toinen testausvaihe eli integraatiotestaus. Jokaisen testauksen jälkeen havaitut virheet korjataan ja tehty testaus uusitaan, kunnes toiminto havaitaan toimivaksi. Ainakin integraatiotestaus pyritään suorittamaan eri henkilön toimesta kuin varsinainen ohjelmointi. Eri henkilön tekemä kattava testaus parantaa testauksen luotettavuutta sekä nostaa mahdollisten virheiden löytymistodennäköisyyttä ennen järjestelmätestausta eli beta-testausta (kolmas testausvaihe). (Haikala & Märijärvi 2006, 40.)

Viimeinen testausvaihe on asiakkaan toteuttama tai yhdessä asiakkaan kanssa toteutettu beta-testaus, jossa asiakas testaa valmista järjestelmää todellisessa käytössä. Asiakas voi tässä vaiheessa syöttää järjestelmään jo tuotantoaineistoa ja käyttää järjestelmää todellisten tarpeidensa mukaisesti, jolloin mahdolliset viat sekä väärin toimivat toiminnot saadaan paremmin paikallistettua. Kaikkien virhetilanteiden raportointi ohjelmoijille on tärkeää, jotta virheet saadaan korjattua ennen järjestelmän tuotantokäyttöön asennusta.

Yksi tärkeä koko ohjelmistotuotannon läpäisevä prosessi on tehtyjen suunnitelmien ja toteutuksen dokumentointi. Määrittelyvaiheen dokumentit ovat ohjelmistoprojektin onnistumisen kannalta kriittisimpiä, mutta dokumentointia tarvitaan myöhemmissäkin vaiheissa. Erilaisten dokumenttien tuottamisessa voidaan käyttää valmiita pohjia tai käytetty mallinnusmenetelmä voi edellyttää tietynlaisia dokumentteja. Myös ohjelmakoodin kommentit ja käyttöohjeet kuuluvat dokumentaatioon. Kunnolla tehty dokumentointi ja siihen liittyvä versionhallinta helpottaa oleellisesti ohjelmiston ylläpitoa. Myös dokumentteja täytyy ylläpitää.

Edellä esitettyä vesiputousmallia sovelletaan usein käytännön työssä varsinkin suuria ohjelmistoja tuottaessa. Vesiputousmallin käyttö asiakasprojektissa edellyttää mallin mukaisiin käytäntöihin sitoutumista myös asiakkaan taholta. Pienemmissä ohjelmistoissa saatetaankin tarvita kevyempiä malleja ja mallinnusmenetelmiä, koska asiakas ei osaa tulkita formaalin menetelmän (esim. UML) mukaisia dokumentteja. Joissain tapauksissa standardien mukaisten menetelmien käyttö saattaa hidastaa määrittelyvaihetta enemmän kuin se nopeuttaa toteutusvaihetta. Lisäksi nykyisille ohjelmistoille on tyypillistä, että niiden toiminnalle asetetut vaatimukset saattavat muuttua hyvin nopeasti. Tällöin kaikenkattava malli jouduttaisiin muodostamaan osittain uu-

delleen ja vesiputousmallissa siirtymään ”vastavirtaan”. Siksi ohjelmistotuotantoon on kehitetty ns. ketteriä menetelmiä (agile software development), jotka soveltuvat paremmin pienempien kehitysprojektien käytännön tarpeisiin, erityisesti web-sovellusten tekoon. Beck ym. (2001) esittelivät manifestissaan ketterän ohjelmistokehityksen periaatteet seuraavasti:

- Yksilöt ja vuorovaikutus ovat tärkeämpiä kuin prosessit ja työkalut.
- Toimiva ohjelmisto on tärkeämpi kuin kattava dokumentaatio.
- Yhteistyö asiakkaan kanssa on tärkeämpää kuin sopimusneuvottelut.
- Muutokseen vastaaminen on tärkeämpää kuin suunnitelman seuraaminen.

Ketterä ohjelmistokehitys sallii perinteisiä menetelmiä paremmin muutoksia ohjelmistolle asetettuihin vaatimuksiin kesken tuotantoprosessin; itse asiassa muutokset ovat jopa toivottuja. Koko ajan pyritään tuottamaan asiakkaalle jotain valmista, jota sitten voidaan muuttaa. Nopealla reagoinnilla pystytään tuottamaan ohjelmistoon asiakkaalle enemmän lisäarvoa. Jatkuva vuorovaikutus on avainasemassa, ja asiakas voidaan jopa ajatella osaksi tuotantotiimiä. Turhan monimutkaisuuden välttäminen, parhaiden käytäntöjen valitseminen ja niiden tehokas soveltaminen tuottavat ketterässä ohjelmistokehityksessä parhaan lopputuloksen. (Beck ym. 2001.)

Protoilu on yksi vesiputousmallista kehitetty menetelmä, joka soveltuu myös ketterään ohjelmistokehitykseen. Siinä ohjelmisto pyritään esittämään valmiin näköisenä heti määrittelyvaiheessa ja toimintoja määritellään niiden toteutuksen kautta. Tällöinkin ohjelmakoodi kirjoitetaan vasta sitten, kun toiminto on määritelty. Protoilussa yksi iteraatiokierros kattaa vesiputousmallin kaikki vaiheet ja jopa järjestelmän käyttöönotto voidaan vaiheistaa eli ensin tuotantoon otetaan järjestelmän perusversio, johon sitten lisätään ominaisuuksia tarpeen mukaisessa järjestyksessä.

Lopuksi vielä muutama sana avoimen lähdekoodin (open source) ohjelmistoista, joiden käyttäjä saa käyttöönsä myös kääntämättömän ohjelmakoodin. Lähdekoodia voi käyttää esim. omien tarpeiden mukaisten muutosten tekemiseen ja jatkolevitykseen. Laajojen avoimien ohjelmistojen (esim. Linux-käyttöjärjestelmä) toteutus tehdään usein hajautetusti, mikä aiheuttaa erityisvaatimuksia niiden tuotannossa käytetyille

menetelmille. Versionhallinta ja yhteensopivuuden varmistaminen ovat sitä keskeisempiä mitä laajemmasta ja useamman kehittäjän projektista on kyse.

Avoimen lähdekoodin ratkaisut ovat lisääntyneet viime vuosina ja ne tarjoavat kilpailukykyisen vaihtoehdon kaupallisille ohjelmistoille silloin, kun löytyy ohjelmistolle asetetut vaatimukset mahdollisimman hyvin täyttävä valmis kokonaisuus. Monesti asiakaskohtaisen ohjelmiston tuottaminen avoimen koodin osasista kokoamalla ei kuitenkaan tuota yhtä hyvää lopputulosta kuin sovelluksen ohjelmointi alusta alkaen tai tuotteistetun ohjelmiston asiakaskohtainen räätälöinti. Useimmiten suljettuun koodiin perustuvat kaupalliset ohjelmistot ovatkin edelleen käytettävyydeltään ja valmiustasoltaan avoimen koodin ratkaisuja edellä, vaikka välimatka on pienentynyt viimeisen kymmenen vuoden aikana.

2.4.3 Projektinhallinta

Ohjelmistot toteutetaan usein projekteina, jolloin ohjelmistotalot ovat tuotekehitystai toimitusprojekteja toteuttavia projektiorganisaatioita. Projektiorganisaatiossa yksittäisen projektin projektiryhmään valitaan projektin ajaksi tarvittavat asiantuntijat sekä projektipäällikkö. Projektille on määritelty tarjouksen, sopimuksen tms. yhteydessä aikataulu- ja kustannusraamit, jotka tarkentuvat projektisuunnitelmaan. Projektipäällikkö vastaa suunnitelman seuraamisesta, projektin valvonnasta ja raportoinnista. Tärkeä osa projektisuunnitelmaa ovat aika-, resurssi- ja kustannusohjauksen menetelmät. Työkokonaisuudet jaetaan itsenäisesti suunniteltaviin ja toteutettaviin tehtäviin, joiden kestot (kustannusvaikutus) ja tarvittavat resurssit (esim. ohjelmoijat) määritellään. Resurssien kuormitusta täytyy seurata erityisesti silloin, jos monet eri projektit käyttävät samoja resursseja. Tavoitteena on, että kuormitus olisi tasaista, mutta ei aivan maksimissa. (Pelin 2004, 25, 85, 93.)

Tehtävillä on keskinäisiä riippuvuuksia, jotka määrittelevät niiden suoritusjärjestyksen ja aikataulun. Kriittinen polku on niiden tehtävien optimoitu ketju, joiden on valmistuttava ajoissa, jotta koko projekti ei viivästy. Kriittistä polkua voi yrittää suojata aika- ja resurssipuskureilla, jolloin projektin sisään jätetään pelivaraa mahdollisten myöhästymisten kiinnisaamiseksi. Kriittisen polun toteutumista seurataan välita-

voitteiden (milestone) avulla, jolloin jokainen välitavoite muodostaa tarkistuspisteen koko projektille. (Pelin 2004, 121-123.)

Myös riskien hallinta on oleellinen osa projektisuunnitelmaa. Riski on määritelmänsä mukaan mahdollinen negatiivinen poikkeama projektin tavoitteista. Riskit voivat liittyä itse projektiin, siinä käytettyyn tekniikkaan tai projektia toteuttavan yrityksen, projektin tilanneen yrityksen tai kolmannen osapuolen liiketoimintaan tai organisaatioon. Riskit tunnistetaan, analysoidaan ja niiden hallinta suunnitellaan projektin alussa sekä tarvittaessa uudelleen projektisuunnitelmaan tehtävien muutosten yhteydessä. Riski on sitä tärkeämpi mitä suurempi sen todennäköisyys tai merkitys projektille on. Tärkeät riskit voidaan joissain tapauksissa kokonaan poistaa suunnitelmia muuttamalla tai riskien todennäköisyyttä pienentää niiden syitä poistamalla. Riskejä voidaan myös siirtää tai jakaa toiselle osapuolelle. Joskus riskin toteutuminen vain hyväksytään ja siihen mahdollisesti valmistaudutaan. Projektipäällikön tehtäviin kuuluu riskien hallinnan budjetti-, aikataulu- ja laatuvaikutusten seuraaminen. (Pelin 2004, 199-204.)

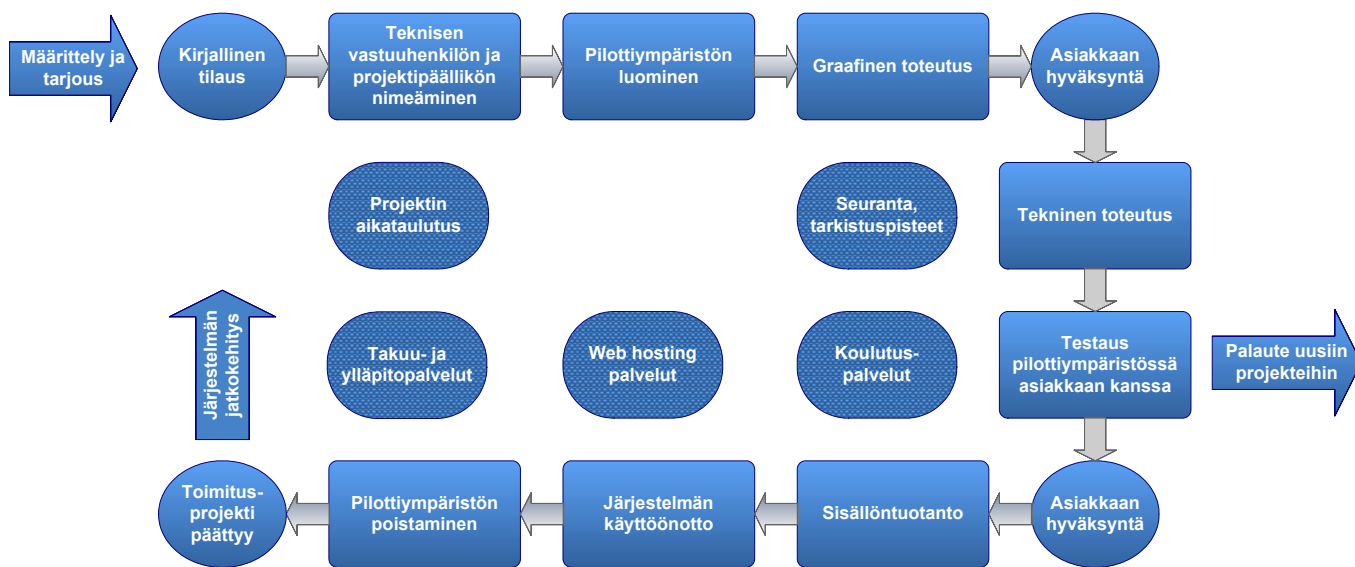
2.5 Hypermedia Oy:n käytännöt ohjelmistotyössä

Tässä kappaleessa kuvataan Hypermedia Oy:n ohjelmistokehitysprosesseja ja järjestelmäprojekteja edellä esitetyn pohjalta. Hypermedia Oy:n toimittamiin selainkäyttöisiin ohjelmistoihin toteutetaan aina enemmän tai vähemmän asiakaskohtaisia räätälöintejä. Sovellusten ulkoasu on yleensä täysin tilaajakohtainen, vaikka sovellus muuten olisi pitkälle tuotteistettu kuten intranet-sivusto. Osa toimitettavista järjestelmistä toteutetaan alusta alkaen asiakkaan toiveiden mukaisesti. Asiakaskohtaisten ohjelmistojen määrittely voidaan toteuttaa erillisenä esiselvitysprojektina käyttäen Hypermedia Oy:n kehittämää Direct1 -menetelmää, jossa vaatimusmäärittely ja toiminnallinen määrittely toteutetaan muutaman yhteisen neuvottelun ja sähköpostitse kerätyn palautteen perusteella.

Asiakaskohtaisen ulkoasun mallintaminen on helppoa, koska luonnoksia voi esitellä valmiin näköisinä kuvina. Uusien ominaisuuksien ja toimintojen mallinnuksessa tarvitaan sen sijaan erilaisia havainnollistamismenetelmiä, joista Hypermedia Oy:n ylei-

simmin käyttämä on käyttöliittymäkuvilla protoilu. Tässä menetelmässä toiminto esitellään asiakkaalle staattisina web-sivuina, joissa tarvittavat tekstikentät, valintapainikkeet ym. elementit ovat suunnilleen oikeassa järjestyksessä ja seuraavalla web-sivulla on havainnollistettu edellisen sivun valintojen vaikutusta. Malliin voi nopeasti tehdä muutoksia tai lisää esimerkkisivuja. Tällainen lähestymistapa tarjoaa vähemmän teknistä näkökulmaa omaaville asiakkaille helpomman, joskin hieman epätarkemman, lähestymistavan kuin formaalien menetelmien käyttö.

Hypermedia Oy:n tuottama valmis toiminnallinen määrittely kuvaa toiminnot käyttöliittymäkuvien lisäksi tekstimuodossa sekä tarvittaessa kaavioina, jotka havainnollistavat tekstiä. Dokumentissa ei oteta kantaa teknisiin toteutustapoihin muuta kuin mahdollisen järjestelmäintegraation osalta asiakkaan käytössä oleviin järjestelmiin (tekniset reunaehdot). Asiakas voi käyttää erillisen esiselvitysprojektin tuloksena syntyneitä määrittelyä oman tarjouspyyntönsä liitteenä, jonka perusteella Hypermedia Oy laatii sitovan tarjouksen järjestelmästä.



Kuva 2. Hypermedia Oy:n toimitusprojektin kulku pääpiirteittäin

Järjestelmän varsinainen toimitus alkaa asiakkaan kirjallisella tarjouksen hyväksymisellä eli tilauksella. Toimitusprojektin tärkeimmät vaiheet on esitetty kuvassa 2. Projektille laaditaan aikataulu ja tarvittava määrä tarkistuspisteitä. Hypermedia Oy:ssä projektin teknisenä vastuuhenkilönä toimii yksi henkilö, joka vastaa projektin tekni-

sestä toteutuksesta projektin alusta loppuun saakka. Tällä henkilöllä on useimmiten parhaiten asiantuntemusta kehitettävästä ohjelmistosta ja hän toimii koko järjestelmätoimituksen projektipäällikkönä. Jos järjestelmä sisältää paljon graafisia elementtejä, projektille määritellään Hypermedia Oy:n graafinen yhteyshenkilö, joka vastaa projektin graafisesta suunnittelusta yhdessä tilaajan kanssa. Graafinen yhteyshenkilö voi toimia myös projektin projektipäällikkönä. Tilaaja ilmoittaa omat yhteyshenkilönsä projektiin liittyvissä kysymyksissä.

Hypermedia Oy toteuttaa asiakaskohtaiset web-järjestelmät sekä tuotteistettuihin sovelluksiin tehtävät räätälöinnit toiminnallisen määrittelyn valmistuttua hyvin suoraviivaisesti, ketterän ohjelmistokehityksen periaatteiden mukaisesti. Projektit toteutetaan pienessä tiimissä, jolloin projektiin liittyvien henkilöiden välinen kommunikointi on nopeaa ja vaivatonta. Graafinen suunnittelu toteutetaan ennen järjestelmän teknistä osaa tai limittäin sen kanssa. Hypermedia Oy toimii Microsoft-ympäristössä eli käyttää Windowsin eri versioita työasemissa ja palvelimissa sekä Microsoftin palvelintuotteita kuten SQL Serveriä ja Exchange-sähköpostijärjestelmää. Uudet web-järjestelmät toteutetaan ASP.NET -tekniikalla, mutta tarvittaessa vanhempia ASP-järjestelmiäkin ylläpidetään.

Kun graafinen käyttöliittymä on valmis ja tilaajan hyväksymä, se pilkotaan HTML-sivuksi, joka toimii web-pohjaisten järjestelmien visuaalisena alustana. Pilkkomisen tekee graafinen suunnittelija tai projektin tekninen vastuuhenkilö. Pilkkomisessa käytetään HTML-editoria ja Adobe Photoshop -ohjelmaa. Jossain tapauksissa käyttöliittymiä toteutetaan Flash-tekniikalla, jonka on kehittänyt alun perin Macromedia, jonka Adobe sittemmin osti. Flash on kevyt, interaktiivinen web-animaatio, jota useimmat selaimet tukevat ja joka mahdollistaa erittäin näyttävien, interaktiivisten käyttöliittymien rakentamisen. Flash-pohjainen käyttöliittymä on valmis ilman pilkkomista, mutta se voi vaatia ohjelmointivaiheessa vaihtuvien tekstien tai linkkien viemistä parametreina Flash-objekteille. Tulevaisuudessa Microsoft Silverlight saattaa olla vahva kilpailija Flash-tekniikalle, koska se mahdollistaa objektien tehokkaamman ohjelmallisen käsittelyn.

Pilkottua HTML-sivua voidaan käyttää suoraan ohjelmoinnin pohjana. Ohjelmointi tehdään käyttäen VB.NET -ohjelmointikieltä ja Microsoft Visual Studio -sovellus-

kehittäjä sekä SQL Server -tietokantaa. Ohjelmointi tehdään pilottiympäristössä, jonka projektin tekninen vastuhenkilö rakentaa erityiselle testipalvelimelle. Ohjelmisto ohjelmoidaan ja testataan valmiiksi pilottiympäristössä, joka vastaa tulevaa tuotantoympäristöä. Ohjelmointityöhön kuuluva aika on suoraan verrannollinen toteutettavan järjestelmän tai räätälöinnin laajuuteen. Projektinhallinnan tehtävänä on pitää työ aikataulussa ja asiakas tietoisena toteutuksen tilanteesta.

Pilottiympäristö on eristetty tuotantoympäristöstä, joten testauksen voi suorittaa turvallisesti tietäen, että virhetilanteet ja testausta varten luotu sisältö eivät vaikuta tuotantoversioon. Tavoitteena on, että testauksen suorittaa aina eri henkilö kuin ohjelmakoodin kirjoittanut henkilö, jolloin loogiset ristiriidat tulee helpommin havaittua. Myös asiakas osallistuu yleensä testaukseen erityisesti asiakaskohtaisten ominaisuuksien osalta. Tekninen vastuhenkilö saa testauksen aikana raportteja tuotteen toimivuudesta ja pystyy korjaamaan mahdolliset virhetilanteet. Myös järjestelmän käytettävyyttä sekä selainyhteensopivuutta testataan tässä vaiheessa.

Asiakas voi tuottaa pilottiympäristöön tuotantokäyttöön kuuluvaa aineistoa, kun järjestelmän valmiusaste mahdollistaa tarvittavien ylläpitotoimintojen käytön ja asiakkaan edustajille on annettu riittävä koulutus toimintojen käyttöön. Näin toimien sivusto tai järjestelmä on julkaisuvaiheessa valmiin näköinen myös sisällön osalta. Järjestelmälle varataan tarvittaessa oma verkkotunnus (domain) ja se reititetään valmiiksi tuotantopalvelimen IP-osoitteeseen. Verkkotunnukselle voidaan tietoturvan parantamiseksi hankkia SSL-sertifikaatti (Secure Sockets Layer), jonka avulla selaimen ja palvelimen välinen tietoliikenne salataan.

Kun asiakas on hyväksynyt järjestelmän pilottiympäristössä tekemänsä testauksen perusteella ja tarvittavat aineistot on siirretty, voidaan järjestelmä julkaista eli asentaa lopulliseen paikkaansa (edellä mainittu tuotantopalvelimen IP-osoite). Jos kyseessä on asiakkaan omalle palvelimelle tehtävä asennus, asennuspaketista poistetaan lähdekooditiedostot (.vb-tiedostot) ellei asiakkaan kanssa ole erikseen sovittu järjestelmän toimittamisesta avoimena. Pilottiympäristö säilytetään tulevia kehitysprojekteja tai testausta varten. Jos järjestelmä asennetaan Hypermedia Oy:n palvelimelle (web hosting), koko järjestelmä kopioidaan tuotantopalvelimelle. Tällöin pilottiympäristö

poistetaan, koska tuotantoympäristöstä saa tehtyä tarvittaessa kopion uudeksi pilot-ti ympäristöksi, jos järjestelmää muutetaan.

Käyttöönoton yhteydessä suoritetaan myös tuotannollinen koeajo tilaajan muihin järjestelmiin tehdylle integraatiolle. Samalla huolehditaan, että tietokannasta otetaan säännöllisesti varmuuskopiot ja että mahdollisten päivitysten asennus sujuu asiakkaan teknisen yhteyshenkilön toimesta. Päivitykset lähetetään yleensä sähköpostin liitteinä: sovelluksen muuttuneet tiedostot ja tietokantaan ajettava SQL-tiedosto. Normaalisti järjestelmille tarjotaan kolmen vuoden takuu sekä yhden vuoden maksuttomat tukipalvelut pääkäyttäjälle.

Hypermedia Oy on toteuttanut Microsoftin ASP-tekniikkaan pohjautuvia dynaamisia web-sivustoja 1990-luvun lopusta saakka ja yrityksen Hyperscriptor -julkaisu-järjestelmä perustui ASP-tekniikkaan vuoteen 2005 saakka. Silloin julkaistiin Hyperscriptorin kolmosversio, jota oli kehitetty vuoden verran puhtaalta pöydältä Microsoftin seuraavan sukupolven tekniikan päälle. Nykyisen Hyperscriptor XG versio 3.5:n teknisenä alustana käytetään Microsoft .NET Frameworkia ja sen tarjoamia tietokantarajapintoja SQL Serveriin. Sovellus on kokonaan selainkäyttöinen Windows Serverin IIS-palvelimella toimiva ASP.NET -sovellus, joka on toteutettu VB.NET -ohjelmointikielellä käyttäen Visual Studio -kehitysympäristöä.

Hypermedia Oy:n omat ohjelmistotuotteet perustuvat suljettuun lähdekoodiin, mutta yritys on toimittanut asiakkailleen myös sovelluksia, joissa lähdekoodi on mukana. Molemmissa tapauksissa voidaan käyttää yrityksen kehittämiä valmiita koodikirjastoja. Sovelluksiin kuuluvat kolmansien osapuolten toimittamat ohjelmistokomponentit ovat yleensä suljettuja ja Hypermedia Oy:llä on niihin ns. royalty-free jakelulisenssit, jolloin asiakkaan ei tarvitse maksaa asennuskohtaista lisenssimaksua kolmannen osapuolen komponenteista, vaikka asennus tehtäisiin asiakkaan palvelimelle.

Tämän projektin yhteydessä mietittiin myös oikeita ohjelmistotuotannon menetelmiä. Pitkäjänteisessä kehitystyössä ohjelmakoodin ymmärrettävyyttä pidettiin tärkeänä, koska kaikkia yksityiskohtia ei voida dokumentoida muulla tavoin. Siksi päädyttiin käyttämään sovittuja nimeämiskäytäntöjä muuttujien ja objektien nimissä. Kolmikirjaiminen englannista johdettu lyhenne kertoo heti minkä tyyppisestä muuttujasta tai

objektista on kyse, esim. rptProductFields on repeater-tyyppinen objekti, joka esittää tuotemallin mukaiset kentät.

Ohjelmakoodissa kaikki nimetään englanniksi ja myös kommentit kirjoitetaan englanniksi, jotta kansainväliselle yhteistyölle ei aseteta raja-aitoja. Käytännön syistä ylemmän tason määrittelyt, joita käytetään myös tarjousten liitteinä, kirjoitetaan tois- taiseksi suomeksi ja käännetään vasta tarvittaessa. Graafisten suunnitelmien lisäksi kaikki ohjelmistokehitykseen liittyvä dokumentaatio päätettiin koota palvelimelta jaettuun kansioon, johon on luotu alikansiot jokaista projektia varten.

2.6 Uuden Captum2:n tavoitteet

Erään verkkokaupan valmisohjelmistoja tuottavan yrityksen teettämän tutkimuksen mukaan joka viides kotimainen yritys käy kauppaa verkossa, mutta suomalaisten verkkokauppojen tarjonnan niukkuuden vuoksi monet kuluttaja-asiakkaat ovat pää- tyneet ostoksille ulkomaisiin verkkokauppoihin. Vuonna 2008 verkkokaupan arvioi- daan kasvavan lähes neljä kertaa nopeammin kuin tavallisen vähittäiskaupan. Erityi- sesti vähittäis- ja tukkukaupassa sekä tietotekniikka-, majoitus- ja ravitsemisaloilla sekä moottoriajoneuvojen kaupassa toimivat pk-yritykset aikovat vuoden 2008 aika- na siirtyä verkkoliiketoimintaan. (Soprano Oyj 2007.) Vaikka Hypermedia Oy:n pää- kohderyhmänä ovatkin hieman suuremmat, räätälöityjä verkkokaupparatkaisuja tar- vitsevat yritykset, niin tutkimustulokset osoittavat, että sähköisen kaupankäynnin järjestelmille on lähitulevaisuudessa kysyntää.

Captum2:n uudistusprojektin alkaessa vuonna 2006 järjestelmän versio 1.0 oli ollut käytössä monilla sivustoilla useita vuosia. Siitä oli kertynyt kokemuksia ja asiakaspa- lautetta, ja Hypermedia Oy:llä oli näkemys järjestelmän tulevaisuudesta. Verkkokauppa koettiin jo silloin välttämättömänä osana yrityksen valikoimaa, mutta sen toteutus ei enää vastannut ohjelmistotuotteille asetettuja vaatimuksia. Captum2 ver- sio 1.0 toimi kyllä luotettavasti, mutta sen suurimmiksi ongelmiksi koettiin:

- Tuotteistaminen ei ollut läheskään samalla tasolla kuin julkaisujärjestelmä Hyperscriptorissa, mikä teki Captum2:n käyttöönotosta työlästä.

- Ominaisuudet olivat osittain hyvinkin rajoittuneita, esim. yleiskäyttöinen myynnin raportointi puuttui kokonaan.
- Em. syistä Captum2:ta käyttävät sivustot sisältävät paljon asennuskohtaista räätälöintiä, mikä vaikeuttaa niiden ylläpitoa.
- Ylläpitosivuston ulkoasu ja toiminta eivät vastanneet nykyisille järjestelmille asetettuja kriteerejä helppokäyttöisyyden ja monipuolisuuden osalta.
- Asiakkaan ja ylläpitäjän näkymissä ei voinut käyttää eri kieliversioita kuten Hyperscriptorissa (käyttöliittymän käännökset tietokannassa ja kielen valinta käyttäjäkohtaisesti).
- Vanha versio alkoi olla myös teknisesti vanhentunut käytettäväksi yhdessä Hyperscriptorin kanssa.

Jos yrityksen tuotteilla ei ole valmista kehityspolkua, niihin kehitetään helposti tilaa-
jakohtaisia toimintoja, joiden hyödyntäminen jatkossa saattaa olla vaikeaa. Yleis-
käyttöisen ratkaisun sijaan pyörä keksitään aina uudelleen. Siksi nähtiin tarpeelliseksi
määrittellä perusteellisesti uudistettu Captum2 versio 3.5, jossa on mietitty monia
eri käyttötarkoituksia verkkokaupasta tuotteiden ja palveluiden esittelyjärjestelmäksi.
Näin järjestelmäratkaisu on sovitettavissa monien eri toimialojen tarpeisiin.

Captum2:ssa tuotteet voivat olla fyysisten tuotteiden ohella myös verkosta ladattavia
aineistoja tai palveluita, esim. koulutuksen myyntiä. Yhtenä keskeisenä kehityssuun-
tana nähtiin myös tilanne, jossa järjestelmää käytetään tuotteiden myynnin sijaan
vuokrausten tai lainausten järjestelmänä. Tällöin tarvitaan varaus- ja palautustoimin-
toja lukumäärällisesti tai ajallisesti rajallisten resurssien käsittelyyn. Tarvittaessa
ladattavien aineistojen tekijänoikeudet tulee suojata erilaisten DRM-tekniikoiden
(Digital Rights Management) avulla.

Myös varastonseurannalle tai integraatiolle varastonseurantajärjestelmään saattaisi
olla kysyntää. Kauppatoimintojen, kuten ostoskorin ja tilausprosessin, lisäksi Cap-
tum2:ta voidaan käyttää tuhansien tuotenimikkeiden esittelyyn. Tällöin tiedon siirto
muista järjestelmistä, ryhmittely ja helppo julkaiseminen ovat tärkeitä. Yhtenä käyt-
tömahdollisuutena nähtiin myös Captum2:n toimiminen intranetin osana.

Captum2 on kuitenkin hyvin monipuolinen ja laaja ohjelmisto, joka vaatii enemmän tilaajakohtaisia ominaisuuksia kuin julkaisujärjestelmä Hyperscriptor. Lisäksi Captum2:ta myydään parhaimmillaankin vain muutamia asennuksia vuodessa, joten massiiviseen kehitystyöhön ei ollut käytettävissä niin paljon resursseja kuin Hyperscriptorin kolmosversion kohdalla muutama vuosi aiemmin. Captum2:n käyttöönotto vaatii myös asiakkaalta enemmän sitoutumista kuin normaalin web-sivuston pystyttäminen. Captum2 voi olla integroitu osaksi yrityksen muuta toimintaa, koska se saattaa liittyä niin kirjanpidon, varastoseurannan kuin tuotetiedon hallinnankin järjestelmiin.

2.7 Ohjelmistoprojektin käynnistys

Kappaleessa 2.6 esitettyjen tavoitteiden toteuttamiseksi käynnistettiin Hypermedia Oy:ssä tuotekehitysprojekti kesälomien jälkeen syksyllä 2006 Captum2 versio 3.5:n tuottamiseksi. Samalla päätettiin, että tämä opinnäytetyö seuraa kehitysprojektiä. Loppuvuoden aikana pidettiin yrityksen sisäisiä neuvotteluja 1-2 viikon välein. Neuvotteluissa hahmoteltiin Captum2:n uuden version ominaisuuksia, toteutusmahdollisuuksia ja aikataulua. Alkuperäisen suunnitelman mukaan toiminnot piti saada määriteltyä talven aikana ja toteutustyö aloittaa maaliskuussa 2007. Projektiakatauluun kirjattiin, että samoihin aikoihin aloitettaisiin myös uuden version myynti, jotta ensimmäinen tilaaja saataisiin kesään mennessä. Tilaajakohtaisen pilottiversion oli määrä valmistua syksyyn mennessä ja ensimmäisen toimitusprojektin päättyä vuoden 2007 aikana. Hypermedia Oy:n asiakaskunnassa oli orastavaa kiinnostusta uutta versiota kohtaan jo vuoden 2006 aikana, mutta kauppoja ei vielä syntynyt.

Ohjelmistoprojektiin käytettyä työaikaa ja budjetin toteutumista päätettiin seurata kirjaamalla tehdyt päivät ja tunnit Excel-taulukkoon yksittäisen työtehtävän (esim. tuoteluettelotoiminnon suunnittelu ja toteutus) tarkkuudella. Kun kaikkien projektiin osallistuvien työtunnit kerätään samaan taulukkoon, projektin tilanne on varsin reaaliaikaisesti selvillä. Työtehtävien järjestyksellä ja värikoodauksella osoitetaan kriittinen polku. Myös FileMaker Pro sekä Microsoft Project -ohjelmistojen käyttöä harkittiin aikataulujen suunnitteluun ja seurantaan, mutta niiden käyttöönotto vaatisi oman

projektinsa. Project-osaaminen mahdollistaisi kuitenkin monien manuaalista laskentaa edellyttävien seurantavaiheiden automatisoinnin.

Yksi osa projektisuunnitelmaa on riskien kartoitus. Captum2-projektin suurimmat riskit liittyvät aikataulun pitävyyteen ja resurssien riittävyyteen. Alla on tarkasteltu projektin riskien todennäköisyyttä ja vaikutusta asteikolla 1-5 sekä riskien tärkeyttä näiden tulona asteikolla 1-25:

1. Pilottiasiakkaan investointipäätös viivästyy
todennäköisyys 3 x vaikutus aikatauluun 5 = 15
2. Töiden ruuhkautuminen, priorisointiongelmat
todennäköisyys 3 x vaikutus aikatauluun 4 = 12
3. Kolmansien osapuolten ohjelmistokomponentteihin liittyvät riskit
todennäköisyys 2 x vaikutus tekniikkaan ja kustannuksiin 3 = 6
4. Uudesta tekniikasta johtuvat riskit (osaaminen, tekniset ongelmat)
todennäköisyys 2 x vaikutus tekniikkaan ja kustannuksiin 2 = 4

Pilottiasiakkaan investointipäätöksen viivästyminen aiheuttaa sen, että Captum2:n valmistuminen siirtyy, koska ohjelmisto toteutetaan loppuun asti vasta tulorahoituksella. Muuten taloudellinen riski todelliselle asiakkaalle sopimattoman ohjelman kehittämisestä muodostuu liian suureksi. Investointipäätöksen viivästymisen merkitys projektille on suuri, mutta sen syihin ei pääse vaikuttamaan eikä riskiä voi siirtää. Riskiin varaudutaan siten, että tilauksen siirtyessä toteutetaan vain tarkennettu toiminnallinen määrittely sekä keskeisimmät toiminnot demosivustoon myynnin tueksi. Tämän riskin toteutuminen nostaa seuraavan riskin todennäköisyyttä.

Jos pilottiversioiden toteutus siirtyy syksyyn 2007, saattaa tulla ongelmia muiden päällekkäisten projektien kanssa, jotka jakavat samat henkilöresurssit. Tämä riski pienenee, kun mahdollisimman paljon töitä siirretään hiljaisempiin kuukausiin. Jos riski kuitenkin toteutuu, sen vaikutuksia voidaan estää kuormituksen tasaamisella eri henkilöiden kesken ja mahdollisesti lisärekrytoinneilla tai ylitöillä.

Kolmansien osapuolten ohjelmistokomponentteihin liittyvät riskit ovat huomattavasti edellisiä pienempiä. Jonkin projektissa käytetyn komponentin osalta voi tulla tilanne,

että uutta asennusta varten ei ole saatavissa lisenssiä tai komponentin saatavissa oleva versio ei olekaan yhteensopiva aikaisemmalle versiolle tehdyn ohjelmakoodin kanssa. Tämä riski voidaan poistaa siten, että käytetään vain sellaisia ohjelmistokomponentteja, joihin on hankittu tai voidaan hankkia royalty-free jakelulisenssi.

Viimeisenä ja vähäisimpänä ovat uudesta tekniikasta johtuvat riskit, joita tällaiseen projektiin aina liittyy. Ongelmana ei niinkään ole se, miten asiat saa tehtyä kuin se, miten ne kannattaa tehdä tehokkaimmin. Tekniikasta johtuvia riskejä voidaan pienentää koulutuksella ja osaamisen jakamisella.

3 OHJELMISTOPROJEKTIN TULOKSET

Toiminnallisen määrittelyn ensimmäinen versio valmistui aikataulun mukaisesti maaliskuussa 2007 ja kevään aikana valmisteltiin myös muita myynnin tueksi tarvittavia materiaaleja, esim. tekniikkaa ja ohjelmistoprojektien laatua käsittelevät dokumentit. Hiljaisempi kesäkausi teki kuitenkin tuloaan, joten varsinainen Captum2:n myynnin aloitus päätettiin siirtää syksyyn. Järjestelmä oli kuitenkin koko ajan esillä esim. Hypermedia Oy:n yritysseiteissä ja web-sivuilla.

Seuraavissa kappaleissa käydään läpi Captum2:n toiminnot yleisellä tasolla puuttumatta yksityiskohtiin tai toteutukseen. Sen jälkeen kappaleessa 3.3 tarkastellaan projektin toteutusvaihetta tarkemmin. Captum2 sisältää kaikki julkaisujärjestelmä Hyperscriptor XG:n ominaisuudet, joten sitä voi käyttää kaikissa niissä web-ratkaisuissa, joihin Hyperscriptor soveltuu (esim. intranet).

Järjestelmä on kokonaan web-pohjainen eli sitä käytetään selaimella. Käyttöliittymältään Captum2 jakautuu asiakkaan ja ylläpitäjän näkymiin, jotka on optimoitu normaalikokoisella näytöllä (vähintään XGA-resoluutio eli 1024 x 768 pikseliä) tarkasteltaviksi. Tarvittaessa näkymistä voidaan toteuttaa pienemmät mobiiliversiot, joiden sisältö ja toiminnot ovat vastaavat. Asiakkaan näkymä voi olla julkinen,

avoimesti internetiin näkyvä, tai suljettu, extranet-tyyppinen ratkaisu, jolloin asiakkaat kirjautuvat sisään omilla käyttäjätunnuksillaan. Asiakkaan näkymä muokataan aina järjestelmän tilaajan toiveiden mukaisesti ja on ulkoasultaan yksilöllisesti suunniteltu.

Ylläpitäjillä on erilaisia rooleja, joiden mukaan käytettävissä olevat toiminnot määräytyvät. Ylimmän tason toiminnot suorittaa pääkäyttäjä. Tuotetiedon esittelyyn, tilausten hallintaan ja raportointiin liittyvät toiminnot on integroitu julkaisujärjestelmään siten, että kaikkien toimintojen käyttölogiikka ja ulkoasu ovat samanlaisia.

3.1 Asiakkaan näkymät

Asiakkaan käyttöliittymä noudattaa sivustolle suunniteltua graafista ilmettä. Järjestelmän saa tarvittaessa monikielisenä ja useampaa valuuttaa käyttävänä. Kaikki käyttöliittymän vakiotekstit ovat tietokannassa niin, että ne on helppo siirtää tekstimuodossa käännöstä varten. Sivuston eri kieliversiot voivat olla laajuudeltaan erilaisia ja sisältää eri tuotteita.

Jos samaa järjestelmää käytetään useammalle kohderyhmälle (esim. jälleenmyyjät ja kuluttajat), niin käytössä voi olla useita eri ulkoasuja. Samaan järjestelmään voidaan kytkeä myös useita verkkotunnuksia (domain), jolloin yhden ylläpito näkymän kautta voidaan hallita useita kauppapaikkoja. Tarvittaessa asiakas kirjautuu sisään ennen kuin hän pääsee selaamaan tuotetietoja, jolloin näkymä voi olla profiloitu asiakkaan mukaan ja esim. asiakaskohtaisia alennusprosentteja voidaan käyttää. Järjestelmä lähettää unohtuneen salasanan asiakkaan pyynnöstä sähköpostiin automaattisesti.

Sivustolle voidaan räätälöidä erillinen etusivu, jossa näkyvät esim. uusimmat tai osatetuimmat tuotteet tai ylläpitäjän määrittelemät ajankohtaiset tarjoukset. Tietoja ja tuotteita voi tuoda näkyviin ja poistaa näkyvistä ajastetusti.

3.1.1 Tuoteryhmät ja hakutoiminto

Tuotetietoja voi etsiä tuoteryhmittäin tai hakutoiminnolla. Liitteessä 1 on kuva tuoteryhmästä eli yhdestä julkaisujärjestelmällä tuotetusta sivusta. Tuoteryhmävalikko eli tuotteita sisältävä sivuhaara voi olla koko ajan näkyvissä sivuston päävalikossa, jos sivustossa on paljon muutakin sisältöä. Extranet-tyyppisessä järjestelmässä tuoteryhmät voivat olla asiakaskohtaisia, esim. projekteina toimitettuja laitteita, joihin järjestelmän kautta tilataan varaosia. Tuoteryhmähierarkia on monitasoinen ja tuotteet voivat sijoittua mille tahansa sivuston sivulle. Tuotteen voi määrittellä näkymään tuoteryhmässä, vaikka se ei olisi enää tai vielä myynnissä.

Ylläpitäjä voi muodostaa mistä tahansa julkaisujärjestelmällä tehdystä sivusta tuoteryhmäsivun liittämällä siihen halutut tuotteet. Tuotteet näkyvät asiakkaalle ylläpitäjän määrittelemässä järjestyksessä perustuoteluettelona tai laajennettuna tuoteluettelona. Perustuoteluettelossa tiedot esitetään taulukkomuodossa siten, että jokainen tuote muodostaa yhden rivin ja sarakkeissa näytetään pääkäyttäjän perustuoteluetteloön määrittelemät tiedot. Perustuoteluettelo voi koostua myös pelkistä tuotteiden nimistä ja linkeistä tuotekorteille.

Laajennetun tuoteluettelon ulkoasu määritellään yhdessä tilaajan kanssa sivuston tyyliin sopivaksi esimerkiksi siten, että tuotteet esitetään kahdessa palstassa. Pääkäyttäjä määrittelee laajennetuissa tuoteluetteloissa näkyvät kentät, esim. nimi, kuva, lyhyt esittelyteksti ja hinta. Molemmat tuoteluettelot voi lisäksi sivuttaa alasisivuihin, jolloin yhdessä tuoteryhmässä voi olla kymmeniä tuotteita. Ylläpitäjä valitsee, onko sivutus käytössä ja minkä kokoisia alasisivuja (montako tuotetta kerralla) asiakkaalle näytetään.

Oletuksena yhdessä tuoteryhmässä eli sivulla voi olla vain yhden tuotemallin mukaisia tuotteita. Tuotemalli on luettelo tuotetietoihin kuuluvista kentistä, jotka pääkäyttäjä määrittelee. Tuotemallien avulla erilaisista tuotteista pystytään esittämään strukturoidusti yksityiskohtaisempia tietoja (mm. vertailutoiminnon tarpeisiin) ja perustuoteluettelossa kaikilla riveillä on samat sarakkeet. Jos samaan tuoteryhmään tarvitaan useamman tuotemallin mukaisia tuotteita, tilaajan kanssa täytyy määrittellä miten erilaiset kentät esitetään samassa perustuoteluettelossa yms. ristiriitatilanteet.

Tarvittaessa tuoteryhmät eli kaikki tuotetietoja sisältävät sivut saa tulostettua asiakkaan tai ylläpitäjän näkymässä PDF-muotoisena luettelona, jossa tuotetiedot on esitetty perustuoteluetteloina. Tällaista luetteloa voidaan käyttää vaikkapa hinnastona. Yksittäisen tuoteryhmän tai muun näkymän voi tulostaa sivun tulostustoiminnolla, joka on julkaisujärjestelmän vakio-ominaisuus. Tulostusnäkyvässä sivulta poistetaan valikot ja teksti esitetään siten, että se leveydeltään täyttää A4-arkin. Tuoteluettelo näkyy tulostussivulla kokonaan ilman sivutusta, jolloin se sivuttuu arkeille optimallisesti.

Hakutoiminnolla asiakas voi etsiä tuotteita ja muuta sivustolla olevaa tietoa vapaasanahauulla. Haun yhteydessä valitaan etsitäänkö hakusanaa vain tuotetiedoista vai kaikesta sisällöstä, jolloin haettu sana voi löytyä esim. toimitusehtoja esittelevältä sivulta. Haku löytää myös tuotteet, joita ei ole liitetty tuoteryhmiin. Hakutulos näyttää löytyneet tuotteet aakkosjärjestyksessä sivutettuna luettelona, jossa on tuotteiden nimet sekä tieto sivusta tai sivuista, joihin tuote on liitetty. Pääkäyttäjä määrittelee, montako hakutulosta yhdellä sivulla näytetään.

Haluttaessa hakutulosta voidaan räätälöidä esim. siten, että tuotteista näytetään myös kuvat. Sekä tuoteryhmässä että hakutuloksessa tuotteen nimi toimii linkkinä tuotekortille. Vapaasanahaku etsii tekstiä myös sivujen ja tuotetietojen liitetiedostoista, jos ne ovat PDF-muotoisia tai Office-asiakirjoja. Hakutoiminto voidaan laajentaa kenttäkohtaiseksi hauksi, jolloin haettavaa tietoa etsitään tuotemallin yksittäisestä kentästä.

3.1.2 Tuotekortti

Kun asiakas valitsee yksittäisen tuotteen tuoteryhmästä tai hakutuloksesta, hän näkee koko tuotekortin, joka koostuu kaikista tuotteeseen liittyvistä tiedoista, esim. kuva, nimi, paino, mitat, hinta ja kuvausteksti. Mukana voi olla myös suurempia kuvia ja muita liitetiedostoja. Kentät perustuvat pääkäyttäjän määrittelemään tuotemalliin ja ne esitetään tuotekortin ulkoasun puitteissa pääkäyttäjän määrittelemässä järjestyksessä. Liitteessä 2 on kuva tuotekortista. Tuotekortin voi tulostaa siistin näköisenä

erillisestä tulostusnäkyvästä, jossa ei ole mukana navigointiin liittyviä elementtejä (valikot ja toimintonapit).

Vertailutoiminnolla asiakas voi valita useita samaa tuotemallia käyttäviä eli samat tietokentät sisältäviä tuotteita vertailtavaksi rinnakkain. Jos järjestelmään lisätään varastoseuranta, asiakas näkee onko tuotetta saatavilla tai koska seuraava erä saapuu. Varastoseurantaan liittyy toiminto, joka lähettää ylläpitäjälle sähköpostia, kun tuotekohtainen varastosaldon hälytysraja alittuu tai tuote myydään loppuun. Tuotekortti voi näyttää, jos tuote on loppuunmyyty eikä anna tilata tuotetta. Ylläpitäjä voi myös määritellä loppuunmyytyt tuotteet poistumaan näkyvistä tuotekohtaisesti. Tuotteen eri kieliversiot käyttävät samaa varastosaldoa. Vertailutoiminto ja varastoseuranta määritellään tarvittaessa tilaajan kanssa tarkemmin. Captum2 on myös integroitavissa olemassa olevaan varastojärjestelmään.

3.1.3 Keräilylista (ostoskori)

Keräilylistatoiminnolla asiakas voi lisätä haluamansa tuotteet virtuaaliseen ostoskoriin tuotekortilta tai suoraan tuoteryhmästä tai hakutuloksesta. Liitteessä 3 on kuva keräilylistasta. Keräilylistalla olevista tuotteista on linkit niiden tuotekortteihin, ja asiakas voi muuttaa poimitujen tuotteiden määrää tai poistaa tuotteita keräilylistalta. Jos tuotteet on hinnoiteltu, keräilylista näyttää tilauksen yhteissumman lisättynä toimituskuluilla. Jos tuotteita ei ole hinnoiteltu, keräilylistalta voi lähettää tarjouspyynnön. Keräilylistalla olevien tuotteiden määrä ja mahdollinen hinta voidaan näyttää kaikilla sivuilla.

Jos Captum2:a käytetään varausjärjestelmänä, keräilylistalle siirtyviin tietoihin liittyy myös päivämäärä ja mahdollisesti kellonaika, jolloin varaus tehdään. Näiden valitsemiseksi tuotekortille tai keräilylistanäkymään voidaan liittää kalenteritoiminto, joka näyttää vapaat ajat.

Jos asiakas ei tee tilausta tai tarjouspyyntöä, keräilylistalle jätetyt tiedot poistetaan ajastetusti tai, jos järjestelmään kirjaudutaan extranet-tyyppisesti, tiedot voidaan tal-

lentaa asiakkaan käyttäjätunnuksen yhteyteen. Jos kirjautuminen ei ole käytössä eli asiakasta ei tunnisteta, keräilylistan sisältö on selainkohtainen.

Kun asiakas tunnistetaan, keräilylistan voi tallentaa toivelistaksi, jolloin keräilylistalle poimitut tuotteet saa palautettua myöhemmin. Yhdellä asiakkaalla voi olla useita toivelistoja ja ylläpitäjä voi tehdä niitä asiakkaalle valmiiksi. Toimintoa voi käyttää vakioitujen tilausten tekemiseen, esim. määräaikaishuollossa tarvittavien varaosien tilaamiseen. Toivelistatoiminnon kautta voidaan myös toimittaa hinnoitellut tarjoukset asiakkaan tarjouspyyntöihin. Tällöin asiakas ei voi muuttaa toivelistalta keräilylistalle palauttamiaan tilausrivejä.

Varastosaldoa vähentävä tuote pysyy varattuna asiakkaalle puoli tuntia, kun se on poimittu keräilylistalle tai palautettu toivelistalta. Jos tuotetta ei tässä ajassa siirrytä tilaamaan, tuote poistuu keräilylistalta ja asiakkaan täytyy varata se uudelleen. Tilauslomakkeelle siirryttyään asiakkaalla on toinen puoli tuntia aikaa tehdä tilaus.

Keräilylistan saa tallennettua Excel-muodossa tai lähetettyä sähköpostitse. Kun kaikki tuotteet ovat keräilylistalla, asiakas siirtyy tilaus- tai tarjouspyyntölomakkeelle.

3.1.4 Tilauslomake ja tarjouspyyntölomake

Mikäli käytössä on sähköinen tilaustoiminto, keräilylistalla olevat tuotteet voi tilata verkkokaupasta. Liitteessä 4 on kuva tilauslomakkeesta. Tilaussivu voi vaatia asiakkaan kirjautumisen henkilökohtaisella tunnuksella ja salasanalla ellei kirjautumista ole vaadittu jo aiemmin kauppaan tullessa. Järjestelmä voidaan määritellä myös täysin avoimeksi, jolloin jokainen tilaaja täyttää tyhjän tilauslomakkeen ja käyttäjätunnukset eivät ole käytössä.

Asiakaskohtainen kirjautuminen mahdollistaa asiakasrekisterin tiedoilla valmiiksi täytetyn tilauslomakkeen näyttämisen, erilaisten räätälöityjen näkymien tekemisen ja tilaushistorian seurannan sekä tilausvalmiiden tarjousten saamisen tarjouspyyntöihin ja tilauskohtaisten reklamaatioiden tekemisen. Tarvittaessa keräilylistalle tai tilaus-

lomakkeelle voidaan räätälöidä erilaisia lisävalintoja liittyen esim. toimitustapaan ja -aikatauluun.

Tilauksen yhteydessä asiakkaalta kysytään pääkäyttäjän määrittelemät tiedot ja sen jälkeen asiakas valitsee tuotteiden maksutavan. Käytävissä ovat suomalaisten pankkien verkkomaksut, yleisimmät luottokortit sekä tilaus postiennakolla tai laskulla. Tilauksen ja mahdollisen maksun jälkeen asiakas saa tilausvahvistuksen sähköpostiinsa ja keräilylista tyhjenetään. Tilauksesta voi lähteä sähköposti myös ylläpitäjälle.

Jos asiakas on valinnut keräilylistalle yhdenkin tuotteen, jota ei ole hinnoiteltu, tilauksen sijaan asiakas tekee keräilylistan sisällöstä tarjouspyynnön. Pääkäyttäjä voi määrittellä tarjouspyyntölomakkeelle eri kenttiä kuin tilauslomakkeelle, mutta molemmissa on yhteiset peruskentät (mm. asiakkaan nimi ja yhteystiedot), joista muodostuu asiakasrekisteri. Tarjouspyyntölomake näyttää keräilylistan sisällön ja kysyy tarjouksen pyytäjän tiedot, mutta ei sisällä maksutavan valintaa. Tarjouspyyntö tallennetaan ja näytetään asiakkaalle. Samalla keräilylista tyhjenetään ja tarjouspyynnöstä lähtee sähköposti asiakkaalle ja ylläpitäjälle.

3.2 Ylläpitäjän näkymät

Ylläpitäjät kirjautuvat salasanasuojatuille ylläpitosivuille omalla henkilökohtaisella tunnuksellaan. Käytävissä olevat toiminnot riippuvat ylläpitäjän rooleista, esim. ylläpitäjällä voi olla oikeudet tilausten käsittelyyn ja raportointiin, mutta ei tuotetietojen ylläpitoon. Tuoteryhmissä on lisäksi julkaisujärjestelmän hierarkkisesti periytyvät sivukohtaiset ylläpito-oikeudet. Tämä tarkoittaa, että ylemmän tason sivulle annetut oikeudet oikeuttavat ylläpitämään myös sivun alisivuja.

Pääkäyttäjä suorittaa ylimmän tason toiminnot (mm. tuotemallin muokkaus, josta on kuva liitteessä 5) ja määrittelee käyttäjätunnukset rooleineen. Pääkäyttäjä voi myös ylläpitää samoja tietoja kuin tavalliset ylläpitäjät kaikissa eri rooleissa. Liitteessä 5 näkyy pääkäyttäjän toimintovalikko ja liitteissä 6-8 ylläpitäjän toimintovalikko.

Ylläpitäjillä on käytettävissä kaikki Hyperscriptor XG:n julkaisutyökalut, joten esim. verkkokaupan toimitusehtojen tai yhteydenottolomakkeen lisääminen sivustolle on helppoa. Julkaisujärjestelmän toiminnoilla sivustoon voi lisätä sivuja ja niille tekstikappaleita, kuvia, linkkejä, lomakkeita ja yhteystietoja. Sivuja ja kappaleita voi järjestellä sekä siirtää tai kopioida paikasta toiseen. Liitteessä 7 on kuva yhden sivun (tuoteryhmän) muokkausnäköistä. Avatussa ylävalikossa näkyy toimintoja, joita sivulla voi suorittaa.

Ylläpitäjä voi selata ja ylläpitää asiakkaalle näkyvää tuoteryhmähierarkiaa ja muita sivustolle lisättyjä sivuja Windowsin resurssienhallintaa muistuttavan hakemistopuun kautta, joka muodostuu web-sivujen otsikoista. Jokaiseen tuoteryhmään eli web-sivulle voi liittää rajoituksettoman määrän saman tuotemallin mukaisia eli samat tietokentät sisältäviä tuotteita ja sama tuote voi näkyä useammassa paikassa, jos se on liitetty monelle sivulle. Tuotteita voi liittää tuote kerrallaan tai sivulle voi määrittellä hakuehdon, jonka täyttävät tuotteet näytetään dynaamisesti sivulla. Näin esim. kaikki tarjoustuotteet eri tuoteryhmistä voidaan näyttää kootusti yhdellä sivulla. Myös tuote kerrallaan liitettäessä ylläpitäjä voi käyttää hakua ja valita liitettävät tuotteet hakutuksesta. Liitteessä 8 on kuva tuotteiden liittämistä tuoteryhmään.

Tuote kerrallaan liitettäessä ylläpitäjä voi järjestää tuotteet haluamaansa järjestykseen numeroimalla ne kuten kappaleita tai sivuja järjestettäessä. Oletuksena uusi tuote lisätään tuoteryhmän alkuun. Hakuehdolla dynaamisesti liitettäessä tuotteet näytetään tuoteryhmässä aakkosjärjestyksessä.

Yksittäisen sivun tai kokonaisen sivuhaaran voi poistaa ilman että tuotetietoja poistetaan. Sivuja voi myös piilottaa julkisesta näköistä tai ajastaa näkymään tiettyinä aikoina. Niitä voi vapaasti nimetä uudestaan tai siirtää tai kopioida paikasta toiseen. Koska tuoteryhmät ovat julkaisujärjestelmän sivuja, niille voi helposti liittää tuoteryhmäkohtaista tekstiä, kuvia ja liitetiedostoja, esim. kaikille ryhmän tuotteille yhteisen asennusohjeen.

3.2.1 Tuotteiden lisäys ja muokkaus

Uuden tuotteen lisäys tapahtuu omalla toiminnollaan, jonka voi käynnistää myös suoraan sivun muokkauksesta, jolloin tuote tulee oletuksena kyseiseen tuoteryhmään. Olemassa olevan tuotteen tiedot voi kopioida uuden tuotteen pohjaksi. Tuotteesta annetaan tietoja, joista osa on pakollisia (esim. tuotteen nimi) ja osa pääkäyttäjän määrittelemiä erityyppisiä valinnaisia kenttiä (esim. tietoa tuotteen soveltuvuudesta tiettyyn käyttötarkoitukseen). Osa kentistä voi olla metatietoa, joka ei näy asiakkaalle, mutta jonka perusteella hakutoiminto löytää tuotteen.

Tuotetietojen kentät pohjautuvat pääkäyttäjän määrittelemään tuotemalliin. Liitteessä 5 on kuva tuotemallin muokkauksesta. Nimi on pakollinen tieto, joka yksilöi tuotteen. Järjestelmä ei anna lisätä kahta täysin samannimistä tuotetta. Jotkut kentät, kuten hinta ja pieni tuotekuva, ovat tuotemalleissa valmiina, ja pääkäyttäjä valitsee ovatko kentät käytössä ja näytetäänkö ne tuoteluetteloissa tai tuotekortilla. Lisäksi tuotteeseen liittyy joukko tuotteen tilaa määrittäviä valintoja (mm. onko tuote myynnissä). Muuten pääkäyttäjä voi määritellä tuotemalliin kuuluvat kentät vapaasti. Kentät voivat näkyä ylläpitäjille mm. tekstikenttinä, monivalintakenttinä tai kalenterikenttinä (ajan valinta kalenterista).

Kaikista kentistä on tarvittava määrä kopioita tuotteen eri kieliversioita varten. Tuotteen voi liittää esim. englanninkieliseen tuoteryhmään (sivulle) vain, jos tuotteesta on tallennettu tietoja englanniksi. Jos tuotetiedot on täytetty yhdellä kielellä, kaikki kentät voi kopioida toiseen kieliversioon. Näin esim. tuotteeseen liitetyt kuvatiedostot saa nopeasti monistettua.

Jos tuotemalleja on vain yksi, ylläpitäjän ei tarvitse erikseen valita sitä. Jos tuotemalleja on useita, ylläpitäjä valitsee tuotetta lisätessään minkä mallin mukaisen tuotteen hän lisää ja täyttää vähintään pakolliset pääkäyttäjän määrittelemät tiedot. Ylläpitäjä voi lisätä tuotteeseen liitetiedostoja (kuvia, PDF-dokumentteja, videoita jne.) tai määritellä, millä sivuilla tuote näkyy asiakkaalle. Asiakkaalle näkyvillä sivuilla tuotteet esitellään ylläpitäjän valitsemassa muodossa perustuoteluettelona tai laajennettuna tuoteluettelona, ks. kappale 3.1.1 Tuoteryhmät.

Tuotetiedoissa on kenttä tuotteiden ryhmittelyyn ylläpitäjiä varten (vapaa tekstikenttä ja pudotusvalikko, josta voi valita jonkin aiemmin käytetyistä arvoista). Kenttään syötetyn tiedon perusteella ylläpitäjä voi etsiä samanlaisia, esim. saman tavarantoimittajan tuotteita. Tätä kenttää voi myös käyttää erityisen easy shop -toiminnon yhteydessä, jolloin järjestelmä muodostaa automaattisesti kaikista eri ryhmistä asiakkaalle näkyvät tuotesivut. Easy shop -kaupassa asiakkaalle näkyvä tuotehierarkia on yksitasoinen ja vastaa ylläpidon ryhmittelytietoja. Ryhmittelytiedot voi tuoda järjestelmään Excel-muodossa, joten easy shop -toiminto mahdollistaa kauppapaikan rakenteen tuomisen kerralla järjestelmään.

Tuotetietojen muokkaus tapahtuu hakemalla tuote joltain sivulta tai etsimällä se ylläpitäjän tuotehaulla. Kaikkia tuotetietoja voi myös selata aakkosjärjestyksessä nimen mukaan jättämällä hakukentän tyhjäksi. Liitteessä 6 on kuva tuotteen muokkauksesta. Tuotteen muokkaussivulta näkee, mille sivuille kyseinen tuote on liitetty. Tuotteen poistaminen poistaa sen kaikilta sivuilta, joille tuote on määritelty näkymään. Poistettujen tuotteiden nimet ja hinnat säilyvät tilaushistoriassa, koska tiedot on kopioitu tilausriveille. Tuotteen voi myös piilottaa näkyvistä ja tuoda näkyviin ajastustusti kuten tuoteryhmänkin. Tuotteen tuotemallia ei voi vaihtaa.

Tuotteen toimituskulut asiakkaalle voidaan määritellä maksutapakohtaisesti kiinteiksi tai tuotteiden määrään tai painoon perustuvaksi. Painon mukaan määräytyvät toimituskulut muodostuvat siten, että pääkäyttäjä määrittelee painotaulukon esim. postin hinnaston mukaisesti ja tuotetietoihin kirjataan jokaisen tuotteen paino. Tuotteilla voi olla minimi tilausmäärä, jota ei voi alittaa, tai tilauksen alittaessa pääkäyttäjän määrittelemän rajan peritään pientoimituslisä.

Tuotetiedot saa tulostettua CSV- (Comma Separated Values) tai XML-tekstitiedostoksi tai Excel-taulukoksi, jonka avulla tuotetietoja voi siirtää järjestelmästä ulos ja sisään. Esim. 5 % hintojen korotus voidaan suorittaa lataamalla ylläpitäjän hakutoiminnolla etsityt tuotteet Excel-taulukkona, päivittämällä hintasarakkeen tiedot Excel-kaavalla ja tallentamalla taulukko takaisin järjestelmän sisäiseen tietokantaan. Tarvittaessa tämän tiedonsiirtomahdollisuuden avulla voidaan toteuttaa integraatio toiseen järjestelmään, jolloin tietoja siirretään automaattisesti ajastuksen mukaan.

3.2.2 Tarjouspyynnöt ja tilaukset

Tarjouspyyntö- ja tilauslomake muodostuvat järjestelmän tilaajan kanssa sovittavista kiinteistä peruskentistä sekä pääkäyttäjän lomakekohtaisesti määrittelemistä lisäkentistä. Peruskenttiin täytetyt tiedot tallentuvat asiakasrekisteriin. Asiakasrekisterin ylläpitoon on omat toimintonsa kuten asiakastietojen lisäys, haku minkä tahansa kentän mukaan, muokkaus ja tulostus Excel-muotoon, josta saa tehtyä esim. joukkopostitustarrat.

Ylläpitäjät näkevät keräilylistojen kautta tehdyt tarjouspyynnöt ja tilaukset, joista laskutettavat ja postiennakkotilaukset voi erikseen kuitata maksetuiksi. Laskutettavien tilausten tiedot saa siirrettyä Captum2:sta järjestelmän tilaajan kanssa sovitussa muodossa. Tilaukset kuitataan toimitetuiksi yksi kerrallaan. Tarvittaessa tehtyjä tilauksia voi muuttaa niin, että näkyviin jää sekä alkuperäinen että muutettu tilausrivi. Tilauksen ja tarjouspyyntöjen yhteyteen tallentuu muutoshistoria, joka sisältää ylläpitäjien nimet ja muutosajat sekä muutokseen liittyvät kommentit.

Jos toimitus tapahtuu postitse, ylläpitäjä voi lähettää asiakkaalle paketin lähetystunnuksen järjestelmän kautta sähköpostitse. Myös tarjouspyyntöön liittyvän tarjouksen saa lähetettyä järjestelmän kautta asiakkaan ilmoittamaan sähköpostiosoitteeseen sekä asiakkaan käyttäjätunnuksella näkyviin tulevaksi tilausvalmiiksi toivelistaksi. Kaikki tarjouspyynnöt ja tilaukset arkistoituvat järjestelmään, ja ylläpitäjä voi hakea niitä ajankohdan tai asiakkaan mukaan tai vapaasanahauilla. Tilajarekisterin ylläpitoon on omat toimintonsa ja sitä voi käyttää myös asiakasrekisterinä esim. joukkopostituksia varten.

Pankkien verkkomaksuihin voidaan liittää automaattinen virhetilaan jääneiden tilausten tarkistus. Tämä tarkoittaa sitä, että jos asiakas ei palaa pankin sivuilta kauppaan ja siten kuittaa tilausta maksetuksi, keskeneräisiksi jääneiden maksusuoritusten tila tarkistetaan kerran tunnissa automaattisesti pankin järjestelmästä. Samalla vapautetaan asiakkaiden varaamat tuotteet, joita ei ole tilattu määräajassa. Pankkiyhteyksien käyttöön liittyvät tilikohtaiset tunnisteet on tallennettu järjestelmän asennuskohtaisiin konfiguraatietietoihin.

3.2.3 Raportointi ja tilastot

Riittävät oikeudet omaava ylläpitäjä voi tuottaa järjestelmästä kattavat raportit. Järjestelmään kuuluu ainakin tilausten ja tarjouspyyntöjen raportointi, mutta raportointia voidaan räätälöidä monipuolisesti. Tilauksia ja tarjouspyyntöjä voi raportoida vuoden, kuukauden, viikon tai päivän tarkkuudella ylläpitäjän valitsemalla aikavälillä. Ajat tulevat raportin sarakkeiksi, esim. vuoden aikaväliltä kaikki kuukaudet. Tilaukset ja tarjouspyynnöt voi ryhmitellä tuotemallin, tuoteryhmän (ylläpitäjän ryhmitteilykenttä), yksittäisen tuotteen tai yksittäisen asiakkaan mukaan. Ryhmittely tulee raportin riveiksi, esim. tilaukset tuoteryhmittäin.

Yksittäisen asiakkaan mukaan ryhmiteltäessä saman yrityksen eri yhteyshenkilöiden tilaukset näkyvät raportissa eriteltyinä yrityksen summarivin jälkeen. Yritys yksilöidään Y-tunnuksen perusteella. Tilauksia voi raportoida euromäärän tai lukumäärän mukaan, tarjouspyyntöjä ainoastaan lukumäärän mukaan. Määrät näytetään raportissa edellä kuvattujen rivien ja sarakkeiden leikkauskohdissa lukuina. Lisäksi voi muodostaa raportin tilauksista, jotka on tehty tarjousten perusteella. Raportit tulostuvat Excel-taulukkona, ja niihin voidaan tarvittaessa toteuttaa peruskaavioita. Järjestelmästä saa selaimen myös listaukset suosituimmista tuotteista ja eniten tilanneista asiakkaista tietyltä aikaväliltä.

Verkkokaupparaporttien lisäksi järjestelmässä on Hyperscriptorin kävijätilastointitoiminnot ja viimeksi muokattujen sivujen (tuoteryhmien) luettelo. Ylläpitäjät voivat lisätä tuoteryhmiin ja muihin sivuihin muistiinpanoja, jotka näkyvät ylläpitäjille tietojen muokkauksen yhteydessä. Kaikista muistiinpanoista saa myös koosteraportin.

Järjestelmää voidaan laajentaa esim. vertailevilla raporteilla, muokattavilla raporttipohjilla tai raporteihin voidaan lisätä grafiikkaa. Raportointitietoja on mahdollista siirtää muihin järjestelmiin tai integroida raportointi esim. Crystal Reportsiin. Osa raporteista voidaan myös julkaista asiakkaan näkyvässä.

3.3 Ohjelmiston toteutus

Määrittelyvaiheen jälkeen ohjelmistotuotannon seuraavat vaiheet ovat suunnittelu ja toteutus. Koska Captum2:n myynnin tueksi ei tarvittu laajempaa dokumentointia ja Hypermedia Oy:llä oli runsaasti kokemusta aikaisemman version toiminnasta ja vastaavista projekteista, päätettiin suunnitteluvaihe dokumentoida kevyesti ja siirtyä nopealla aikataululla varsinaiseen ohjelmointityöhön.

Määrittelyn pohjalta laadittiin muutamia käyttöliittymämalleja ja Captum2:n ominaisuusluettelosta valittiin keskeisimmät toiminnot, jotka päätettiin toteuttaa loppukesästä 2007. Koska Hypermedia Oy:n ohjelmistoprojekteihin liittyy aina tilaajakoh-
taisia räätälöintejä, ei nähty tarpeelliseksi toteuttaa koko järjestelmää valmiiksi ennen ensimmäistä tilausta. Tämä päätös oli projektin riskien hallinnan suunnitelman mukainen. Keskeiset toiminnot olisi ollut hyvä saada toimimaan demosivustolla, mutta Captum2-projektista johtumattomista syistä niiden toteutus kuitenkin viivästyi, koska laskutettavat työt olivat etusijalla tuotekehitykseen verrattuna. Vaikutti siltä, että myyntityö pitäisi aloittaa hyvän määrittelyn ja PowerPoint-esitysten pohjalta.

Syyskuussa alkoi kaikesta huolimatta näyttää siltä, että potentiaaliset asiakkaat olivat kiinnostuneet Captum2:sta. Tarjoukset laadittiin perusmäärittelyn ympärille ja aikaisemman version hinnoittelun pohjalta. Lopulta ensimmäinen tilaussopimus uudesta ohjelmistosta allekirjoitettiin lokakuun lopussa. Siinä vaiheessa Captum2:n toteutus ei ollut enää pelkkää tuotekehitystä, vaan myös laskutettavaa työtä. Alkuperäisen suunnitelman mukaisesta tavoitteesta, jonka mukaan tilaajakohtainen toimitusprojekti valmistuisi vuoden 2007 aikana, täytyi joustaa muutamalla kuukaudella.

Ensimmäinen Captum2 versio 3.5:n tilaaja oli Porin Seudun Matkailu Oy. Yritys tarvitsi järjestelmän, joka soveltuu matka- ja kokouspakettien myyntiin sekä varausten tekemiseen. Captum2 soveltui perusvaatimukseen hyvin, mutta sitä piti laajentaa ja räätälöidä jonkin verran. Niinpä toimitusprojekti käynnistyi tilaajakohtaisten toimintojen määrittelyllä. Samaan aikaan Hypermedia Oy:ssä käytiin läpi Captum2:n ominaisuusluettelo ja valittiin projektiin sisältyvät Captum2:n vakio-ominaisuudet ja optiot sekä määriteltiin niiden toteutusjärjestys. Näistä ominaisuuksista päätettiin

toteuttaa demosivustolle jäävä versio ja jatkaa sitten siitä tilaajan edellyttämällä ominaisuuksilla varustetun version kehittämistä.

Koska järjestelmän tilaajan aikataulu oli suhteellisen tiukka ja Captum2:n ohjelmointityö vielä melko alkuvaiheessa, tarvittiin toimitusprojektiin kaikkien Hypermedia Oy:n sovelluskehittäjien työpanosta. Toiminnot jaettiin selkeisiin kokonaisuuksiin, esim. tuotemallin lisäys ja muokkaus, ja niiden toteutusjärjestyksestä ja aikataulusta laadittiin kehittäjäkohtaiset Excel-taulukot. Jotta kaikki pysyisivät ajan tasalla ohjelmiston kehitysvaiheesta ja tehdyistä suunnittelupäätöksistä, alettiin marraskuussa pitää sisäistä seurantalaveria noin kerran viikossa.

Joulukuun 2007 aikana valmistuivat Captum2:n perustoiminnot, joista on kuvaruutukaappauksia tämän opinnäytetyön liitteinä. Liitteiden kuvat ovat siis toimivasta ASP.NET -järjestelmästä, ne eivät ole enää ulkoasuprototyyppejä. Toteutuksen yksityiskohtiin esim. tietokantasuunnittelun tai käytettyjen algoritmien osalta ei tässä opinnäytetyössä puututa, koska työ on kokonaan julkinen. Integraatiotestauksen jälkeen perustoiminnot voidaan esitellä järjestelmän tilaajalle omassa pilottiympäristössään. Koska opinnäytetyö valmistui alkuperäisen aikataulun mukaisesti vuoden 2008 alussa, Captum2:n tilaajakohtaisten räätälöintien toteutus jäi tämän työn valmistuessa vielä työn alle. Räätälöintien valmistumista seuraa vaiheittainen käyttöönotto sekä järjestelmän ylläpitovaihe. Uudistettu Captum2 on valmis kohtaamaan tulevien vuosien haasteet.

4 JOHTOPÄÄTÖKSET

Tässä yhteenvedokappaleessa analysoidaan lyhyesti edellä esitettyjä tuloksia ja verrataan niitä viitekehyksessä esitettyihin asioihin. Opinnäytetyön aikana tehtyä Captum2:een liittyvää työtä voidaan tarkastella ainakin kolmella eri tasolla:

- Opinnäytetyön näkökulma, joka muodostuu teoreettisen viitekehyksen muodostamisesta ja soveltamisesta käytäntöön konstrukttiivisen tutkimusotteen mukaisesti.
- Hypermedia Oy:n näkökulma, joka muodostuu uuden ohjelmistotuotteen lanseerauksesta ja siihen liittyvästä tuotekehityksestä sekä yrityksen teknisen ajanmukaisuuden säilyttämisestä.
- Tilaajan näkökulma, joka muodostuu tarpeita vastaavan ohjelmistoratkaisun hankkimisesta ja käyttöönotosta eli investointiprojektista.

Teoriaosuudessa selvitettyjen asioiden tutkiminen on osaltaan auttanut mm. oikeiden ohjelmistotuotannon menetelmien mielessä pitämisessä silloin, kun käytännön tarpeet pyrkivät oikaisemaan liikaa. Toisaalta ketterien menetelmien soveltaminen on Captum2:n kaltaisissa projekteissa hyvin luontevaa. Opinnäytetyö valmistui aikataulun mukaisesti ja kattaa projektisuunnitelman mukaiset asiat pois lukien Captum2:n käyttöohjeen valmistelu, joka ei ehtinyt toteutukseen vuoden 2007 loppuun mennessä. Työn rakenne oli suunniteltu siten, että se olisi ollut mahdollista tehdä valmiiksi, vaikka tilausta Captum2:sta ei olisi vielä saatukaan.

Hypermedia Oy:n tuotekehitys on sujunut aikaisemmista kehityskohteista (mm. Hyperscriptor) saatujen kokemusten pohjalta varsin hyvin, vaikka kehitystyön resurssien riittävyys oli edelleen pieni ongelma. Captum2:n tuotteistuksessa on saavutettu vuoden 2007 aikana merkittäviä tuloksia aikaisempaan versioon verrattuna, ja Captum2 on tämän opinnäytetyön valmistuessa kaikilta perusominaisuuksiltaan valmis. Osa alun perin määritellyistä version 3.5 toiminnoista, esim. ylläpitäjän automaattinen easy shop -toiminto, on toistaiseksi jäänyt toteuttamatta. Niihin varmasti palataan viimeistään seuraavan versiopäivityksen yhteydessä.

Järjestelmän ensimmäisen tilaajan näkökulmasta toimitusprojekti on tämän opinnäytetyön valmistuessa vielä kesken, koska tilaajakohtaisia räätälöintejä ei ole toteutettu. Ne ovat kuitenkin määritelty valmiiksi ja toteutetaan tilaajan pilottiversioon alkuvuoden 2008 aikana. Toiminnot laajentavat Captum2:n toimintaa mm. laskutusjärjestelmän suuntaan. Osa tilaajakohtaisissa projekteissa kehitettävistä lisäominaisuuksista voidaan tulevaisuudessa sisällyttää ohjelmiston seuraavaan versioon, kuten on tehty Hyperscriptorin kohdalla. Näin Hypermedia Oy:lle kertyy monikäyttöinen oma

työkalu ja siihen liittyvä osaaminen, joiden avulla se voi tarjota ratkaisuja hyvin erilaisiin tarpeisiin.

Kritiikkiä voidaan esittää siitä, että viitekehyksessä ei ole käsitelty kovin laajasti erilaisia teknisiä perusvaihtoehtoja dynaamisen web-sivuston toteuttamiseksi, vaan sitä voi jopa väittää Microsoft-painotteiseksi. Tämä johtuu siitä, että opinnäytetyön toimituksessa työelämän kehittämistehtävänä tarkoituksena ei ollut etsiä vaihtoehtoja Hypermedia Oy:ssä käytetyille menetelmille vaan toteuttaa niiden avulla uusi ohjelmistoversio. Hypermedia Oy:n Microsoft-osaaminen on tämän projektin aikana kasvanut, mikä oli myös yksi opinnäytetyön tavoitteista. Vastaavan järjestelmän toteuttaminen esim. Java-kielellä tai Microsoft Commerce Server -alustalle olisi täysin mahdollista, mutta edellyttäisi lähtökohtaisesti toisenlaista teknologista osaamisprofiilia.

Kritiikkiä voi esittää myös siitä, että Captum2:n uudessa versiossa ei ole laajemmin hyödynnetty AJAX-tekniikan mahdollisuuksia. Niiden toteutus kireän aikataulun puitteissa olisi kuitenkin ollut liian suuri riski uuden version käytettävyydelle ja toisaalta perinteisemmät ratkaisut riittivät vielä hyvin. Tämäkin on asia, johon varmasti palataan tuotekehityksen jatkuessa. Web 2.0 -konseptin mukainen vuorovaikutteisuus on toteutettavissa esim. kommentoitavilla tuotetiedoilla tai toivelistoilla.

Captum2 versio 3.5:n käyttöliittymä ei hyödynnä kuin murto-osan kaikesta siitä potentiaalista, jonka nykyiset tietokoneet tarjoavat, mutta samalla se toimii tehokkaasti myös hieman vanhemmilla mikroilla ja hitaammilla yhteyksillä. Palvelinpään vaatimuksetkin ovat varsin kohtuulliset ja muodostuvat tavanomaisilla käyttökuormilla lähinnä käyttöjärjestelmän, web-palvelimen ja tietokannan asettamista vaatimuksista. Tehokkaalla laitteistolla ja nopealla yhteydellä järjestelmä skaalautuu suuriinkin tapahtumamääriin ja esim. liikkuvan kuvan jakeluun.

Captum2:n kaupalliset näkymät näyttävät uuden version myötä aikaisempaa paremmilta. Se on yhtä kilpailukykyinen tuote kuin Hyperscriptor XG, esim. käyttöönotto on merkittävästi aikaisempaa versiota nopeampaa, mikä oli yksi päivitysprojektin tavoitteista. Captum2, Hyperscriptor XG ja muut web-järjestelmät ovat osa sitä kokonaisuutta, joka muodostaa ympärillemme sähköisen todellisuuden, joka ei ole sidottu aikaan tai paikkaan.

LÄHTEET

Ahonen, T., Hämeen-Anttila, T. & Åstrand, K. 2003. JavaServlets. Jyväskylä: Docendo Finland Oy.

Beck, K., Beedle, M., Bennekum, A. et al. 2001. Manifesto for Agile Software Development [online, viitattu 3.1.2008]. Saatavissa <http://www.agilemanifesto.org>

Falck, K. 1999. Tietokantojen WWW-julkaisun vaihtoehdot. Tietokone 18 (6-7), 105-110.

Haikala, I. & Märijärvi, J. 2006. Ohjelmistotuotanto. 11. painos. Helsinki: Talentum Media Oy.

Hintikka, K. 2007. Web 2.0 sisältää elämän koko kirjon. Helsingin Sanomat 23.1.2007, D1.

Koikkalainen, P. & Orponen, P. 2002. Tietotekniikan perusteet [online]. Jyväskylän yliopisto, Teknillinen korkeakoulu. [Viitattu 3.1.2008]. Saatavissa <http://erin.mit.jyu.fi/pako/kurssit/perusteet/kirja/kirja.html>

Lukka, K. 2001. Konstruktiivinen tutkimusote [online]. www.metodix.com [Viitattu 3.1.2008].

Maurer, D. 2006. Usability for Rich Internet Applications [online]. Digital Web Magazine. [Viitattu 3.1.2008]. Saatavissa http://www.digital-web.com/articles/usability_for_rich_internet_applications

Pelin, R. 2004. Projektihallinnan käsikirja. 4. uudistettu painos. Helsinki: Projekti-johtaminen Oy.

Refsnes Data W3Schools. ASP.NET Tutorial: Differences between ASP and ASP.NET [online, viitattu 3.1.2008]. Saatavissa http://www.w3schools.com/aspnet/aspnet_vsasp.asp

Schneider, R. 1997. Microsoft SQL Server, Planning and Building a High Performance Database. New Jersey: Prentice-Hall, Inc.

Soprano Oyj 2007. Verkkokauppa nousee Suomessa lähes neljään miljardiin euroon [online, viitattu 3.1.2008]. Lehdistötiedote, saatavissa <http://www.soprano.fi/recent/156-1169-13536.html>

Sähköisen kaupan palvelukeskus 2007. Suomi nousee vuonna 2010 Euroopan verkkokaupan huipulle [online, viitattu 3.1.2008]. Lehdistötiedote, saatavissa <http://www.e-finland.org/center/etusivu/informaatiota/uutispalvelu/tiedotteet/?ID=press:3401>

Wikipedia. Unified Modeling Language [online, viitattu 3.1.2008]. Saatavissa http://en.wikipedia.org/wiki/Unified_Modeling_Language

Tuoteryhmä

Kirjakauppa Atlantis - Windows Internet Explorer
http://www.keittokirja.com/hyperscriptor3.5/sivu.aspx?taso=0&id=23

Kirjakauppa Atlantis x Hyperscriptor 3.5 XG - Yllä...
Kirjakauppa Atlantis Yhteystiedot Extranet

Hyperscriptor 3.5

Kirjakauppa Atlantis

Hae

- Sivukartta
- Suomeksi
- In English

Ostoskoriin sisältö

Tuotteita	1
Summa	10,87 €

Tuotteet:
Blink: The Power ...

Siirry koriin

Kirjakauppa Atlantis

Kirjavuosi 2008 tuo mukanaan paljon kiinnostavia uutuuksia niin romaaniin kuin tietokirjallisuuteenkin ystäville. Alla muutamia kiinnostavia polimintoja valikoimastamme.

► Päivän tarjoukset

► Ajankohtaista

Clinically Oriented Anatomy

Hinta	57 €
Tekijä	Keith L. Moore
Sivuja	1104
Kategoria	Lääketiede

Lisää koriin

The Seven Daughters of Eve: The Science That Reveals Our Genetic Ancestry

Hinta	18,95 €
Tekijä	Bryan Sykes
Sivuja	320
Kategoria	Arkeologia

Lisää koriin

The Elegant Universe

Hinta	13,57 €
Tekijä	Brian Greene

Valmis

Internet | Suojattu tila: Käytössä

Tuotekortti

Kirjakauppa Atlantis Yhteistyöedot Extranet

Hyperscriptor 3.5

► Päivän tarjoukset
► Ajankohtaista

The SEVEN DAUGHTERS OF EVE

THE SEVEN DAUGHTERS OF EVE: THE SCIENCE THAT REVEALS OUR GENETIC ANCESTRY

1 kpl. á 18,95 €
Lisää korin

Hinta	18,95 €
Tekijä	Bryan Sykes
Sivuja	320
Varastosaldo	37
Kategoria	Arkeologia
Sisäitökuvaus	The Seven Daughters of Eve is a book by Bryan Sykes that presents the theory of Human mitochondrial genetics to a general audience. Sykes explains the principles of genetics and human evolution, the particularities of mitochondrial genetics, and analyses of ancient DNA to genetically link modern humans to prehistoric ancestors. Following the developments of mitochondrial genetics, Sykes traces back human migrations, discusses the out of Africa theory and refutes Heyerdahl's theory of the Peruvian origin of the Polynesians, which opposed the theory of their origin in Indonesia. He also describes the use of mitochondrial DNA in identifying the remains of Czar Nicholas II, and in assessing the genetic makeup of modern Europe.

Ostoskorin sisältö

Tuotteita	1
Summa	10,87 €

Tuotteet:
Blink: The Power ...

Siirry korin

Tuosta tuotekortti

Hypermedia Oy | Tiedepuisto | 28600 Porri | 02-529 8860 | hypermedia@hypermedia.fi

Valmis

Internet | Suojattu tila: Käytössä

Keräilylista (ostoskori)

Kirjakauppa Atlantis - Windows Internet Explorer
http://www.keittokirja.com/hyperscriptor3.5/captum_tuotekoritti.aspx?tas=0&id=23&tid=103

Kirjakauppa Atlantis x Hyperscriptor 3.5 XG - Yllä...
Kirjakauppa Atlantis Yhteistyöt - Extranet

Hyperscriptor^{xg} 3.5

Kirjakauppa Atlantis Yhteistyöt Extranet

- Päivän tarjoukset
- Ajankohtaista

Ostoskori

Hae

- Sivukatta
- Suomeksi
- In English

Ostoskorin sisältö

Tuote	Ä-hinta	Määrä	Hinta	Poisia
Blink: The Power of Thinking Without Thinking	10,87 €	1	10,87 €	Poisia
The Seven Daughters of Eve: The Science That Reveals Our Genetic Ancestry	18,95 €	2	37,90 €	Poisia

Yhteensä: 48,77 €

Tuotteet:
Blink: The Power ...

Siirry koriin

Pyydä tarjous

Hypermedia Oy | Tiedepuisto | 28600 Pori | 02-529 8860 | hypermedia@hypermedia.fi

Internet | Suojattu tila: Käytössä

Valmis

Tilauslomake

Kirjakauppa Atlantis - Windows Internet Explorer
http://www.keittokirja.com/hyperscriptor3.5/captum_tuotekorppi.aspx?taso=0&id=23&trid=103

Kirjakauppa Atlantis x Hyperscriptor 3.5 XG - Yllä...
Kirjakauppa Atlantis Yhteystiedot Extranet

Hyperscriptor^{xg} 3.5

Kirjakauppa Atlantis Yhteystiedot Extranet

- Päivän tarjoukset
- Ajankohtaista

Tilauslomake

Nimi *
Lähiosoite *
Postinumero *
Postitoimituspaikka *
Sähköposti
Lisätietoja

Hae

- Sivukatta
- Suomeksi
- In English

Ostoskorin sisältö

Tuotteita 3
Summa 48,77 €

Tuotteet:
Blink: The Power ...

Siirry koriin

Tuote	Yksikköhinta	Määrä	Hinta
Blink: The Power of Thinking Without Thinking	10,87 €	1	10,87 €
The Seven Daughters of Eve: The Science That Reveals Our Genetic Ancestry	18,95 €	2	37,90 €

* Merkityt kentät ovat pakollisia

Lähetä

Hypermedia Oy | Tiedepuisto | 28600 Pori | 02-529 8860 | hypermedia@hypermedia.fi

Valmis

Internet | Suojattu tila: Käytössä

Tuotemallin muokkaus

Pääkäyttäjä - Windows Internet Explorer
 http://www.keittokirja.com/hyperscriptor3.5/admin/paakayttaja/captum_tuotemallimuokkaa2.aspx?d=37

Kirjakauppa Atlantis

Hyperscriptor^{xc}

Sivuston pääkäyttötoimet
 Versio: Hyperscriptor XG 3.5
 Pääkäyttäjä: Hypermedia Oy

Sivut
 Käyttäjätunnukset
 Otsikot ja käyttöoikeudet
 Kirjautumisallasana
 Yleiset
Tuotemallit
 Tilauslomakkeet
 Siirry ylläpitoon

Kirjautuu ulos

Pääkäyttäjä

Tuotemalli - muokkaa kenttiä

Lisää kenttä

Järjestys	Tyyppi	Kentän nimi	Kentän muoto	Poista kenttä
1	*	Nimi Name	<input type="checkbox"/> Yksirivinen tekstikenttä	
2	*	Hinta Price	<input type="checkbox"/> Yksirivinen tekstikenttä	
3		Tekijä [Empty]	<input type="checkbox"/> Yksirivinen tekstikenttä	
4		Sivuja [Empty]	<input type="checkbox"/> Yksirivinen tekstikenttä	
5	*	Pieni kuva Small image	<input type="checkbox"/> Liitetiedosto	
6	*	Varastosaldo Quantity	<input type="checkbox"/> Yksirivinen tekstikenttä	
7		Kategoria [Empty]	<input type="checkbox"/> Pudotusvalikko	
8	Y	Tuote on myynnissä Active product	<input type="checkbox"/> Rasti ruutuun -kenttä	
9	Y	Tuote on näkyvässä Product is visible	<input type="checkbox"/> Rasti ruutuun -kenttä	
10		Sisältökuvaus Content description	<input type="checkbox"/> Monirivinen tekstikenttä	

Tallenna tiedot

Kirjautuu ulos

Internet | Suojattu tila: Käytössä

Valmis

Tuotteen muokkaus

Hyperscriptor 3.5 XG - Ylläpito - Windows Internet Explorer

http://www.keittokirja.com/hyperscriptor3.5/admin/captum_lisaamuokkautuote2.aspx?Id=103

Hyperscriptor 3.5 XG - ... X

Kirjakauppa Atlantis

Hyperscriptor^{xg}

Sivuston ylläpito
 Versio: Hyperscriptor XG 3.5
 Ylläpitäjä: Hypermedia Oy

Sivut

- Suomi
 - Kirjakauppa Atlantis
 - Yhteyshiedot
 - Extranet
- English
 - Blue wings
 - City of Duck
 - Taloville

Captum2

- Tuotteiden hallinta
- Tilaukset ja tarjouspyynnöt
- Raportit

Lisätoiminnot ▾

Omat tiedot ▾

MUOKKAA TUOTETTA

Valitse tuotemalli: Kirjakauppa

Valitse kieli: Suomi

Nimi: The Seven Daughters o*

Hinta: 18,95 *

Tekijä: Bryan Sykes

Sivuja: 320

Pieni kuva (max. leveys 150 px)

Varastosaldo: 35

Kategoria: Arkeologia

Tuote on myynnissä

Tuote on näkyvässä

Sisältökuvaus: The Seven Daughters of Eve is a hank hy

Kirjakauppa ulos

Internet | Suojattu tila: Käytössä

Valmis

Tuoteryhmän muokkaus (Muokkaa-valikko avattuna)

The screenshot displays the Hyperscriptor 3.5 XG web application interface. The browser window shows the URL <http://www.keittokirja.com/hyperscriptor3.5/admin/muokkaa sivu.aspx?taso=0&id=23>. The page title is "Kirjakauppa Atlantis".

The main content area features a product description for "Kirjakauppa Atlantis" and a "Muokkaa" (Edit) button. The "Muokkaa" menu is open, showing options: "Iirota lomake", "Ominaisuuksia", "Yhteystietoja", "Ajastusta", and "Tuotteita".

The sidebar on the left contains a "Sivut" (Pages) menu with the following items: "Suomi", "Kirjakauppa Atlantis", "Päivän tarjoukset", "Ajankohtaista", "Lisää sivu", "Sivujen järjestys", "Yhteystiedot", and "Extranet".

The bottom of the page shows the "Sivuston ylläpito" (Site Management) section, including "Versio: Hyperscriptor XG 3.5", "Ylläpitäjä: Hypermedia Oy", and a "Kirjakauppa Atlantis" logo.

The browser's address bar and the application's navigation menu are also visible, showing the current page is "Kirjakauppa Atlantis".

Tuotteiden liittäminen tuoteryhmään

The screenshot shows the Hyperscriptor 3.5 XG web application interface. The browser window title is "Hyperscriptor 3.5 XG - Vlläpito - Windows Internet Explorer". The address bar shows the URL: "http://www.keittokirja.com/hyperscriptor3.5/admin/captum_liitatuotteita.aspx?taso=0&id=23&m=1".

The main content area is titled "LIITÄ TUOTTEITA". It contains a form with the following fields and options:

- Tuotemalli: Kirjakauppa
- Nimi: [Text input field]
- Hinta: [Text input field]
- Varastosaldo: [Text input field]
- Tuote on myynnissä:
- Tuote on näkyvissä:
- Tekijä: [Text input field]
- Sivuja: [Text input field]
- Kategoria: [Dropdown menu]
- Sisältökuvaus: [Text input field]
- Hae: [Search button]

Below the form, there is a section titled "Sivulle on liitetty seuraavat tuotteet" (Products added to the page). It contains a list of products:

- 1 Clinically Oriented Anatomy
- 2 The Seven Daughters of Eve: The Science That Reveals Our Genetic Ancestry
- 3 The Elegant Universe
- 4 Your Inner Fish
- 5 Blink: The Power of Thinking Without Thinking

At the bottom of this list is a button labeled "Tallenna tiedot" (Save data).

On the left side of the interface, there is a sidebar with the following sections:

- Sivuston ylläpito** (Version: Hyperscriptor XG 3.5, Ylläpitäjä: Hypermedia Oy)
- Sivut** (Pages):
 - Suomi
 - Kirjakauppa Atlantis
 - Päivän tarjoukset
 - Ajankohtaista
 - Lisää sivu
 - Sivujen järjestys
 - Yhteystiedot
 - Extranet
 - English
 - Blue wings
 - City of Duck
 - Taloville
- Captum2
- Lisätoiminnot
- Omat tiedot

The bottom right corner of the browser window shows the status "Valmis" and "Internet | Suojattu tila: Käytössä".