



Ohjelmointioppaan kirjoitusprosessi PHP ja MySQL -kirjan kirjoittaminen

Tampereen ammattikorkeakoulu
Viestinnän koulutusohjelman opinnäytetyö
Vuorovaikutteisuuden suunnittelun
suuntautumisvaihtoehto
Syksy 2007
Ilkka Rauta

OPINNÄYTETIIVISTELMÄ

Osasto Viestintä	Erikoistumisala Vuorovaikutteisuuden suunnittelu
Tekijä Ilkka Rauta	
Työn nimi Ohjelmointioppaan kirjoitusprosessi – PHP ja MySQL -kirjan kirjoittaminen	
Lopputyön laji Mediateko	
Työn valmistumisaika 10.12.2007	Sivumäärä 26
<p>Tiivistelmä</p> <p>Opinnäytteeni koostuu kirjallisesta osuudesta sekä mediateosta. Mediateko on neljännen, lähes täysin uudistetun painoksen käsikirjoitus web-ohjelmoinnin perustason oppaana toimivaan kirjaan "PHP ja MySQL: tietokantapohjaiset verkkopalvelut". Kyseessä on yhteisprojekti kirjan aiemmat painokset kirjoittaneen Rami Heinisuon kanssa.</p> <p>Mediateon tietopohjana on käytetty ensisijaisesti ohjelmistojen omia virallisia ohjelmointioppaita. Asiat käydään läpi kirjassa ensisijaisesti koodiesimerkkien ja esimerkkisovellusten avulla monimutkaisuusjärjestyksessä alkaen perusteista.</p> <p>Kirjallisessa osuudessa selostetaan kirjoitusprosessia tarkemmin sekä vertaillaan kirjaa kolmen muun jossain määrin vastaavan kirjan kanssa. Vertailun perusteella on mahdollista saada uudenlaista näkökulmaa lopputulokseen ja näin saadun tiedon avulla tuottaa parempaa jälkeä tulevaisuudessa.</p> <p>Johtopäätöksenä todetaan että vaikka päätavoitteisiin päästiin, hyvää ohjelmointitapaa olisi voitu käsitellä yleisesti enemmän, esitellä syvällisemmin oikeaoppisempia tapoja toteuttaa www-sovellus sekä tarjota vinkkejä ja lisätietoa usein tehdyistä, mutta esimerkkisovelluksiin sopimattomista asioista "keittokirjamaisesti".</p>	
Aineisto	
Asiasanat kirjoittaminen, tietokirjallisuus, ohjelmointi	
Säilytyspaikka TAMK / Taide ja viestintä	
Muita tietoja	

THESIS	SUMMARY
Department Media Programme	Area of specialisation Interactivity Design
Author Ilkka Rauta	
Title Writing process of a programming guide – writing the book PHP ja MySQL	
Sort of Final Thesis (Written / Project / Portfolio) Project	
Date 10.12.2007	Number of pages 26
<p>Summary:</p> <p>My thesis is composed of a written part and a project. The project is script for fourth, almost completely rewritten edition of a book named "PHP ja MySQL: tietokantapohjaiset verkkopalvelut" ("PHP and MySQL: database-based online services"). The script is written in collaboration with Rami Heinisuo, who has written the three previous editions.</p> <p>The official programming manuals of the software used in the book are forming the knowledge base of the project. The book starts from the basics and advances to more complex topics with the help of code samples and example applications.</p> <p>The written part of the thesis describes the writing process in detail and compares the book with other three books that concentrate roughly on the same topic. Based on the comparison it is possible to get new perspective to the results of the project, and new information gained can be used to produce better results in the future.</p> <p>As a conclusion it is stated that although the main goals were met, good programming practices could have been handled more and more orthodox ways to accomplish things could have been described in depth. In addition, tips and information on often used techniques, which did not fit in the example applications could have been presented in a "cookbook-like" way.</p>	
Material	
Key words writing, non-fiction literature, programming	
Filing TAMK University of Applied sciences / Art and Media	
Other information	

Sisällysluettelo

1. Johdanto.....	5
2. Tietoperusta.....	7
3. Kirjoitusprosessi ja esimerkkikoodin kehitys.....	9
4. Tarkempaa tarkastelua.....	13
5. Vertailua toisiin kirjoihin.....	16
5.1 Java, Prolog ja Python.....	16
5.2 CGI-ohjelmointi.....	19
5.3 Java-ohjelmoinnin peruskirja.....	21
6. Johtopäätökset.....	23
7. Lähteet.....	26

1. Johdanto

Tutkintotyökseni olen päivittänyt PHP ja MySQL -kirjan neljännen painoksen käsikirjoitusta ajantasaiseksi, vastaamaan nykyisiä versioita kirjan nimessä mainituista ohjelmistotuotteista. Kirja on käytännönläheinen opas web-sovellusten ohjelmointiin PHP-ohjelmointikieltä ja MySQL-tietokantajärjestelmää käyttämällä. Kirjan aiempia painoksia on käytetty kurssikirjoina ja eri painoksia nykyinen mukaan luettuna yhdistää se, että niissä käydään asioita läpi perusteista alkaen. Tämä tutkintotyön kirjallinen osuus pyrkii selvittämään sen, millaisen prosessin kautta tällaisen kirjan käsikirjoitus syntyy ja miten kirjoitustyön lopputulosta olisi mahdollista parantaa.

Kirja keskittyy selittämään lukijalle miten tietokantapohjaiset web-sovellukset toimivat käytännön tasolla. Kyseessä ei ole missään nimessä lajinsa ainoa edustaja, mutta esimerkiksi Ajax-tekniikoiden tai MySQL-tietokannan kehittyneempien uusien ominaisuuksien käsittely on käsittääkseni vielä melko vähäistä suomenkielisessä tietokirjallisuudessa, joten kirja tuskin vielä hukkuu täysin samoja aiheita käsittelevien teosten joukkoon.

Vaikka kirjasta onkin olemassa kolme aiempaa julkaistua painosta, kyseessä on silti lähes täysin uusi kirja. Käsitellyt tekniikat ovat muuttuneet merkittävästi niiden edellistä painosta seuranneiden muutaman vuoden aikana, joten isommalle päivitykselle oli todella tarvetta. Tämä tarkoittaa sitä että kyseessä ei ole pelkkä pieniä muutoksia sisältävä korjauksenomainen päivitys vaan lopulta tekstistä suurin osa kirjoitettiin kokonaan uusiksi vanhan tekstin jäädessä sirpaleiksi sinne tänne.

Kirjasta on siis kolme aiempaa painosta, vuosilta 2001, 2003 ja 2004. Edellisestä painoksesta on kulunut kolme vuotta, ja se puolestaan on ollut enemmänkin asteittaisesti ensimmäisestä painoksesta eteenpäin viety versio verrattuna nyt tulleeseen lähes täysin uusittuun painokseen.

Tutkintotyöni on määritelty mediateoksi, vaikka se on aivan selvästi luonteeltaan kirjallinen. Syynä tälle on se, että työ ei kuitenkaan olisi sellaisenaan kelvollinen kirjalliseksi tutkintotyöksi sillä kirja ei ole otteeltaan tutkiva, mikä on yksi kirjallisen tutkintotyön tekijöistä. Kun kyseessä on oppimateriaalimaisen kirjan käsikirjoituksen

tuottaminen, lopputulosta pitäisi muokata merkittävästi jotta se kävisi kirjallisesta tutkintotyöstä. Mediateon määritelmän pitäisi kuitenkin täytyä aivan riittävän hyvin, joten tätä työtä on tuskin syytä pitää mitenkään erityisesti rajatapauksena näiden kahden suoritustavan välillä.

Toinen syy sille, että työ on juuri mediateko eikä kirjallinen tutkinto, on se etten ole sen ainoa kirjoittaja. Toisena tekijänä on Rami Heinisuo, jonka aloitteesta ylipäättään päädyin tähän kirjoitustyöhön. Tästä huolimatta valtaosa neljännen painoksen tekstistä on minun kirjoittamaani, ja vaikka monessa kohdassa taustalla onkin myös edellinen painos, suoraan sellaisenaan käytettyä tekstiä on mukana vain vähän.

Syksyllä 2006 Rami tiedusteli minulta kiinnostusta tähän kirjoitusprojektiin. Samaan aikaan minun piti tehdä päätös siitä minkä aiheen valitsen tutkintotyölleni, ja päätös lähteä mukaan oli helppo tehdä. Suunnitelma kirjan rakenteesta tehtiin jonkin verran ennen vuodenvaihdetta ja varsinainen kirjoitustyö alkoi vuoden 2007 puolella. Eteneminen tapahtui pääosin alusta loppua kohden tasaiseen tahtiin kesään asti. Alun perin tavoitteena oli saada kirja valmiiksi kesäkuun alussa, mutta tähän tavoitteeseen ei päästy. Lopulta kirja saatiin painoon vasta juuri ennen elokuun alkua. Oma osuuteni päättyi käytännössä siinä vaiheessa kun käsikirjoitustiedostot ja oheis-CD-ROM:n tiedostot lähetettiin viimeisen kerran kustantajalle – ennen viimeisiä versioita tiedostoja pompoteltiin edestakaisin viimeisiä korjauksia ja pikkumuutoksia varten.

Sisällysluettelon merkintöjen tasolla katsottuna työt jakautuivat näin (sivumäärät ovat sisällysluettelosta laskettuja ja summittaisia):

Lähtökohtia (6 sivua): Rami

Asennukset ja määrittelyt (48 sivua): Alkuosuus pohjautuu pitkälti Ramin tekstiin, muuten minä

PHP-ohjelmointi käytännössä (34 sivua): Pohjautuu erittäin paljon edelliseen painokseen alkupuolella

MySQL-tietokannan käyttö (84 sivua): Alkuosa pohjautuu edelliseen painokseen, mutta vähemmän loppupuolelta

Ensimmäinen sovellus – "Rakas päiväkirjani" (68 sivua): Minä

Toinen sovellus – Kalenteri monelle käyttäjälle (132 sivua): Minä

Sovellusten arviointi ja kehittäminen (24 sivua): Rami ja joiltain osin minä

Liite: Siivoaminen eli PHP:n, Apachen ja MySQL:n poistaminen (4 sivua): Minä

Alkupuoli kirjasta perustuu enemmän edelliseen painokseen mutta loppua kohden on jo siirrytty käytännössä täysin uuteen sisältöön. Alku- ja loppupään kappaleet ovat ensisijaisesti Ramin käsialaa, kun minä taas tein alun ja lopun välissä olevaa osuutta. Jaksojen sivumäärät kasvavat loppua kohden, tosin kaksi viimeistä ovat kuitenkin lyhyempiä "lopettelupätkiä". Kirjan pääosien rakenne on ensi sijassa Ramin suunnittelema, mutta olen itse pääosin vastuussa tekemiäni osien sisäisestä rakenteesta.

2. Tietoperusta

Kahtena ensisijaisena tietolähteenä ovat olleet ajan kuluessa karttunut kokemuspohja sekä kirjassa käsiteltyjen tekniikoiden oma dokumentaatio. Ensiksi mainittu on jonkinlainen edellytys sille, että tämä projekti olisi ylipäättään mahdollinen. Asiaa on suhteellisen vaikea selittää lukijalle riittävän tarkasti, ellei asia ole itsellä riittävän hyvin hallussa. Toisena mainittu lähde, tekniikoiden dokumentaatio on tämän luonteisessa kirjassa myös varsin välttämätön, sillä kaikkea ei voi missään nimessä muistaa tarpeeksi hyvin ja toisaalta, käytännössä vain viralliset lähteet pystyvät antamaan riittävästi oikeaa informaatiota vaikka jossain tapauksessa myös ulkopuolisista lähteistä on hyötyä.

Oma henkilökohtainen kokemuspohja on ollut tässä työssä välttämätön edellytys onnistumiselle. Ilman kokemusta olisin periaatteessa melko lailla samalla viivalla kuin lukijatkin, mikä ei ainakaan periaatteessa vaikuta toimivalta lähtökohdalta, enkä pidä todennäköisenä että pelkkää teknistä dokumentaatiota lukien tehty teksti olisi yhtä hyvää kuin kokemuksen ja dokumentaation pohjalta tehtynä. Kokemuksesta on apua myös dokumentaatioiden lukemisessa, vaikka MySQL:n ja PHP:n www-dokumentaatiot ovatkin varsin informatiivisia ja helppoja käyttää.

Tekstin tuottaminenkin on viime kädessä jonkin olemassa olevan tiedon käsittelyä – ainakin tässä tapauksessa kun tajunnanvirtakirjoittaminen ei käytännössä käy päinsä. Alla käsitellään ensin lähinnä tiedon keruuta ja "sisääntulevan" tiedon käsittelyä.

Mitä eri lähteistä kerätyn tiedon käsittelyyn tulee, tekninen dokumentaatio on sinänsä useimmiten yksiselitteistä faktatietoa, joka ei vaadi erityistä analyysiä tai purkamista jotta koko totuus selviäisi – poikkeuksena tähän tietysti se, että yhdessä paikassa ei aina välttämättä mainita kaikkea kyseiseen aiheeseen liittyvää tietoa joten joskus asioista saattaa jäädä myös pelkkä osittainen kuva. Vakavia puutteita tiedoissa ei tällä tavalla pääse kovin helposti syntymään jos pystyy arvioimaan suurin piirtein millaisia rajoituksia tai mahdollisuuksia kullakin tekniikalla on. Esimerkiksi MySQL:n uusien ohjelmointiominaisuuksien toimintaa pystyy vertailemaan aiemmin tutuksi tulleiden toisten ohjelmointikielien vastaavien rakenteiden kanssa.

Itselleni PHP ja MySQL ovat entuudestaan tuttuja ja ne ovat olleet käytännössä pääasiallisia työkalujani koulun ohella tekemässäni työssä. Molempia tekniikoita on käsitelty myös koulussa sinä aikana kun olen opiskellut TTVO:lla. JavaScript-osaamiseni ei ole aivan samalla tasolla edellä mainittujen kanssa, ja tämän lisäksi erityisesti Ajax-käsitteen alle kuuluvat JavaScript-tekniikat eivät olleet minulle entuudestaan tuttuja. Kirjassa Ajaxiin liittyvistä kehittyneemmistä asioista käsitellään lähinnä XmlHttpRequest-tekniikkaa eikä esimerkiksi drag & drop -toiminnallisuutta HTML-sivulla. Oman kokemattomuuden, mitä Ajaxiin tulee, vaikutuksen lisäksi kehittyneemmät Ajax-asiat on jätetty vähemmälle käytännön syistä – Ajaxissa on itsessään asiaa monen kirjan aiheiksi ja toisaalta tässä kirjassa ei ole mahdollisuuksia kaivautua kovin syvälle aiheeseen.

Itse käsikirjoituksessa selitetään ne käsitteet joita siinä käsitellään ainakin pintapuolisesti; tarkoituksena on antaa riittävästi tietoa eikä selittää asioita joiden ymmärtämisestä ei ole välttämättä mitään suoranaista hyötyä. Toki kirja olettaa lukijaltaan jonkintasoista pohjatietämystäkin, mutta viime kädessä tähän kirjan ajatuksena on juuri selittää asioita. Maininnan arvoisena pidän tätä siksi että yleensä ottaen mediatekoa, kuin automaattisesti myöskään kirjallista tutkintotyötä, ei voida mitenkään välttämättä pitää luonteeltaan tällaisena. Koska asioiden selittäminen onnistuneesti on kirjan päämääränä, sitä voidaan pitää myös kriteerinä tuotoksen arvioinnissa.

Lukijalta oletetaan jonkin asteista tietoa asioista etukäteen. Käytännössä odotukset voitaisiin tiivistää muotoon "lukijan tulisi osata tuottaa staattisia HTML-sivuja". Peruskäsitteitä joita lukijan tulisi ymmärtää ovat tiivistetyt HTML, yleiskuva palvelin-

asiakas-suhteesta WWW-sivujen tapauksessa – joskin tätä asiaa myös selitetään kirjan alkupuolella – ja tietysti käsitys siitä mikä on vaikkapa selain ja palvelin. Mistään kovin monimutkaisista teknisistä yksityiskohdista ei tarvitse olla tietoinen, ja todennäköisesti "matalimman tason" asia, jota kirjassa sivutaan pikaisesti, on tietoliikenteessä käytettävä käsite portti. (Tarkalleen ottaen kyseessä on TCP-portti, mutta edes tätä asiaa ei mainita.)

3. Kirjoitusprosessi ja esimerkkikoodin kehitys

Menetelmät ja työtavat ovat tässä tapauksessa monessa suhteessa itsestään selviä, mitä käytännön suoritukseen tulee: teksti tuotetaan tekstinkäsittelyohjelmalla ja esimerkkitehtävät tuotetaan pääosin kirjassa käsitellyillä työkaluilla.

Tekstinkäsittelyohjelmana toimi OpenOffice.org Writer, mutta loppujen lopuksi tiedostot oli käytävä läpi ja muokattava lopulliseen muotoonsa Microsoft Wordin avulla, sillä käytetty taitto-ohjelma eivät tunnista OpenOffice.org Writerin tiedostot toisin kuin Word-tiedostot. Taitto-ohjelmalla tuotetaan PDF-tiedosto joka sitten lopulta painetaan. Oma osuuteni prosessissa päättyi tuotetun Word-tiedoston lähettämiseen kustantajalle – tämän takia tämän työn kuvaus on käsikirjoituksen tuottaminen eikä kirjan tekeminen. OpenOffice.org Writerilläkin saadaan aikaan Word-tiedostoja, mutta näiden toimivuudesta ei ole samanlaista takuuta kuin itse Wordilla tehtyjen tiedostojen kanssa on.

Itse kirjassa käsiteltävien työkalujen ja niiden versioiden valinta melko vähän harkintaa; valinnat olivat itse asiassa melko lailla itsestään selviä: PHP:sta, MySQL:stä ja Apachesta valittiin kirjoituksen alkuvaiheessa tuoreimmat vakaat versiot joissa pysyttiin jotta välttyttäisiin "liikkuvan alustan" mukanaan tuomia ongelmia. Tämä yhteen versioon jääminen aiheutti lopulta tosin myös ongelmia. PHP:sta ei ole erilaisia variaatioita siinä mielessä kuin MySQL:stä on joten sen suhteen valintaa ei oikeastaan tarvinnut tehdä. MySQL:stä valittiin ilmainen yhteisöversio. Apachen tuorein versio on sopiva, vaikka todennäköisesti vanhempikin versio olisi toiminut yhtä hyvin, Apache on palvelinalustana varsin vakaa ja rauhallisesti muuttuva. Apuvälineohjelmien (tiedostonsiirto-ohjelma ja koodieditori) kanssa valinta oli yhtä lailla helppo tehdä – uusin vakaa versio on todennäköisesti paras. Ohjelmat ovat myös tunnettuja ja hyvämaineisia. Virtuaalikoneohjelmistosta oli tullut juuri sopivaan aikaan uudempi versio, samoin

virtuaalikoneessa käytettävästä CentOS Linuxista, joten nämä tuoreimmat versiot valittiin käytettäväksi virtuaalikoneohjelmistoiksi.

Esimerkkien sisältö määräytyi pitkälti siitä että lähtötaso on suhteellisen matala; ensimmäiset esimerkit selostavat perustason asioita, joten ne täytyy pitää selkeinä jottei käsiteltävä asia sekoitu esimerkin kanssa. "Montako omenaa Kallelle jäi jäljelle?" -henkeä noudattavat esimerkit ovat tästä syystä melko käytettyjä naiiviudestaan huolimatta. Esimerkiksi pankkimaailmaan sijoittuvat esimerkit ovat oman kokemukseni mukaan kuivia, ja hiukan lapselliset esimerkit ovat toivottavasti sopivaa kevennystä muuten melko raskaaseen ja kuivaan tekstiin. Lukijalle mahdollisesti tuntemattomaan ympäristöön – vaikkapa juuri pankkimaailmaan eli myös minulle outoon ympäristöön – sijoittuvat esimerkit saattavat aiheuttaa lisäsekaannusta koska esimerkissä käsiteltävä ohjelmointiin liittyvä käsite ja esimerkkitapaus saattavat sekaantua toisiinsa.

Kielellisesti vastaan tuli joitain mielenkiintoisia tilanteita, sillä kaikille termeille ei ole niin vakiintuneita vastineita suomen kielestä että esimerkiksi internetistä haettaessa löytyisi merkittävältä vaikuttavaa määrää tuloksia. Tällaisia tapauksia ovat viestitiivistealgoritmit (message digest algorithm, tietoturvasanastoa), soketti (socket, verkkotekniikkasanastoa) ja vaikkapa vasemmanpuoleinen ulkoliitos (left outer join, SQL-kielen ilmaus). Viimeisintä näistä pidän sopivana käännöksenä, sillä sitä näytettäisiin käytettävän yliopistojen oppimateriaaleissa. Sokettikin vaikuttaisi suhteellisen käytetyltä käännökseltä, vaikka onkin loppujen lopuksi melko tökerö väännös englannista. Viestitiivistealgoritmin olisi voinut kääntää pelkäksi viestitiivisteeksi, mutta silloin tiiviste-sana olisi voinut saada siinä määrin lisää painoa yhdyssanassa jolloin lukijalle voisi tulla mieleen tiivistäminen pakkausmenetelmänä.

Koodiesimerkit ovat lyhyitä koodinpätkiä jotka esittelevät jotain yksittäistä asiaa tai käsitettä. Koodiesimerkkejä käytetään esittelemään perusasioita joita tarvitaan myöhemmin esimerkkisovellusten kanssa. Kaikki koodiesimerkit eivät kuitenkaan ole sellaisia että niitä käytettäisiin oikeasti sovelluksissa; varsinkin MySQL-tietokantajärjestelmän uusia monimutkaisia toimintoja ei nähdä kirjassa minkäänlaisena käytännön sovelluksena. Niitä käyttävät kokonaiset esimerkkisovellukset olisivat todennäköisesti liian monimutkaisia käsiteltäväksi tarpeeksi lyhyesti.

Esimerkkisovellukset ovat puolestaan suurempia kokonaisuuksia, joissa pääpaino on toimivan kokonaisuuden aikaan saamisella. Kirjassa on kaksi esimerkkisovellusta joista ensimmäisen on tarkoitus vain antaa perusteita PHP-MySQL-yhdistelmän käytöstä ja se onkin varsin rajallinen sovelluksena. Toinen sovellus taas on laajempi ja se on huomattavasti lähempänä oikeasti käyttökelpoista web-sovellusta.

Esimerkkisovelluksissakin tulee toki uutta informaatiota lukijalle, mutta asioita käsitellään silti pääosin suuremman kokonaisuuden kannalta. Toisen esimerkkisovelluksen kanssa uutta tietoa tulee kuitenkin selkeästi ensimmäistä sovellusta enemmän, joten sitä käsittelevän luvun alussa on vielä oma koodiesimerkeistä koostuva jaksonsa jossa käsitellään uusia asioita perustasolla.

Testaaminen on koodiesimerkkien tapauksessa pääosin helppoa, sillä lyhyt esimerkki vaatii vain vähän koodia ympärilleen muodostaakseen riittävän kokonaisuuden testaamista varten. Poikkeuksena tähän kuitenkin ovat MySQL-tietokantaa käsittelevässä luvussa käsitellyt kehittyneemmät toiminnot, muun muassa funktiot ja herättimet, sillä tietokannan toimintojen testaamista varten pitää olla käytössä riittävästi testitietoja joita hakea ja käsitellä. Monimutkaisemmat toiminnot vaativat myös ohjelmointia, sillä kehittyneiden ominaisuuksien tapauksessa kyse on käytännössä ohjelmoinnista tietokannassa. Tietokanta ei kuitenkaan ole ympäristö jota on suunniteltu nopeata kehitystä varten, joten testaamisessa on oma vaivalloisuutensa.

Vaikka kirjan mukana tuleva oheis-CD-ROM ei kuulu suoranaisesti kirjan käsikirjoitukseen, on se silti käytännössä osa kirjan ja tutkintotyön kokonaisuutta. CD-ROM sisältää asennettavat ohjelmat, esimerkkisovellusten tiedostot sekä virtuaalikoneen, joka on vaihtoehtoinen järjestely suoraan lukijan koneelle asennettavien palvelinohjelmien sijaan.

Virtuaalikone on VMware Playerin avulla käytettävä virtuaalinen tietokone. Player-ohjelma toimii alustana varsinaiselle virtuaalikoneelle, jonka tiedostot löytyvät kirjan ohessa toimitettavalta CD-ROM:lta. Karkeasti ottaen Player vastaa virtuaalikoneen laitteistoa ja siinä käytettävät tiedostot koneen kiintolevyn sisältöä eli virtuaalikoneessa ajettavaa ohjelmistoa. Player-ohjelma oli alun perin tarkoitus toimittaa myös CD:llä, mutta koska VMware Inc. ei jostain syystä vastannut kyselyihin levitysluvasta, kirjaan jouduttiin lisäämään vain ohjeet ohjelman lataamiseksi. CD:llä toimitetaan siis vain itse

tehty virtuaalikone yhtenä pakattuna tiedostona eikä sitä siis valitettavasti ole mahdollista käyttää ilman että lukija lataa Player-ohjelman Internetistä.

Virtuaalikoneen käyttöjärjestelmänä on CentOS Linux, joka antaa suhteellisen hyvän kuvan siitä millainen oikea palvelinympäristö olisi. Linux-jakelun valintaan vaikuttaa myös kaksi muuta tekijää: se on molemmille kirjan tekijöille tuttu ja ilmainen. Tuttuus on tärkeä asia, sillä tuntemattomasta asiasta on vaikea opettaa mitään.

Koodin testaus ja kehitys esimerkkejä varten on yllättävän haastavaa. Ei riitä että ohjelma tai koodiesimerkki vain toimisi, vaan sen on käytännössä toimittava monessa eri ympäristössä – vähintään Windows-ympäristössä ja kirjan mukana tulevassa virtuaalikoneessa. Näissä molemmissa on hiukan eri versiot ohjelmistoista ja on myös mahdollista että lukija hankkii lukuhetkellä uusimmat versiot ohjelmista. Toisaalta yhteensopivuus varsinkin käsiteltyjen ohjelmistojen tulevia versioita ajatellen on jotain minkä mahdollistaminen tekee kirjasta ylipäätään paremman tuotteen sillä näin sen sisältö on pätevää ainakin jonkin aikaa myös tulevaisuudessa.

Kirjassa on jonkin verran kuvia, joskin vähemmän kuin aiemmassa painoksessa. Kaikki kuvat ovat ruutukaappauksia, ja suurin osa niistä on käytettävien ohjelmien asennuksia käsittelevässä luvussa. Tuon luvun kuvat esittävät lähes poikkeuksetta asennusohjelmien eri vaiheita. Muualla olevat ruutukaappaukset sisältävät useimmiten selaimen jossa on auki jokin kirjan esimerkkisovelluksen sivu. Selaimena ruutukaappauksia varten on käytetty Mozilla Firefoxia. Selain ei ole suosituin (tällä hetkellä Microsoftin Internet Explorer -selaimen kuudes ja seitsemäs versio ovat selkeästi suosituimmat selaimet). Sitä käytetään kuitenkin siksi että sen ulkoasu on selkeä eikä ole todennäköisesti muuttumassa radikaalisti – Internet Explorerin kanssa pitäisi valita että käytetäänkö harvinaistuvampaa kuutosversiota vai jonkin verran sekavamman näköistä seitsemättä versiota. Jonkin verran vaikutusta on ollut myös Firefoxin maineella turvallisempana ja paremmin standardien kanssa yhteensopivana selaimena.

4. Tarkempaa tarkastelua

Vaikka uskon että osaan ja ymmärrän PHP:ta ja MySQL:ää ainakin riittävällä tasolla, on Ajaxin kanssa toisin. Käytännössä minulla ei ennen tähän työhön ryhtymistä ollut mitään käytännön kokemusta Ajaxista muuten kuin Ajax-sovellusta käyttäneen näkökulmasta, mikä todennäköisesti ei ole ollut avuksi. Ajaxia käytetään kirjassa varsin pintapuolisesti ja tavalla joka ei välttämättä anna realistista kuvaa siitä miten sitä pitäisi käyttää: jonkin Ajax-kirjaston avulla. Kirjassa käytetään itse XMLHttpRequest-rajapintaa jolla tehdään HTTP-pyyntöjä palvelimelle ja tämän lisäksi manipuloidaan dokumenttia – verkkosivua – pyynnön palauttaman datan avulla.

Todellisessa tilanteessa Ajaxia käyttävä sovellus ei tekisi kovinkaan paljoa matalan tason toiminnoista itse, vaan todennäköisemmin käyttäisi jotain olemassa olevaa Ajax-kirjastoa tai ainakin jotakin kevyttä itse tehtyä kirjastoa joka kuitenkin on todennäköisesti selvästi laajempi kuin varsin pieni systeemi jota tässä kirjassa on esitelty. Ajax-kirjaston käyttämisessä on omia puolia joita ei kirjassa edes sivuttu – esimerkiksi Prototype JavaScript Framework tai lyhyemmin Prototype on kirjasto, joka antaa käyttäjälleen huomattavasti valtaa mutta myös "rikkoo" monia JavaScriptin perustoimintoja jolloin kirjaston käyttäjän on toimittava Prototypen tekijöiden valitsemalla tavalla.

Vaikka odotukset lukijan tietojen tasosta onkin tiivistettävissä melko hyvin aiemmin mainittuun "osaa tehdä staattisia HTML-sivuja" -ilmaukseen, kyseessä ei kuitenkaan ole mikään kirjoitusvaiheessa erityisesti ajateltu asia. Tässä suhteessa edellisen painoksen vaikutus näkyy, eli perusasioissa on lähdetty suurin piirtein samalta tasolta. Nykyisessä painoksessa kuitenkin on se ero, että erityisesti tietokannan käytön suhteen mennään huomattavasti pidemmälle, sillä kirjassa käsitellään melkoinen osa MySQL:n viidennen version uusista ominaisuuksista. On varsin mahdollista että kehittyneemmät asiat menevät lukijalta yksinkertaisesti ohi, toisaalta siitä syystä että ne ovat melko monimutkaisia varsinkin aloittelijan näkökulmasta ja toisaalta minulla ei ole ollut kirjoittajana kokemusta tämän tyyppisten asioiden opettamisesta.

Kirja on kirjoitettu käytännössä kokonaan OpenOffice.org Writerillä, ja omat osuuteni tein käytännössä kokonaan Linuxissa, esimerkkitsovellusten pyöriessä virtuaalikoneessa olevassa Windowsissa. Koska tekstiin ei ole ollut tarkoitus tehdä minkäänlaista taittoa,

vaan se tehtiin erillisellä ohjelmalla, järjestely toimi varsin hyvin. Jos OpenOffice.org Writerillä olisi yritetty tehdä jotain monimutkaisempia tekstin ulkoasuun liittyviä asioita, Word-tiedostojen tuottaminen olisi tuskin onnistunut yhtä hyvin. Kirjoitustyön loppuvaiheissa oli kuitenkin siis pakko vaihtaa käyttämään Wordia, sillä taitto-ohjelmat eivät juuri ymmärrä OpenOffice.org Writerin omia tiedostomuotoja. Kun tiedostoja läheteltiin useita kertoja edestakaisin loppuvaiheiden isompia ja pienempiä korjauksia varten, käytettiin muutosten jäljitys -toimintoa jota ilman sinne tänne ripoteltujen pikkumuutosten seuraaminen olisi ollut todennäköisesti hyvin hankalaa.

Myös OpenOffice.org Writer osaa ainakin jollain tasolla käyttää Word-tiedostojen muutosten jäljitystä, mutta asian kanssa ei otettu mitään riskejä. Tietyissä mielessä oli siis turhaa käyttää toista ohjelmaa sillä Wordin käyttö oli joka tapauksessa välttämätöntä.

Kirjan oppimiskäyrä saattaa olla kokonaisuutta katsottaessa erikoisen muotoinen, sillä MySQL-luvun loppupuoli on jo varsin monimutkaista asiaa, kun taas sitä seuraava ensimmäinen esimerkkisovellus (päiväkirja) on taas selvästi yksinkertaisempi. Tätä on kompensoitu sillä, että MySQL-luvussa huomautetaan että monimutkaisempia asioita ei ole pakko lukea jotta kirjan lukemista voisi jatkaa. Vaikeusaste muodostaa siis MySQL-osuuden lopussa "huipun" jolta laskeudutaan taas, kun mielekkäämmältä tuntuisi tasaisesti eikä välillä huippuja muodostava oppimiskäyrä. Ensimmäisen ja toisen sovelluksen välinen ero monimutkaisuudessa voi olla myös turhan iso, mutta oma arviointikyky ei auta tässä. En kuitenkaan ole tähän mennessä saanut vielä kuulla että tästä olisi tullut merkittävää ongelmaa.

Ohjelmistot joita kirjan kanssa on käytetty kehittyvät nopeaan tahtiin. Toisaalta, melko aikaisessa vaiheessa tapahtui versioiden "jäädymä" kirjan näkökulmasta, eli PHP:ta ja MySQL:ää ei päivitetty kesken kirjan kirjoittamisen. Tämä on järkevää siinä mielessä, että alustan muuttumisesta johtuvia ongelmia ei tarvitse selvittää koko aikaa. Huonona puolena on se, että tällaisessa tilanteessa esimerkkeihin saattaa tulla kohtia jotka riippuvat ominaisuudesta joka on jätetty seuraavassa versiossa pois tai joka ei peräti toimi kuin nimenomaan kehitykseen käytettävällä versiolla. Jälkimmäinen tilanne voi vaikuttaa epätodennäköiseltä, mutta niin kuitenkin kävi.

PHP:n PDO-lisäosan avulla voidaan tehdä tietokantahakuja turvallisesti valmisteltujen lausekkeiden avulla. Valmistellun lausekkeen idea on se, että lausekkeeseen kirjoitetaan arvojen sijasta merkinnät, joiden kohdalle varsinaiset arvot sijoitetaan lausekkeen luonnin jälkeen. Paikat merkitään lausekkeeseen omalla syntaksillaan. Arvot sijoitetaan lausekkeeseen sitomalla ne oman kutsunsa avulla, antamalla nimi johon arvo sidotaan ja muuttuja jossa arvo sijaitsee (esimerkiksi "SELECT * FROM taulu WHERE id = :id" jossa :id on paikka johon muuttujan arvo sijoitetaan).

Alun perin käytin lausekkeessa samaa nimeä useaan kertaan, siten että samaa muuttujaa ei tarvitsisi sitoa lausekkeeseen useaan kertaan, erikseen joka kohtaan jossa kyseistä muuttujaa käytetään (esimerkiksi "SELECT * FROM taulu WHERE sarake1 < :muuttuja AND sarake2 > :muuttuja"). Tämä toimi kirjoitusaikana kun käytin vain yhtä versiota PHP:sta. Viimeistelyvaiheessa kun oli tarkoitus varmistaa esimerkkien toimivuus virtuaalikoneessa, selvisi että saman nimen uudelleenkäyttö yhden lausekkeen sisällä oli tuettuna vain siinä versiossa PHP:ta jota olin käyttänyt. Aiempiin versioihin sitä ei ollut vielä lisätty ja seuraavasta versiosta se oli jätetty pois tietoturvasyistä.

Esimerkkisovelluksista päiväkirjasovellus on käytännössä raakile, eikä sitä voisi käyttää mitenkään mielekkäästi todellisuudessa muuten kuin ehkä täysin henkilökohtaisena muisti- tai päiväkirjana. Sovelluksen ei ole tietenkään tarkoitus kuin esitellä kuinka joitain perusasioita hoidetaan PHP-MySQL-ympäristössä, mutta sen voisi todennäköisesti ilmaista selkeämminkin ja mahdollisesti useaan kertaan jottei kokemattomalle lukijalle tulisi väärää kuvaa asioista. Tärkein puute sovelluksessa on käyttäjän tunnistuksen puute, eli kuka tahansa joka vain pääsee sovelluksen sivuille, voi tehdä mitä tahansa mitä kuka tahansa muukin, mukaan lukien sovelluksen "omistaja". Minkäänlaisesta julkaisujärjestelmästä tai blogiohjelmistosta ei siis ole kysymys.

Ensimmäisen sovelluksen puutteita paikataan toisessa sovelluksessa, jossa on jopa yksinkertainen käyttäjähallinta. Nykyisellään käyttäjähallinnasta vastaa yksi tiedosto. Käyttäjähallinnan yksinkertaisuus on valitettavaa, sillä se voisi olla myös selkeämmin toteutettu. Kirjan tekstissä kerrotaan, että varsinaiset hallintasivut ja omien tietojen hallintaan tarkoitettut sivut olisi hyvä erotella erillisiksi, mutta tämä ei välttämättä ole oikea lähtökohta – esimerkkisovelluksen tulisi olla esimerkillinen. Selitys tilanteelle on

se, että kirja oli pakko saada valmiiksi kohtuullisen ajan sisään ja toisaalta kalenterisovelluksen sivumäärä kokonaisuudessa on jo nykyisellään melko iso.

Esimerkkisovellukset eivät käytä kovin monimutkaisia tietokantarakenteita. Tämä selittyy sillä että sovellukset on pidettävä käytännössä mahdollisimman yksinkertaisina. Päiväkirjasovelluksessa lukija näkee kuinka tietokantaa ylipäätään käytetään PHP-sovelluksessa oikeasti, joten sen tapauksessa useita osia sisältävän tietokantarakenteen esittely ei vaikuta toimivalta ratkaisulta. Toisessa sovelluksessa keskitytään pitkälti selainpään tapahtumiin, joten siinäkään ei ole ollut perustetta kovin monimutkaiselle tietokantarakenteelle. On kuitenkin jossain määrin kyseenalaista että riittävätkö tietokantaa käsittelevän jakson monimutkaisemmat esimerkit lukijalle, jolla ei välttämättä ole vielä kovin paljoa kykyä soveltaa saatua tietoa muuten kuin esimerkikoodia lähtökohtana käyttämällä.

5. Vertailua toisiin kirjoihin

Vertailu toisiin ohjelmointikirjoihin on yksi keino etsiä näkökulmia omaan kirjoitustyöhöni. Vertailukohtina on käytetty kirjoja Java, Python ja Prolog (Ilkka Kokkarinen, 2004) ja CGI-ohjelmointi (Niklas Frykholm, 1998) sekä Java-ohjelmoinnin peruskirja (Juha Peltomäki, Pekka Malmirae, 1999). Pyrin hakemaan sekä erilaisia että samankaltaisia kirjoja niin aiheiden kuin asioiden käsittelynkin puolesta. Ensimmäinen kirjoista käsittelee useita kieliä, mutta toisaalta kielet eivät ole samoja kuin mitä PHP ja MySQL käsittelee. CGI-ohjelmointi esittelee Perl-kielen käyttöä web-ympäristössä, joten sen voisi olettaa olevan varsin samankaltainen kirja. Java-ohjelmoinnin peruskirja on puolestaan toinen yksittäiseen kieleen keskittyvä kirja, jonka lähtötaso on myös varsin matalalla.

5.1 Java, Prolog ja Python

Java, Prolog ja Python on Ilkka Kokkarisen kirjoittama kirja vuodelta 2004. Takakannen kuvauksen mukaan "kirja on tarkoitettu oppaaksi niin ohjelmoinnin aloittelijalle kuin koodauksen ammattilaiselle". Kuvaus sisältää myös maininnan että kirja sopii oppikirjaksi ammattikorkeakoulu- ja yliopistotasolla. Tämä tekee kirjasta mielekkään vertailukohdan, sillä myös PHP ja MySQL on jossain määrin oppikirjamainen.

Ilkka Kokkarisen kirja koostuu kolmesta osasta, joissa kussakin käsitellään yhtä kieltä, samassa järjestyksessä kuin kirjan nimessäkin mainitaan. Ensimmäisessä Java-kieltä käsittelevässä osiossa on melkein puolet kirjan sivuista. Loppuosa ei jakaudu pelkästään kahden muun kielen kesken, vaan lopussa on myös merkittävän pitkä osuus joissa tarjotaan vastauksia kirjan harjoitustehtäviin. Python-osiossa käsitellään myös Ruby-kieltä, joten kirjassa käsitellään itse asiassa neljää eri kieltä.

Java, Prolog ja Python -kirjan osiot eivät liity toisiinsa erityisen kiinteästi. Myöhemmät osat käyttävät aiemmissa osissa käsitellyjä pohjana, mutta osiot eivät esimerkiksi yritä ratkaista jotain yhteistä ongelmaa. PHP ja MySQL -kirjassa ajatuksena on nimenomaan käyttää tekniikoita yhdessä, mutta silti PHP ja MySQL käsitellään ensin yksinään ilman erityistä kosketuspintaa käytännön sovellutuksiin. Kokkarisen kirjan lähtökohtana näyttäisi olevankin enemmän ohjelmoinnin periaatteiden selvittäminen kuin jonkin toimivan sovelluksen aikaan saaminen. PHP ja MySQL ei käsittele läheskään yhtä paljon periaatetason asioita, ja yleisemmän tason neuvotkin liittyvät enemmän esimerkiksi tietoturvaan kuin hyvään ohjelmointitapaan.

Kirjan jakaminen erillisiin osioihin sinänsä perusteltua, sillä käsitellyt kielet ovat lähtökohdiltaan kaukana toisistaan. Johdannossa eri kielten käsittelyä samojen kansien välissä perustellaan sillä, että näin lukijalle ei välittyisi yksipuolista kuvaa ohjelmointikielien luonteesta. Esimerkiksi logiikkaohjelmointikieli Prolog on varsin erilainen verrattuna Javaan ja PHP ja MySQL -kirjassa käsiteltyihin kieliin. Jo käytännön syyt sanelevat että kirja, jossa käsitellään web-ohjelmointia, ei voi esitellä mielekkäästi vaikkapa Prologin tapaisia kieliä ilman että hypätään kohtuuttomasti sivupoluille varsinaisesta aiheesta. Monet niin sanotusti erikoisemmat kielet eivät ole sinänsä käyttökelvottomia, mutta niillä ei yleensä pystytä ratkaisemaan yhtä hyvin web-ympäristöissä esiintyviä ongelmia. Tietyissä mielessä PHP ja MySQL ja Java, Prolog ja Python -kirjojen lähtökohdat ovat ristissä toisiinsa nähden: ensin mainittu keskittyy yhteen kapeaan alueeseen käytännön näkökulmasta katsoen kun taas jälkimmäinen pyrkii antamaan laajemman ja periaatetason asioita korostavan näkökulman.

Kokkarisen kirjan Java-osion ohjelmistovalinta tukee ajatusta periaatteiden opettamisesta. Käytetty BlueJ-ympäristö mahdollistaa Java-luokkien tekemisen ilman että pitäisi tehdä niitä yhteen sitovaa sovelluskoodia. Näin on mahdollista aloittaa heti olio-ohjelmointi sen

sijaan että tehtäisiin perinteinen proseduraalista ohjelmointitapaa esittelevä Hei maailma! -ohjelma ja edettäisi siitä vasta myöhemmin luokkien käyttöön. Myös toisten kielten tapauksessa on esillä samantapainen ajattelu. Python-kieli on myös varsin sopiva opettelukieleksi, sillä siihen kuuluu omana osanaan komentorivitulkki jonka avulla uutta suoritettavaa koodia voi kirjoittaa suoraan, ilman että sitä kierrätetään edes tiedoston kautta Python-tulkille.

Kirjassa käytetyt esimerkkitaupukset ovat myös melko pitkälti yksinkertaistettuja tosimaailman asioita. Yhtenä olio-ohjelmoinnin periaatteita kuvaavana esimerkkinä on käytetty postilaitosta: postin palveluiden käyttäjä ei tiedä miten posti kulkee perille asti – hiukan samaan tapaan olioita käyttävän ohjelmoijan ei tarvitse tietää mitä jonkin olion sisällä tapahtuu. Kirjan Prolog-osion esimerkit eivät ole yhtä arkisia, mutta tämä johtuu pitkälti kyseisen ohjelmointikielen luonteesta. Java-osion esimerkit näyttäisivät olevan kuitenkin samoilla linjoilla kuin PHP ja MySQL -esimerkit.

Java, Prolog ja Python -kirjassa on myös harjoitustehtäviä. Tehtävät ovat luonteeltaan pitkälti ajattelutehtäviä, joiden tarkoituksena on saada lukija perustelemaan asioita itselleen. Osa tehtävistä on myös ohjelmointitehtäviä, ja kaikki tehtävät eivät keskity esimerkiksi algoritmeihin tai suunnittelumalleihin vaan vaikkapa ohjausrakenteiden käyttämiseen. Kirjan lopussa on myös vastauksia tehtäviin, tosin ei aivan jokaiseen. Vastauksettomia tehtäviä voitaneen pitää jonkinlaisina retorisia kysymyksiä ja ajatuksien herättäjinä kuin varsinaisina harjoitustehtävinä. Mitään varsinaisia harjoitustehtäviä ei löydy PHP ja MySQL -kirjasta, vaikka koodiesimerkit ja esimerkkisovellukset ovat enemmän tai vähemmän epäsuorasti tarkoitettu lukijan kokeiltaviksi.

Kirjan loppupuolella olevassa Python-osiossa käsitellään Python-ohjelmointikielen lisäksi myös toista kieltä, Rubya. Python ja Ruby eivät ole varsinaisesti sukulaiskieliä, mutta Python on antanut vaikutteita tuoreempaan Rubyyn. Python ja Ruby ovat dynaamisia ja oliopainotteisia kieliä. Molemmissa on myös taustalla ajatus tehdä ohjelmoinnista ohjelmoijaystävällisempää.

Kolmas osio sisältää luonnollisesti siinä käsiteltyjen kielten perusteet. Itse kielten perusasioiden lisäksi käydään myös lyhyesti läpi myös sellaisia asioita joita ei aiemmin

vielä käsitelty, esimerkiksi tiedostojen lukeminen ja kirjoittaminen ja säännölliset lausekkeet. Tällaisten asioiden läpikäyminen sekä kahden kielen käsittely osion sisällä antaa osiosta sellaisen vaikutelman, että siihen on koottu tarkoituksella sekalaisia jäljellä olevia asioita itse kielten perusteiden lisäksi. Tämä on sikäli hyvä ratkaisu, että näin menetellessä itse kieliä käsittelevät osuudet pysyvät siistimpinä.

PHP ja MySQL -kirjassa kieliä esittelevät osuudet ovat myös melko keskittyneitä itse kielen käyttöön, joskin toki käytännöllisemmästä näkökulmasta kuin Kokkarisen kirjassa. Esimerkkisovellusten tapauksessa ei kuitenkaan ole päästy tällaiseen "puhtauteen", vaan tiettyä sovellusta kuvaavan tekstin joukossa on myös yleispätevämmäksi tarkoitettuja kohtia. Näitä olisi mahdollisesti kannattanut esimerkiksi pakata kutakin sovellusta käsittelevän tekstin alkuun tai ainakin erottaa muusta tekstistä selkeämmin.

5.2 CGI-ohjelmointi

CGI-ohjelmointi on Niklas Frykholmin web-ohjelmointia käsittelevä kirja vuodelta 1998. Tämä kirja on iästään huolimatta monessa suhteessa lähellä PHP ja MySQL -kirjaa: molemmat käsittelevät web-ohjelmointia, kirjojen kielet PHP ja Perl ovat sukulaiskieliä keskenään. Kuriositeettitason yhtäläisyytenä myös Frykholm on ollut opiskelija silloin kun kirja on julkaistu.

Yksi olennainen ero kuitenkin näyttäisi olevan se, että tämä kirja on tähdätty huomattavasti kokeneemmille käyttäjille. Ainakin takakannen perusteella kohderyhmänä ovat kokeneet käyttäjät ja ammatillaiset siinä missä PHP ja MySQL on tähdätty enemmänkin aloittelijoille. Myös CGI-ohjelmointi käsittelee ensin aivan perustason asioita, eli ainakaan kirjassa käsiteltyjen asioiden tunteminen ennalta ei ole vaadittua. Perusasiat käydään läpi kuitenkin nopeammin ja käytännössä lukijalla pitäisi olla etukäteistuntemusta ohjelmoinnista.

Kirjassa on neljä osaa: Johdanto on lyhyt kirjan aloittava osio jossa varsinaisen johdantotekstin lisäksi käsitellään myös lyhyesti ohjelmistojen asennus. PHP ja MySQL -kirjassa tämä käsitellään huomattavasti pidemmän kaavan mukaan, ja lukijalle esitellään ohjelmistojen asennus ruutukaappauksien avulla. Periaatteessa lyhyempikin asennusten käsittely olisi ollut aivan mahdollinen. Toisaalta, kirjan alkuvaiheessa lukijalla on

todennäköisesti vielä epäselvä kuva asioista ja sovellusten asennuspaketit tarjoavat varsin paljon mahdollisuuksia vaikuttaa asennuksen konfiguraatioon.

Toinen CGI-ohjelmointi-kirjan osio käsittelee Perl-kieltä yleisesti. Ensimmäisenä käsitellään muuttujat, sitten ohjausrakenteet ja matemaattiset toiminnot ja rakenteet. Näiden jälkeen tulee merkkijonot (sisältäen säännöllisten lausekkeiden perusteet), taulukot ja lopulta myös aliohjelmat. Tämä rakenne muistuttaa pitkälti sitä mitä PHP ja MySQL käyttää: edetään yksinkertaisemmista rakenteista monimutkaisempiin. CGI-ohjelmointi-kirja käy kuitenkin asiat läpi paljon kattavammin suhteellisen nopeaan tahtiin ja yhtenä ryhmittelyn perusteena näyttäisi olevan kirjan hakuteosmaisen käytön helpottaminen. Tähän ei ole pyritty erityisesti PHP ja MySQL -kirjan tapauksessa, mutta jonkinlaiset tiivistelmät kustakin käsitellystä pääasiasta voisivat olla avuksi lukijalle jonka ei tarvitsisi käyttää aikaa niinkään paljoa tiedon etsimiseen. SQL-kieltä käsittelevässä osiossa tätä on jossain määrin harjoitettukin.

Frykholmin kirjan varsinainen CGI-ohjelmointi-osio keskittyy enemmän web-ohjelmointiin Perlillä. Osiossa käydään läpi web-ohjelmointiin liittyviä asioita melko matalalta tasolta lähtien – ensimmäisten käsiteltävien asioiden joukossa ovat HTTP-statuskoodit. Tämä ei luonnollisestikaan olisi ollut sopiva lähestymistapa PHP ja MySQL -kirjan tapauksessa erilaisen kohdeyleisön takia.

CGI-ohjelmointi-osiossa on useita projekteiksi kutsuttuja esimerkkisovelluksia. Projekti käyttävät aiemmin käsiteltyjä yksittäisiä koodiesimerkkejä hyväkseen ja ovat lyhyitä. Käytännössä ne ovat siis yksittäisen koodiesimerkin ja kokonaisen itsenäisen esimerkkisovelluksen välimuotoja. Myöhemmät projektit ovat pidempiä, mutta kokonaisista sovelluksista ei niidenkään tapauksessa ole kyse. PHP ja MySQL -esimerkkisovellukset eivät myöskään ole täydellisiä sovelluksia, mutta ne ovat laajempia eivätkä käytä samaan tapaan aiempaa koodia sellaisenaan apunaan. Tällainen tarkempi rajaaminen on uskoakseni parempi valinta ainakin PHP ja MySQL -kirjan tapauksessa.

CGI-ohjelmointi-osiossa käsitellään myös tietoturva-asioita, eli käytännössä käyttäjän antaman syötteen tarkistamista. PHP ja MySQL antaa myös ohjeita tietoturvan suhteen, mutta nämä ohjeet on levitetty laveammin kirjan alueelle; kirjan lopussa on näiden lisäksi muutaman sivun mittainen erityisesti tietoturvalle omistettu osuus. Tietokantoja ja niiden

käyttöä esitellään CGI-ohjelmointi-kirjassa lyhyesti, mutta SQL-kieltä käyttävät järjestelmät on jätetty pelkäksi maininnaksi – kirjassa käsitellään lähinnä tekstitiedostopohjaisia tietokantoja. Näin myöskään erityisesti SQL:n avulla käytettäviin tietokantoihin liittyviä tietoturvaseikkoja ei käsitellä.

Kirjan neljäs osa on otsikoltaan Kehittynyt CGI-ohjelmointi. Osion alussa käsitellään jonkin verran sovelluksen kehitystä laajemmasta näkökulmasta keskittymättä itse ohjelmointiin. Sovellussuunnittelua erityisesti käsittelevä osuus ei kuitenkaan ole erityisen pitkä. PHP ja MySQL ei käsittele sovellussuunnittelua käytännössä ollenkaan, ja joitain tämän tason asioita todennäköisesti olisi voinut ollut syytä käsitellä. Muutoin Kehittynyt CGI-ohjelmointi -osio keskittyy lähinnä kirjastojen käyttöön ja toisten järjestelmien (esimerkiksi sähköpostipalvelinten) kanssa kommunikointiin.

CGI-ohjelmointi on kirja, joka käsittelee lähes samaa asiaa kuin PHP ja MySQL. Silti erot kohderyhmässä erottavat kirjoja olennaisilla tavoilla: ensin mainittu on monessa suhteessa hakuteosmainen vaikka monet myöhemmät kohdat viittaavatkin aiemmin käsiteltyihin asioihin kun taas jälkimmäinen kirjoista ohjailee lukijaa huomattavasti enemmän. Uskoakseni myös kirjojen ikäerolla on vaikutusta, ja web-sovellusten kehitykseenkin on muodostunut omia mallejaan – näitä malleja olisi voitu käsitellä enemmän myös PHP ja MySQL -kirjassa vaikkei se suoraan käsittelekään suurien sovellusten tekemistä.

5.3 Java-ohjelmoinnin peruskirja

Java-ohjelmoinnin peruskirja on Juha Peltomäen ja Pekka Malmirakeen kirjoittama Java-kieltä käsittelevä kirja vuodelta 1999. Kirja keskittyy yksinomaan Javaan. Lukijalta ei odoteta käytännössä mitään erityisiä esitietoja ohjelmoinnista. Kirjassa on kuusi pääosiota, nimiltään Java – ohjelmointikieli, Java-kielen perusteet, Olio-ohjelmointia Javalla, Graafinen ohjelmointi Javalla, Java-sovelluskehitys ja Java-ohjelmoijan työkalupakki.

Ensimmäinen osio toimii käytännössä johdantona jossa käsitellään Javan historiaa ja kykyjä, mm. alustariippumattomuutta, turvallisuutta ja muita vastaavia ominaisuuksia. Johdannoksi alkuosa on melko pitkä, PHP ja MySQL -kirjassa ohjelmistojen ja ohjelmointikielten vastaavat esittelyosiot on sijoitettu hajautetusti ennen kunkin ohjelmiston asennusohjeita asennuksia käsittelevässä luvussa.

Java-kielen perusteissa aloitetaan Java-lähdekoodin kääntämisellä, mille ei luonnollisestikaan löydy vertailukohdetta PHP ja MySQL -kirjasta – tosin jos PHP olisi käännettävä eikä tulkittava kieli, koodin kääntäminen todennäköisesti käsiteltäisiin yhtenä ensimmäisistä asioista. Debuggerin käyttöä tuskin kuitenkaan käsiteltäisiin, tosin kuin Java-kirjassa on tehty. Kielen perusteet käsitellään tässä kirjassa hiukan sekä CGI-ohjelmointi- että PHP ja MySQL -kirjojen tapaan, eli yksinkertaisista kieliopillisista asioista kohti monimutkaisempia. Alussa ei keskitytä vielä olio-ohjelmoinnin ajattelutapaan, vaan tätä käsitellään seuraavassa osiossa.

Kirjan kolmannessa osiossa keskitytään aiemmin selvitettyjen syntaktisten asioiden jälkeen oliopohjaiseen ajatteluun. Kuten totesin jo Java, Prolog ja Python -kirjan kohdalla, tällaista kohtaa tai sen proseduraaliseen ajatteluun opastavaa vastinetta ei löydy PHP ja MySQL -kirjasta. Se, kannattaisiko PHP:n tapauksessa keskittyä proseduraaliseen tyyliin vai olio-ohjelmointiin, on kysymys johon en osaa vastata. Proseduraalinen tyyli sopii monessa mielessä PHP:hen paremmin – vaikka PHP:n oliotuki on parantunut paljon, tuki ei silti ole missään nimessä "aitojen oliokieli" tasolla – mutta pitkällä tähtäimellä olioista voi kuitenkin olla enemmän hyötyä.

Siinä missä Java-kirjan kolmas osio keskittyy olio-ohjelmoinnin perusasioihin, neljännessä osiossa on kuitenkin jo graafisen ohjelmoinnin vuoro. Tässä osiossa käydään läpi grafiikan ja äänien käyttäminen Java-sovelmissa. Jos kyseessä olisi web-ohjelmointia käsittelevä kirja, tämä osio käsittelisi todennäköisesti kielen mahdollistamia web-aiheisia toimintoja siinä missä aiemmin käsiteltiin kieltä yleisemmällä tasolla. Java-kirjan osio ei kuitenkaan keskity erityisesti minkään kokonaisen sovelluksen tekemiseen, joten sitä ei voi verrata PHP ja MySQL -kirjan esimerkkisovelluksiin.

Esimerkkisovelluksia Java-kirja kuitenkin tarjoaa viidennessä osiossaan, jossa toteutetaan useita pieniä sovelluksia. Ensimmäinen esimerkkisovellus, matopeli, ei ole mitenkään erityisen oliopohjainen, ja sellaisena sen voi rinnastaa PHP ja MySQL -kirjan päiväkirja-esimerkkisovelluksen kanssa – molempien ajatuksena on näyttää sellaisten perusasioiden toiminta, joita aiemmin ei ole käsitelty mutta joiden tuntemus on silti välttämätöntä. Seuraavassa Java-kirjan sovelluksessa käytetään jo enemmän oliotekniikoita, mutta koska

kaikki esimerkkisovellukset ovat melko pieniä, monimutkaisempiin oliorakennelmiin ei ehditä tutustua.

Kirjan viimeisessä jaksossa käsitellään Java-kielen ohjelmointityökaluja. PHP ja MySQL -kirjassa nämä asiat käsitellään kirjan alussa asennusten yhteydessä, pois lukien MySQL-tietokantapalvelimen hallintaan tarkoitettu phpMyAdmin, joka esitellään MySQL-luvun alussa. PHP ja MySQL -kirjan loppuun tarkoitettu keittokirjaliite ei kuitenkaan ole kovin kaukaa haettu esimerkki, sillä Java-kirjan loppu käsittelee asioita myös enemmän tai vähemmän sekalaisena kokoelmana asioita.

6. Johtopäätökset

Se, päästiinkö tavoitteisiin, riippuu siitä millä tarkkuudella asiaa katsotaan. Tavoitteeseen päästiin jos katsotaan vain sitä, että valmistuiko kirja. Aikataulu venyi alkuperäisestä – alun perin oli tavoitteena saada kirjoitustyöt valmiiksi kesäkuun alkupuolella, mutta lopulta lopullisessa muodossa oleva materiaali saatiin lähetettyä vasta heinäkuun ja elokuun taitteessa. Tätä ei voi kuitenkaan pitää varsinaisena epäonnistumisena ainakaan kustannustoimittajalta saadun palautteen perusteella: siinä ei ollut mitään viittausta että kirja olisi myöhästynyt vaan itse asiassa päinvastoin.

Tarkemmin katsottuna kirjassa on jouduttu tekemään paljon kompromisseja kun verrataan lopputulosta alkuperäiseen suunnitelmaan, johon kuului muun muassa "keittokirjaliite" jossa tarjotaan pieniä koodinpätkiä, esimerkiksi henkilötunnuksen tarkistamiseksi tai laskun viitenumeron tarkistusnumeron laskemiseksi. Aiemmin mainitsinkin jo monimutkaisempien tietokantarakenteiden käyttämisen esimerkkisovelluksissa. Tällaiset poisjättämiset eivät välttämättä näy lukijalle, mutta tekijän näkökulmasta katsottuna lopputulos ei ole aivan sitä mitä sen piti olla.

Kirjaa suunniteltaessa pyrittiin karkeasti ottaen ottamaan saamaan aikaan tietty sivumäärä. Vähimmäisvaatimuksena oli että kirjassa olisi ainakin yhtä paljon sivuja (ja siten ainakin yhtä paljon sisältöä) kuin edellisessä painoksessa. Tavoitteena oli kuitenkin ylittää sivumäärä, ja tässä onnistuttiin: edellisen painoksen viimeinen sivunumero on 227 ja uuden painoksen 412. Uudessa versiossa on siis vajaat puolet enemmän sivuja. Vaikka

suuri sivumäärä ei ole mitenkään itsestään selvästi positiivinen asia, on kuitenkin melko turvallista olettaa että myös sisällön määrä on kasvanut.

Vertailllessani omaa tekstiäni verrokkikirjoihin, näyttäisi siltä että kirjasta melko suuri osa keskittyy koodiin. Esimerkkisovelluksissa käydään pääasiassa läpi lähdekooditiedostoja sekä kohta kohdalta että kokonaisuutena. CGI-ohjelmointi-kirjassa on myös melko pitkälti vastaava ote, kun taas toisissa oppikirjamaisemmissa kirjoissa koodiesimerkit puolestaan palvelevat muuta tekstiä.

En ole ennen kirjoittanut tämän tyyppistä tekstiä, enkä myöskään näin suurta määrää tekstiä yhteen kokonaisuuteen. Käytännössä oman kirjoitustyön arviointi on siis vaikeata, vaikka minulla onkin jonkinlainen mielikuva siitä missä kohdissa ei ole onnistuttu yhtä hyvin kuin joissain paremmin onnistuneissa kohdissa. Uskoisin kuitenkin että kokemattomuuteni kirjoittajana näkyy tekstissä ainakin jossain määrin, todennäköisesti erityisesti niissä osissa joissa en itsekään tunne asiaa erityisen hyvin.

En ole vielä tätä kirjoittaessani saanut minkäänlaista palautetta kirjan lukeneilta ihmisiltä vaikka kirja on ollut kaupoissa jo jonkin aikaa, mutta odotan mielenkiinnolla saavani joskus kuulla mitä mieltä lukijat ovat tekstistä. Vaikka osat, joista kirjoitusprojekti koostuu, ovatkin entuudestaan enemmän tai vähemmän tuttuja, kokonaisuus on silti ollut poikkeuksellinen kokemus minulle – ehkä kyse on mittakaavasta tai tekstin aihepiiristä.

Erityisesti vertailu toisiin kirjoihin on antanut ajatuksia siitä, mihin suuntaan esimerkiksi seuraavaa painosta PHP ja MySQL -kirjasta voisi viedä tai siitä, mitä asioita voisin jossain muussa mahdollisessa ohjelmointiaiheisessa tekstissä painottaa nykyistä enemmän.

Vaikka PHP ja MySQL ei ole tarkoitettu hakuteosmaiseksi, aloittelevalle ohjelmoijalle on luultavasti hyödyksi että olennaiset asiat löytyisivät kirjasta suhteellisen helposti. Tätä varten kirjassa voisi olla vaikkapa jonkinlaisia tiivistelmäosioita joissa käydään läpi lyhyesti käsiteltyjä asioita.

Keittokirjaliite on mainittu tässä tekstissä useammankin kerran. Osioita, jossa käydään läpi niitä asioita jotka eivät sopineet mihinkään suurempaan kokonaisuuteen, ei välttämättä

kannata ajatella minkäänlaisena jämäpalojen kokoelmana, sillä sekalaisenakin tiedonpalasten kokoelmana se antaa kuvan kielten ja niiden kirjastojen kyvyistä. Kaikkea mahdollista ei luonnollisesti ole järkevääkään puristaa esimerkkisovelluksiin ja ohjelmointikielten perusasioita käsitteleviin osuuksiin.

Sivupohjamoottoreita käsitellään kirjassa vain hyvin lyhyesti. Niiden pidempi käsittely voidaan melkeinpä laskea hyvien ohjelmointitapojen selittämiseksi. Web-ohjelmoinnissa sovelluslogiikan ja ulkoasun erottelu on asia joka ansaitsee enemmän painoa, varsinkin silloin kun kyseessä on PHP:n kaltainen kieli jossa on erittäin helppoa johtaa itsensä umpikujaan tekemällä koodia jossa HTML-koodin tuottaminen ja sovelluksen logiikka nivoutuvat toisiinsa liian tiukasti. Sivupohjamoottorin käytön yhteydessä voisi olla hyvä mahdollisuus käsitellä myös oliopohjaisen web-sovelluksen tekemistä.

Todennäköisesti tärkein asia jota tällaisessa kirjassa tulisi korostaa enemmän, on hyvä ohjelmointitapa. Vaikka nykyisellään huonoihin tapoihin kannustaminen on mielestäni melko vähäistä, varsinaista oikeiden tapojen esittelyä ja selittämistä ei kuitenkaan ole välttämättä vielä tarpeeksi. Ainakin esimerkkeinä toimivien verrokkikirjojen perusteella olisi todennäköisesti järkevää ensin selittää asia periaatteen tasolla ja sitten esitellä kuinka asia tehdään käytännössä. Aivan alussa voi kuitenkin olla parempi esitellä vain syntaksia – jo pelkässä PHP:n kieliopissa lienee paljon sulateltavaa ohjelmointia tuntemattomalle lukijalle. Ensimmäinen periaatteita enemmän esittelevä tekstijakso voisi tulla esimerkiksi ennen perusohjausrakenteiden jälkeen, ennen funktioiden käsittelyä, jotta lukijalla olisi parempi pohja ymmärtää proseduraalista ohjelmointia.

Periaatetason asioiden selittäminen kuitenkin nostaa esille ongelman: Pitäisikö käsitellä vain proseduraalista ohjelmointia vai käsitellä asioita lähes pelkästään olio-ohjelmoinnin näkökulmasta. Käytännössä katsoen jonkinlainen välimuoto lienee lähes järkevin vaihtoehto, mutta tällöinkin pitää käytännössä valita, kumpi saa suuremman painon. Itse olisin taipuvainen valitsemaan proseduraalisen tavan, vaikka PHP nykyisellään tukeekin olio-ominaisuuksia suhteellisen hyvin. Syynä tähän on se, että proseduraalinen ajattelutapa on todennäköisesti helpompi omaksua PHP:n kaltaisessa ympäristössä. En pitäisi kuitenkaan mahdottomana ajatuksena että kirjassa olisi myös vaikkapa yksi nimenomaan oliopohjaisesti toteutettu esimerkkisovellus.

7. Lähteet

Kokkarinen, Ilkka. 2004. *Java, Prolog ja Python*. 1. painos. Helsinki: IT Press.

Frykholm, Niklas. 1998. *CGI-ohjelmointi*. 1. painos. Espoo: Pagina.

Peltomäki, Juha. 1999. *Java-ohjelmoinnin peruskirja*. 1. painos. Jyväskylä: Teknolit.