



MONITOROINTIJÄRJESTELMÄN KÄYTETTÄVYYDEN PARANTAMINEN

(julkinen osa)

Harri Berglund

Opinnäytetyö
Elokuu 2012
Tietotekniikan koulutusohjelma
Ohjelmistotekniikan suuntautumis-
vaihtoehto

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Ohjelmistotekniikan suuntautumisvaihtoehto

HARRI BERGLUND:

Monitorointijärjestelmän käytettävyyden parantaminen (julkinen osa)

Opinnäytetyö 26 sivua, josta liitteitä 1 sivu
Elokuu 2012

Opinnäytetyön tavoitteena oli parantaa toimeksiantajan kehittämän monitorointijärjestelmän käytettävyyttä sekä loppukäyttäjän että ylläpitäjän näkökulmasta. Tavoitteeseen pääsemiseksi suunniteltiin ja toteutettiin uusi web-sovellus, joka voidaan myöhemmin integroida osaksi järjestelmää.

Sovelluksen tarkoitus oli nopeuttaa sellaisen suuren datamäärän muokkaamista, jonka käsittelyyn ei entuudestaan ollut tarkoituksenmukaista työkalua. Kyseisen datan muokkaaminen tuli sovelluksen avulla olla helppoa sekä vähemmän inhimillisille virheille altista.

Tässä työn kirjallisessa raportissa kuvataan, mitkä kriteerit vaikuttivat käytettyjen ohjelmistojen, tekniikoiden ja menetelmien valintaan. Lisäksi kerrotaan sovelluksen kehittämiseen liittyneistä vaiheista ja haasteista. Lopuksi esitellään ongelmiin löydettyjä ratkaisuja, pohditaan työssä opittuja asioita sekä arvioidaan suunnitteluvaiheen valintojen onnistuneisuutta ja sovelluksen jatkokehittelymahdollisuuksia.

Sovelluksen suunnittelua ja toteutusta tarkemmin käsittelevät luvut on rajattu pois työn julkisesta osasta luottamuksellisuussyistä. Lisäksi osa liitteistä on luottamuksellisia.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Option of Software Engineering

HARRI BERGLUND:

Improving the usability of an online monitoring system (public copy)

Bachelor's thesis 26 pages, appendices 1 page
August 2012

The objective of this thesis work was to improve the usability of an online monitoring system by developing a new web-based application which could later be integrated with the system. This application was meant to simplify and speed up the process of editing a large set of data. Using the application should also make this process less error-prone.

Programming languages and technologies used in this project are discussed in detail. Reasons for choosing an option over another are provided along with reflections on lessons learned. Possible future improvements to the newly written application are also discussed.

This thesis contains information which was considered confidential and as such was left out of the public version of this document. Above information concerns the design and implementation of the new application.

Key words: monitoring systems, linux, csv, ajax, php

ALKUSANAT

Parempi myöhään kuin ei milloinkaan, kuuluu vanha sanonta. Opinnäytetyön aiheen löytyminen vei oman aikansa, mutta sellaisen sitten viimein löydyttyä, osoittautui se varsin mielenkiintoiseksi. Tästä haluankin kiittää aiheen tarjonnutta yritystä, Proact Finland Oy:tä. Ohjauksesta ja palautteesta osoitan kiitokseni toimeksiantajan edustajalle Janne Rantaselle sekä työn ohjaajalle Jari Mikkolaiselle. Lisäksi tuesta työn aikana haluan kiittää perhettäni.

Tampereella elokuussa 2012

Harri Berglund

SISÄLLYS

1	JOHDANTO.....	8
1.1	Työn aiheesta	8
1.2	Toimeksiantaja Proact yrityksenä.....	8
1.3	Monitorointijärjestelmät	9
2	SUUNNITTELU (luottamuksellinen).....	10
3	OHJELMISTOT	11
3.1	Linux.....	11
3.1.1	Debian	12
3.1.2	Red Hat Enterprise Linux, CentOS ja Fedora.....	12
3.2	Apache HTTP Server.....	13
4	OHJELMOINTIKIELET JA -KIRJASTOT	14
4.1	Bash	14
4.2	PHP	14
4.2.1	Syntaksi.....	15
4.2.2	Huomioita.....	16
4.3	JavaScript.....	17
4.3.1	JQuery	18
4.3.2	DataTables.....	19
5	MUUT TEKNIIKAT.....	20
5.1	CSV-tiedostot.....	20
5.2	Tiedostojen lukitus ja lukkotiedostot	21
5.3	AJAX	21
5.4	JSON.....	22
6	TOTEUTUS (luottamuksellinen).....	23
7	JOHTOPÄÄTÖKSET JA POHDINTA	24
	LÄHTEET.....	25
	LIITTEET	26
	Liite 1. Proact Online Monitoring -palvelun esite.....	26

LYHENTEET JA TERMIT

AJAX	(engl. Asynchronous JavaScript And XML) Joukko web-sovelluskehityksessä vuorovaikutteisuuden lisäämiseksi käytettyjä tekniikoita
Bash	GNU-projektin tuottama komentotulkki
BSD-lisenssi	Vapaa ohjelmistolisenssi, joka sallii ohjelmakoodin muokkaamisen ja uudelleenkäytön niin avoimen lähdekoodin projekteissa kuin kaupallisissa tuotteissakin
CentOS	Yhteisön ylläpitämä Linux-jakelu, joka pohjautuu Red Hat Enterprise Linuxin lähdekoodiin
CSV	(engl. Comma-Separated Values) Yleisesti käytössä oleva tekstipohjainen tiedostoformaatti, jossa tieto on jaettu kenttiin käyttäen rajamerkkejä
Fedora	Red Hat:n sponsoroima ja yhteisön kehittämä Linux-jakelu
GNU	(engl. GNU's Not Unix) Vapaiden ohjelmistojen kehittämiseen keskittynyt ohjelmistoprojekti
GPL-lisenssi	GNU-projektin yleinen ohjelmistolisenssi
GUI	(engl. Graphical User Interface) Graafinen käyttöliittymä
HTTP	(engl. HyperText Transport Protocol) Hypertekstin siirtoprotokolla
JavaScript	EcmaScript-standardiin pohjautuva komentosarjakieli, jota käytetään erityisesti verkkosivujen interaktiivisuuden lisäämiseen
JQuery	Ohjelmakirjasto JavaScript-kielelle
kernel	Käyttöjärjestelmän ydin, joka toimii ikään kuin siltana sovelusten ja laitteistotasolla tapahtuvan tiedonkäsittelyn välillä
komentotulkki	Tekstipohjainen tietokoneohjelma, jolla ohjataan käyttöjärjestelmää

PHP	(engl. PHP: Hypertext Preprocessor) Komentosarjakieli, jonka merkittävin käyttökohde on dynaamisten verkkosivujen kehitys
Red Hat	Yhdysvaltalainen ohjelmistoyritys
RHEL	(engl. Red Hat Enterprise Linux) Red Hat:n kehittämä yrityskäyttöön suunnattu käyttöjärjestelmä
RPM	Paketinhallintajärjestelmä, jota käytetään mm. RHEL:ssa
skripti	Komentosarjakielen ohjelma, jota ei tarvitse kääntää, mutta jonka ajoon tarvitaan kielelle sopiva tulkki
XML	(engl. eXtensible Markup Language) Merkintäkieli, jolla isoja tietokokonaisuuksia saatetaan jäsentää selkeämmin

1 JOHDANTO

1.1 Työn aiheesta

Työn aihe löytyi, kun Proact Finland Oy:n Tampereen toimipisteeseen haettiin helmikuussa 2012 opinnäytetyöntekijää. Alustavien keskusteluiden jälkeen sopimus opinnäytetyön tekemisestä sekä sen aiheesta pääpiirteittäin tehtiin kuun lopulla. Tuolloin oli tarkoitus, että työ jakautuisi kahteen osaan. Näistä ensimmäinen olisi käsittänyt monitorointijärjestelmän taustatehtäviä hoitavien Bash-skriptien kääntämistyön Python-kielelle sekä samalla olemassa olevan ja uuden lähdekoodin perusteelliseen dokumentointiin. Toinen osa olisi pitänyt sisällään uuden, kyseisen järjestelmän osaksi kaavailun, editorisovelluksen suunnittelun ja toteutuksen.

Ennen kuin näitä tehtäviä lähdettiin työstämään, tultiin siihen tulokseen, että edellä mainitun editorin toteuttaminen olisi ensisijaista. Tähän päädyttiin, koska olemassa olevat Bash-skriptit hoitivat tehtävänsä hyvin ja niiden toimintaan tutustumiseen sekä uudelleen kirjoittamiseen Pythonilla olisi arviolta kulunut huomattavasti pidempään. Editorin toteutukseen vaadittavaan ohjelmointityöhön arvioitiin kuluvan aikaa yhdestä kahteen kuukauteen. Lisäksi aikaa tiedettiin tarvittavan tämän raportin kirjoittamiseen, jolloin optimistinen arvio valmistumisajankohdasta olisi sijoittunut toukokuulle.

1.2 Toimeksiantaja Proact yrityksenä

Proact on vuodesta 1994 toiminut tiedon tallennukseen, varmuuskopiointiin sekä virtuaalisointiin keskittyvä IT-palveluiden tarjoaja. Pilvi-infran ja tallennusratkaisujen toteuttajana Proact lukeutuu Euroopan johtaviin toimijoihin (Tietoja Proactista 2012). Yritys tarjoaa tukipalveluja paikallisella kielellä 13 maassa ja tukea on asiakkaan tarpeiden mukaisesti saatavilla ympärivuorokautisesti (Yritystason tekninen tuki Proactilta 2012).

Tämän työn kannalta keskeisessä osassa on Proactin tarjoama PoMo-palvelu (liite 1), johon työssä toteutettu editori voidaan jatkossa integroida. PoMo kattaa erittäin monipuolisen monitorointijärjestelmän sekä siihen liittyvät tukipalvelut.

1.3 Monitorointijärjestelmät

Monitorointijärjestelmän avulla voidaan mm. tarkkailla tietoverkon tilaa reaaliajassa, lähettää verkon ylläpitäjälle automaattisia huomautuksia vikatilanteista esimerkiksi tekstiviestitse tai sähköpostitse sekä koostaa raportteja verkon toiminnasta pidemmällä aikavälillä. Yleisiä tarkkailtavia seikkoja ovat palvelinten osalta mm. jäljellä oleva vapaa levytila sekä prosessorin käyttöaste. Usein tarkkailtavana voi olla myös muita laitteita, kuten tulostimia tai vaikkapa rakennuksen kattava ilmastointijärjestelmä.

Markkinoilla on lukuisia kaupallisia monitorointijärjestelmiä, kuten IBM Tivoli Network Manager, sekä useita avoimeen lähdekoodiin pohjautuvia vaihtoehtoja. Esimerkkinä suositusta avoimen lähdekoodin monitorointijärjestelmästä voidaan mainita GPL-lisenssillä julkaistu Nagios. Nagioksesta on olemassa myös laajempi kaupallinen versio. Alun perin Linuxille suunniteltua sovellusta voidaan ajaa myös muilla Unix-pohjaisilla käyttöjärjestelmillä (About Nagios 2012).

2 SUUNNITTELU

Tämä luku on jätetty pois työn julkisesta versiosta luottamuksellisuussyistä.

3 OHJELMISTOT

3.1 Linux

Linux on avoimen lähdekoodin käyttöjärjestelmäydin eli kerneli (engl. kernel), jonka ensimmäisen version kehitti ja julkaisi vuonna 1991 suomalainen Linus Torvalds. Hän on ollut mukana kehittämässä Linuxia siitä lähtien, mutta tähän mennessä korjauksia ja lisäyksiä ovat toteuttaneet jo tuhannet muutkin kehittäjät. Linuxia pidetäänkin yhtenä oivallisena esimerkkinä avoimen lähdekoodin projekteista puhuttaessa.

Linux-nimitystä käytetään yleisesti viitattaessa Unixin kaltaisiin käyttöjärjestelmiin, jotka hyödyntävät em. samannimistä kerneliä. Näitä käyttöjärjestelmiä, joista käytetään nimitystä Linux-jakelu (engl. Linux-distribution), on olemassa mittava määrä. Monet niistä perustuvat johonkin aiempaan jakeluun ja niiden jo valmiiksi laajoihin pakettivarastoihin. Esimerkiksi Debian ja Red Hat Enterprise Linux ovat toimineet perustana lukuisille muille jakeluille.

Linux-jakelut tarjoavat paketinhallintajärjestelmän (engl. package management system), joka auttaa ohjelmien asentamisessa, päivittämisessä, poistamisessa jne. Paketinhallinnan tehostamiseksi on saatavilla erilaisia apuohjelmia, jotka hyödyntävät paketinhallintajärjestelmän tarjoamia toimintoja. Monet näistä apuohjelmista ovat komentorivipohjaisia, mutta osalle on saatavissa myös erillinen graafinen käyttöliittymä eli GUI (engl. Graphical User Interface).

Paketinhallintajärjestelmä käyttää hyödykseen pakettivarastoja, joista se voi etsiä käyttäjän haluamaa uutta ohjelmapakettia tai tarkistaa onko jostakin ohjelmapaketista tarjolla uudempaa versiota. Asennusten ja päivitysten yhteydessä paketinhallintajärjestelmän tehtävä on tarkistaa myös ohjelmapakettien väliset riippuvuussuhteet.

Ohjelmapaketin kehittäjä voi määritellä paketille riippuvuuksia, mikäli sen sisältämä ohjelma vaatii toimiakseen esimerkiksi kolmannen osapuolen kehittämän ohjelmakirjaston, jota käyttäjän järjestelmästä ei välttämättä entuudestaan löydy. Päivitykset voidaan asettaa tapahtumaan automaattisesti tai vain käyttäjän toiveesta. Tarpeettomiksi käyneiden ohjelmapakettien poistaminen voidaan myös automatisoida haluttaessa.

3.1.1 Debian

Debian on pitkään kehitetty Linux-jakelu, jonka ensimmäinen versio ilmestyi jo vuonna 1994. Debian on yksi käytetyimmistä Linux-jakeluista – pitkän historian lisäksi tähän voidaan pitää syinä jakelun tavoitetta tarjota mahdollisimman vakaa käyttöjärjestelmä, laajaa tukea erilaisille prosessoriarkkitehtuureille sekä ohjelmapakettien vaikuttavaa määrää (Debian GNU/Linux 2012).

Debianin paketinhallintajärjestelmä dpkg käyttää deb-päätteisiä paketteja, jotka ovat käytössä myös mm. Debianiin pohjautuvassa Ubuntuissa sekä sen rinnakkaisversioissa. Debianissa paketinhallintaan voidaan hyödyntää tunnettua APT-apuohjelmaa (engl. Advanced Packaging Tool). APT:sta on saatavilla versio myös RPM-paketteja hyödyntäville käyttöjärjestelmille.

3.1.2 Red Hat Enterprise Linux, CentOS ja Fedora

Red Hat Enterprise Linux on yhdysvaltalaisen Red Hat:n kehittämä yrityskäyttöön suunnattu Linux-jakelu. Yleisesti käytetty lyhenne jakelulle on RHEL. Ensimmäinen versio RHEL:n edeltäjästä, aikanaan suositusta Red Hat Linuxista julkaistiin 1995.

Red Hat:n kehittämä heti käyttövalmis versio RHEL:sta on maksullinen eikä siten ole vapaasti levitettävissä kuten monet muut Linux-jakelut. Red Hat myös myy tuki- ja konsultointipalveluja RHEL:n käyttäjille. Red Hat kuitenkin julkaisee jakelun ohjelmistojen lähdekoodin ja ulkopuoliset kehittäjät voivat näin ollen kääntää ja koota periaatteessa identtisen käyttöjärjestelmän – sillä erotuksella, että se ei sisällä Red Hat:n tavaramerkkejä, kuten esimerkiksi yrityksen logoa.

Yksi merkittävä kokonaan RHEL:n lähdekoodiin perustuva, vapaasti levitettävä käyttöjärjestelmä on CentOS (engl. Community enterprise Operating System). Käytännössä CentOS:n ja RHEL:n välillä ei siis juuri ole varsinaisia eroja. Red Hat:n tarjoaman tuen sijaan CentOS:n käyttäjät voivat tarvittaessa etsiä apua yhteisön keskustelufoorumeilta.

Fedora on Red Hat:n sponsoroima ja Fedora-projektin kehittämä, RHEL:n ja CentOS:n tapaan RPM-paketteja hyödyntävä käyttöjärjestelmä. Fedora-projektin kehittäjiin lukeu-

tuu sekä Red Hat:n työntekijöitä että vapaaehtoisia kehittäjiä ympäri maailmaa (Fedora Project 2012). Fedora-projektin tavoite on kehittää ainoastaan avoimeen lähdekoodiin perustuva käyttöjärjestelmä, jonka sisältämät ohjelmistot edustavat samalla teknisesti ajan kirkkainta kärkeä.

3.2 Apache HTTP Server

Apache Software Foundationin johdolla yhteisövoimin kehitettävä Apache HTTP Server on ollut suosituin HTTP-palvelinsovellus vuodesta 1996 lähtien (March 2012 Web Server Survey 2012). HTTP-palvelimella voidaan ylläpitää mm. käyttäjien henkilökohtaisia blogeja ja kotisivuja tai suurta yhteisöportaalia kuten YouTubea.

Tästä palvelinsovelluksesta käytetään yleisesti pelkistettyä Apache-nimitystä. Apache on saatavilla lukuisille eri käyttöjärjestelmille, mukaan lukien Unix, Linux, Solaris ja Windows -käyttöjärjestelmät. Suurella osalla Apache-palvelimista on käytössä jokin Linux-jakelu (Sander 2012).

4 OHJELMOINTIKIELET JA -KIRJASTOT

4.1 Bash

Bash:llä voidaan tietotekniikassa viitata Unixin kaltaisille käyttöjärjestelmille saatavan komentotulkin lisäksi tulkin ymmärtämään skriptikieleen. Bash on akronyymi sanoista Bourne-again shell, joka viittaa vanhempaan Bourne shell -komentotulkkiin. Syntaksi ja perusominaisuudet ovat molemmissa samat, jonka lisäksi Bash:stä löytyy runsaasti lisäominaisuuksia ja -toimintoja.

Työssä oli alun perin tarkoitus tutustua Bash-kielen saloihin varsin syvällisesti, mutta editorin toteutuksen osoittauduttua ensisijaiseksi tehtäväksi, jäi kieli vähäisemmälle huomiolle. Työn edetessä toteutettiin kuitenkin monia lyhyitä työskentelyä helpottavia Bash-skriptejä, joista ensimmäinen oli kuviossa 1 näkyvän skriptin kaltainen.

```

user@host:~/examples$ ls
bash-example.sh source
user@host:~/examples$ tail bash-example.sh
#!/bin/sh

suffix=$(date +%d%m%y)".tar.gz"
name="$1-"$suffix

tar -cvzf $name $1
user@host:~/examples$ ./bash-example.sh source
source
user@host:~/examples$ rm source
user@host:~/examples$ ls
bash-example.sh source-220812.tar.gz
user@host:~/examples$ tar -zxvf source-220812.tar.gz
source
user@host:~/examples$ rm source-220812.tar.gz
user@host:~/examples$ ls
bash-example.sh source

```

KUVIO 1. Kuvakaappaus Bash-komentotulkista

4.2 PHP

PHP on suosittu komentosarjakieli, jolla kirjoitetut ohjelmat tulkitaan vasta, kun ne suoritetaan. Ohjelmakoodia ei siis tarvitse erikseen kääntää (engl. compile), jotta ohjelma

olisi suoritettavissa, toisin kuin esimerkiksi C++ -kielisten ohjelmien kohdalla. Tällä on tietenkin omat hyvät ja huonot puolensa.

Valmiiksi konekielelle käännetyt ohjelmat suoriutuvat tehtävistään usein huomattavasti nopeammin kuin tulkattavilla kielillä kirjoitetut vastaavan toiminnallisuuden tarjoavat ohjelmat, mikäli ohjelman on esimerkiksi suoritettava paljon prosessoritehoa vaativia monimutkaisia laskutoimituksia. Toisaalta tulkattavaa kieltä käytettäessä ohjelmoijan ei tarvitse lähdekoodimuutosten jälkeen odotella, että kääntäjä (engl. compiler) saa työnsä tehtyä ja muutosten vaikutukset ohjelman toimintaan voidaan havaita ja arvioida.

Selkeästi yleisin käyttökohte PHP:lle ovat dynaamiset web-sivustot. PHP ei kuitenkaan ole täsmäkieli (ts. tarkkaan rajatun ongelman ratkomiseen tai vain tietylle sovellusalueelle suunniteltu kieli). Mahdollisten käyttökohteiden skaala on erittäin laaja, sillä PHP:tä käyttäen voidaan kirjoittaa komentoriviohjelmaa hoitamaan järjestelmän ylläpitotehtäviä tai luoda käyttöjärjestelmäriippumattomia työpöytäsovelluksia. PHP:llä työpöytäsovellusten luominen on mahdollista PHP-GTK -laajennusta käyttäen, vaikka se ei olekaan kielen vahvin alue (What can PHP do? 2012).

PHP:tä voidaan käyttää lukuisilla eri käyttöjärjestelmillä, joihin lukeutuvat mm. Mac OS X, Linux, OpenBSD, Solaris ja Windows. PHP tukee myös suurinta osaa käytössä olevista web-palvelinohjelmistoista. Kielen monikäyttöisyyttä lisää se, että ohjelmoijissa voidaan valita joko proseduraalinen tai oliopohjainen ohjelmointitapa, tai näiden sekoitus. (What can PHP do? 2012.)

4.2.1 Syntaksi

Kielen syntaksi on monilta osin hyvin samanlainen mm. C:n, Javan ja Perlin kanssa. Tämän vuoksi aiemmin ohjelmointia edes jonkin verran harrastanut henkilö voi yleensä aloittaa hyödyllisten ohjelmien tuottamisen PHP:llä varsin nopeasti. Erona edellä mainittuihin kieliin on, että PHP-koodia voidaan upottaa suoraan HTML-dokumentteihin. PHP-tulkki etsii dokumentista aloitus- ja lopetustageja ja suorittaa vain niiden väliin sijoitetun ohjelmakoodin (kuvio 2).

```

<!DOCTYPE html>
<html>
  <head>
    <title>PHP example</title>
  </head>
  <body>
    <?php

    $sum = $_GET['a'] + $_GET['b'];
    $var = 3 * ($sum / 2);
    echo $var;

    $var = "suddenly, a string appears!";
    printf("<br />%s<br />", $var);

    ?>
  </body>
</html>

```

KUVIO 2. Esimerkki HTML-dokumentin sisään upotetusta PHP-koodista

Muuttujien nimet aloitetaan \$-merkillä. Toisin kuin mm. C-kielessä, PHP:ssä muuttujien tyyppiä ei tarvitse määritellä etukäteen ja muuttujan sisältämä data voi myös vaihtaa tyyppiä myöhemmässä vaiheessa, vaikka luvusta merkkijonoksi tai toisin päin. Monet muista kielistä tutut funktiot, kuten kuviossa 2 näkyvä *printf*, löytyvät PHP:stä. Tekstiä voidaan tulostaa näytölle myös *echo*-komennolla, joka ei ole varsinainen funktio, vaan yksi PHP:n tarjoamista ns. kielirakenteista (engl. language construct).

4.2.2 Huomioita

PHP:tä on kritisoitu vuosien varrella useista heikkouksista, kuten huonosta tietoturvasta, joka on kuitenkin parantunut merkittävästi kielen alkuajoilta. Esimerkiksi PHP:n versiosta 4.2.0 lähtien loppukäyttäjät eivät oletusarvoisesti enää pysty syöttämään muuttujia PHP-skriptiin ajonaikana (PHP4 ChangeLog 2002). PHP5:ssä ei ole suoraa Unicode-tukea; tuki on tulossa PHP6:een (Minutes PHP Developers meeting 2005). Lisäksi sisäänrakennettujen (engl. built-in) funktioiden nimien sekavuus on herättänyt arvostelua.

Funktioiden nimissä esiintyvistä epä johdonmukaisuuksista esimerkkinä käyvät seuraavat funktioparit: *strpos* ja *strpos* sekä *strcmp* ja *strcasecmp*. Molempien parien funktiot ovat toiminnaltaan keskenään muuten identtisiä, mutta ensimmäiset funktiot tekevät eron käsittelemiensä merkkijonojen pien- ja suuraakkosten välillä, kun jälkimmäiset funktiot puolestaan tulkitsevat a:n ja A:n samaksi merkiksi.

Tämä pieni ero saattaa aiheuttaa päänvaivaa, mikäli funktioiden nimien kanssa ei ole jatkuvasti tarkkana. Mahdollisesti käytössä oleva kehitystyökalu saattaa varoittaa asiasta välittömästi ja viime kädessä PHP-tulkin antama virheilmoitus ohjaa virheen jäljille.

4.3 JavaScript

JavaScript on erittäin suosittu erityisesti web-sovelluksissa käytetty komentasarjakieli. Kielen tärkein käyttökohde on verkkosivujen dynaamisuuden lisääminen. Hyvä esimerkki dynaamisesta toiminnallisuudesta on lomakkeen syötteiden järjestyksen tarkistus jo ennen kuin tietoja lähetetään palvelimelle. Lomake voidaan tarkistaa mm. siltä varalta, ettei käyttäjä ole jättänyt mitään pakollista kenttää täyttämättä tai vahingossa syöttänyt kirjaimia puhelinnumerokenttään. Näin voidaan vähentää palvelimelle aiheutuvaa kuormaa ns. turhista pyynnöistä. Huomioitavaa on, että käyttäjän syöttämät tiedot tulee kuitenkin aina tarkistaa myös palvelimella ajettavassa ohjelmakoodissa.

Oliopohjaisuudesta huolimatta JavaScript erottuu useimmista olio-ohjelmointikielistä siinä, että sen oliomalli pohjautuu prototyyppeihin (engl. prototype) luokkien sijaan. Kuvion 3 lähdekoodiesimerkissä luodaan ensin prototyypin pohjalta olio nimeltä *board*, jonka jälkeen prototyyppiin lisätään ominaisuus nimeltä *year*. Lopuksi *board*-olion *year*-ominaisuuden arvo kirjoitetaan dokumenttiin, johon ohjelmakoodi on upotettu.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Prototype example</title>
  </head>
  <body>
    <script type="text/javascript">
      function Snowboard(manufacturer, model) {
        this.manufacturer = manufacturer;
        this.model = model;
      }

      var board = new Snowboard("Burton", "Clash");
      Snowboard.prototype.year = null;
      board.year = 2003;

      document.write(board.year);
    </script>
  </body>
</html>
```

KUVIO 3. JavaScript-esimerkkikoodi prototyyppien käytöstä

4.3.1 JQuery

JQuery on avoimen lähdekoodin JavaScript-ohjelmakirjasto, jonka avulla voidaan mm. käsitellä DOM-elementtejä (kuvio 4) sekä toteuttaa Ajax-pyyntöjä. Sitä käytetään tutkimuksen mukaan noin puolessa kaikista web-sivustoista (Usage of JavaScript libraries for websites 2012).

Monien JavaScript-kirjastojen tapaan JQuery on saatavilla yhtenä js-päätteisenä tiedostona, joko selkeästi luettavassa muodossa tai valmiiksi tiivistettynä. Tiivistetystä versiosta on poistettu kommentit ja ylimääräiset välimerkit, jolloin tiedoston koko pienenee ja sen lataamiseen vaadittava aika lyhenee. Tiedosto sisällytetään web-sivuun lisäämällä sivulle linkki tiedostoon joko paikallisella palvelimella tai jollakin julkisella CDN:lla (engl. Content Delivery Network), kuten kuviossa 4.

```
<!DOCTYPE html>
<html>
  <head>
    <title>jQuery example</title>
    <script type="text/javascript"
      src="http://code.jquery.com/jquery-1.7.2.js"></script>
  </head>
  <body>
    <div id="click_me">
      Click here
    </div>
    <p id="hidden_paragraph">
      Lorem ipsum dolor sit amet, consectetur adipisicing elit,
      sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
      Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
      nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
      reprehenderit in voluptate velit esse cillum dolore eu fugiat
      nulla pariatur. Excepteur sint occaecat cupidatat non proident,
      sunt in culpa qui officia deserunt mollit anim id est laborum.
    </p>
    <script type="text/javascript">
      $("#hidden_paragraph").hide();

      $("#click_me").click(function() {
        $("#hidden_paragraph").slideDown("slow");
      });
    </script>
  </body>
</html>
```

KUVIO 4. Esimerkki DOM-elementtien käsittelystä JQueryä käyttäen

4.3.2 DataTables

DataTables on avoimen lähdekoodin (GPL v2 -lisenssi tai BSD-lisenssi) JQuery-liitännäinen, jonka avulla voidaan luoda dynaamisia HTML-taulukoita. Käyttäjä voi näin järjestää (engl. sort), suodattaa (engl. filter) ja selata suurtakin määrää dataa. Liitännäinen ei välttämättä luo taulukkoa kaikesta tutkittavasta datasta kerralla, vaan jakaa datan sivuihin, joiden välillä käyttäjä voi liikkua. Scroller-lisäosan avulla data voidaan sivuihin jakamisen sijaan esittää vieritysnäkymässä.

Liitännäisen käyttämä data voidaan kirjoittaa valmiiksi HTML-taulukkoon – kun liitännäinen sitten alustetaan JavaScriptissä, se korvaa tämän perinteisen HTML-taulukon interaktiivisella versiolla (kuvio 5). Liitännäinen voidaan kuitenkin alustaa ennen kuin käytettävästä datasta on mitään havaintoa. Tällöin voidaan JavaScriptissä tehdä Ajax-pyyntö esimerkiksi palvelimella olevalle PHP-skriptille, joka hakee datan tietokannasta, CSV-tiedostosta tai muusta tietolähteestä.

Show entries Search:

Rendering engine	Browser	Platform(s)	Engine version	CSS grade
KHTML	Konqueror 3.1	KDE 3.1	3.1	C
KHTML	Konqueror 3.3	KDE 3.3	3.3	A
KHTML	Konqueror 3.5	KDE 3.5	3.5	A
Misc	Links	Text only	-	X
Misc	Lynx	Text only	-	X
Gecko	Mozilla 1.0	Win 95+ / OSX.1+	1	A
Gecko	Mozilla 1.1	Win 95+ / OSX.1+	1.1	A
Gecko	Mozilla 1.2	Win 95+ / OSX.1+	1.2	A
Gecko	Mozilla 1.3	Win 95+ / OSX.1+	1.3	A
Gecko	Mozilla 1.4	Win 95+ / OSX.1+	1.4	A

Showing 21 to 30 of 57 entries ◀ Previous Next ▶

KUVIO 5. DataTables-liitännäinen vakioasetuksilla

5 MUUT TEKNIIKAT

5.1 CSV-tiedostot

CSV (engl. Comma-Separated Values) on pitkään laajassa käytössä ollut tekstipohjainen tiedostoformaatti, jonka tiedostorakenteelle ei ole olemassa tarkasti noudatettua universaalia määrittelyä. Internet-protokollien standardoinnista vastaava IETF-organisaatio on julkaissut asiakirjan, joka sisältää kuvauksen formaatin yleisimmistä piirteistä (RFC4180 2005). Tämä asiakirja on kuitenkin verrattain tuore ja siinä kuvattua tiedostorakenteesta on eri yhteisöjen ja yritysten käytössä lukemattomia variaatioita. CSV-tiedostoille yhteistä on taulukkorakenne, jossa rivit ja sarakkeet on erotettu toisistaan ns. rajamerkeillä (kuvio 6).

```
doe, john, 123, unknown  
crane, denny, 9001, owner  
freeman, gordon, "0, 5", mute  
carter, sam, 100, colonel  
soap, bob, 555, unknown  
smith, mr, 101110, agent  
cobb, jayne, 2002, hero
```

KUVIO 6. Esimerkki CSV-tiedostosta

Perinteisesti sarakkeiden erottamiseksi toisistaan on käytetty pilkkua ja rivien erottamiseksi yksinkertaisesti rivinvaihtoja. On kuitenkin olemassa myös toteutuksia, joissa yksi taulukon rivi saattaa ulottua useammalle riville tiedostossa. Näin voi olla esimerkiksi silloin, kun jokin sarake sisältää tekstiä, jonka muotoilu halutaan säilyttää. Erikoismerkkejä, kuten juuri rivinvaihtoja tai pilkkuja, sisältävät kentät ympäröidään sovitulla merkillä, useimmiten lainausmerkein (kuvio 6). Tällöin lainausmerkkien sisällä olevia erityismerkityksen omaavia merkkejä ei huomioida, kun tiedostoa luetaan ohjelmallisesti.

Useimmat taulukkolaskentaohjelmat osaavat jäsentää oikein CSV-muotoillun datan, joka kopioidaan tekstitiedostosta suoraan taulukkoon. Monet tietokannan hallintajärjestelmät tukevat niin ikään CSV-formaattia. Tuki rajamerkkien määrittelylle vaihtelee sovelluksesta tai tietokannasta riippuen.

5.2 Tiedostojen lukitus ja lukkotiedostot

Tiedoston lukituksella tarkoitetaan mekanismeja, jonka avulla prosessi voi varata tiedoston tai jopa kokonaisen hakemiston käyttöönsä. Lukituksen tarkoituksena on estää mm. seuraavan skenaarion mahdollisuus: ensimmäinen ohjelma lukee tiedostosta rivin käsittelyä varten, toinen ohjelma kirjoittaa samalle riville, ja lopulta ensimmäinen ohjelma kirjoittaa tiedostoon kumoten riville toisessa ohjelmassa tehdyt muutokset.

Mikäli tämän kaltainen tilanne pääsee syntymään, on tiedostossa tapahtuman jälkeen oleva data joko vanhaa (toisen ohjelman tallentama tieto ei tallentunut) tai pahimmassa tapauksessa käyttökelvotonta; ensimmäisen ohjelman tekemä kirjoitus saattaa muuttaa tiedoston lukukelvottomaksi, mikäli se olettaa, että tiedoston sisältö on säilynyt samana kuin mitä se oli sen lukiessa tiedostoa hetkeä aiemmin.

Yksinkertaisimmillaan tiedostojen lukituksen voi toteuttaa ns. lukkotiedoston (engl. lock file) avulla: prosessi luo tiedoston tai vaihtoehtoisesti kansion sen merkiksi, että jotain toista tiedostoa käsitellään, ja poistaa lukkotiedoston käsittelyn päätyttyä. Tätä ennen prosessi tarkistaa, onko lukkotiedosto jo olemassa, ja odottaa tarvittaessa kunnes tiedosto poistetaan toisen prosessin toimesta. Tämä ns. ohjeellinen tapa hoitaa tiedoston lukitus vaatii sen, että kaikki tiedostoa käsittelemään pyrkivät prosessit noudattavat samaa kaavaa lukituksessa.

5.3 AJAX

AJAX lyhenne tulee sanoista Asynchronous JavaScript And XML, mutta sillä voidaan viitata yleisesti tekniikoihin, joita yhdessä käyttämällä voidaan luoda vuorovaikutteisia web-sovelluksia; esimerkiksi nimessä esiintyvän XML:n sijaan käytetään usein JSON:ia. Vuorovaikutteisuuteen pyritään välttämällä tilanteita, joissa koko web-sivu joudutaan lataamaan uudelleen.

Mahdollisuuksien mukaan selaimen ja palvelimen välillä lähetetään dataa ns. Ajax-pyyntöinä ja -vastauksina, jolloin selain ei jää odottamaan vastausta juuri lähetettyyn pyyntöön vaan jatkaa tarpeen mukaan sivun lataamista tai JavaScript-koodin tulkitsemista. Palvelimen lähettäessä vastauksen, siirrytään skriptin suorituksessa ns. takaisin-

kutsumetodiin (engl. callback method). Takaisinkutsumetodi saa parametrina palvelimen palauttaman vastauksen, jonka sisältö voidaan siten tutkia. Sisällöstä tehtyjen havaintojen perusteella saatetaan esimerkiksi päivittää nykyisellä sivulla olevan taulukon sisältö tai näyttää käyttäjälle virheilmoitus uudessa selainikkunassa.

5.4 JSON

JSON eli JavaScript Object Notation on avoin standardi tiedonsiirtoon. JSON-muotoiltu data on ihmiselle helposti luettavissa ja kirjoitettavissa, ja lisäksi tietokoneen näkökulmasta helposti jäsennettävissä ja tuotettavissa (Introducing JSON 2012). Nimestään huolimatta formaatti ei ole sidoksissa JavaScriptiin, vaan se on täysin ohjelmointikieliriippumaton.

JSON-objekti koostuu yksinkertaisimmillaan avain-arvo pareista. Lisäksi objekti voi sisältää listoja. Avaimen ja arvon väliin merkitään kaksoispiste ja avain-arvo parit erotetaan toisistaan pilkulla. Arvo voi olla lainausmerkein ympäröity merkkijono, luku, totuusarvo (*true* tai *false*), *null*, objekti tai lista. Kuviossa 7 nähdään hyvin suppea esimerkki.

```
{
  "character": "Harold Mayborne",
  "appearances": [
    {
      "episode": "Paradise Lost",
      "season": 6
    },
    {
      "episode": "It's Good to Be King",
      "season": 8
    }
  ]
}
```

KUVIO 7. Esimerkki JSON-objektista

6 TOTEUTUS

Tämä luku on jätetty pois työn julkisesta versiosta luottamuksellisuussyistä.

7 JOHTOPÄÄTÖKSET JA POHDINTA

Työn tuloksena toteutettiin uusi web-pohjainen sovellus toimeksiantajan käyttöön. Huomion arvoista on, että ohjelmointityöhön kului loppujen lopuksi huomattavasti kauemmin kuin aluksi optimistisesti arvioitiin. Kehitystyön aikana sovellukseen lisättiin joitakin ylimääräisiä ominaisuuksia sitä mukaa, kun ne nähtiin tarpeellisiksi.

Jatkossa sovellusta pystyttäisiin hiomaan monin tavoin: optimoimalla raskaimpia kohtia ohjelmakoodista, kiinnittämällä enemmän huomiota käyttöliittymän ulkoasuun sekä toteuttamalla käyttäjien toivomia lisätoiminnallisuuksia. Monet käyttöliittymän muutokset ja osa esille tulleista lisätoiminnallisuuksista olisi verrattain nopea toteuttaa. Työn aikana alulle laitettun ohjeen parantaminen voisi olla jatkotoimenpiteistä ensimmäinen.

Varsinaisen ohjelmointityön lisäksi tämän opinnäytetyön tekoon liittyi vahvasti useiden Linux-jakeluiden käyttö sekä erilaisten ohjelmien asentaminen ja asetusten säätäminen. Linuxista minulla oli entuudestaan kokemusta lähinnä Debianista ja sen paketinhallintajärjestelmästä, niinpä RPM-paketteja hyödyntävien käyttöjärjestelmien parissa täytyi aikaa käyttää myös uuden opetteluun. Lisäksi Bash:n syntaksi ja sen erot moniin muihin komentosarjakieliin tulivat työn kuluessa aiempaa tutummaksi.

Kaiken kaikkiaan koin, että työ opetti paljon uutta ja hyödyllistä. Monia tuttujakin asioita sai opetella uudestaan, vaikka ne joskus kuvitteli opetelleensa huolellisesti – kertaus on todella opintojen äiti.

LÄHTEET

About Nagios. 2012. Nagios Enterprises. Luettu 30.07.2012.
<http://www.nagios.org/about>

Debian GNU/Linux. 2012. DistroWatch. Luettu 30.07.2012.
<http://distrowatch.com/table.php?distribution=debian>

Fedora Project. 2012. Wikipedia. Luettu 17.07.2012.
http://en.wikipedia.org/wiki/Fedora_Project

Introducing JSON. 2012. JSON.org. Luettu 30.07.2012.
<http://www.json.org/>

March 2012 Web Server Survey. 2012. Netcraft. Luettu 30.07.2012.
<http://news.netcraft.com/archives/2012/03/05/march-2012-web-server-survey.html>

Minutes PHP Developers Meeting. 2005. PHP.net. Luettu 16.04.2012.
<http://php.net/~derick/meeting-notes.html>

PHP4 ChangeLog. 2002. PHP.net. Luettu 16.04.2012.
<http://www.php.net/ChangeLog-4.php>

Proactin sähköiset esitteet. 2012. Proact. Tulostettu 06.06.2012.
<http://www.proact.fi/asiantuntijapalvelut/>

RFC4180. 2005. Internet Engineering Task Force. Luettu 27.07.2012.
<http://tools.ietf.org/html/rfc4180>

Tietoja Proactista. 2012. Proact. Luettu 17.07.2012.
http://www.proact.fi/Tietoja_Proactista/

Temme, Sander. 2012. Apache HTTP Server Usage Survey Results. Julkaistu 28.02.2012. Luettu 30.07.2012.
https://blogs.apache.org/httpd/entry/apache_http_server_usage_survey

Usage of JavaScript libraries for websites. 2012. W3Techs. Luettu 27.07.2012.
http://w3techs.com/technologies/overview/javascript_library/all

What can PHP do? 2012. PHP.net. Luettu 16.04.2012.
<http://fi2.php.net/manual/en/intro-whatcando.php>

Yritystason tekninen tuki Proactilta. 2012. Proact. Luettu 17.07.2012.
http://www.proact.fi/suuryritysluokan_tuki/

LIITTEET

Liite 1. Proact Online Monitoring -palvelun esite (Proactin sähköiset esitteet 2012)

PROACT we secure
mission-critical
information

Ota käyttöön PoMo-palvelu!

Se monitoroi IT-ympäristösi tilaa 24/7 ja reagoi proaktiivisesti.

PoMo-palvelu (Proact Online Monitoring) on Proactin kehittämä 24/7 valvonta- ja hallintapalvelu, joka monitoroi IT-ympäristöäsi vuoden jokaisena päivänä ja vuorokauden jokaisena sekuntina. PoMon avulla varmistat tallennus-, palvelin- ja virtualisointiratkaisujesi käytettävyyden, ehdottoman tietoturvallisesti. Kaikki tarvittava tieto voidaan tiivistää vaikka yhteen graafiin, jos niin haluat.

Ja mikä parasta, niin PoMo reagoi aina proaktiivisesti. Levytilasi ei siis koskaan enää täyty yllättäen.

PoMo kerää tietoa ympäristöstäsi yhdessä määrittelemien suorituskykyindikaattoreiden (Key Performance Indicators, KPIs) avulla. Jos haluat vaikkapa nähdä Kiinan tuotantoyksikköjesi tilanteen, klikkaat monitorissa näkyvää palloa. Eteesi avautuu nopealukuinen graafi, joka antaa toimipistekohtaisen tiedon IT-ympäristön tilasta.

Kaikki seurattava tieto tallentuu yhdeksi vuodeksi, ja tuotettuun tietoon on helppo palata.

PoMo on siis myös kuin päiväkirja verkkosi tapahtumista.



Suomeksi ympäri vuorokauden

Proactin Service Desk palvelee sinua suomeksi ympäri vuorokauden.

Seuraamme puolestasi muun muassa seuraavia asioita:

- Työasema- ja palvelinvirtualisointiratkaisut
- Levjärjestelmät
- Palvelimet
- Konesalverkot
- Pilvipalvelut

Kommentteja asiakkailta:

- *"Toimii moitteettomasti. Meille tärkeintä on mahdollisuus saada selkeä kuva pidemmän aikavälin kehityksestä."*
- *"Palvelu on helppokäyttöinen ja joustaa muuttuvien tarpeidemme mukaan."*
- *"Toimintaympäristömme luonteen takia, meille on tärkeää saada palvelua ympäri vuorokauden suomeksi."*
- *"PoMo-palvelu on auttanut meitä karsimaan ylimääräiset valvontaohjelmistot."*
- *"Palvelun tehokas viestinvälitys on säästänyt meidät muutaman kerran varmalta käyttökatkolta."*

PoMo-palvelussa **jokaisella asiakkaalla on omat palvelun SLA-sopimukset**, joissa määrittelemme haluamasi vasteajan hälytyksiin, mitkä ovat hälytysrajat ja millaisiin ongelmiin reagoimme välittömästi sekä millaisia huoltotoimenpiteitä teemme.

Esimerkiksi Platinum Basic -valvontapalvelu ympäristössä, jossa on kahdennettu levjärjestelmä sekä 50 virtuaalipalvelinta, maksaa 1.646,00 euroa kuukaudessa.

Palvelullamme on tyytyväisyystakuu. Kysy lisätietoa PoMosta tai pyydä meiltä tarjous sales@proact.fi tai soittamalla numeroon 020 7919 200!

