

Timo Hoppendorff

# WEB-POHJAISEN RAPORTOINTIJÄRJESTELMÄN KEHITYS

Tietotekniikan koulutusohjelma  
2014

## ALKULAUSE

Tämä opinnäytetyö on tehty Cimcorp Oy:n Ulvilassa toimivalle pääkonttorille. Työn valvojana on Satakunnan Ammattikorkeakoulun Porin tekniikan yksikön puolelta toiminut Petteri Pulkkinen ja Reino Aarinen. Cimcorp Oy:n puolelta valvojana on toiminut Sakari Mikkola ja lisäksi työssä on ollut mukana Cimcorp Oy:n henkilökuntaan kuuluvia. Esitän kaikille mukana olleille suuret kiitokset saamastani tuesta.

Lisäksi haluan kiittää jokaista henkilöä, joka on omalla panoksellaan mahdollistanut työn onnistumisen. Ilman satunnaista muistuttelua tämä kirjallinen osuus ei olisi ehkä koskaan valmistunut. Kiitos siitä.

# WEB-POHJAISEN RAPORTOINTIJÄRJESTELMÄN KEHITYS

Hoppendorff, Timo  
Satakunnan Ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Asiakas: Cimcorp Oy  
Ohjaaja: Pulkkinen Petteri ja Aarinen Reino  
Sivumäärä: 73 sivua, 1 liitettä  
Kesäkuu 2014

Asiasanat: Web Service, Progress OpenEdge, Dokumentointi

---

Tämän opinnäytetyön aiheena oli kehittää Cimcorpin laitteiden ohjausjärjestelmän raportointia. Aikaisemmin raportit oli sisällytetty Cimcorpin laitteiden ohjausjärjestelmään ja niiden käyttö oli rajoitetun käyttäjäkunnan saatavilla. Raportit haluttiin siirtää pois laitteiden ohjausjärjestelmästä ja tuoda ne www-sivuille laajemman käyttäjäkunnan saataville.

Raporttien kehitys jakautui kahteen osaan, jotka olivat raporttien ulkoasu ja tekniikka jolla raportit tuotiin Progressin tietokannasta www-sivuille. Www-sivujen ulkoasu toteutettiin käyttäen HTML ja PHP:tä ja tekniikaksi valittiin Progressin tarjoama web service rajapinta, jolla Progressin tietokantaan saatiin suoritettua tietokantakyselyitä suoraan www-sivuilta.

Raportteja oli kolme joista kaksi näytettiin käyttäjälle taulukkomuotoisena ja yksi joka näytettiin käyttäjälle PDF muotoisena. Taulukkomuotoiset raportit oli mahdollista tallentaa PDF tai CSV muodossa.

## DEVELOPMENT OF WEB BASED REPORT SYSTEM

Hoppendorff, Timo

Satakunnan Ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Information Technology

Client: Cimcorp Oy

Supervisor: Pulkkinen Petteri and Aarinen Reino

Number of pages: 73 pages, 1 attachment

June 2014

Key words: Web Service, Progress OpenEdge, Documentation

---

The subscriber of the work was Cimcorp Oy. The purpose of this thesis was to develop and modify Cimcorp automatic robot system reporting. Earlier the reports were integrated to robot systems user interface and the reports could be accessed only with a limited number of users. Reports wanted to move away from user interface and bring reports to web page.

The reports development was divided to two different parts. First part was reports visual look and second part was technology how to bring data from Progress database to web page. Web page visual look implementation was done using HTML and PHP code and technology was chosen to web service interface provided by Progress. Using web service interface Progress database can be accessed directly from web page.

There were three kind of reports of which two was showed to user using table form at web page and one was showed to user using PDF form. Reports with table form were possible to store using PDF and CSV form.

# SISÄLLYSLUETTELO

LYHENNE- JA TERMILUETTELO .....	7
1. JOHTANTO .....	9
2. KÄYTETYT TEKNIIKAT .....	10
2.1 Hyper Text Markup Language HTML.....	10
2.2 Hypertext Preprocessor PHP .....	11
2.2.1 nuSOAP .....	11
2.2.2 ezPDF.....	12
2.3 Cascading Style Sheets CSS .....	12
2.4 JavaScript ja JQuery.....	12
2.5 XML.....	13
2.5.1 Schema.....	13
2.6 Web service.....	15
2.7 Apache Tomcat .....	15
2.8 Java Servlet .....	15
2.9 Progress ohjelmisto ja sen komponentit.....	16
2.9.1 Progress Database .....	17
2.9.2 Progress Nameserver.....	17
2.9.3 Progress appserver .....	18
2.9.4 Progress Web service adapter (WSA).....	19
2.9.5 Proxy generator toolkit.....	20
2.9.6 Hallinnointikomponentit .....	21
2.10 SOAP.....	23
2.10.1 Nimiavaruudet.....	24
2.10.2 Header .....	24
2.10.3 Body.....	24
2.10.4 Fault .....	24
2.11 Web service Description Language (WSDL).....	25
2.12 Python ja reportlab – kirjasto .....	26
3. TARKOITUS .....	27
4. TOTEUTUS .....	29
4.1 Käyttöliittymä.....	30
4.1.1 Varaston karttaraportti .....	32
4.1.2 Robottien sisältöraportti .....	38
4.1.3 Virheraportti.....	42
4.2 Palvelin ja sen ohjelmistot .....	45
4.2.1 Tomcat ja Progress servletti.....	46
4.2.2 Progressin Web service adapter (WSA).....	49
4.2.3 Tietokannan kyselykoodit .....	62
4.3 Projekti ja koodin uudelleen käytettävyys .....	66
4.3.1 mkmp.py .....	66
4.3.2 wsproxy.py.....	66
4.4 Havaitut ongelmat .....	67

5.	JATKOKEHITYS .....	69
6.	TULOKSET .....	70
	LÄHTEET .....	71

## LYHENNE- JA TERMILUETTELO

CSS	Cascading Style Sheets. Tyylisivut ovat yksinkertainen mekanismi tyylien lisäämiseksi web-sivuihin.
HTML	HyperText Markup Language on rakenteellinen merkkauskieli, yleisesti käytetty verkkosivuilla.
PHP	Hypertext Preprocessor. HTML-merkkaukseen upotettava skriptauskieli. PHP-koodi suoritetaan palvelimen palvelimella ja siitä luodaan HTML-merkkkaus, joka lähetetään asiakkaalle. Käytetään dynaamisten sivujen luomiseen.
Progress®	Progress Software Corporationin tietokanta, joka sisältää ohjelmistokehitysympäristön.
Python	Python on oliopohjainen ohjelmointikieli.
URL	Uniform Resource Locator. <a href="#">URL:n</a> avulla osoitetaan haluttuun tiedostoon WWW:ssä.
WWW	World Wide Web.
XML	eXtensible Markup Language. Rakenteellinen merkintäkieli.
WCS	Cimcorp Oy:n ohjelmistokokonaisuus, jolla ohjataan automaattisia keräily- ja varastojärjestelmiä.
WWP	Warehouse Web Portal. Cimcorpin luoma sivusto joka projektille, jolla saadaan kaikki tarvittava tieto samaan paikkaan ja helposti saataville.

WSDL	WSDL on W3C määrittämä XML-pohjainen kieli, jolla kuvataan internetissä oleva web teknologiaan pohjautuva palvelu eli web service.
CVS	CVS lyhenne tulee sanoista Concurrent Versioning System. CVS on laajalti käytetty avoimen lähdekoodin versionhallintaohjelma.
DTD	Document Type Definition eli DTD on yksi SGML- ja XML- kielten yhteydessä käytetty rakennemäärittelytapa. DTD määrittelee dokumentin sallitut ilmenemismuodot elementeille ja attribuuteille.
SGML	Standard Generalized Markup Language on yleinen dokumenttien rakenteen kuvaamisen kieli.
AJAX	Asynchronous JavaScript And XML. Tekniikka mahdollistaa nopeuden ja dynaamisten web-sivujen luomisen. Tekniikka mahdollistaa web-sivujen vain osittaisen päivittämisen päivittämättä koko sivua samalla kertaa.
HTTPD	Apache HTTP palvelin. Avoimen lähdekoodin ohjelmisto joka käsittelee asiakasohjelman HTTP kyselyt ja vastaukset.
SSH	Secure Shell on salattuun tietoliikenteeseen tarkoitettu protokolla.
Java JDK	Java Development Kit on javan kehitysympäristö jota tarvitaan java ohjelmien luomiseen.



## 1. JOHTANTO

Työn tilaajana oli Cimcorp Oy. Työn tarkoituksena oli kehittää parempi tapa automaattirobottijärjestelmän raportointia varten. Ideana oli hakea tiedot Progress OpenEdge tietokannasta ja näyttää ne järjestelmän käyttäjälle. Raportit on haettu Progress tietokannasta käyttäen web service tekniikkaa ja näytetään käyttäjälle tavallisella web-sivulla. Web sivut on toteutettu käyttäen HTML ja PHP kieltä. Raportteja on kolme ja kaksi niistä näytetään käyttäjälle taulukkomuodossa ja yksi raportti suoraan PDF tiedostona.

Aikaisemmin raportit oli integroitu suoraan robottijärjestelmän käyttöliittymään ja niitä pystyi käyttämään vain henkilöt, joilla oli pääsy järjestelmän käyttöliittymään. Työn tarkoituksena oli muuttaa raporttien esittäminen niin, että kuka tahansa samassa verkossa pääsisi tietoihin käsiksi.

Tämä opinnäytetyö liittyy toiseen opinnäytetyöhön. Toisen työn aiheena oli mm. Internet sivujen ulkoasu ja muut ominaisuudet joten tässä työssä keskitytään enemmän raporttien luomiseen ja sen tarvitsemiin tekniikoihin ja käytäntöihin. Työssä toteutetut raportit on upotettu toisen työn aiheena olevien internet sivujen sisään.

Seuraavassa luvussa käyn läpi työssä käytettyjä tekniikoita ja ohjelmia. Kolmannessa luvussa käyn läpi työn tarkoitusta ja neljännessä luvussa toteutusta. Viides luku käsittelee työn jatkokehitystä ja kuudes- ja seitsemäs luku työn tuloksia.

## 2. KÄYTETYT TEKNIIKAT

Sain tutustua työssäni useaan eri ohjelmakieleen ja niihin liittyviin ohjelmistoihin ja tekniikoihin. Työni on siis kokonaisuus johon liittyy monta eri osa-aluetta, jotta kokonaisuus olisi toimiva. Tässä luvussa kerrotaan pintapuolisesti työssäni käytetyistä ohjelmista ja tekniikoista.

### 2.1 Hyper Text Markup Language HTML

HTML on merkintäkieli internet sivujen sisällön näyttämiseen /7. HTML koodi kertoo internet selaimelle, miten näyttää sisältöä käyttäjälle. HTML kieli koostuu elementeistä joilla kaikilla on oma tehtävä. Elementit on ennalta määrätty, ja ne suorittavat tietyn tehtävän. Yleensä elementeillä on aloitus- sekä lopetuselementti. Elementtien sisälle laitetaan tekstiä, joka näytetään käyttäjälle. Elementit voivat sisältää attribuutteja, jotka tarkentavat elementin sisältöä /1.

HTML dokumentti koostuu perus elementeistä jotka ovat HTML, HEAD, TITLE ja BODY /1.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
<html>  
<head>  
<title></title>  
</head>  
<body></body>  
</html>
```

Yllä on esitetty yksinkertainen HTML sivu, sivulla ei ole sisältöä /2.

Ylimpänä elementtinä tulee olla DOCTYPE määrittely joka ei ole varsinaisesti HTML elementti vaan kertoo selaimelle millä versiolla sivut on kirjoitettu. HTML 4.01 DOCTYPE määrittely viittaa DTD tiedostoon. DTD määrittää HTML kielen säännöt ja sen kuinka tarkasti selain seuraa HTML määrittelyä. HTML 4.01 on käytössä kolmea eri DOCTYPE määrittelyä, mutta HTML 5 käytetään vain yhtä. DOCTYPE määrittelyllä ei ole lopetuselementtiä /2. Työssäni en syvenny tarkemmin DOCTYPE määrittelyihin.

HTML elementti kertoo tulkille, yleensä selaimelle, kyseessä olevan HTML merkintäkieli. HEAD elementti kertoo selaimelle, että elementti pitää sisällään otsikkotietoja. BODY elementti pitää sisällään varsinaisen sivun näytettävän sisällön /1.

## 2.2 Hypertext Preprocessor PHP

PHP on yleiskäyttöinen komentosarjakieli, joka soveltuu erityisesti web-sivujen kehittämiseen. PHP syntaksi muistuttaa C, Java ja Perl kieliä. PHP on lisenssivapaa vapaan lähdekoodin tuote, ja se on vapaasti kaikkien hyödynnettävissä. PHP tuki on saatavissa kaikkiin yleisempiin web-palvelimiin ja käyttöjärjestelmiin /3.

PHP koodi voidaan kirjoittaa HTML koodin sekaan. PHP koodi käännetään ja suoritetaan palvelimessa ja lopputulos palautetaan käyttäjälle. PHP koodi on olemassa ainoastaan palvelimella ja loppukäyttäjä ei näe sitä missään vaiheessa /3.

PHP on laajasti käytetty kieli palvelimissa ja sitä voidaan käyttää monissa käyttöjärjestelmissä ja ympäristöissä mm. Linux, Unix, Microsoft Windows ja Mac Os tukevat PHP:tä. PHP tukee useita tietokantoja ja on siksi laajalti käytetty palvelinpuolen toteutuksissa.

Työssäni päädyin käyttämään melko paljon PHP koodia toiminnallisuuksien luomiseen. Suurin osa käyttäjän toimista tai ”taustaälystä” on toteutettu PHP:llä. Työssä olisi voinut käyttää muutakin kieltä, mutta osaamistasoni ja tutun syntaksin johdosta päädyin käyttämään PHP:tä.

### 2.2.1 nuSOAP

nuSOAP on joukko PHP luokkia, jotka mahdollistavat web servicen kutsun ja toteutuksen PHP:stä. nuSOAP:a ei tarvitse asentaa palvelimelle joka tekee siitä helpon ottaa käyttöön. NuSOAP on avoimen lähdekoodin projekti joka aloitettiin vuoden 2002 aikana. NuSOAP mahdollistaa sovelluskehittäjän kehittää ja käyttää sovelluksia jotka tukevat SOAP 1.1, WSDL 1.1 ja HTTP 1.0 standardeja /4.

Työssäni päädyin käyttämään nuSOAP laajennusta huomattessani, että web-sivun tarvitsee kutsua Progressin tarjoamaa web service rajapintaa suoraan PHP:stä. NuSOAP on tehokas ja helppo käyttää internet-sivuilla.

### 2.2.2 ezPDF

ezPDF on joukko PHP luokkia, jotka mahdollistavat PDF tiedostojen luonnin suoraan PHP koodista. NuSOAP:n tapaan myöskään ezPDF tukea ei tarvitse asentaa erikseen palvelimelle. Riittää, kun ottaa PHP luokat käyttöön. Luokat on kehittänyt R&OS ltd /6.

Työssäni käytin ezPDF raporttien muuttamiseen PDF formaattiin ja varastokartan luomiseen PDF muodossa.

## 2.3 Cascading Style Sheets CSS

CSS on W3C kehittämä tekniikka. CSS on tekniikka, jolla määritellään ulkoasu HTML sivustolle. HTML kuvailee mitä sisältö on ja CSS määrittelee miten se näytetään käyttäjälle /7.

CSS kehittäminen alkoi 1990- luvun puolessa välissä, kun HTML kieli ei enää vastannut kaikkia sille asetettuja vaatimuksia.

CSS:n käyttö helpottaa www-sivuston hallintaa, koska tieto siitä miten HTML elementti näytetään loppukäyttäjälle, voidaan sisällyttää yhteen tiedostoon ja on näin helposti muokattavissa.

Käytin CSS:ä työssäni PHP sivujen ulkoasuun.

## 2.4 JavaScript ja JQuery

JavaScript on ohjelmointikieli, jota yleensä käytetään selainskriptien toteuttamiseen. JavaScriptiä ei tule sekoittaa Javaan vaikka nimessä yhtäläisyyksiä onkin. JavaScript on tulkattava skriptikieli, jolla useimmin tarkoitetaan web-sivujen sisään upotettua koodinpätkää, joka suoritetaan asiakkaan selaimessa. Java taas on ohjelmointikieli, jolla voidaan toteuttaa itsenäisiä ohjelmia toisin kuin JavaScriptillä joka vaatii selainta toimiakseen. Java koodit pitää myös kääntää ennen ajoa /9 sivu 3.

JQuery on nopea, kevyt ja ominaisuuksiltaan rikas JavaScript kirjasto. JQuery mahdollistaa monimutkaisten tehtävien suorittamisen vain muutamalla rivillä koodia. JQuery -kirjasto on helppo ottaa käyttöön web-sivuilla. JQuery kirjasto mahdollistaa mm seuraavat asiat: HTML ja CSS manipulaation, HTML eventtien luomisen, erilaisten tehosteiden ja animaatioiden käytön ja AJAX toiminnon. JQuery:n vahvuus on, että se toimii samalla tavalla kaikilla yleisimmillä selaimilla. JQuery:n suurimpia käyttäjiä ovat Google, Microsoft, IBM ja Netflix /10.

Työssäni käytin JavaScriptiä ja JQuerya muutamaan eri kohtaan tuomaan lisäarvoa käyttäjälle. JavaScriptin valinta oli helppo, sillä se on kevyt ja tehokas kieli, jolla voi tehdä web-sivuista rikkaamman käyttäjäkokemuksen.

## 2.5 XML

EXtensible Markup Language (XML) on helppo ja erittäin joustava tekstiformaatti. XML on ympäristöriippumaton ja näin ollen hyvä tiedon esitysmuoto eri järjestelmissä ja siksi laajasti käytetty verkossa ja eri järjestelmien tiedonvaihdossa /11.

HTML:n ja XML:n ero on se, että HTML kielessä tunnisteet määrittelevät tietojen ulkoasun ja tyylin eli otsikon paikat, kappaleen alkamiskohdan tai rivivaihdon. XML kielellä tunnisteet määrittelevät tietojen rakenteen ja merkityksen. XML on tehty tiedon kuvaamiseen /12.

XML tiedostoja on helppo lukea automaattisesti erilaisilla ohjelmilla, mikä selittää niin suuren käytettävyyden tiedon esittämiseen verkossa ja eri järjestelmien välillä /12.

### 2.5.1 Schema

XML schema kuvaa XML dokumentin rakenteen. XML Schema on XML -pohjainen vaihtoehto DTD:lle. XML schemalla on mahdollista määritellä elementit ja attribuutit, jotka voivat esiintyä dokumentissa, millä elementeillä voi olla lapsielementtejä, elementtien järjestyksen, elementin sisällön tai elementtien ja attribuuttien tyypit /13.

```

<?xml version="1.0"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>

```

Yllä kuvattu yksinkertainen XML dokumentti /14.

XML dokumentti alkaa XML määrittelyllä, jonka alla on note – elementti. Note elementeillä on to, from, heading ja body -lapsielementit.

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">

  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

Yllä on kuvattu XML dokumentin schema määrittely /14.

ComplexType:llä määritellään elementti, jonka alla on lapsielementtejä. Esimerkissä note elementin alla on to, from, heading ja body elementit, joiden kaikkien tyyppinä on merkkijono. Lapsielementit on määritetty singleType:llä, koska niillä ei ole lapsielementtejä /14.

## 2.6 Web service

Mikä on Web Service? Web service on ohjelma, jota voidaan ajaa ja käyttää verkon yli käyttäen laajasti standardoituja protokollia. Ohjelman käyttöön asiakas tarvitsee operaatioita, jotka on kuvailtu esimerkiksi käyttäen Web Service Description Language:a (WSDL) ja lähetetty ohjelmalle käyttäen Simple Object Access Protocol (SOAP). Ohjelma suorittaa tietyn toiminnon verkon yli ja palauttaa vastauksen asiakkaalleen. Web service tarjoaa standardoidun tavan viestiä erilaisten ohjelmien välillä käyttäen SOAP protokollaa. /15 sivu 84 6-2.

Avainsana web servicen määrittämiseen on laajasti standardoitu viestintä. Web servicet ovat hyvin standardoitu menetelmä, jonka avulla voidaan viestitellä erilaisten laitteiden ja käyttöjärjestelmien välillä laitteistoriippumattomasti /15 sivu 84 6-2.

Työssäni käytin web service toteutusta, joka mahdollisti tietokantakyselyt Progressin tietokannasta.

## 2.7 Apache Tomcat

Apache Tomcat on avoimen lähdekoodin internet palvelin ja servletti säiliö, jonka on kehittänyt Apache Software Foundation. Palvelin toteuttaa Java Servlet ja JavaServer Pages spesifikaation. Palvelimesta on saatavissa omat versiot Windowsille ja Linuxille /17.

Työssäni käytin Apache Tomcat:a Progressin tarjoaman web service servletin ajoon.

## 2.8 Java Servlet

Java Servlet on dynaamisten web-sovellusten kehittämiseen tarkoitettu jakelun rajapinta. Servletit soveltuvat parhaiten datan prosessointiin palvelimella. Palvelinsovelma on rakenteeltaan normaali Java-luokka, joka määrittelee servletin ja servlet säiliön toiminnallisuuden. Servlet säiliö on HTTP palvelimen komponentti, joka vastaa servletin ja palvelimen välisestä viestinnästä. Tunnetuin HTTP palvelin on Apache Tomcat jota myös käytin työssäni servlet säiliönä. Servlet säiliö luo servlet

ilmentymän ja ohjaa servletin toimintaa koko sen elinkaaren ajan. Elinkaaren loppuun servlet ilmentymä poistetaan muistista. Servlet säiliö tarjoaa servletille mm. HTTP-pyyynnön ja HTTP vastauksen välittämisen /33.

Servletin elinkaari on aika, jonka servlet toimii. Elinkaaren päätteeksi servleti tuhoetaan. Servletin elinkaaren osat ovat alustus, palvelu ja tuhoutuminen. Servletin alustusta kutsutaan vain kerran servletin luonnissa. Servletin palvelu on servletin päämetodi, joka suorittaa varsinaisen toiminnallisuuden. Servlet säiliö kutsuu palvelumetodia käsitelläkseen asiakassovelluksen lähettämät pyynnöt ja palauttaa asiakassovellukselle vastaukset. Servletin elinkaaren loppuun kutsutaan tuhoa -metodia, jossa servletin tietokantayhteydet tai muut avatut yhteydet katkaistaan. Servletin tuhoutumisen jälkeen Javan roskienkeruu vapauttaa muistin /34 /36.

## 2.9 Progress ohjelmisto ja sen komponentit

Progress Software Corporation on amerikkalainen yhtiö ja se on perustettu vuonna 1981. Progress Softwaren tuotteita käyttää lähes 140 000 organisaatiota yli 180 eri maassa ympäri maailman /16.

Progress Openedge on yksi monista ohjelmistoista joita Progress Software kuluttajille tarjoaa. Progress Openedgen monipuolinen ohjelmistokokonaisuus koostuu useista komponenteista. Komponentit ovat tietokanta (Progress Database), nimipalvelin (Progress Nameserver) ja appserver (Progress Appserver). Appserverin käyttöä voidaan tarvittaessa laajentaa ottamalla käyttöön lisäkomponentteja, jotka laajentavat Progressin ohjelmistokehityskokonaisuutta. Komponentit ovat web service adapter (WSA), appserver internet adapter (AIA) ja Sonic ESB. Progress Openedge tukee sekä Windows sekä Unix käyttöjärjestelmiä, ja ohjelmiston voi asentaa molempiin käyttöjärjestelmiin.

Keskityn jatkossa vain seuraaviin komponentteihin: tietokantaan, nimipalvelimeen, appserveriin sekä appserveriä laajentavaan komponenttiin web service adapter (WSA), joka mahdollistaa web servicen rakentamisen ja käytön Progressissa.

Progress OpenEdge ohjelmistokokonaisuudesta on olemassa eri versiota. Työssäni käytin versiota 10.1c. Kirjoitushetkellä uusin versio on 11. Toiminnallisuus on kuitenkin pysynyt samanlaisena ja



tässä työssä käytettyjä tekniikoita voidaan käyttää myös Progressin myöhemmissä versioissa. Työn ruutukaappaukset on otettu Progressin versiolla 10.2b.

Progress OpenEdgen kulmakivi on voimakas ohjelmointikieli Advanced Business Language ABL. ABL kieli on erityisesti kehitetty ohjelmien, käyttöliittymien ja tietokannan hallintaan /18.

### 2.9.1 Progress Database

Progress Database on yrityksille suunnattu suorituskykyinen ja skaalautuva relaatiotietokanta järjestelmä. Progress tietokantaan tehdään tietokantahakuja (database query) käyttäen progressin omaa ohjelmointikieltä Advanced Business Language (ABL), jota aiemmin 4GL:ksi kutsuttiin /19. Progress database tukee tietokannan transaktioita, joiden avulla tehdyt muutokset voidaan perua virhetilanteen sattuessa.

### 2.9.2 Progress Nameserver

Progress nameserver pitää sisällään nimipalvelimen, johon kaikki käytettävät appserverit liittyvät ja rekisteröityvät /15 sivu 37 2-5.

Nameserver on prosessi, joka hyväksyy asiakkaan yhteyspyynnön appserveriin ja ohjaa pyynnön oikealle appserverille. Käytettäessä useita appserver ilmentymiä nameserveriä on käytettävä tehokkaaseen yhteyksien hallintaan appserver ilmentymien välillä /15 sivu 37 2-5.

Nameserverin tilan ja sille rekisteröityneet appserver ilmentymät voidaan listata komennolla nsman -i <nimipalvelun nimi> -q. Esimerkiksi nsman -i NS1 -q.

```
$ nsman -i NS1 -q
OpenEdge Release 10.2B06 as of Mon Mar 19 19:14:48 EDT 2012

Connecting to Progress AdminServer using rmi://localhost:20931/Chimera (8280)
Searching for NS1 (8288)
Connecting to NS1 (8276)

NameServer NS1 running on Host hkiwcssrv01 Port 5162 Timeout 30 seconds.
Application Service      UUID                               Name      Host      Port  Weight  Timeout
AS.d92600p1
  fb6714531d7bfd32:2be41acf:13cba6fb71f:-8000  AS.d92600p1  hkiwcssrv01/10.94.19.20  39260  0      30
AS.[Default]
  fb6714531d7bfd32:2be41acf:13cba6fb71f:-8000  AS.d92600p1  hkiwcssrv01/10.94.19.20  39260  0      30
```

Kuva 1. Nameserverin tilatiedot.

Kuvassa näkyy, että NS1 nimiselle nameserverille on rekisteröitynyt d92600p1 niminen appserver.

### 2.9.3 Progress appserver

Appserver on OpenEdge ohjelman ydinkomponentti. Appserver komponentti toimii moottorina ajettaessa ABL:ä. Appserver on ABL ajonaikainen asiakas jolla ei ole käyttöliittymää vaan se tarjoaa keinot suorittaa tietokantahakuja /21 sivu 36 1-2.

Appserver pitää sisällään kaksi komponenttia. Appserver agentti on prosessi, joka suorittaa pyydyt ABL koodit ja palauttaa vastauksen. Appserver broker on prosessi, joka hoitaa asiakassovelluksen ja appserverin yhteyksiä ja kutsuu appserver agenttia ajamaan kutsuttua tietokantakoodia. Yksi appserver ilmentymä pitää sisällään yhden appserver brokerin ja appserver asetuksista riippuen lukuisia appserver agenteja /15 sivu 37 2-5.

Appserverillä on useita tiloja joihin se voidaan asettaa käynnistyksessä tiedonvälitystä ajatellen. Tila määrää miten Appserver käsittelee asiakasohjelman pyynnöt. Appserver tukee neljää eri tilatyyppeä. Tilasta riippuen tiedonvälitys Appserverin ja asiakassovelluksen välillä mukautuu kahteen eri istuntomalliin. Mallit ovat Session managed ja Session-free /23 sivu 25 1-1.

Session managed:ssa yksi appserveri käsittelee kaikki yhden asiakassovelluksen kutsut, ja asiakassovellus hallitsee fyysisen yhteyden appserveriin. Session-free:ssä yksi tai useampi appserveri voi käsitellä kaikki yhden asiakassovelluksen kutsut ja asiakassovelluksen ei tarvitse hallita fyysistä yhteyttä /23 sivu 25 1-1.

Appserveriin on saatavissa adaptoreita joilla voi laajentaa appserverin toimintaa. Adapterit ovat Appserver Internet Adapter (AIA), OpenEdge Adapter for Sonic ESB ja Web Services Adapter (WSA) /15 sivu 41 2-9.

Tässä työssä keskityn vain viimeiseen eli WSA adapteriin, koska käytin sitä työssäni web servicen toteutukseen.

Appserverin tilan, käynnistämisen, pysäyttämisen ja sen agenttien tilan tarkastaminen onnistuu komennolla `asbman -i <appserver-ilmentymä> -q`. Esimerkiksi `asbman -i d92600p1 -q`.

```

$ asbman -i d92600p1 -q
OpenEdge Release 10.2B06 as of Mon Mar 19 19:14:48 EDT 2012

Connecting to Progress AdminServer using rmi://localhost:20931/Chimera (8280)
Searching for d92600p1 (8288)
Connecting to d92600p1 (8276)

Broker Name           : d92600p1
Operating Mode       : Stateless
Broker Status        : ACTIVE
Broker Port          : 39260
Broker PID           : 2897
Active Servers       : 10
Busy Servers         : 0
Locked Servers       : 0
Available Servers    : 10
Active Clients (now, peak) : (63, 79)
Client Queue Depth (cur, max) : (0, 14)
Total Requests       : 66259495
Rq Wait (max, avg)   : (15877 ms, 0 ms)
Rq Duration (max, avg) : (20428 ms, 28 ms)

PID  State      Port  nRq   nRcvd  nSent  Started      Last Change
02967 AVAILABLE 02047 6626562 7557080 9546178 Mar 21, 2014 12:00 May 4, 2014 13:59
02969 AVAILABLE 02066 6623749 7557342 9544955 Mar 21, 2014 12:00 May 4, 2014 13:59
02971 AVAILABLE 02153 6625214 7557089 9543298 Mar 21, 2014 12:00 May 4, 2014 13:59
02973 AVAILABLE 02154 6623421 7551821 9532168 Mar 21, 2014 12:00 May 4, 2014 13:59
02975 AVAILABLE 02155 6627732 7558283 9542099 Mar 21, 2014 12:00 May 4, 2014 13:59
02977 AVAILABLE 02156 6626367 7558028 9544464 Mar 21, 2014 12:00 May 4, 2014 13:59
02979 AVAILABLE 02157 6628330 7558574 9541729 Mar 21, 2014 12:00 May 4, 2014 13:59
02982 AVAILABLE 02158 6629007 7560118 9545720 Mar 21, 2014 12:00 May 4, 2014 13:59
02984 AVAILABLE 02179 6627330 7558895 9546781 Mar 21, 2014 12:00 May 4, 2014 13:59
02986 AVAILABLE 02186 6625214 7555574 9538614 Mar 21, 2014 12:00 May 4, 2014 13:59

```

Kuva 2. Appserverin tilatiedot.

Kuvasta näkyy, että d92600p1 niminen appserver on käynnissä ja valmis suorittamaan tietokantakyselyitä. Appserverin tila on stateless, ja sillä on kymmenen agenttia vapaana. Asbman näyttää myös muuta tilastotietoa käytetyistä yhteyksistä ja niiden suoritusajoista.

#### 2.9.4 Progress Web service adapter (WSA)

Progress web service adapter (WSA) on java servletti joka toteuttaa ja mahdollistaa web servicen käytön Progressissa käyttäen Web Service Description Languagea. WSA tarvitsee toimiakseen java servlet säiliön, jonka päällä java servlet toimii. Java servletti toimii asiakasohjelman ja progressin appserverin välissä ja kommunikoi asiakasohjelman ja appserverin välillä /21 sivut 121 5-1 /21 sivut 38 1-4.

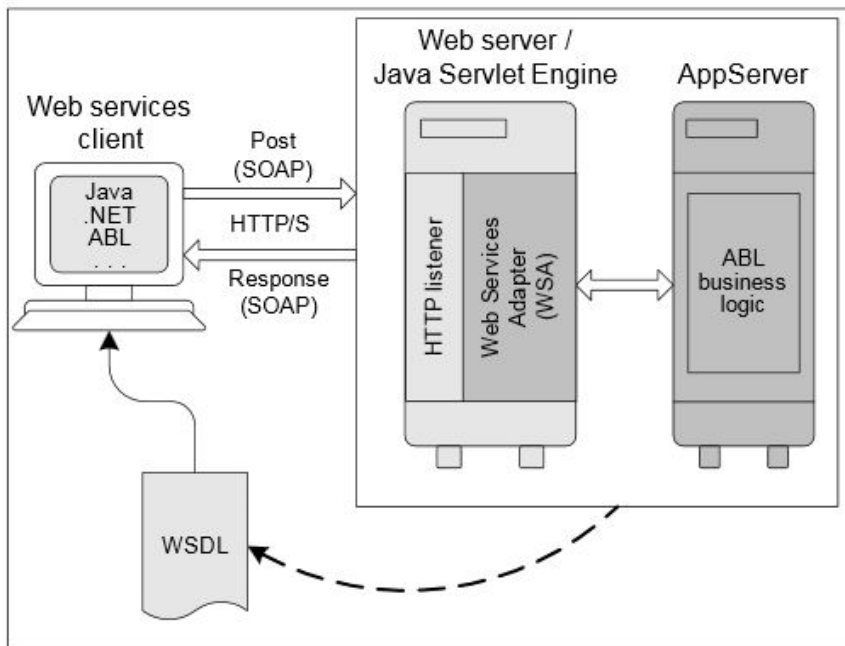


Figure 1-1: OpenEdge Web services architecture

Kuva 3. Web service arkkitehtuuri.

Kuvassa näkyy Progress web service arkkitehtuuri. Progress web service koostuu Java moottorista ja appserverista jotka keskustelevat asiakasohjelman välillä käyttäen SOAP protokollaa ja HTTP tiedonsiirtoa /47 sivu 37 1-7.

### 2.9.5 Proxy generator toolkit

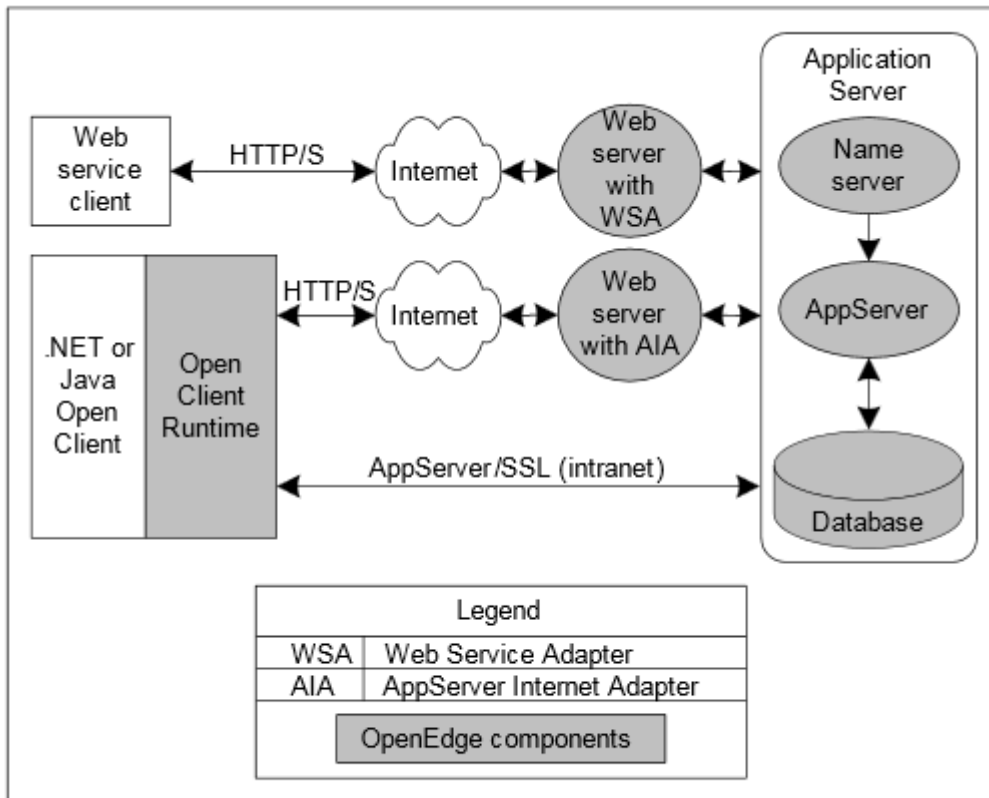
Proxygen on Progressin kehitystyökalu, jolla voidaan mahdollistaa muiden kuin Progressin omien ohjelmistojen tietokantahaut. Proxy generator ohjelma tulee Progressin asennuksen yhteydessä.

Kehitystyökalu tukee useita eri ohjelmointikieliä ja niillä tehtyjä ohjelmistoja. Tuettuja kieliä ovat mm. Java, .NET ja Web services. /31 sivu 19 1-1.

Proxygenistä on olemassa kahta eri versiota toinen Windowsille ja toinen Unixille. Windowsille kehityssä proxygenissä on graafinen käyttöliittymä, ja sillä voi luoda proxytiedostoja kaikille tuetuille alustoille Java, .NET ja Web Services. Unix versio on riisutumpi, eikä pidä sisällään graafista käyttöliittymää, vaan toimii komentoriviltä ja on siksi oivallinen työkalu skripteissä. Unix versiolla voi luoda proxytiedostoja vain Java ja Web Services alustoille. Unix versio ei myöskään salli

määrittysten muutoksia eikä määrittysten tallennusta myöhempää käyttöä varten /31 sivut 39–40 3-5 3-6.

Proxygen on tehokas työkalu, jossa on monia ominaisuuksia, mutta tulen keskittymään jatkossa vain oleellisiin ominaisuuksiin web services proxytiedostojen teon kannalta.



**Figure 1–1: Open Client architecture**

Kuva 4. Progressin peruskomponentit.

Kuvassa näkyy kaikki peruskomponentit jotka tarvitaan onnistuneeseen tietokantahakujen kutsuun muista kuin Progressin omista ohjelmistoista /31 sivu 20 1-2.

## 2.9.6 Hallinnointikomponentit

Progress tarjoaa useita hallinnointikomponentteja joilla voidaan hallita ja tarkkailla pääkomponenttien tilaa.

### 2.9.6.1 Wsaman

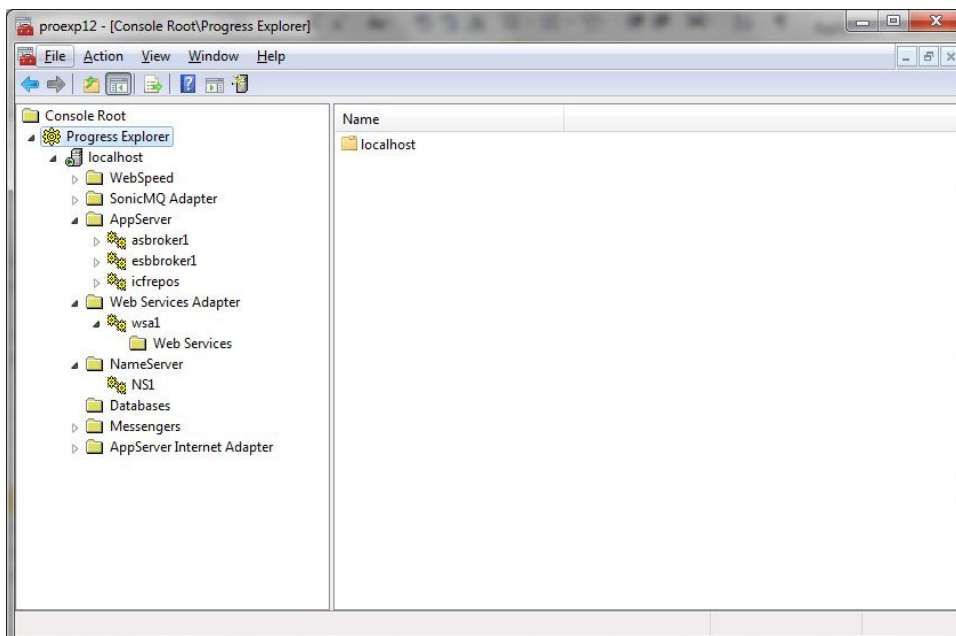
Wsaman on komentorivityökalu jolla voidaan hallinnoida ja tutkia web service adapterin tilaa. Työkalulla voidaan tehdä monia samoja asioita kuin Progress Explorer toolilla. Wsaman työkalu on oivallinen työkalu skripteissä. Työkalulla voidaan mm. tarkistaa web servicen tila, käynnistää web service, julkaista web service tai tarkastella web servicen статистиikkaa /21 sivu 176 8-2.

Käytin työssäni wsaman työkalua käyttöä helpottavissa skripteissä.

### 2.9.6.2 Progress Explorer Tool

Progress Explorer Tool on Windowsille kehitetty visuaalinen työkalu Progressin komponenttien hallintaan. Työkalulla voidaan hallinnoida ja seurata Progressin komponentteja mm. tietokantaa, appserveriä ja web serviceä /15 sivu 45 2-13.

Explorer Toolilla tehdyt asetusten muutokset tallentuvat ubroker.properties tiedostoon, joka sijaitsee Progressin asennuskansion alla olevassa properties kansiossa. Tiedostoa voi muuttaa tarvittaessa myös käsin käyttäen normaalia tekstieditoria.



Kuva 5. Progress Explorer Tool.

Kuvassa Progress Explorer Toolin pääikkuna. Ikkunassa näkyy kaikki Progressin tarjoamat komponentit. Ylimpänä näkyy palvelu johon Progress Explorer Toolilla on yhteys. Tässä tapauksessa Explorer Tool on yhdistetty paikalliseen koneeseen.

### 2.9.6.3 bprowsddoc

Bprowsddoc on progressin tarjoama WSDL analysointityökalu, jolla voidaan luoda suoraan WSDL tiedosta HTML -pohjainen dokumentaatio, jolla web serviceä voidaan hyödyntää suoraan Progressin omissa tietokannan koodeissa /29.

## 2.10 SOAP

SOAP on lyhenne sanoista The Simple Object Access Protocol. SOAP on kevyt ja pohjautuu XML kieleen, ja se on suunniteltu vaihtamaan tietoa hajautetuissa järjestelmissä /37 sivu 10. SOAP viestit voidaan välittää normaalin HTTP liikenteen mukana, mikä tekee siitä hyvän tiedonvälityskanavan. SOAP liikenne välitetään HTTP liikenteen mukana jolloin SOAP viestit läpäisevät myös palomuurit mikäli normaali HTTP portti 80 on päästetty läpi.

SOAP viesti on XML dokumentti nimeltään envelope, joka pitää sisällään SOAP viestin. SOAP envelope:n sisältö on määritelty SOAP spesifikaatiossa. SOAP sallii viestin lähettäjän ja vastaanottajan viestiä keskenään ohjelmointikieliriippumattomasti. Esimerkiksi lähettäjä voi kirjoittaa oman viestinsä Pythonilla ja viestin vastaanottaja voi käsitellä viestin Javalla. Kummankaan osapuolen ei tarvitse välittää, miten viesti on kirjoitettu tai vastaanotettu, sillä viestin rakenne on tarkasti määritelty SOAP spesifikaatiossa, jota kumpikin osapuoli noudattaa /37 sivu 14.

SOAP viestin pääelementti on envelope, joka pitää löytyä viestistä, jotta viesti on laillinen ja hyväksyttävä. Envelope elementin alla voi esiintyä header elementti, mutta se ei ole pakollinen. Header elementin ollessa viestissä sen pitää olla heti envelope elementin alla. Envelope elementin pitää sisältää body elementti. Jos envelope pitää sisällään header elementin body elementti pitää olla heti header elementin jälkeen, muussa tapauksessa body elementin pitää olla heti envelope elementin jälkeen /37 sivu 18.

### 2.10.1 Nimiavaruudet

XML nimiavaruuksien ymmärtäminen on tärkeää, SOAP:n ymmärtämisen kannalta. Periaatteessa nimiavaruudet on keino välttää sekaannus XML elementtien ja attribuuttien välillä. Nimiavaruudet helpottavat ymmärtämään sisällön tai elementin tarkoituksen. Nimiavaruudet ovat vastaavaa kuin muuttujat ohjelmointikielessä /37 sivu 18.

Tyypillinen SOAP nimiavaruusmäärittäminen on muotoa `xmlns:id="URI"`. Määrittäminen XMLNs kertoo XML käsittelijälle, että olemme määrittelemässä nimiavaruutta. Id on määritetty tunniste. Viimeinen osa URI määrittää nimiavaruuden sisällön. Usein URI on URL, mutta on tärkeä tietää että se on vain tunniste. Viestin vastaanottaja ei yritä ladata mitään URI:sta eikä sitä yleensä edes ole olemassa. Nimiavaruustunniste on useasti tarpeellinen määrittelemään elementtien tai attribuuttien tarkoituksen /37 sivut 18 ja 19.

### 2.10.2 Header

Header elementti on valinnainen. Header elementtiä käyttäessä elementin nimi pitää olla header ja se on ensimmäinen alielementti envelope elementissä. Yleisesti header elementissä välitetään transaktiotietoja, tunnistuksia tai muuta viestin tarvitsemaa tietoa. Header elementtiä tarvitaan, sillä body elementti pitää sisällään vain viestin sisällön eikä sinne kirjoiteta muuta tietoa /37 sivu 21.

### 2.10.3 Body

Toisin kuin header elementti, body elementti on pakollinen. Mikäli viestissä ei ole header elementtiä, body elementti on ensimmäinen alielementti envelope elementissä. Muussa tapauksessa body elementti pitää olla heti header elementin jälkeen. Body elementti pitää sisällään varsinaisen viestin, yleensä pyynnön ja vastauksen /37 sivu 25.

### 2.10.4 Fault

Fault elementti on valinnainen lapsielementti. Elementtiä käytetään välittämään viestin virhetietoja. Fault elementti on heti body elementin lapsielementtinä ja se voi esiintyä vain kerran. Fault elementin tarkoitus on välittää lähettäjälle tieto tapahtuneesta viestin käsittelyvirheestä ja mahdollisuus toipua virheestä tai käsitellä se ja tehdä tarvittavat korjaukset /37 sivu 25. Fault elementti pitää sisällään elementin Faultcode. Faultcode on pakollinen ja kertoo virheen laadun. Faultcode



elementin arvoja on neljä ja ne kaikki on määritetty SOAP nimiavaruudessa. Elementin arvot voivat olla: VersionMismatch, MustUnderstand, Client ja Server. Virhekoodeja on mahdollista laajentaa vaadittaessa tarkempaa tietoa virheestä /37 sivu 26.

```
<?xml version="1.0"?>
<soap:Envelope
XMLNs:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
  ...
  </soap:Header>
  <soap:Body>
  ...
  </soap:Body>

</soap:Envelope>
```

Kuva yksinkertaisen SOAP viestin rungosta /30.

## 2.11 Web service Description Language (WSDL)

Web Service Description Language (WSDL) on W3C:n määrittämä XML-perustainen kieli, jolla kuvataan internetissä tarjolla oleva web-teknologioihin perustuva palvelu, eli Web Service. Kuvaus-tieto koostuu abstraktista ja konkreettisesta osasta. Abstrakti kuvaus määrittelee palvelun rajapinnan sitomatta palvelua mihinkään tiettyyn osoitteeseen tai käytettävään yhteyskäytäntöön /38 Sivun 2. WSDL paras vahvuus on se, että se on hyväksytty standardi web servicen kuvaukseen /37 sivu 170, 171. WSDL määritelmästä on käytössä kaksi eri versiota. WSDL 1.1 ja WSDL 2.0.

Työssäni käytin WSDL versiota 1.1 joten keskityn jatkossa vain siihen.

WSDL tiedosto on jaettu seitsemään eri osaelementtiin. Juurielementti Definitions jonka alla on seitsemän pääelementtiä Types, Message, PortType, Operation, Binding, Service ja Port.

Types, määrittelee palvelussa käytetyt tietotyypit ja niiden määritykset.

Message, määrittelee abstraktit viestit jotka tarvitaan operaatioihin. Operaatioiden tietotyypit on kerrottu types elementissä.

PortType, määrittelee abstraktit operaatiot jotka yksi tai useampi määritetty palvelu tarjoaa.

Binding, määrittelee konkreettisen protokollan jolla viestit välitetään.

Operation, määrittelee abstraktin operaation palvelun ja asiakkaan välillä.

Service, määrittelee miten palveluun saadaan yhteys.

Port, määrittelee palvelun verkko-osoitteen.

/37 sivu 172

## 2.12 Python ja reportlab – kirjasto

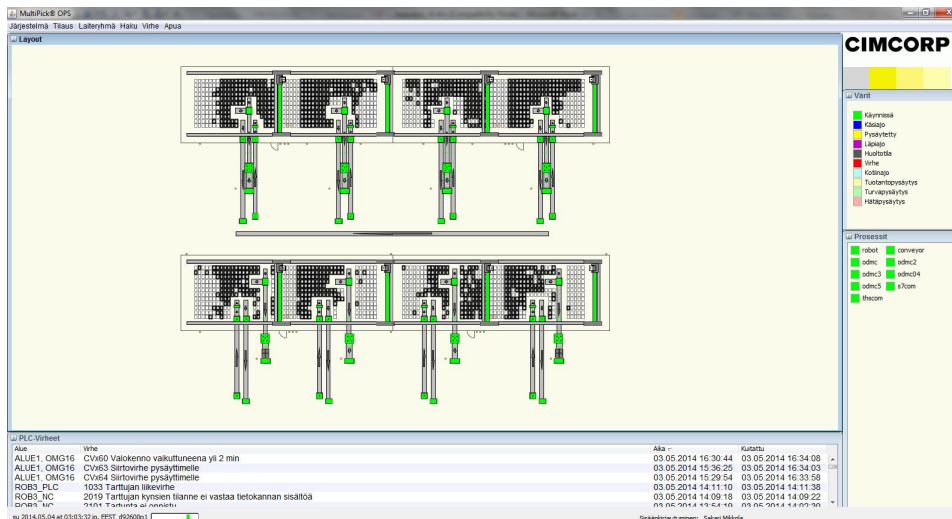
Python on monipuolinen ohjelmointikieli. Python ohjelmointikieli on helppo oppia, ja sitä käytetään laajalti /42. Guido von Rossum loi Pythonin vuonna 1990 ja se nimettiin komediasarja Monty Pythonin lentävä sirkuksen mukaan. Python on kehitetty laajalti vapaaehtoisin voimin, ja se on vapaasti saatavissa Python Software Foundationista. /40.

Reportlab-kirjasto on Pythonin laajennus, joka on pääasiassa suunniteltu PDF raporttien luontiin. Reportlab on avoimen lähdekoodin projekti. Se on kirjoitettu Python ohjelmointikielellä. Perusversio reportlabista on ilmainen ja perustuu avoimeen lähdekoodin. Reportlabista on myös olemassa maksullinen versio, joka tuo lisää ominaisuuksia /41. Reportlab on nopea, joustava ja laitteistoalusta riippumaton ratkaisu luoda PDF tiedostoja.

### 3. TARKOITUS

Tässä luvussa käydään läpi tarkoitusta miksi työ tehtiin, mitä hyötyjä siitä saadaan ja miten se palvelee loppukäyttäjää.

Cimcorpin liiketoiminta koostuu pääosin robottijärjestelmien myynnistä ja toimittamisesta asiakkaille. Robottijärjestelmä pitää sisällään tyypillisesti robottivaraston ja siihen liittyvät robotit ja kuljettimet sekä ohjelmistot niiden käyttämiseen. Cimcorpin kehittämä ja käyttämä järjestelmä robottien hallinnointiin on nimeltä WCS.



Kuva 6. WCS käyttöliittymä.

Kuva WCS käyttöliittymästä jolla hallinnoidaan robotteja ja muita laitteita.

Jokaisesta myydystä järjestelmästä luodaan oma projekti. Toimitetun järjestelmäkokonaisuuden tarvittavat käyttöohjeet ja dokumentit voivat erota suuresti eri projekteissa. Tätä eroavaisuutta varten Cimcorp kehitti jokaiselle projektille räätälöidyn web sivun nimeltään WWP. Tämä sivusto pitää mm. sisällään kaiken projektin tarvitsevan dokumentaation. WWP pitää sisällään projektin käyttöohjeet, järjestelmän laitteet, raportit ja статистиikan. Asiakkaan on helppo hakea tarvittava tieto yhdestä paikasta. Raporttien tai käyttöohjeiden päivitys on helppoa. WWP sivujen teko ei ole tämän työn aiheena, vaan se kuului toisten osapuolten tehtäväksi. Työssä toteutetut raportit tulivat WWP

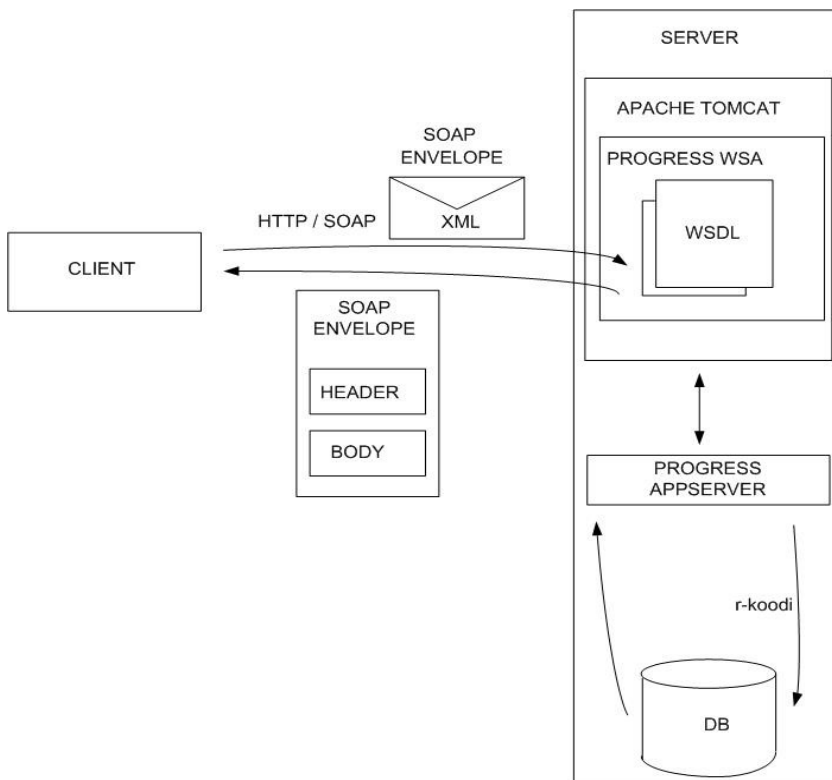
sivujen sisään upotettuna web-sivuna. Jatkossa keskityn raporttien tekoon ja siihen, miten ne näkyvät käyttäjälle.

Aikaisemmin raporttien näyttäminen oli sidottu WCS käyttöliittymään. Raportit luotiin suoraan WCS:stä josta käyttäjän piti valita raportin esitysmuoto. Valittavat esitysmuodot olivat CSV tai PDF tiedosto. Valinnan jälkeen käyttöliittymä suoritti Progress tietokantaan kantanäkyselyn, jolla luotiin XML tiedosto palvelimelle. XML tiedoston luonnin jälkeen kutsuttiin Pythonilla rakennettua toteutusta, jolla luettiin XML tiedoston sisältö ja luotiin PDF tiedosto palvelimelle. Lopuksi vielä haettiin luotu PDF tiedosto palvelimelta WCS käyttöliittymä koneelle käyttäen SSH yhteyttä ja näytettiin käyttäjälle. Tällä tavalla toteutettu raporttien luonti ja tarkastelu oli vain pienen kohderyhmän saatavilla. Tekniikkaan jolla raportit luotiin liittyi myös monta ongelmaa, joista yksi ja suurin oli, että raporttien syntyminen oli epävarmaa. Raportit luotiin käyttäen monta eri tekniikkaa. Jonkun kohdan epäonnistuessa raporttia ei syntynyt ja käyttäjä ei tiennyt miksi näin tapahtui.

Työn tarkoitus oli selkeyttää raporttien luontia, esittämistä ja tuoda raportit laajemman kohderyhmän saataville. Tämä onnistui muuttamalla raporttien esittäminen ja luonti kokonaan palvelimelle. Kaikki toimenpiteet tapahtuvat palvelimella. Lisäksi raporttien luontiin käytettävät työvaiheet vähentyivät ja mahdolliset virhetilanteet saatiin paremmin hallintaan.

## 4. TOTEUTUS

Tässä luvussa käyn läpi raportointijärjestelmän toteutusta ja siihen liittyviä asioita. Luku on jaettu kolmeen päälukuun joista jokainen käsittelee tiettyä aluetta työstä. Ensimmäinen pääluku käsittelee käyttöliittymää ja toinen palvelinpuolen ohjelmistoa. Kolmas luku käsittelee työn upottamista erilaisiin projekteihin, ja kertoo miten sitä on helpotettu tekemällä erinäisiä automaattisia skriptejä ja työkaluja, joilla toteutusta voidaan helposti laajentaa eri projekteissa.



Kuva 7. Työn osa-alueet.

Kuvassa näkyy työn kaikki osa-alueet ja miten käytetyt osat liittyvät toisiinsa.

Asiakasohjelma raporttien näyttämiseen asiakkaalle on toteutettu käyttäen HTML ja PHP ohjelmointikieltä, ja se on rakenteeltaan tavallinen web sivu. Asiakasohjelmasta käsin tehdään tietokantakyselyitä Progress tietokantaan. Kyselyt Progress tietokannan ja PHP asiakassovelluksen välillä on toteutettu käyttäen web service rajapintaa, ja rajapinta on toteutettu käyttäen WSDL

määrittystä. Web service kutsut asiakasohjelman ja Progress tietokannan välillä koostuu XML tiedostoista joiden sisältö on tarkkaan määritetty SOAP protokollassa. SOAP protokollan viestejä käsittelee Progressin päässä Progressin tarjoama Web service adapter, joka kutsuu Progressin Appserver prosessisovellusta ja suorittaa tietokantakyselyt ja palauttaa kyselyn paluuarvot Web service adapterille ja sieltä takaisin asiakasovellukselle. Progressin web service adapteri on tavallinen java servlet – sovellus, joka pyörii palvelimella Apache Tomcat ohjelmiston päällä.

#### 4.1 Käyttöliittymä

Tässä luvussa käyn läpi mitä vaadittiin, jotta työn käyttöliittymä saatiin toteutettua sekä palvelimella olevan tiedostorakenteen ja eri tiedostojen ja kansioden sisällöt. Työn tarvitsemat tiedostot ja kirjastot on tallennettu palvelimelle tiettyyn hakemistorakenteeseen.

Selainpohjaisia raportteja suoritetaan palvelimessa, johon on asennettu Linux käyttöjärjestelmä, Apachen HTTP palvelimenohjelma ja siihen PHP laajennus. Web-sivujen päähakemisto sijaitsee palvelimella polussa /var/www/html/<projektinnumero> jonka alta löytyy kaikki web-sivujen tuottamiseen ja näyttämiseen tarvittavat koodit. Päähakemiston ilmoittaminen projektinumerolla mahdollistaa, että projektipalvelin voisi pitää sisällään enemmän kuin yhden WWP sivun. Kaikki raporttien tarvitsemat tiedostot ja kirjastot on palvelimella tallennettu päähakemiston alle WS kansioon. WS kansion alta löytyy PHP – tiedostoja jotka tarvitaan raporttien näyttämiseen.

Käyn seuraavaksi lyhyesti läpi kansioden sisällöt ja tarkoituksen. WS – kansion alta löytyvä 4gl kansio pitää sisällään raporttien tarvitsemat Progressin tietokantakyselykoodit. Js – kansion alta löytyy JavaScript koodit joita työssä tarvittiin. Lang - kansion alta löytyy raporttien tarvitsemat kielitiedostot. Lib kansio pitää sisällään raporttien tarvitsemia kirjastoja esimerkiksi kirjastot ezPDF:n sekä nuSOAP käyttöön. Web service palvelun luomat XML tiedostot tallentuvat XML kansioon ja CSV – tiedostot CSV kansioon.

Kansioden lisäksi WS kansion juuresta löytyy useita tiedostoja. Conf.php pitää sisällään mm. tietokantayhteyden luomiseen tarvitsemia tietoja ja ses.php pitää sisällään mm. vakioita joilla voidaan säädellä muutamia lisäominaisuuksia, jotka näkyvät käyttäjälle sekä raporttien kehityksessä helpottavaa debug tilan arvoa. Mikäli debug tila on aktivoitu web service säilyttää luodut XML tiedostot kutsun jälkeen. Tämä on erityisen tärkeää kehitysvaiheessa ja helpottaa vian etsintää.

Raportointijärjestelmän käyttöliittymä on selain-pohjainen. Tällä on pyritty siihen, että raportointijärjestelmää voidaan käyttää mahdollisimman monesta paikasta ja käyttö ei vaadi erilaisten ohjelmien asennusta. Käyttöliittymä on toteutettu käyttäen PHP ja HTML -kieltä. Raportit esitetään käyttäjälle taulukkomuodossa ja PDF tiedostona. Taulukkoraportin voi tallentaa tarvittaessa CSV tai PDF muodossa.

Käyttäjälle näytettäviä raportteja on kolmea erilaista tyyppiä, ja ne tulevat käyttöön jokaisessa Cimcorpin projektissa. Raportit ovat varaston karttaraportti, robottien sisältöraportti ja virheraportti.

Varaston karttaraportti näyttää halutun robotin varaston sisällön käyttäjälle. Raportista näkee, missä kohtaa varastoa mitäkin tuotetta on ja kuinka paljon. Varastokartta näkyy käyttäjälle PDF tiedostona, johon piirretään robotin jokaisen varastopaikan kuva ja siinä olevan tuotteen nimi ja kappalemäärä. Raportti olisi voinut myös olla pelkkä kuva, mutta PDF tiedosto valittiin paremmaksi vaihtoehdoksi, koska siitä voi kopioida tekstiä sitä tarvittaessa. PDF:ssä on myös mahdollista zoomata tiedostoa jolloin saa paremman kuvan halutusta alueesta.

Robottien sisältöraportti on toteutettu taulukkomuotoisena ja näyttää varaston tuotteet ja niiden määrät eri varastoissa. Raportin tarkoitus on näyttää käyttäjälle tämän hetkinen tilanne varastosta. Jokaisen robotin tuotemäärät näytetään käyttäjälle taulukon sarakkeissa. Raportin taulukon yläotsikko koostuu sarakkeista joissa on listattu tuotenimi, robotit ja viimeisenä on sarake, joka kertoo tuotteen kokonaismäärän kaikilta roboteilta. Varsinainen taulukon sisältö on listattu alla otsikon sisällön mukaan. Taulukon viimeisellä rivillä on kerrottu tuotteiden kokonaismäärät robotti-kohtaisesti. Raportti on mahdollista tallentaa CSV tai PDF muotoon napin painalluksella.

Virheraportissa käyttäjä voi valita yhden päivän tai vaihtoehtoisesti ajanjakson, jonka ajalta virheet halutaan nähdä. Raportin tarkoitus on näyttää käyttäjälle järjestelmän virheet valitulta ajanjaksolta. Raportti on toteutettu samalla tavalla taulukkomuotoisena kuin sisältöraporttikin. Raportin taulukon yläotsikko koostuu sarakkeista joissa on listattu aika jolloin virhe tapahtui, virhenumero, virheteksti ja mahdollinen ratkaisuehdotus jolla virheestä voidaan selvittää. Taulukon sisällössä näkyy järjestelmän virheet rivi riviltä. Raportti on mahdollista tallentaa CSV tai PDF muotoon napin painalluksella.

#### 4.1.1 Varaston karttaraportti

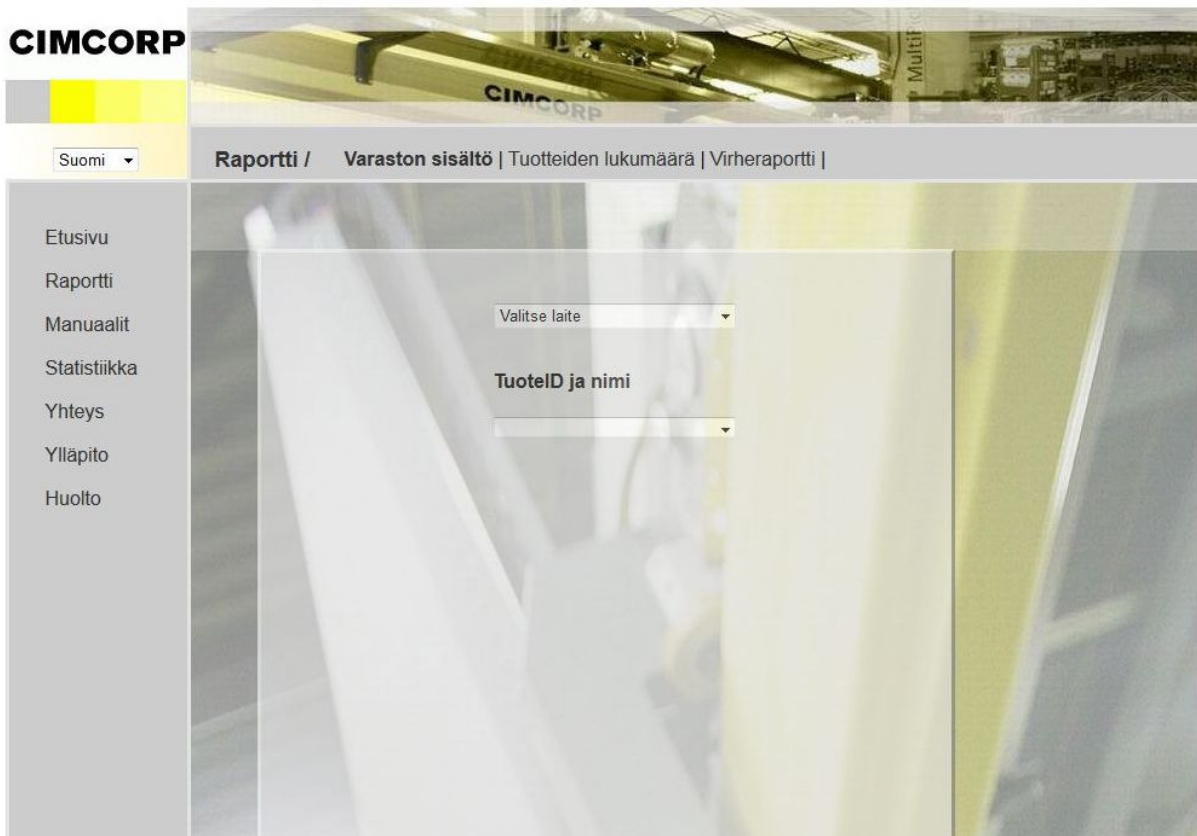
Tässä luvussa käyn lähemmin varastokartan tekemistä ja tekemiseen kuuluvia työvaiheita ennen kuin raportti oli toimintakunnossa ja valmis käyttäjän käytettäväksi.

Varaston kartta on erilainen raportti kuin raporttien sisältöraportti ja virheraportti. Varaston karttaraportissa luodaan suoraan PDF tiedosto tietokannan palauttamista tiedoista. Kartan toteuttamista varten piti määritellä raportin ulkoasu. Pienen mietinnän jälkeen uusien raportin ulkoasu kopioitiin pääpiirteissään vanhoista raporteista.

Vanhat raportit oli toteutettu käyttäen HTML kieltä, Pythonia ja Pythonin reportlab laajennusta, joka mahdollisti PDF tiedoston luomisen Python koodilla. Pythonista haluttiin eroon, jolloin ei myöskään ollut mahdollista käyttää reportlabia ja piti keksiä keino jolla saataisiin suoraan luotua PDF tiedostoja PHP:stä.

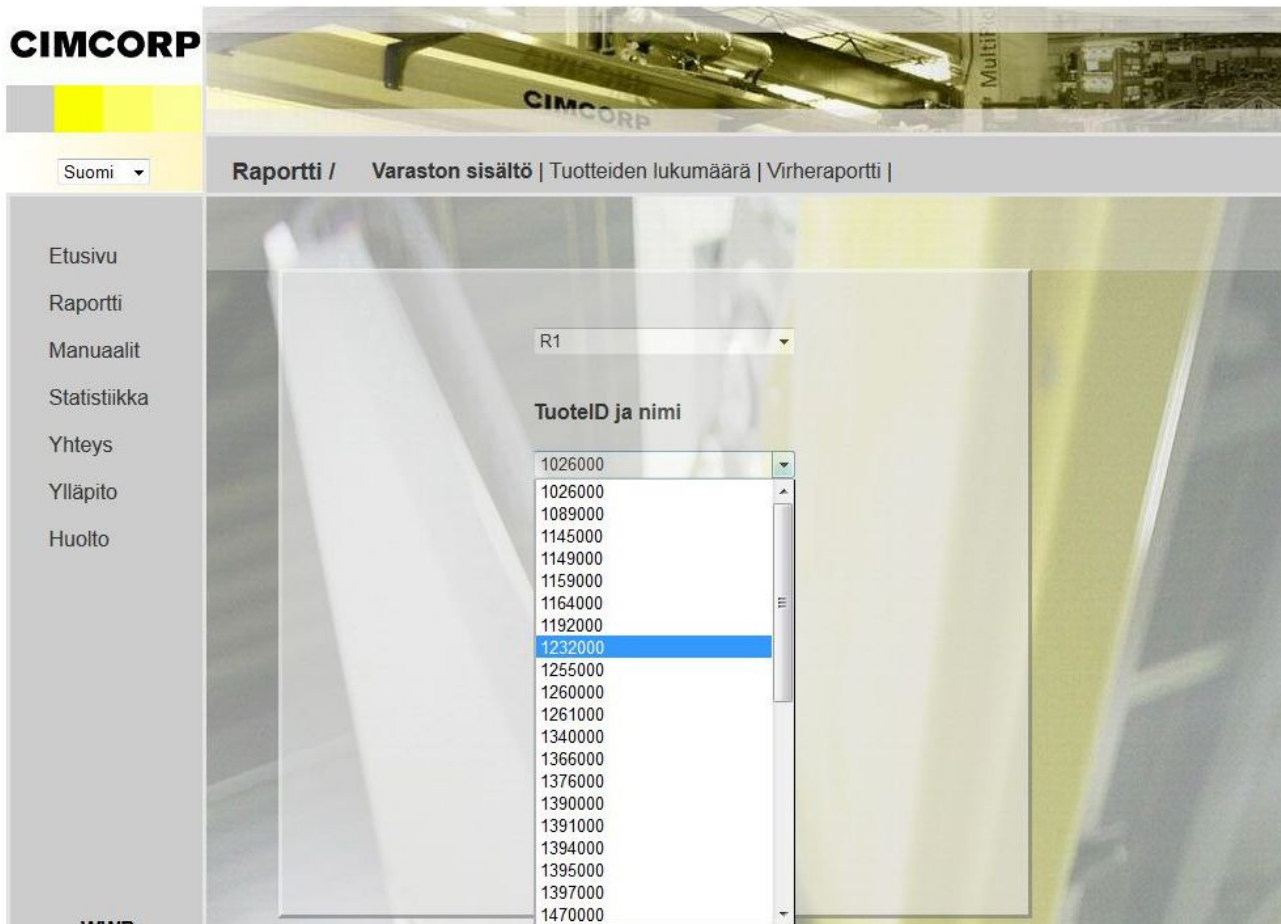
Tämä ongelma ratkesi käyttäen ezPDF laajennusta jota voidaan kutsua suoraan PHP koodista. Mahdollisista lisäosien asentamista ei enää tarvittu, sillä ezPDF on avoimen lähdekoodin projekti, joka mahdollistaa laajennuksen suoran käytön ilman laajennuksen asentamista. Laajennuksen käyttökuntoon saamiseksi riittää, että lataa tarvittavat PHP-tiedostot ja sisällyttää ne PHP koodiin, jolloin voidaan suoraan kutsua PDF luontiin tarvittavia funktioita.





Kuva 8. Varastokartta.

Varastokartta koostuu kolmesta tiedostosta joita kutsutaan tietystä järjestyksessä, jotta kartta saataisiin näytettyä käyttäjälle. Aluksi käyttäjä valitsee varaston sisältö linkin web sivulta, joka kutsuu storagemap.php sivua ja hakee käyttäjälle kaikki järjestelmän robotit listattuna alasvetovalikossa.



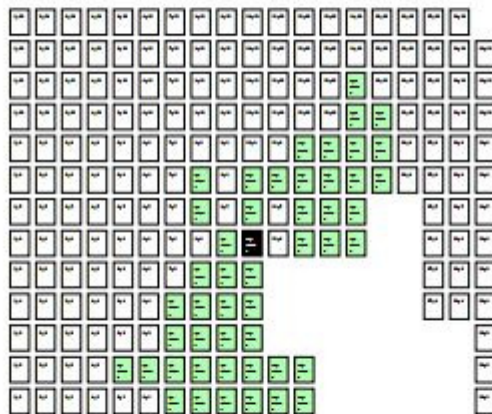
Kuva 9. Varastokartan tuotelistaus.

Tämän jälkeen käyttäjä valitsee halutun robotin ja sivu listaa alle kaikki varastossa olevat tuotteet joista käyttäjä voi halutessa valita tuotteen. Tuotteen valinta ei ole pakollista, mutta valittu tuote näytetään kartassa eri värillä.

Valintojen jälkeen käyttäjä painaa sivun alaosassa olevaa OK painiketta, ja järjestelmä kutsuu web service rajapinnan kautta `WS_getstoragemap_php.p` nimistä tietokantakyselykoodia ja suorittaa tietokantakyselyn. Tietokantakyselykoodi luo XML tiedoston, joka sisältää tarvittavan tiedon. Tämän jälkeen järjestelmä kutsuu `storagemap2.php` koodia ja XML tiedostosta piirretään PDF tiedosto `ezPDF` laajennusta hyväksi käyttäen.

Varaston sisältö  
Robotti 1

AV	Työ
AV	Huolto
AV	Ennen kuin otetaan käyttöön
AV	Puhdas kone
AV	Uudelle alustamiselle

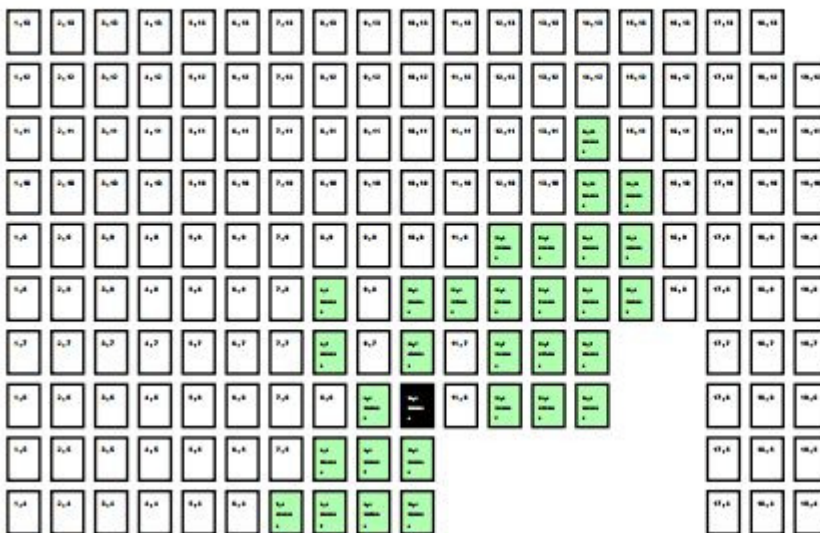


04.05.2014 18:05

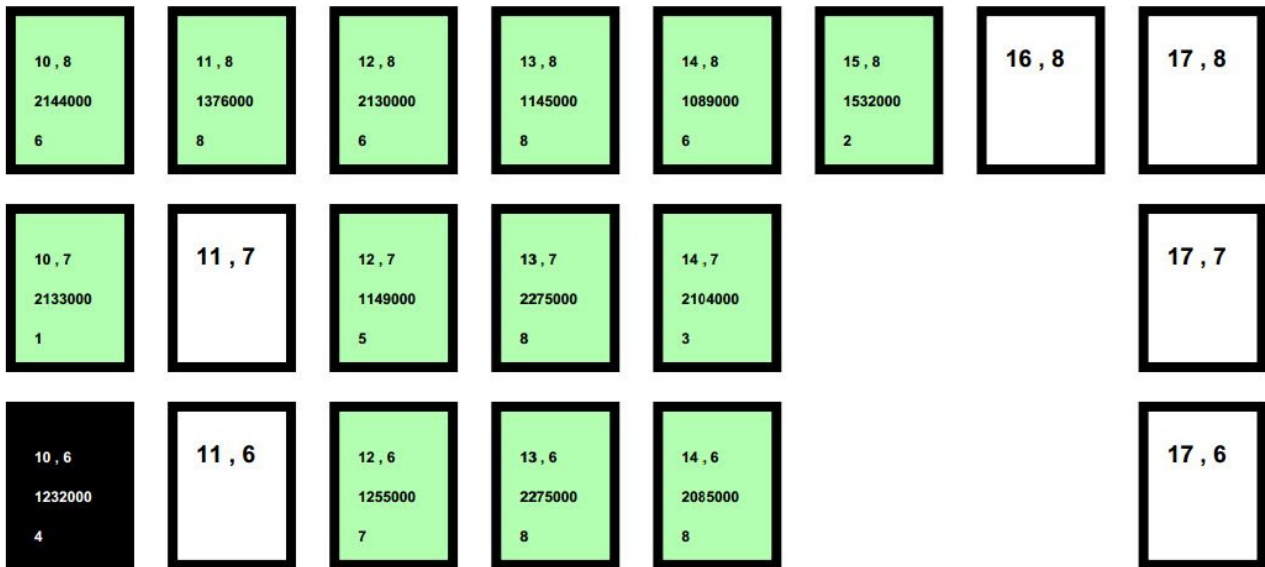
Kuva 10. Varastokartan PDF tulostus.

# Varaston sisältö

## Robotti 1



Kuva 11. Varastokartassa näkyvä valittu tuote.



Kuva 12. Varastokartta varastopaikat.

Kuvissa 10 – 12 näkyy yhden varstorobotin sisältö varastopaikoittain.

Storagemap2.php koodi koostuu kolmesta osasta. Ensin luodaan tyhjä PDF tiedosto ja annetaan paperin koko ja tekstin kirjainkoko. Paperikokona käytettiin A4:sta, koska se on helpoin tulostaa millä tahansa tulostimella. Tämän jälkeen alustetaan muuttujat ja asetetaan paperin reunaan jäävä tyhjä tila, jotta tiedoston tulostus sujuisi onnistuneesti ja kaikki tieto mahtuisi yhdelle sivulle. Näiden toimenpiteiden jälkeen luetaan aikaisemmin luodun XML tiedoston sisältö elementti elementiltä ja tallennetaan ne taulukkoon. Tietojen haussa taulukkoon haetaan XML tiedoston suurin ja pienin arvo, jotta piirto PDF tiedostoon saadaan skaalattua oikein. Tämän jälkeen piirretään paikat PDF tiedostoon ja viimeiseksi piirretään yläotsikko ja alaotsikko, johon tallennetaan raportin luonnin aikaleima.

#### 4.1.2 Robottien sisältöraportti

Tässä luvussa käyn lähemmin robottien sisältönäytön tekemistä ja mitä työvaiheita siihen kuului ennen kuin raportti on toimintakunnossa ja valmis käyttäjän käytettäväksi.

Varaston sisältöraportti koostuu yhdestä tiedostosta, joka ladataan ja näytetään käyttäjälle sivua kutsuttaessa. Aluksi sivu on tyhjä ja se ei pidä sisällään muuta kuin sivun otsikon linkin CSS tiedostoon. Sivua ladattaessa haetaan tiedot Progress tietokannasta käyttäen web service rajapintaa. Sivulla kutsutaan funktiota checkFile, joka löytyy functions.php tiedostosta. Funktio tekee kaiken tarvittavan. Funktion sisään syötetään lista parametreja joiden tuloksena syntyy HTML taulukko tai CSV-tiedosto tietokannan sisällöstä.

```
checkFile(webservice('WS_printstoragecont_php', array('sessionid'=>$sesid, 'lang'=>$lang,
'productnum'=>$productnum), $debug), "XML/" . $sesid . "storage.xml", "html", 0, $debug, 0, "" ,
"" , "storage");
```

Parametrit järjestyksessä:

1. webservice('WS\_printstoragecont\_php', array('sessionid'=>\$sesid, 'lang'=>\$lang,
'productnum'=>\$productnum), \$debug)

Webservice -funktio käyttää suoraan nuSOAP PHP laajennuksen valmiita funktioita, joiden avulla kutsutaan Progressin tarjoamaa web-service rajapintaa ja suoritetaan tietokantakyselyitä. Ensiksi funktiossa määritellään tietokannan yhteystiedot joihin kuuluu, tietokannan käyttäjänimi, salasana ja appserverin osoite sekä portti ja kutsuttavan tietokannan nimi. Nämä tiedot haetaan conf.php tiedostosta. Tämän jälkeen luodaan uusi nuSOAP objekti jolle annetaan WSDL tiedoston polku josta määrittely löytyy. Objektiin luonnin jälkeen kutsutaan Connect\_ws\_reports WSDL tapahtumaa, jolla luodaan yhteys tietokantaan. Paluuarvona saadaan tietokannan käyttämä UUID tunniste, jonka avulla avattu tietokantayhteys voidaan tunnistaa myöhemmin. Yhteyden luonnin jälkeen UUID tunniste tarvitsee liittää tietokantakyselyihin, jotta appserver tunnistaa yhteyden.

Seuraavaksi kutsutaan tietokantakoodia `WS_printstoragecont_php.p`, joka tekee tietokantakyselyn ja luo tietokannan kyselyn tuloksena XML tiedoston josta haluttu tieto löytyy. Tietokantakutsu saa syötteenä kolme parametria, jotka tarvitaan XML tiedoston luonnissa. Ensimmäinen parametri on istuntotunnus, joka haetaan PHP:stä ja tämä yksilöi kunkin selainistunnon. Toinen parametri on valittu kieli, jonka voi muuttaa web -sivun käyttöliittymästä. Kolmas parametri on projektinnumero, joka tarvitaan XML tiedoston tallettamisessa oikeaan hakemistoon. XML tiedoston rakenne pitää olla oikeanlainen, jotta XML voitaisiin lukea PHP päässä ja näyttää käyttäjälle taulukkomuodossa. Lopuksi kutsutaan `Release_ws_reports WSDL` tapahtumaa jolla yhteys tietokantaan suljetaan.

2. `"XML/" . $sesid . "storage.xml"`

Parametri kertoo polun ja tiedostonimen, jonka nimisen XML tiedoston tietokantakysely luo.

3. `"html"`

Parametri kertoo halutun tiedostomuodon. Parametrin arvo on joko `"csv"` tai `"html"` riippuen halutaanko käyttäjälle näyttää kyselyn tulos HTML taulukkomuodossa vai CSV muodossa.

4.0

Parametri ei ole nykyään enää käytössä. Parametrin tarkoitus oli luoda taulukkodatan otsikoista linkit.

5. `$debug`

Parametri kertoo säilytetäänkö XML tiedostot niiden käsittelyn jälkeen vai ei. Tämä on pääasiassa kehitystä helpottava toiminto ja normaalisti debug arvo on 0 joka tarkoittaa, että tiedostoa ei säilytetä kutsun jälkeen. `$debug` muuttujan arvo määritetään `ses.php` tiedostossa.

6.0

Parametri ei ole nykyään enää käytössä.

## 7. ""

Parametri kertoo valitun CSV tiedoston nimen. Käytetään vain jos toiseksi parametriksi on valittu "csv" muutoin muuttujan arvo on tyhjä.

## 8. ""

Parametri kertoo CVS tiedoston tiedon erottelumerkin. Erottelumerkki annetaan tässä parametrissa. Oletuserottimena käytetään ";" merkkiä jos ei muuta merkkiä ole annettu. Käytetään vain jos toiseksi parametriksi on valittu "csv" muutoin muuttujan arvo on tyhjä.

## 9. "storage"

Käytetään yksilöimään kutsun tarkoitus. Tällä hetkellä muuttujan arvo voi olla joko "storage" jos kutsutaan varaston sisältöraporttia tai "error" jos kutsutaan virheraporttia.

## 10.

Valinnainen parametri, jolla voidaan antaa taulukolle class määrittely CSS tiedostoa varten. Arvon ollessa tyhjä käytetään oletusarvoa "scrolltable". Taulukon ulkoasu on määritetty CSS tiedostossa ja ulkoasua voidaan muuttaa tarvittaessa.



**CIMCORP**

Suomi ▾

Raportti / Varaston sisältö | Tuotteiden lukumäärä | Virheraportti |

CSV  
PDF

**Tuotteiden lukumäärä**

SKU ID	THS Tuote	R1	R2	R3	R4	R5	R6	Kpl	kpl	Yksiköintiin	SKU suljettu
1026000		3	8						11	no	no
	40F0023	3	8					11			
1030000							6		6	no	no
	40L4622						6	6			
1087000							5		5	no	no
	40L4422						5	5			
1088000						5			5	no	no
	40L4322					5		5			
1089000		6							6	no	no
	40L4122	6						6			
1090000						5			5	no	no
	40L4222					5		5			
1121000						5			5	no	no
	34L4422					5		5			
1143000			5						5	no	no
	33L4122		5					5			
1144000			4						4	no	no
	33L4222		4					4			
1145000		8		8		6	8		30	no	no

Kuva 13. Varaston tuotteet.

Kuvassa näkyy järjestelmän robottien varaston tuotemäärät ja niiden kokonaismäärät.

### 4.1.3 Virheraportti

Tässä luvussa käyn lähemmin virheraportin tekemistä ja mitä työvaiheita siihen kuului ennen kuin raportti on toimintakunnossa ja valmis käyttäjän käytettäväksi.

Virheraportti koostuu kahdesta tiedostosta. Ensimmäinen tiedosto error.php, jossa on raportin virheiden aikavälin valinta ja client.php jossa luodaan virheet taulukkomuotoon. Aikavälin valinta on toteutettu käyttäen PHP:tä ja linkitystekniikkaa. Päivämäärän valinta olisi voitu toteuttaa myös käyttäen JavaScriptin tarjoamaa valmista funktiota.

Käyttäjä voi valita päivän virheet joko yhdeltä päivältä tai vaihtoehtoisesti pidemmältä ajalta. Päivämäärän valinta tapahtuu valitsemalla haluttu aloituspäivä, jolloin valittu päivä merkataan punaisella värillä. Aloituspäivän valinnan jälkeen käyttäjä valitsee päättymispäivän, jonka jälkeen alku ja loppupäivämäärän välinen aika värjätään punaisella. Jos alkupäivämäärää ei valita, järjestelmä valitsee automaattisesti valintahetken päivän.

Aikavälin valinnan jälkeen painetaan OK nappia ja haetaan virheet valitulta aikaväliltä ja näytetään ne käyttäjälle taulukkomuodossa.

```
checkFile(webservice('WS_printererror_php', array('daStartDate'=>$start, 'daEndDate'=>$end, 'sessionid'=>$sesid, 'sequenceid'=>'HTML', 'lang'=>$lang, 'productnum'=>$productnum),$debug), "XML/" . $sesid . "HTML.xml", "html", 0, $debug, 0, "", "", "error", "scrolltable");
```

Sivua ladattaessa haetaan tiedot Progress tietokannasta käyttäen web service tekniikkaa. Sivulla kutsutaan funktiota checkFile joka löytyy functions.php tiedostosta. Funktio tekee kaiken tarvittavan. Funktion sisään syötetään lista parametreja joiden tuloksena syntyy HTML taulukko tai CSV-tiedosto tietokannan sisällöstä samalla tavalla kuin sisältöraportissa.

**CIMCORP**

Suomi ▾

**Raportti /** Varaston sisältö | Tuotteiden lukumäärä | **Virheraportti** |

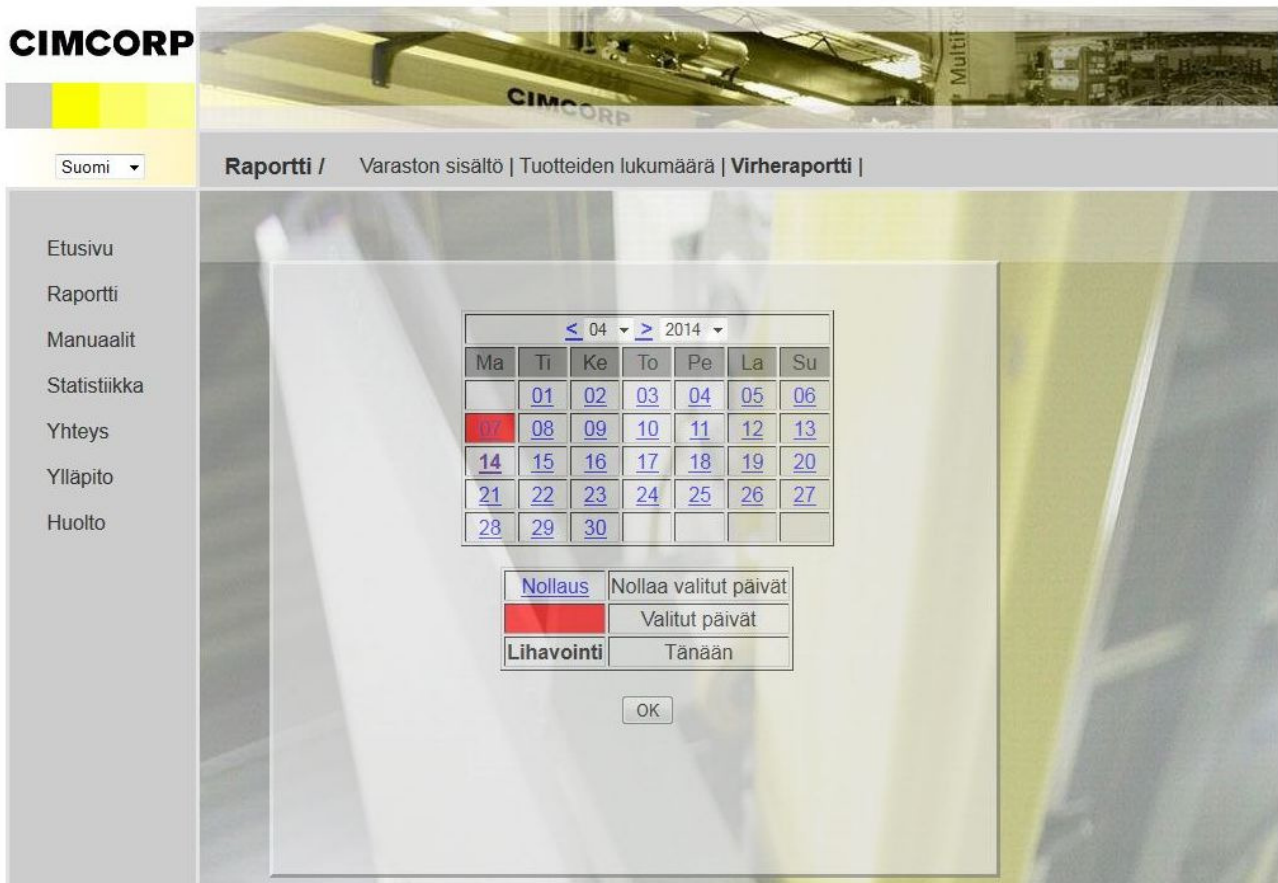
Etusivu  
Raportti  
Manuaalit  
Statistiikka  
Yhteys  
Ylläpito  
Huolto

≤ 04 ▾ ≥ 2014 ▾						
Ma	Ti	Ke	To	Pe	La	Su
	01	02	03	04	05	06
07	08	09	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

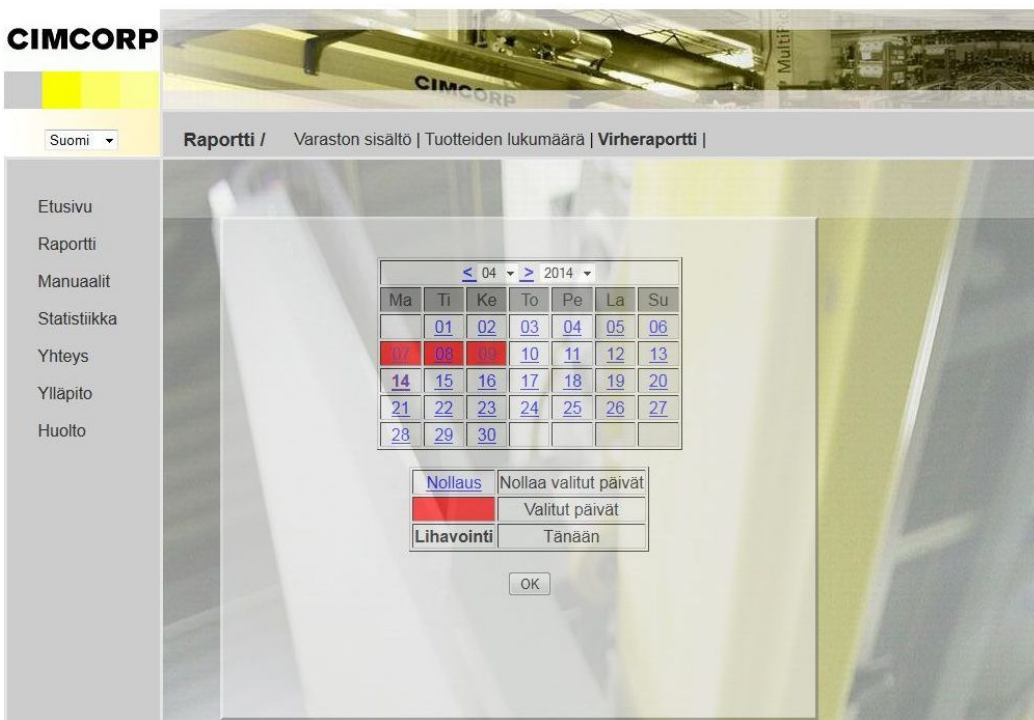
Nollaus	Nollaa valitut päivät
<input checked="" type="checkbox"/>	Valitut päivät
Lihavointi	Tänään

OK

Kuva 14. Virheraportin päivämäärän valinta.



Kuva 15. Päivämäärän valinta.



Kuva 16. Ajanjakson valinta.

**CIMCORP**

Suomi ▾ Raportti / Varaston sisältö | Tuotteiden lukumäärä | Virheraportti |

Etusivu  
Raportti  
Manuaalit  
Statistiikka  
Yhteys  
Ylläpito  
Huolto

CSV  
PDF

**Järjestelmän virheet 07/04/2014 00:00:00,000 - 10/04/2014 00:00:00,000**

Pvm - Aika	Laite	Virhe	Virheteksti	Ratkaisuehdotus
09/04/2014 - 04:51	R5	3103	Robotissa on TU:ita, mutta robotin tyhjentävää tehtävää ei pystytä tekemään.	Tarkista työkierron tilan perusteella mikä estää jättötehtävän tekemisen.
09/04/2014 - 04:40	R5	3019	Järjestelmä asettanut mittauspyynnön varastopaikkaan, koska varastopaikkaan kohdistunut tehtävä on päättynyt virheeseen.	Robotti suorittaa mittauksen ko. varastopaikkaan heti, kun se on suorittanut työkiertonsa loppuun.
09/04/2014 - 04:40	R5	2311	Varastopaikassa liian vähän TU:ita.	
07/04/2014 - 05:30	0	2208	Tilaustenhallintajärjestelmän taustaprosessissa tapahtunut virhe.	
07/04/2014 - 05:29	0	2208	Tilaustenhallintajärjestelmän taustaprosessissa tapahtunut virhe.	
07/04/2014 - 05:29	0	2208	Tilaustenhallintajärjestelmän taustaprosessissa tapahtunut virhe.	
07/04/2014 - 05:29	0	2208	Tilaustenhallintajärjestelmän taustaprosessissa tapahtunut virhe.	
07/04/2014 - 05:29	0	2208	Tilaustenhallintajärjestelmän taustaprosessissa tapahtunut virhe.	
07/04/2014 - 05:28	0	2208	Tilaustenhallintajärjestelmän taustaprosessissa tapahtunut virhe.	
07/04/2014 - 05:27	0	2208	Tilaustenhallintajärjestelmän taustaprosessissa tapahtunut virhe.	
07/04/2014 -	0	2208	Tilaustenhallintajärjestelmän taustaprosessissa	

Kuva 17. Virheraportti.

Kuvassa näkyy järjestelmän robottien virheet valitulta ajanjaksolta.

## 4.2 Palvelin ja sen ohjelmistot

Tässä luvussa käyn läpi asioita jotka piti tehdä ennen kuin käyttöliittymästä voidaan suorittaa tietokantahakuja. Ensin keskityn Tomcat ohjelmiston asennukseen palvelimelle ja Progressin tarjoaman servletin pystyttämiseen palvelimelle. Seuraavaksi käyn läpi Progressin tarjoaman web-service rajapinnan ja sen, miten saan progressin ja PHP:n toimimaan yhdessä. Näiden jälkeen käyn läpi tietokantakyselyn koodit ja lopuksi kerron miten kokonaisuus toimii ja miten se rakennetaan eri projekteissa.

Toimiva kokonaisuus koostuu Progressin OpenEdge ohjelmistosta ja siihen liittyvästä web service adapterista, jolla voidaan luoda Progressin tietokantaohjelmistoon web service rajapinta ja sitä voidaan kutsua asiakasohjelmasta. Lisäksi tarvitaan jvasäiliöohjelmisto, jonka päällä voidaan ajaa



web service rajapinnan luontiin tarvittavaa servlettiä. Tämän jälkeen tarvitsee saada vielä PHP ja Progress yhdistettyä toisiinsa toimivaksi kokonaisuudeksi.

En käy läpi Progress ohjelmiston asentamista palvelimelle, sillä se on melko suoraviivainen toimenpide. Windowsille ja Linuxille on molemmille omat asennuspaketit, jotka ladataan kohdekoneelle ja asennetaan.

#### 4.2.1 Tomcat ja Progress servletti

Progressin tarjoama java servlet ei toimi yksinään vaan, se tarvitsee ympäristön jossa se voi toimia. Valitsin java servlet moottoriksi hyvin tunnetun ja käytetyn Apache Tomcat palvelinkokonaisuuden. Apache Tomcat oli minulle tuttu entuudestaan, ja siksi päätin käyttää sitä.

Tomcat palvelimen asennus Linux ympäristöön on tehty helpoksi ja nopeaksi. Ensin pitää hakea Apache Tomcat palvelinpaketti Apachen sivuilta ja tallentaa se palvelimelle. Paketti puretaan palvelimelle haluttuun kansioon. Paketti pitää sisällään kaikki palvelimen käynnistämiseen, pysäyttämiseen ja ajamiseen tarvitsevat tiedostot. Erillistä asennusta ei näin tarvita.

Paketin purun jälkeen bin kansioista löytyy valmiina palvelimen käynnistys- ja pysäytys- skriptit. Ennen kuin palvelin voidaan käynnistää, pitää palvelimelle määrittää ympäristömuuttujat CATALINA\_HOME ja JAVA\_HOME. CATALINA\_HOME muuttujan pitää osoittaa Tomcatin purkukansioon ja JAVA\_HOME:n pitää osoittaa Javan JDK kansioon. Palvelin saadaan käyntiin suorittamalla bin kansiossa oleva startup.sh skripti ja pysäytys tapahtuu suorittamalla shutdown.sh skripti.

Tomcatin asennuksen jälkeen voidaan toteuttaa Progressin tarjoamaa valmista servlettiä. Progressin asennuskansioista löytyy mallipohja servletille, joka kopioidaan Tomcatin webapps kansioon. Progressin tarjoama servlet löytyy Progressin asennuskansion alla kansioista /servlets/wsa ja se kopioidaan palvelimelle \$CATALINA\_HOME/webapps kansioon. Kansion kopioinnin jälkeen luodaan kansioista toinen kopio, mutta muutetaan nimi projektissa käytettäväksi työnumeroksi. Samalla palvelimella voidaan suorittaa periaatteessa useita eri web service servlettejä ja siksi haluttiin tehdä mahdolliseksi monen ilmentymän ajaminen samalla kertaa. Tämä ei ole yleensä tarpeellista, mutta tälle haluttiin rakentaa tuki tulevaisuutta ajatellen. Yleensä riittää yksi ilmentymä

per palvelin ja ilmentymän yksilöivä nimi on projektille annettu työnnumero. Käytin työssäni ilmentymälle työnnumeroa 92600.

Toimenpiteiden jälkeen webapps kansiossa on siis kansiot wsa ja 92600. Muokataan web.xml tiedostoa joka löytyy \$CATALINA\_HOME/webapps/92600/WEB-INF/ kansioista.

```
<init-param>
  <param-name>instanceName</param-name>
  <!-- Enter this WSA servlet instance's name that will be used to locate
        its properties in the Progress ubroker.properties file -->
  <param-value>92600</param-value>
</init-param>
```

Web.xml tiedoston muokattava kohta on <param-value> elementin arvo. Oletuksena arvo on wsa1, mutta arvoksi asetetaan projektille annettu työnnumero jolloin servlettiin saa myöhemmin yhteyden osoitteesta <http://<ip-osoite>:8080/<työnnumero>/wsa1>.

Seuraavaksi pitää vielä lisätä luomamme ilmentymä Progresssin asennustiedostoon. Tiedosto on nimeltään ubroker.properties ja se sijaitsee Progressin asennuskansiossa olevassa properties kansiossa. Ilmentymän lisäys tapahtuu luomalla tiedostoon uusi osio ilmentymälle. Tiedoston lopussa on [WSA] osio. Osiossa määritellään kaikki WSA yhteiset asetukset ja tämän osion alle luodaan uusi luomamme ilmentymän osio.

```
[WSA.<työnnumero>]
  adminAuth=0
  adminEnabled=1
  location=local
  logFile=/var/mp/<työnnumero>/log/<työnnumero>.wsa.log
  loggingLevel=2
  proxyHost=localhost
  proxyPort=8080
  wsaUrl=http://localhost:8080/<työnnumero>/wsa1
```

Lisättävät rivit ubroker.properties tiedostoon. <työnumero> kohdan tilalle kirjoitetaan uusi ilmentymä tässä tapauksessa 92600.

Muutoksen jälkeen Tomcat pitää vielä käynnistää uudelleen, jotta luotu servlet saataisiin käyttöön.

Servletin toiminnan voi testata selaimella menemällä osoitteeseen <http://<ip-osoite>:8080/92600/wsa1> ja servletin pitäisi vastata sivupyyntöön.

```
WSA Web Services
Status:92600:OK:111
```

Servletin oikeanlaisen asennuksen ja toteuttamisen jälkeen ladatun sivun tulostus pitäisi olla yllä olevan mukainen. Ensimmäinen rivi on otsikkotieto ja toinen rivi kertoo luodun servletin tilan. Rivi on eritelty neljään eri osaan. Rivin osat on eritelty kaksoispistemerkillä.

Status – Status otsikko

92600 – Luodun ilmentymän nimi

OK – Servletin tila

Neljäs rivin osa koostuu kolmesta eri arvosta. Arvot voivat olla joko 0 tai 1 riippuen käytetyistä asetuksista. Käytetyt asetukset on tallennettu ubroker.properties tiedostoon ja liittyvät servletin käytön turvallisuusasetuksiin. Mikäli arvot ovat eri kuin esimerkissä näkyvät kolme ykköstä (111) tarvitsee ubroker.properties tiedostoa muokata ja lisätä WSA.wsa1 osion alle rivit:

```
[WSA.92600]
  adminEnabled=1
  webAppEnabled=1
  enableWsd1=1
```

---

## Progress Web Services Adapter

Web Services Adapter Deployed Applications

Name	WSDL File	Target URL
ws_reports	<a href="#">ws_reports.wsdl</a>	urn:cimcorp-com

---



Menemällä osoitteeseen `http://<ip-osoite>:8080/<työnumero>/wsa1/wsdl` voidaan listata kaikki palvelimelle asennettut WSA ilmentymät. Listaus on yllä olevan mukainen. WSDL kuvausta klikkaamalla saadaan haluttu wsdl määrittäminen näkyviin.

#### 4.2.2 Progressin Web service adapter (WSA)

Raporttien toiminnallisuuden kannalta ydinkomponentti on Progressin Web service adapter, joka mahdollistaa Progress tietokannan web servicen rakentamisen ja kutsumisen. WSA ei toimi suoraan vaan se vaatii yllä rakennetun java säiliön toteuttamisen, jonka päälle java servletti pystytetään ja suoritetaan.

Appserverillä on useita eri tiloja joihin se voidaan asetuksista määrittää. Tila määrittää miten appserver broker kutsuu ja käsittelee agenteja sekä miten agentit suorittavat tietokantakyselyitä. Tilan määrittämisellä voi olla suuria vaikutuksia ajettavien tietokantakyselyiden suorituskykyyn. Määritetyn tilan voi selvittää suorittamalla kyselyn appserverille. Tila määritellään `ubroker.properties` tiedostossa. Tilat voivat olla `state-reset`, `state-aware`, `stateless` ja `state-free` /23 sivu 26 1-2.

Käyn seuraavassa läpi tilojen ominaisuuksia ja miten appserveri toimii eri tiloissa. Työssäni jouduin käsittelemään appserveria, jonka tila on `stateless`, koska web service palvelu haluttiin saada toimimaan jo olemassa olevan appserverin kanssa ja välttää uuden appserverin asentamiselta projektiympäristöön vain web serviceä varten. Suorituskyvyn takia ei haluttu luoda uutta appserveriä web palvelua varten, koska käytetyt web service kutsut eivät ole suorituskyvyn kannalta oleellisia.

`State-reset` tilassa olevan appserverin saadessa yhteyspyynnön OpenEdge:ltä, appserver broker yhdistää asiakasovelluksen suoraan yhteen agenttiin appserverin agenttipoolista. Agentti on varattuna asiakasovellukselle koko sen yhteyden elinkaaren ajan. Yhteyksien välillä OpenEdge alustaa appserver istunnon alkutilaan. Tämä tila on sama jossa appserveri on heti appserver agenttien käynnistyksen jälkeen /23 sivu 26 1-2.

`State-aware` tilassa olevan appserverin saadessa yhteyspyynnön OpenEdge:ltä, appserver broker yhdistää asiakasovelluksen suoraan yhteen agenttiin appserverin agenttipoolista. Agentti on

varattuna asiakassovellukselle koko sen yhteyden elinkaaren ajan. Erona state-reset tilaan on, että OpenEdge ei nollaa appserverin istunnon sisältöä yhteyksien välillä. Yhteyden päätyttyä istunnon sisältö on saatavissa toisella yhteydellä /23 sivu 28 1-4.

Stateless eroaa state-reset ja state-aware tilassa siinä, että yhteyspyynnön jälkeen appserver broker ei yhdistä suoraan agenttiin vaan kaikki appserver agentit jäävät vapaiksi. Asiakasohjelman kutsuessa appserveria, appserver broker säilyttää yhteyden itseensä ja asiakassovellukseen. Appserver broker voi tällöin lähettää suorituskäskyn mille tahansa vapaalle agentille /23 sivu 29 1-5

State-free tilassa oleva appserver toimii kuten stateless tilan appserver. Agentit pysyvät vapaana ja voivat suorittaa etäkutsuja. Asiakassovelluksen suorittaessa tietokantakyselyitä OpenEdge välittää tiedon appserverille. Appserver brokerin saadessa pyynnön, pyyntö lähetetään vapaalle agentille joka suorittaa tietokantakyselyn. Agentin suorittama tietokantakysely se palauttaa tulokset appserver brokerille, joka välittää ne takaisin asiakassovellukselle ja agentti vapautuu suorittamaan seuraavaa pyyntöä joka voidaan välittää brokerille /23 sivu 31 1-8.

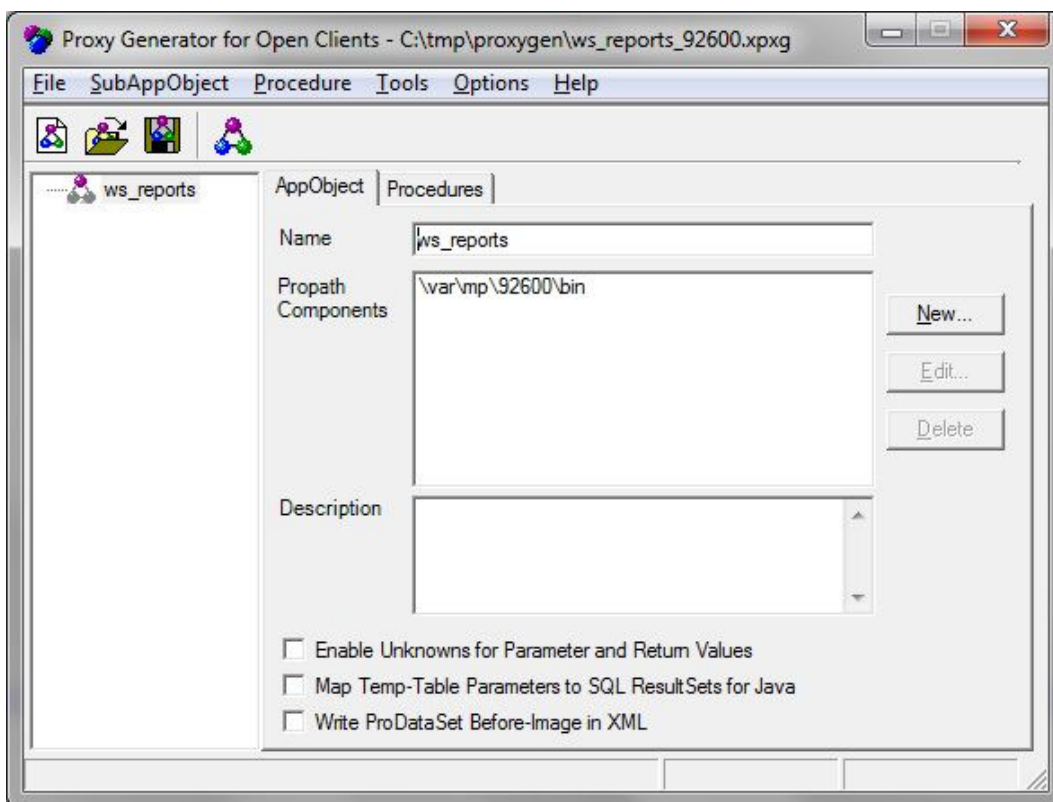
Web servicen pystyttämässä ja sen määrittämässä WSDL tiedostossa on eroa onko valittu appserverin tila session managed vai session-free. Session-free muotoa käytettäessä asiakassovelluksen ei tarvitse välittää Progressille header tietoja SOAP viestissä. Tämä helpottaa kutsun tekemistä PHP:n päässä. Kutsu on yksinkertaisempi ja suoraviivaisempi. Session-managed tilan valittaessa Progressille tarvitsee välittää myös header tietoja SOAP viestissä ja se monimutkaistaa PHP:stä tehtäviä kutsuja. Session-managed tilan appserverille tarvitsee ennen varsinaisen tietokantakyselyn suorittamista kutsua connect-metodia, jolla lähetetään kirjautumistiedot ja appserver palauttaa objektiID:n, joka pitää sisällyttää SOAP headeriin tietokantakyselyitä tehtäessä. Tietokantakyselyiden jälkeen yhteys pitää sulkea kutsumalla release-metodia joka sulkee yhteyden /43.

### 4.2.2.1 Proxy generator

Proxy generaattori eli Proxygen on yksi Progressin tarjoamista työkaluista. Työssäni käytin Proxy generaattoria jolla loin haluamistani tietokantakoodeista WSDL määrittymen joka voidaan antaa WSA:lle.

Proxygen tukee kahta appserverin istuntotilaa. Tilat ovat session managed ja session-free. Tila määrittää sen miten appserver on määritelty ja miten asiakasohjelmisto ja appserver viestivät keskenään. Luotaessa web service palvelua tarvitsee tekovaiheessa valita joko session managed tai session-free tila, koska web service palvelun rajapinta on erilainen kummassakin tilassa /31 sivu 37 3-3.

Seuraavaksi käyn läpi proxygen ohjelmistoa ja miten ohjelmalla luodaan WSDL määrittymen.

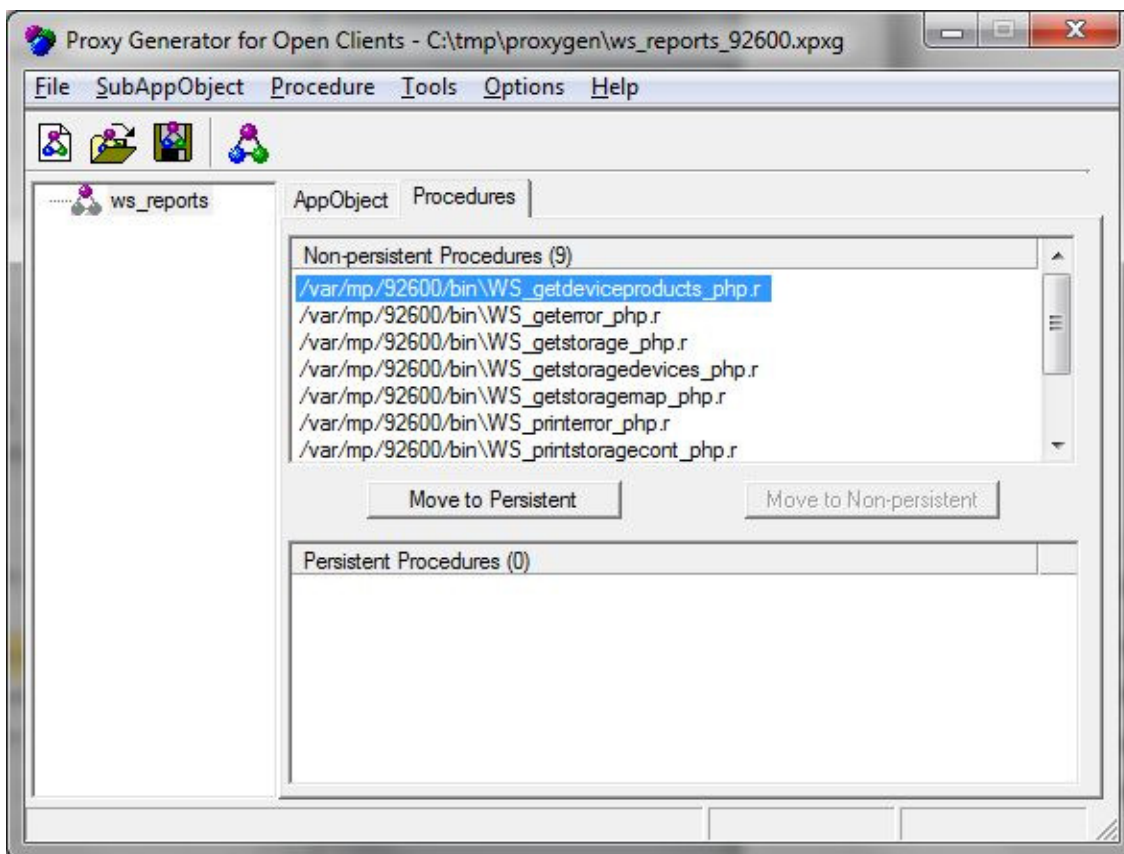


Kuva 18. Proxygen.

AppObject välilehdelle määritellään web service palvelun nimi ja nimeksi valitaan ws\_reports. Tämän jälkeen määritellään kohtaan Propath Components paikka josta käännetty tietokantakysely-

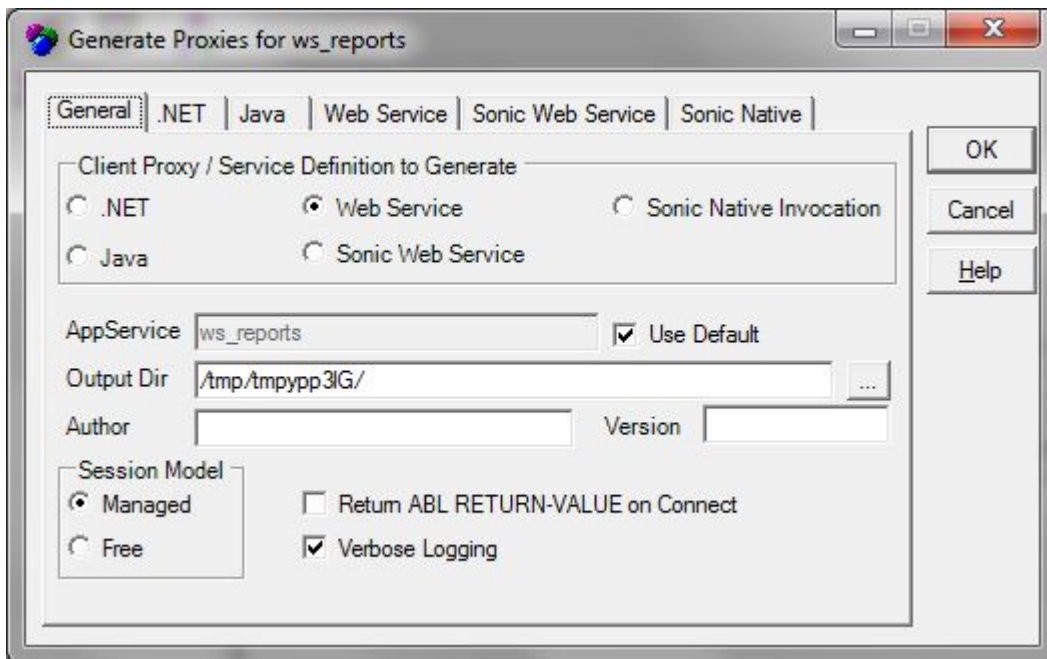
koodit löytyvät ja joita luotava web palvelu suorittaa. Näytön alalaidassa on kolme valintaruutua, joilla voi määrittellä luotavan proxy määrittelyä tarkemmin. Työssäni en käyttänyt lisämäärittelyitä, joten en keskity niihin tarkemmin vaan käyn ne läpi pintapuolisesti /31 sivu 48 3-14.

Valinta ”Enable Unknowns for parameter and return values” liittyy siihen miten asiakasohjelma käsittelee tuntemattomia paluuarvoja. Web services voi aina palauttaa tuntemattoman arvon (?) Progressista riippumatta rajapinnan määrittelystä /31 sivu 49 3-15. Valinta ”Map Temp-Table Parameters to SQL ResultSets for Java” liittyy siihen miten paluuarvotaulukkoa käsitellään Javassa. Työssäni en tarvinnut tätä ominaisuutta /31 sivu 49. Valinta ”write ProDataSet Before-Image in XML” liittyy siihen miten before imagen dataa käsitellään SOAP viestissä /31 sivu 50 3-16. Before imagea käytetään Progressissa, jotta suoritettu transaktio voidaan perua esimerkiksi virheen sattuessa. Koodin transaktio-osa kirjoitetaan ensin before imageen, josta se voidaan palauttaa tarvittaessa /32 sivu 8.



Kuva 19. Proxygen tietokantakyselykoodien määrittely.

AppObject määrittysten jälkeen siirrytetään Procedures välilehdelle ja lisätään sinne halutut tietokannan kyselykoodit, joita web service palvelussa käytetään. Käytetyt tietokannan kyselykoodit pitää löytyä kansioista, joka määriteltiin Propath Components kenttään. Proxy generator tarvitsee Progressin tietokantahakukoodista käännettyt versiot. Tietokantakoodit voidaan kääntää käyttäen Progressin työkaluja.

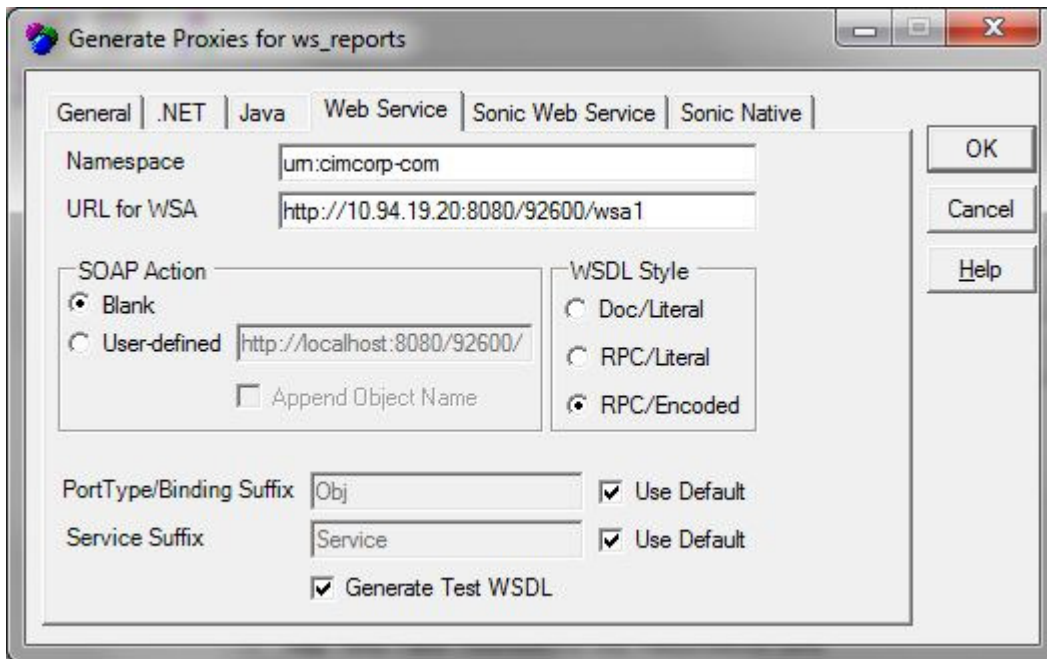


Kuva 20. Proxygen yleisten asetusten määrittäminen.

Määrittysten jälkeen painetaan generate nappia (kolmio merkki ylävalikon alla) ja saadaan näkyviin lisävalinnaiset välilehdet. Jokaiselle avoimelle asiakasrajapinnalle on omat välilehdet joilta voi määrittellä lisäasetuksia tarvittaessa. Välilehdet ovat General, .NET, Java, Web services, Sonic Web Service ja Sonic Native. Työssäni käytin vain General ja Web Service välilehtiä joten keskityn vain niiden sisältöön ja jätän muiden välilehtien sisällön huomioita.

General -välilehti pitää sisällään kaikki yleiset lisämäärittelyt josta myös valitaan haluttu rajapintamäärittely. Client proxy / Service Definition to Generate -osioon valitaan web service, output dir kenttään määritellään kansio johon valmis WSDL – tiedosto, myöhemmin hyödynnettävissä oleva wsm -tiedosto ja ohjelma loki-tiedostot tallennetaan /31 sivu 60 3-26. AppService kenttään voidaan kirjoittaa käytetty appserver nimi joka on kirjautunut nameserveriin.

Käytettäessä aiemmin määritettyä AppObject arvoa valitaan viereinen oletusvalinta valintaruutu /31 sivu 59 3-25. Author ja Version kenttien täyttö on valinnaista ja session modeliksi valitaan managed, koska käyttämämme appserver on määritetty käyttämään stateless tilaa /31 sivu 60 3-26. Return ABL RETURN-VALUE on Connect valinta mahdollistaa paluuarvon lähettämisen web servicen connect metodista. Valinta on mahdollinen vain jos rajapinnan modeliksi on valittu managed /31 sivu 60 3-26. Verbose logging on mahdollista valita siinä tapauksessa jos halutaan saada yksityiskohtaisempaa tietoa tulostettavaan lokiin proxyn luonnista /31 sivu 60 3-26.



Kuva 21. Proxygen web service määrittely.

Viimeiseksi valitaan Web Service välilehti jossa määritellään tarkemmat web service palvelun käyttämät määrittelyt. Määrittelyjen jälkeen painetaan OK painiketta ja Progressin Proxygen luo tarvittavat tiedostot aikaisemmin määritettyyn kansioon.

Namespace:n määrittelyllä valitaan haluttu nimiavaruus. Nimiavaruus määrittelee web servicen ja siinä käytettyjen elementtien nimiavaruuden. Nimiavaruus pitää olla XML nimiavaruus määrittelyn mukainen. URL for WSA määrittelee paikan johon web service luodaan ja mistä sitä voi kutsua /31 sivu 64 3-30. SOAP action määrittää käytettävän soap actionin jota käytetään soap viestin otsikko-elementissä. WSDL Style määrittää web servicessä käytettävän palvelun SOAP viestien muotoilun. Tyylin pitää olla sama mitä käytettävä web palvelun asiakas tukee /31 sivu 65 3-31.



PortType/Binding Suffix määrittää PortType ja Binding nimen jälkiliitteen. Oletusarvo on merkkijono Obj. Service Suffix määrittää Service elementin nimen jälkiliitteen. Oletusarvo on merkkijono Service. Generate Test WSDL valinnalla voidaan luoda määrittelystä WSDL tiedosto kohde kansioon /31 sivu 66 3-32.

Yllä olevien määrittelyiden jälkeen ohjelma luo web service palvelun wsdl -kuvauksen, joka määrittää palvelun ja sen rajapinnan sekä wsm – tiedoston, jolla voi luoda ja julkaista määrittelemämme web palvelun käyttäen Progress Explorer työkalua. Haluttaessa määrittely voidaan tallentaa xpxg tiedostoksi, jota voidaan hyödyntää myöhemmin.

```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:S1="urn:soap-fault:details" xmlns:wsdl="http://schemas.:
xmlns:prodata="urn:schemas-progress-com:xml-prodata:0001" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soa
com:ws_reports" xmlns:tns="urn:cimcorp-com" targetNamespace="urn:cimcorp-com" name="ws_reports">
  <wsdl:documentation>EncodingType=RPC_ENCODED, Proxygen_Product=Progress Version 10.2B</wsdl:documentation>
  + <wsdl:types>
  + <wsdl:message name="ws_reports_WS_printererror_php">
  + <wsdl:message name="ws_reports_WS_getdeviceproducts_php">
  + <wsdl:message name="ws_reports_WS_testconnectionResponse">
  + <wsdl:message name="ws_reports_WS_geterror_phpResponse"> </wsdl:message>
  + <wsdl:message name="ws_reports_WS_getstoragemap_php">
  + <wsdl:message name="ws_reportsConnect">
  + <wsdl:message name="ws_reports_WS_getdeviceproducts_phpResponse">
  + <wsdl:message name="ws_reports_WS_printstoragecont_php">
  + <wsdl:message name="ws_reportsConnectResponse"> </wsdl:message>
  + <wsdl:message name="ws_reports_WS_getstoragedevices_phpResponse">
  + <wsdl:message name="ws_reportsID">
  + <wsdl:message name="ws_reports_WS_geterror_php">
  + <wsdl:message name="ws_reports_WS_getstorage_phpResponse"> </wsdl:message>
  + <wsdl:message name="ws_reports_WS_getstoragedevices_php"> </wsdl:message>
  + <wsdl:message name="ws_reportsRelease"> </wsdl:message>
  + <wsdl:message name="ws_reports_WS_printstoragemap_phpResponse"> </wsdl:message>
  + <wsdl:message name="ws_reports_WS_printstoragecont_phpResponse"> </wsdl:message>
  + <wsdl:message name="ws_reports_WS_printererror_phpResponse"> </wsdl:message>
  + <wsdl:message name="ws_reports_WS_getstorage_php">
  + <wsdl:message name="ws_reports_WS_printstoragemap_php">
  + <wsdl:message name="FaultDetailMessage">
  + <wsdl:message name="ws_reportsReleaseResponse"> </wsdl:message>
  + <wsdl:message name="ws_reports_WS_testconnection"> </wsdl:message>
  + <wsdl:message name="ws_reports_WS_getstoragemap_phpResponse"> </wsdl:message>
  + <wsdl:portType name="ws_reportsObj">
  + <wsdl:binding name="ws_reportsObj" type="tns:ws_reportsObj">
  + <wsdl:service name="ws_reportsService">
</wsdl:definitions>
```

Kuva 22. WSDL listaus.

Kuvassa näkyy luotu wsdl tiedosto. Wsdl tiedosto on vaikeasti tulkittava ja siksi sitä yleensä luutaankin ohjelmallisesti. Kuvassa näkyy vain dokumentin sisältämät pääelementit. Liite 1 pitää sisällään koko wsdl tiedoston.

Seuraavaksi käyn yhden kutsuttavan tietokantakoodin wsdl elementit läpi. Kutsuttava koodi on nimeltään WS\_getdeviceproducts\_php ja sitä kutsutaan suoraan PHP:stä. Koodi hakee annetun laite id:n sisältämät tuotteet ja listaa ne web-sivulla. Koodi ottaa syötteenä yhden parametrin nimeltään

devID, jonka tietotyyppi on kokonaisluku. Koodi palauttaa taulukon, jonka nimi on tt\_sku. Taulukolla on yksi alkio nimeltään skuid jonka tietotyyppi on kokonaisluku.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="ws_reports" targetNamespace="urn:cimcorp-com" XMLNs:tns="urn:cimcorp-com"
XMLNs:S2="urn:cimcorp-com:ws_reports" XMLNs:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
XMLNs:xsd="http://www.w3.org/2001/XMLSchema" XMLNs:prodata="urn:schemas-progress-com:XML-
prodata:0001" XMLNs:soap="http://schemas.xmlsoap.org/wsdl/soap/"
XMLNs:wsdl="http://schemas.xmlsoap.org/wsdl/" XMLNs:S1="urn:soap-fault:details"
XMLNs="http://schemas.xmlsoap.org/wsdl/">
<wsdl:documentation>EncodingType=RPC_ENCODED, Proxygen_Product=Progress Version
10.2B</wsdl:documentation>
```

WSDL dokumentin alussa on XML tiedoston alkumäärittely, jonka jälkeen on definitions elementti. Elementissä määritellään WSDL tiedoston tarvitsemat nimiavaruudet ja Progressin versiotiedot. Wsdl nimi on ws\_reports ja sille käytettävä nimiavaruus urn:cimcorp-com jotka määriteltiin työkalulla aikaisemmin. Näiden lisäksi työkalu luo nimiavaruudet SOAP protokollalle ja sen tarvitsemat muut nimiavaruusmäärittelyt. Dokumentation elementissä on kerrottu Progressin versio-tietoja ja proxygen työkalun nimi jolla wsdl tiedosto on luotu.

```
<wsdl:types>
  <schema elementFormDefault="unqualified" targetNamespace="urn:soap-fault:details"
XMLNs="http://www.w3.org/2001/XMLSchema">
    <complexType name="FaultDetail">
      <sequence>
        <element name="errorMessage" type="xsd:string"/>
        <element name="requestID" type="xsd:string"/>
      </sequence>
    </complexType>
  </schema>
  <schema elementFormDefault="unqualified" targetNamespace="urn:cimcorp-com:ws_reports"
XMLNs="http://www.w3.org/2001/XMLSchema">
    <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
    <complexType name="ws_reportsID">
      <sequence>
        <element name="UUID" type="xsd:string"/>
      </sequence>
    </complexType>
    <complexType name="WS_getdeviceproducts_php_tt_skuRow">
      <sequence>
        <element name="skuid" nillable="true" type="xsd:int"/>
      </sequence>
    </complexType>
    <complexType name="ArrayOfWS_getdeviceproducts_php_tt_skuRow">
      <complexContent>
        <restriction base="soapenc:Array">
          <attribute ref="soapenc:arrayType" wsdl:arrayType="S2:WS_getdeviceproducts_php_tt_skuRow[]" />
        </restriction>
      </complexContent>
    </complexType>
</wsdl:types>
```



Definitions elementin alla olevassa types elementissä on määritelty web servicen tarvitsemat tietotyypit ja määrittelyt. Tietotyypit on määritelty käyttäen XML schema määrittelyä. WS\_getdeviceproducts\_php\_tt\_skuRow on taulukko ja pitää sisällään kokonaislukumuuttujan skuid. Alla on vielä määritelty SOAP protokollan mukaiset taulukkomäärittelyt.

```
<wsdl:message name="ws_reports_WS_getdeviceproducts_php">
  <wsdl:part name="devID" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="ws_reports_WS_getdeviceproducts_phpResponse">
  <wsdl:part name="tt_sku" type="S2:ArrayOfWS_getdeviceproducts_php_tt_skuRow"/>
</wsdl:message>
<wsdl:message name="FaultDetailMessage">
  <wsdl:part name="FaultDetail" type="S1:FaultDetail"/>
</wsdl:message>
```

Message elementeillä kerrotaan kutsuissa käytettävien muuttujien nimet ja tietotyypit. Jokainen kutsuttava koodi on listattu omalla message elementillä. Kutsu on määritelty käyttäen kutsuttavan koodin nimeä ja koodin paluuarvo on määritelty käyttäen kutsuttavan koodin nimeä ja sen perään response tekstiä. Esimerkiksi ws\_reports\_WS\_getdeviceproducts\_php määrittelee kutsuttavan koodin sisäänotettavat muuttujat ja paluuarvo on määritelty käyttäen ws\_reports\_WS\_getdeviceproducts\_phpResponse:a. Virheen sattuessa käytetään FaultDetailMessagea.

```
<wsdl:portType name="ws_reportsObj">
  <wsdl:operation name="WS_getdeviceproducts_php" parameterOrder="tt_sku devID">
    <wsdl:input message="tns:ws_reports_WS_getdeviceproducts_php"/>
    <wsdl:output message="tns:ws_reports_WS_getdeviceproducts_phpResponse"/>
    <wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
  </wsdl:operation>
</wsdl:portType>
```

PortTypellä määritellään kutsuttavien koodien sisään menevät ja paluuarvot jotka on aikaisemmin määritelty message elementeillä. PortType elementin name attribuutin arvona on aikaisemmin proxygen ohjelmistolla määritetty portType / binding kentän arvo.

```

<wsdl:binding name="ws_reportsObj" type="tns:ws_reportsObj">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="WS_getdeviceproducts_php">
    <soap:operation soapAction="" style="rpc"/>
    <wsdl:input>
      <soap:header message="tns:ws_reportsID" part="ws_reportsID" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="urn:cimcorp-com:ws_reports" wsdl:required="true">
      </soap:header>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
    </wsdl:output>
    <wsdl:fault name="ws_reportsFault">
      <soap:fault name="ws_reportsFault" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        namespace="http://localhost:8080/92600/wsa1"/>
    </wsdl:fault>
  </wsdl:operation>
</wsdl:binding>

```

Binding elementillä kerrotaan käytettävästä yhteydestä ja elementin alla on listattu operation elementin yhteydessä kaikkien kutsuttavien tietokantakoodien yhteyskäytännöt. Binding elementin name attribuutin arvona käytetään aikaisemmin proxygenillä määriteltyä nimeä.

```

<wsdl:service name="ws_reportsService">
  <wsdl:port name="ws_reportsObj" binding="tns:ws_reportsObj">
    <wsdl:documentation />
    <soap:address location="http://localhost:8080/92600/wsa1"/>
  </wsdl:port>
</wsdl:service>

```

Service elementin alla on määritetty port -elementissä osoite, josta web serviceä voi kutsua ja josta sitä voidaan käyttää. Service elementin name attribuuttina käytetään nimeä joka määritettiin Proxygen ohjelmiston service kentälle ja port elementin name attribuutin arvona käytetään aikaisemmin Proxygenillä määriteltyä arvoa. Location attribuutin arvona on yhteysosoite josta web service löytyy.

Progress tarjoaa työkalun jolla voidaan luoda suoraan WSDL tiedostosta dokumentaatio ja esimerkkiohjelmat jolla WSDL tiedostoa voi käyttää ja kutsua Progressissa. Ohjelman nimi on bprowsldoc.bat. Ohjelmalle kerrotaan WSDL tiedoston sijainti ja ohjelma luo dokumentaatio-tiedoston annetusta määrittelystä ja tallentaa sen samaan kansioon jossa ohjelma suoritetaan.

```

C:\test\webservice\ws_reports>c:\opt\prgs\dlc102b\bin\bprowsldoc.bat
http://10.x.x.x:8080/92600/wsa1/wsdl?targetURI=urn:cimcorp-com

```

Dokumentti tallentuu samaan kansioon jossa ohjelma suoritetaan. Tässä tapauksessa tiedosto tallentuu polkuun C:\test\webservice\ws\_reports\index.html

Alla on listattuna kaikki web service liikenteen SOAP viestit, jolla luodaan kantayhteys Progress tietokantaan, suoritetaan kysely ja suljetaan kantayhteys.

```
<soapenv:Envelope XMLNs:xsi="http://www.w3.org/2001/XMLSchema-instance" XMLNs:xsd="http://www.w3.org/2001/XMLSchema"
XMLNs:soapenv="http://schemas.xmlsoap.org/soap/envelope/" XMLNs:urn="urn:cimcorp-com:ws_reports">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:Connect_ws_reports soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <userId xsi:type="xsd:string">smik</userId>
      <password xsi:type="xsd:string">smik10</password>
      <appServerInfo xsi:type="xsd:string">Appserver://10.94.19.20:5162/d92600p1</appServerInfo>
    </urn:Connect_ws_reports>
  </soapenv:Body>
</soapenv:Envelope>
```

Esimerkissä lähetetään yhteyden avauspyyntö käyttäen web palvelussa määritettyä Connect\_ws\_reports operaatiota. Yhteyden avaamiseen tarvitsee antaa kirjautumistiedot. Kirjautumistiedot ovat userId, password ja appServerInfo.

```
<SOAP-ENV:Envelope XMLNs:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" XMLNs:xsi="http://www.w3.org/2001/XMLSchema-
instance" XMLNs:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Header>
    <t:ws_reportsID SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" XMLNs:t="urn:cimcorp-com:ws_reports">
      <UUID xsi:type="xsd:string">79e1cc570b3c744e:44df39d1:1405847a672:-7ffa;&lt;ws_reportsIPX-
000353IAO>;IuzGFawGQYtvD8u212Ws0A==</UUID>
    </t:ws_reportsID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <ns1:Connect_ws_reportsResponse SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" XMLNs:ns1="urn:cimcorp-
com:ws_reports"></ns1:Connect_ws_reportsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Yhteyspyynnön vastaukseksi saatiin UUID arvo, joka yksilöi yhteyden appserveriin päin. UUID merkkijono pitää sisällyttää jokaiseen kutsuun, joka tehdään appserverille. UUID merkkijono annetaan SOAP viestin header osioon.

```

<soapenv:Envelope XMLNs:xsi="http://www.w3.org/2001/XMLSchema-instance" XMLNs:xsd="http://www.w3.org/2001/XMLSchema"
XMLNs:soapenv="http://schemas.xmlsoap.org/soap/envelope/" XMLNs:urn="urn:cimcorp-com:ws_reports">
  <soapenv:Header>
    <ws_reportsID xsi:type="urn:ws_reportsID">
      <UUID xsi:type="xsd:string">79e1cc570b3c744e:44df39d1:1405847a672:-7ffa;&lt;ws_reportsIPX-
000353IAO>;IuzGFawGQYtvD8u212Ws0A==</UUID>
    </ws_reportsID>
  </soapenv:Header>
  <soapenv:Body>
    <urn:WS_getdeviceproducts_php soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <devID xsi:type="xsd:int">1</devID>
    </urn:WS_getdeviceproducts_php>
  </soapenv:Body>
</soapenv:Envelope>

```

Kutsutaan WS\_getdeviceproduct\_php operaatiota ja annetaan yhteyden avaamiskutsun paluuarvona saatu UUID merkkijono SOAP viestin header elementtiin. Lisäksi lähetetään robotti id, jonka varaston sisältö halutaan saada paluuarvona. Robottinumeroksi valitaan 1.

```

<SOAP-ENV:Envelope XMLNs:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" XMLNs:xsi="http://www.w3.org/2001/XMLSchema-
instance" XMLNs:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:WS_getdeviceproducts_phpResponse SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" XMLNs:ns1="urn:cimcorp-
com:ws_reports">
      <tt_sku xsi:type="ns1:ArrayOfWS_getdeviceproducts_php_tt_skuRow" SOAP-
ENC:arrayType="ns1:WS_getdeviceproducts_php_tt_skuRow[]" XMLNs:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
        <Item xsi:type="ns1:WS_getdeviceproducts_php_tt_skuRow">
          <skuid xsi:type="xsd:int">1008000</skuid>
        </Item>
        <Item xsi:type="ns1:WS_getdeviceproducts_php_tt_skuRow">
          <skuid xsi:type="xsd:int">1011000</skuid>
        </Item>
        <Item xsi:type="ns1:WS_getdeviceproducts_php_tt_skuRow">
          <skuid xsi:type="xsd:int">1033000</skuid>
        </Item>
        <Item xsi:type="ns1:WS_getdeviceproducts_php_tt_skuRow">
          <skuid xsi:type="xsd:int">1052000</skuid>
        </Item>
        <Item xsi:type="ns1:WS_getdeviceproducts_php_tt_skuRow">
          <skuid xsi:type="xsd:int">1055000</skuid>
        </Item>
        <Item xsi:type="ns1:WS_getdeviceproducts_php_tt_skuRow">
          <skuid xsi:type="xsd:int">1074000</skuid>
        </Item>
        <Item xsi:type="ns1:WS_getdeviceproducts_php_tt_skuRow">
          <skuid xsi:type="xsd:int">1211000</skuid>
        </Item>
        <Item xsi:type="ns1:WS_getdeviceproducts_php_tt_skuRow">

```

```

    <skuid xsi:type="xsd:int">1212000</skuid>
  </Item>
  <Item xsi:type="ns1:WS_getdeviceproducts_php_tt_skuRow">
    <skuid xsi:type="xsd:int">1267000</skuid>
  </Item>
</tt_sku>
</ns1:WS_getdeviceproducts_phpResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

**WS\_getdeviceproducts\_php** operaatio palauttaa kysytyn robotin tuotteet jotka ovat varastossa.

```

<soapenv:Envelope XMLNs:xsi="http://www.w3.org/2001/XMLSchema-instance" XMLNs:xsd="http://www.w3.org/2001/XMLSchema"
XMLNs:soapenv="http://schemas.xmlsoap.org/soap/envelope/" XMLNs:urn="urn:cimcorp-com:ws_reports">
  <soapenv:Header>
    <ws_reportsID xsi:type="urn:ws_reportsID">
      <UUID xsi:type="xsd:string">79e1cc570b3c744e:44df39d1:1405847a672:-7ffa;&lt;ws_reports|PX-
000353|AO>;IuzGFawGQYtvD8u212Ws0A==</UUID>
    </ws_reportsID>
  </soapenv:Header>
  <soapenv:Body>
    <urn:Release_ws_reports soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
  </soapenv:Body>
</soapenv:Envelope>

```

Lopuksi kutsutaan Release\_ws\_reports operaatiota, joka sulkee avatun yhteyden appserveriin.

```

<SOAP-ENV:Envelope XMLNs:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" XMLNs:xsi="http://www.w3.org/2001/XMLSchema-
instance" XMLNs:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <ns1:Release_ws_reportsResponse SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" XMLNs:ns1="urn:cimcorp-
com:ws_reports"></ns1:Release_ws_reportsResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Yhteys on suljettu onnistuneesti.

```

<SOAP-ENV:Envelope XMLNs:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" XMLNs:xsi="http://www.w3.org/2001/XMLSchema-instance" XMLNs:xsd="http://www.w3.org/2001/XMLSchema">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>An error was detected in the Web Service request. (10894)</faultstring>
      <detail>
        <ns1:FaultDetail SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" XMLNs:ns1="urn:soap-fault:details">
          <errorMessage xsi:type="xsd:string">Error in SOAP header: Cannot find object. (10917)</errorMessage>
          <requestID xsi:type="xsd:string">79e1cc570b3c744e:44df39d1:1405847a672:-7ffa#61d</requestID>
        </ns1:FaultDetail>
      </detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Yritettäessä käyttää jo suljetun yhteyden UUID merkkijonoa saadaan Progressista virheilmoitus.

#### 4.2.2.2 Progress Explorer Tool

Työssäni käytin Progress Explorer työkalua web services palveluiden ylläpidossa. Työkalulla voidaan luoda ja muokata tietokantaa, appserveriä tai web service palveluita sekä niiden asetuksia. Työkalu on graafinen ja se on saatavilla Windowsille. Tehdyt muutokset tallentuvat automaattisesti ubroker.properties tiedostoon.

Progress Explorer toolilla voidaan luoda ja ottaa käyttöön web palvelu käyttäen Proxygen työkalulla luotua wsm -tiedostoa.

#### 4.2.3 Tietokannan kyselykoodit

Jotta web palvelua voidaan käyttää pitää web palvelun määritetyt tietokannankyselykoodit toteuttaa. Koodit luodaan käyttäen Progressin tarjoamaa ABL kieltä. Koodien tarkoitus on kerätä tietokannasta tarvittava tieto ja luoda tiedoista XML tiedosto. Koodien kirjoitus onnistuu parhaiten Progressin omalla tietokantaeditorilla jolla pääsee myös tarkastelemaan kannan tauluja ja kenttiä. Editorilla on myös mahdollista tarkistaa kirjoitetun koodin kieliasu. Koodin toteutuksen jälkeen pitää se viedä ympäristöön jossa koodia voidaan suorittaa.

Progressin tietokantakoodit suoritetaan yleensä käännettyinä. Kehittäjä luo koodin ja tallentaa luodun tiedoston p -tiedostopäätteellä. Ennen kuin appserver prosessi ajaa koodia, koodi pitää kääntää

niin sanotuksi r-koodiksi, jota appserver prosessi kutsuu ja suorittaa. Koodien kääntäminen onnistuu Progressin tarjoamilla työkaluilla.

Tietokantakyselykoodit varaston sisältöraporttiin ja virheraporttiin ovat melko samanlaiset. Molempien koodien tarkoitus on luoda XML tiedosto, joka luetaan PHP koodilla ja näytetään käyttäjälle taulukkomuodossa. XML tiedoston rakenne on molemmissa samankaltainen, mutta tiedoston sisältö vaihtelee. Koodi luo ensiksi taulukon yläotsikoiden nimet ja sen jälkeen varsinaisen taulukon sisältödatan.

```
<?xml version="1.0" encoding="utf-8" ?>
<errorlist>
  <headers name="Järjestelmän virheet 07/04/2014 00:00:00,000 - 11/04/2014 00:00:00,000">
    <header name="datetime" order="0" text="Pvm - Aika" visible="1"/>
    <header name="device" order="1" text="Laitte" visible="1"/>
    <header name="herror" order="2" text="Virhe" visible="1"/>
    <header name="errortext" order="3" text="Virheteksti" visible="1"/>
    <header name="solution" order="4" text="Ratkaisuehdotus" visible="1"/>
  </headers>
  <errors>
    <error datetime="09/04/2014 - 04:51" device="R5" errortext="Robotissa on TU:ita, mutta robotin tyhjentävää tehtävää ei pystytä tekemään."
    error="3103" solution="Tarkista työkierron tilan perusteella mikä estää jättötehtävän tekemisen."/>
    <error datetime="09/04/2014 - 04:40" device="R5" errortext="Järjestelmä asettanut mittauspyynnön varastopaikkaan, koska varastopaikkaan
    kohdistunut tehtävä on päättynyt virheeseen." error="3019" solution="Robotti suorittaa mittauksen ko. varastopaikkaan heti, kun se on suorittanut
    työkiertonsa loppuun."/>
    <error datetime="09/04/2014 - 04:40" device="R5" errortext="Varastopaikassa liian vähän TU:ita." error="2311" solution=""/>
  </errors>
</errorlist>
```

Esimerkkitulostus virheraportin XML tiedostosta jota luetaan PHP:stä.

Tiedosto alkaa XML tiedoston määrittelyllä ja merkistömäärittelyllä. Dokumentin pääelementti on nimeltään errorlist joka on kuvaava nimi virheraportille. Pääelementin lapsielementtinä on headers elementti jolla on kaksi attribuuttia. Attribuutit ovat name ja usesorting. Name attribuuttiin tulee virheraportin otsikko ja usesorting attribuutti liittyy taulukon lajitteluun. Headers elementin lapsielementtejä on seitsemän kappaletta joista muodostuu taulukon otsikkorivit. Elementillä on neljä attribuuttia. Attribuutit ovat name, order, text ja visible.

Name attribuuttia käytetään sitomaan otsikkotieto oikeaan dataan. Order attribuutti määrää missä järjestyksessä otsikot näytetään taulukossa. Text attribuutti kertoo otsikon nimen ja visible attribuutti määrää näytetäänkö otsikkoa CSV tiedostossa. Mahdolliset arvot ovat 0 ja 1. Määritetyn arvon ollessa yksi näyttää otsikon CSV tiedostossa ja arvo 0 piilottaa otsikon. Tämä on joskus tarpeen, jos halutaan näyttää taulukkomuodossa tarkempaa sisältöä kun CSV tiedostossa.

Otsikkotietojen jälkeen tulee elementti errors jonka alla on varsinaiset taulukon virherivit. Errors elementillä on virheiden määrästä riippuen yksi tai useampi error elementti jossa on kerrottu virhetiedot. Error elementillä on XML tiedoston otsikoiden määrän verran attribuutteja jotka ovat täsmälleen samannimiset kuin otsikkorivin name attribuutin arvot. Tällöin osataan hakea oikea tieto oikealle sarakkeelle. XML tiedostoa luetaan PHP:ssä niin, että oikeat otsikkotiedot ja virhetiedot osataan hakea oikealle kohdalle eikä PHP koodia tarvitse muuttaa.

```
<?xml version="1.0" encoding="utf-8" ?>
<storage>
  <headers name="Tuotteiden lukumäärä">
    <header name="skuid" order="0" text="Sku ID" visible="1"/>
    <header name="thsproductid" order="1" text="THS Tuote" visible="1"/>
    <header name="robR1" order="3" text="R1" visible="1"/>
    <header name="robR2" order="4" text="R2" visible="1"/>
    <header name="robR3" order="5" text="R3" visible="1"/>
    <header name="robR4" order="6" text="R4" visible="1"/>
    <header name="robR5" order="7" text="R5" visible="1"/>
    <header name="robR6" order="8" text="R6" visible="1"/>
    <header name="thsqty" order="9" text="Kpl" visible="1"/>
    <header name="total" order="10" text="kpl" visible="1"/>
    <header name="puttodolly" order="11" text="Yksiköintiin" visible="1"/>
    <header name="closed" order="12" text="Sku suljettu" visible="1"/>
  </headers>
  <products>
    <product closed="no" puttodolly="no" robR1="3" robR2="8" skuid="1026000" total="11"/>
    <product robR1="3" robR2="8" thsproductid="40F0023" thsqty="11"/>
    <product closed="no" puttodolly="no" robR6="6" skuid="1030000" total="6"/>
    <product robR6="6" thsproductid="40L4622" thsqty="6"/>
    <product closed="no" puttodolly="no" robR6="5" skuid="1087000" total="5"/>
    <product robR6="5" thsproductid="40L4422" thsqty="5"/>
    ...
  </products>
</storage>
```

Esimerkkitulostus robotin sisältöraportin XML tiedostosta.

Robotin sisältöraportin XML tiedoston rakenteen tarkoitus on täsmälleen sama kuin virheraportinkin. Otsikkotiedot haetaan headers elementin alle. Jokainen otsikko on erikseen määritetty käyttäen header elementtiä. Robotin sisältö on määritetty products elementin alle. Jokaisella tuotteella on oma product elementtimäärittäminen.



```

<?xml version="1.0" encoding="utf-8" ?>
<storagemap cratex="300" cratey="400" desc1="Tyhjä" desc2="Haku" desc3="kpl" desc4="Sekapino"
desc5="Sekapino ilman valittua sku:ta" desc6="Puhdas pino" desc7="Valittu sku:ta sekapinossa" header1="Varaston
sisältö" header2="Robotti 1" orientation="portrait" prj="92600" qty="kpl" shape="1" skuid="Sku ID"
timeheader="Aika">
  <stack xindex="1" xmm="2140" yindex="1" ymm="1275"/>
  <stack xindex="1" xmm="2140" yindex="2" ymm="1775"/>
  <stack xindex="1" xmm="2140" yindex="3" ymm="2275"/>
  <stack xindex="1" xmm="2140" yindex="4" ymm="2775"/>
  <stack xindex="1" xmm="2140" yindex="5" ymm="3275"/>
  <stack xindex="1" xmm="2140" yindex="6" ymm="3775"/>
  ...
  <stack qty="5" skuid="1261000" xindex="5" xmm="3780" yindex="2" ymm="1775"/>
  <stack xindex="5" xmm="3780" yindex="3" ymm="2275"/>
  ...
</storagemap>

```

Esimerkkitulostus varastokarttaraportin XML tiedostosta.

Kolmas raportti on varastokarttaraportti joka eroaa kahdesta muusta robotinsisältö ja virheraportista siinä, että XML tiedoston sisältöä ei näytetä käyttäjälle taulukkomuodossa vaan tietokannan datasta luodaan PDF tiedosto joka näytetään käyttäjälle.

Varastokarttaraportin tiedosto alkaa samalla XML määrittelyllä kuin edeltävätkin raportit. XML tiedoston pääelementtinä on storagemap, jolla on useita attribuutteja projektin tarpeista riippuen. Attribuuttien cratex ja cratey arvoja käytetään varastopaikkojen koon piirtoon. Attribuuttien desc1, desc2, desc3, desc4, desc5, desc6 ja desc7 arvoja käytetään PDF tiedoston oikean yläkulman väriselityksiin. Header1 ja header2 elementin arvoja käytetään PDF tiedoston otsikkoon. Otsikot on tiedoston yläosassa keskitettyinä. Orientation attribuutin arvoa käytetään sivun suuntaamiseen, arvo voi olla joko portrait tai landscape riippuen halutusta paperin asennosta. Attribuutin prj arvossa kerrotaan projektin sisäinen työnnumero. Qty ja skuid attribuuttien arvoja käytetään oikean yläkulman tietolaatikkoon. Shape attribuutin arvoa käytetään varastokartan varastopaikkojen piirtoon. Arvon ollessa yksi piirettään varastopaikat neliöinä ja arvolla kaksi ympyröinä.

Storagemap otsikkoelementin jälkeen tulee yksi tai useampi stack elementti jossa on osittain samat attribuutit kuin otsikkotiedoissa. Attribuuttien xindex ja yindex arvot kertovat robotin varastopaikan koordinaatit. Attribuuttien xmm ja ymm arvoja käytetään varastopaikan piirtoon. Skuid attribuutilla kerrotaan tuotteen nimi.

Kartta piirretään suoraan XML tiedoston datasta. Onnistuneen tietokannan tietojen haun ja XML tiedoston luonnin jälkeen tiedostosta luetaan arvot ja piirretään ne PDF tiedostoon ja näytetään käyttäjälle.

### 4.3 Projekti ja koodin uudelleen käytettävyys

Cimporp toimittaa monia projekteja ja kaikkiin näihin halutaan samat ominaisuudet ja siksi WWP:ssä olevien raporttien luomista piti automatisoida mahdollisimman pitkälle. Tämän takia ympäristön ja raporttien toimintavalmiiksi laittoa varten luotiin omat skriptit jolla työ voidaan automatisoida ja näin ollen helpottaa projektin kehittämistä ja suorittamista. Jokaiseen projektiin tulee samat standardiraportit ja tarvittaessa raportteja voidaan muokata tai luoda lisää. Raporttien teko ja muokkaus on pyritty tehdä mahdollisimman helpoksi.

#### 4.3.1 mkmp.py

Ennen jokaisen projektin aloitusta tuotantopalvelimelle ajetaan aloituskoodi joka suorittaa erinäisiä palvelimen alustuksia ja määrittäviä alkaen tietokannan ympäristön rakentamisesta, tiedostorakenteen oikeuksien muuttamiseen, projektin koodien lataamiseen CVS palvelimelta ja projektilaitteiden väliseen kommunikoinnin mahdollistamiseen. Tähän samaan aloituskoodin on lisätty myös raporttien ympäristön ja siihen tarvittavien ohjelmistojen sekä koodien alustus. Aloituskoodi rakentaa web service palvelun ja siihen tarvittavat osat: Progressin web service servletin asennus ja toimintakuntoon ajamalla wsproxy.py ja raporttien tarvitsemien PHP koodien haku CVS palvelimelta.

#### 4.3.2 wsproxy.py

Komentosarjakieli on kehitetty web service palvelun päivittämiseen ja luomiseen. Koodi tarvitsee toimiakseen määrittäykset haettavista koodeista joista web service palvelu rakennetaan. Määrittäykset on kerrottu XML tiedostossa jossa on listattu kaikki palvelun tarvitsemat tietokantakoodit. XML tiedosto on nimeltään wsproxyconfig.xml. Tiedosto suoritetaan aloituskoodin mkmp.py ajon yhteydessä. Web service palvelun pystyttämiseen tai päivittämiseen riittää wsproxy.py koodin ajo. Komentosarjakieli kääntää tarvittavat koodit ja sen jälkeen ottaa web palvelun pois käytöstä ja palauttaa sen päivitetyn version käyttöön.

```

<?xml version="1.0" ?>
<wsproxyGeneratorConfig>
  <proxy name="ws_reports">
    <file>WS_getstagemap_php </file>
    <file>WS_geterror_php</file>
    <file>WS_printstagemap_php</file>
    <file>WS_printerror_php</file>
    <file>WS_printstoragecont_php</file>
    <file>WS_getdeviceproducts_php</file>
    <file>WS_testconnection</file>
    <file>WS_getstorage_php</file>
    <file>WS_getstoragedevices_php</file>
  </proxy>
</wsproxyGeneratorConfig>

```

Esimerkkitulostus wsproxyconfig.xml tiedoston sisältö. XML tiedostossa on kerrottu kaikki tietokantakyselykoodit joita web service rajanpinnan kautta kutsutaan.

#### 4.4 Havaitut ongelmat

Työssäni törmäsin muutamaan ikävään tilanteeseen ja niistä on kerrottu seuraavassa. Työn tarkoituksena oli luoda web sivujen kokonaisuus ja tuoda kaikki tarvittava tieto yhteen paikkaan. Ikäväkseni sain huomata, että CSS tulkinta on hieman erilainen riippuen käytetystä Internet selaimesta tai sen versiosta. Internet sivujen taulukot näkyi eri tavalla riippuen käytetystä selaimesta. Tästä syystä jouduin luomaan työssäni eri CSS määrittelyt pääasiassa Internet Explorer ja Mozilla Firefox selaimille ja vielä niiden eri versioille. Etenkin Internet Explorerin CSS tulkinta vaihteli suuresti eri selainversioiden välillä ja aiheutti ongelmia.

Toinen työhöni liittyvä ongelma oli web servicen palvelussa käytetyn SOAP viestiliikenteen vian selvittely vikatilanteessa. Tähän ratkaisuksi löytyi ohjelma nimeltään SoapUI. SoapUI on avoimen lähdekoodin ohjelma ja laitteistoriippumaton työkalu SOAP liikenteen ja toiminnallisuuden testaamiseen /48. SoapUI:ssa on graafinen käyttöliittymä ja sitä on helppo käyttää. Ohjelmalle annetaan WSDL tiedoston sijainti ja määrittelyn pohjalta ohjelma luo automaattisesti tarvittavat muuttujat ja operaatiot kutsujen testaamiseen. Yksi suurimmista ongelmista oli selvittää missä muodossa SOAP protokolla tarvitsee päivämäärän. Tähän ongelmaan löytyi ratkaisu SoapUI ohjelmistolla, koska ohjelmalla liikennettä pystyi tarkkailemaan ja tutkimaan.

Kolmas ongelma koski SOAP viestin UUID header tiedon välittämistä SOAP viestien mukana. UUID otsikkotieto piti välittää SOAP viestien mukana, koska käytettävä appserver istuntunnon tilana käytettiin statelessia.

## 5. JATKOKEHITYS

Nyt tehty raportointi pitää sisällään vain tietokantadatan näytön käyttäjälle taulukkomuodossa, joka on hyvä, selkeä ja yleensä riittävä tapa esittää tietoa. Kuitenkin virheraportit voisi pitää sisällään myös virheiden määrän kuvaajan ajan jakson funktiona. Tällöin käyttäjä voisi seurata virheiden esiintymistä helposti ja havainnollisesti valittuna ajanjaksona graafisesta käyrästä. Käyrästä näkyisi kaikki aikavälin virheet ja niiden esiintymistiheys. Tällöin käyttäjä näkisi helposti onko esimerkiksi joku tietty aika jolloin virheitä tulee enemmän kun keskimäärin.

## 6. TULOKSET

Työn tavoitteena oli selkeyttää Cimcorpissa käytettävää robottijärjestelmän raportointia ja tuoda asiakkaalle kaikki tärkeät tiedostot yhdelle sivustolle, johon asiakkaalla on laajasti pääsy. Raporttien luontiin tarvittava ympäristö ja oletusraportit tulevat jokaiseen projektiin automaattisesti. Uusien raporttien luonti projektiin on helppoa ja nopeaa.

Raportit on upotettu olemassa olevaan web-sivustoon. Kaikki raportit on esitetty taulukkomuodossa ja raportit voidaan muuttaa joko CSV tai PDF muotoon. CSV tiedosto on mahdollista avata taulukkolaskentaohjelmassa ja raportin sisällön yksityiskohtaisempi tarkastelu on täten mahdollista. PDF tiedosto mahdollistaa tiedon esittämisen helposti ja sen tulostamisen juuri sellaisessa muodossa, jonka käyttäjä näkee. Raporttien muuttaminen PDF muotoon mahdollistaa myös helpon raporttien jakamisen useille käyttäjille. Raporttien muuttamiseen tarvittavat CSV ja PDF painikkeet löytyvät taulukkoraporttien vasemmasta yläkulmasta, jonka vieressä on taulukon otsikot ja alapuolella itse raportin sisältö.

Raportit voidaan luoda suoraan XML tiedostosta. Luku tapahtuu PHP:llä toteutetulla funktiolla, joka lukee suoraan XML tiedoston ja muuntaa sen html taulukoksi, joka voidaan suoraan näyttää käyttäjälle web-sivulla helposti ja nopeasti. Automaattinen XML:n luku on mahdollista jos XML:n tiedostomuoto on oikea.

## LÄHTEET

1 Salakapakka, HTML ja CSS opas www-sivusto, viitattu 10.6.2014 saatavilla:

[http://salakapakka.net/oppaat/html-ja-css-opas/html\\_opas\\_dokumentti.php](http://salakapakka.net/oppaat/html-ja-css-opas/html_opas_dokumentti.php)

2 W3 schools, HTML Doctype declaration www-sivusto, viitattu 10.6.2014 saatavilla:

[http://www.w3schools.com/tags/tag\\_doctype.asp](http://www.w3schools.com/tags/tag_doctype.asp)

3 Johdatus PHP kieleen www-sivusto, viitattu 10.6.2014 saatavilla:

[http://users.jyu.fi/~kolli/ITK215\\_05/php/](http://users.jyu.fi/~kolli/ITK215_05/php/)

4 Sourceforge, NuSOAP toolkit for PHP www-sivusto, viitattu 10.6.2014

<http://nusoap.sourceforge.net/>

6 R&OS, PHP PDF creation www-sivusto, viitattu 10.6.2014 saatavilla: <http://pdf->

[php-sourceforge.net/pdf-php-code/readme.php](http://pdf-sourceforge.net/pdf-php-code/readme.php)

7 W3 schools, HTML & CSS www-sivusto, viitattu 10.6.2014 saatavilla:

<http://www.w3.org/standards/webdesign/htmlcss>

9 Nokian kaupunki, Johdatus JavaScriptiin, viitattu 10.6.2014 saatavilla:

<http://www.koulut.nokiankaupunki.fi/noklu/materiaalit/js.pdf>

10 W3 schools, jQuery introduction www-sivusto, viitattu 10.6.2014 saatavilla:

[http://www.w3schools.com/jquery/jquery\\_intro.asp](http://www.w3schools.com/jquery/jquery_intro.asp)

11 W3 schools, Extensible Markup Language XML www-sivusto, viitattu 10.6.2014 saatavilla:

<http://www.w3.org/XML/>

12 Microsoft, XML perustietoa www-sivusto, viitattu 12.6.2014 saatavilla:

<http://office.microsoft.com/fi-fi/word-help/XML-perustietoja-aloittelijoille-HA010034022.aspx>

13 W3 schools, XML schema www-sivusto, viitattu 12.6.2014 saatavilla:

[http://www.w3schools.com/schema/schema\\_intro.asp](http://www.w3schools.com/schema/schema_intro.asp)

14 W3 schools, XSD how to? www-sivusto, viitattu 12.6.2014 saatavilla:

[http://www.w3schools.com/schema/schema\\_howto.asp](http://www.w3schools.com/schema/schema_howto.asp)

15 Progress Software 2009, OpenEdge Getting Started: Application and Integration Services www-sivusto, viitattu 12.6.2014 saatavilla:

<http://documentation.progress.com/output/OpenEdge102b/pdfs/gsaais/gsaais.pdf>

16 Progress Software, Who we are www-sivusto, viitattu 12.6.2014 saatavilla:

<http://www.progress.com/about-us>

17 Tomcat Apache, Apache Tomcat open source project www-sivusto, viitattu 12.6.2014 saatavilla: <http://tomcat.apache.org/>

18 Progress software, Advanced Business Language ABL www-sivusto, viitattu 12.6.2014 saatavilla: <http://www.progress.com/products/openedge/features/abl>

19 Prodata, About Progress www-sivusto, viitattu 12.6.2014 saatavilla: [http://www.prodata.lt/EN/PROGRESS/about\\_progress.html](http://www.prodata.lt/EN/PROGRESS/about_progress.html)

21 Progress software 2009, OpenEdge Application Server: Administration www-sivusto, viitattu 12.6.2014 saatavilla: <http://documentation.progress.com/output/OpenEdge102b/pdfs/asadm/asadm.pdf>

23 Progress software 2009, OpenEdge Application Server: Developing Appserver Applications www-sivusto, viitattu 12.6.2014 saatavilla: <http://documentation.progress.com/output/OpenEdge102b/pdfs/asaps/asaps.pdf>

29 Progress software, Knowledgebase What is the WSDL analyzer?, viitattu 12.4.2014 saatavilla: <http://knowledgebase.progress.com/articles/Article/P172455>

30 Paavolainen L. Jyväskylän yliopisto tietotekniikan laitos , SOAP-protokolla www-sivusto, viitattu 12.6.2014 saatavilla: [http://www.ad.jyu.fi/digdoc/TJTSD60\\_2005/soap/soap.xml](http://www.ad.jyu.fi/digdoc/TJTSD60_2005/soap/soap.xml)

31 Progress software 2009, OpenEdge Development: Open Client Introduction and Programming www-sivusto, viitattu 12.6.2014 saatavilla: <http://documentation.progress.com/output/OpenEdge102b/pdfs/dvoci/dvoci.pdf>

32 QAD, System administration reference guide www-sivusto, viitattu 12.6.2104 saatavilla: [http://www.qad.com/Documents/mfgpro/90/tech\\_ref/MFGPRO\\_SystemAdministrationReferenceGuide\\_ProgressDatabase\\_UNIXServer\\_IG\\_v09.pdf](http://www.qad.com/Documents/mfgpro/90/tech_ref/MFGPRO_SystemAdministrationReferenceGuide_ProgressDatabase_UNIXServer_IG_v09.pdf)

33 2K mediat, Java servletit ja JSP www-sivusto, viitattu 12.6.2014 saatavilla: <http://www.2kmediat.com/internetohjelmointi/servlets.asp>

34 Tutorialspoint, Servlets – life cycle www-sivusto, viitattu 12.6.2014 saatavilla: <http://www.tutorialspoint.com/servlets/servlets-life-cycle.htm>

36 Katajisto T. 2000, Java-servletit, Tietotekniikan LuK – tutkielma. Jyväskylän yliopisto. Viitattu 12.6.2014. [http://www.mit.jyu.fi/opetus/opinnayte/LuK/Java-servletit/#\\_Toc501443614](http://www.mit.jyu.fi/opetus/opinnayte/LuK/Java-servletit/#_Toc501443614)

37 Englander R 2002, Java and SOAP. O'reilly & Associates, Inc.

38 Kesola V 2013, Web-palveluiden koostaminen: web- ja RESTful-palvelut. Seminaari. Helsingin yliopisto 17.2.2013. Viitattu 15.6.2014. <http://www.cs.helsinki.fi/u/vkesola/semma/seminaari-ws.pdf>

40 About.com, What is python www-sivusto, viitattu 15.6.2014 saatavilla: [http://python.about.com/od/gettingstarted/ss/whatispython\\_2.htm](http://python.about.com/od/gettingstarted/ss/whatispython_2.htm)



41 Reportlab.com, ReportLab open-source PDF toolkit www-sivusto, viitattu 15.6.2014 saatavilla:  
<http://www.reportlab.com/opensource/>

42 Python.org, What is python? www-sivusto, viivattu 15.6.2014 saatavilla:  
<http://docs.python.org/2/faq/installed.html#what-is-python>

43 Progress software, Knowledgebase Accessing a session managed web service from a 4gl client  
www-sivusto, viitattu 15.6.2014 saatavilla:  
<http://knowledgebase.progress.com/articles/Article/P72655>

47 Progress software 2009, OpenEdge Development: Web Services www-sivusto, viitattu  
15.6.2014 saatavilla:  
<http://documentation.progress.com/output/OpenEdge102b/pdfs/dvwsv/dvwsv.pdf>

48 SoapUI.org, What is SoapUI? www-sivusto, viitattu 15.6.2014 saatavilla:  
<http://www.soapui.org/About-SoapUI/what-is-soapui.html>

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="ws_reports" targetNamespace="urn:cimcorp-com" XMLNs:tns="urn:cimcorp-com" XMLNs:S2="urn:cimcorp-com:ws_reports" XMLNs:soapenc="http://schemas.xmlsoap.org/soap/encoding/" XMLNs:xsd="http://www.w3.org/2001/XMLSchema" XMLNs:prodata="urn:schemas-progress-com:XML-prodata:0001" XMLNs:soap="http://schemas.xmlsoap.org/wsdl/soap/" XMLNs:wsdl="http://schemas.xmlsoap.org/wsdl/" XMLNs:S1="urn:soap-fault:details" XMLNs="http://schemas.xmlsoap.org/wsdl/">
<wsdl:documentation>EncodingType=RPC_ENCODED, Proxygen_Product=Progress Version 10.2B</wsdl:documentation>
<wsdl:types>
<schema elementFormDefault="unqualified" targetNamespace="urn:soap-fault:details"
XMLNs="http://www.w3.org/2001/XMLSchema"><complexType name="FaultDetail"><sequence><element name="errorMessage"
type="xsd:string"/><element name="requestID" type="xsd:string"/></sequence></complexType></schema>
<schema elementFormDefault="unqualified" targetNamespace="urn:cimcorp-com:ws_reports"
XMLNs="http://www.w3.org/2001/XMLSchema"><import namespace="http://schemas.xmlsoap.org/soap/encoding/"><complexType
name="ws_reportsID"><sequence><element name="UUID" type="xsd:string"/></sequence></complexType><complexType
name="WS_getdeviceproducts_php_tt_skuRow"><sequence><element name="skuid" nillable="true"
type="xsd:int"/></sequence></complexType><complexType
name="ArrayOfWS_getdeviceproducts_php_tt_skuRow"><complexContent><restriction base="soapenc:Array"><attribute ref="soapenc:arrayType"
wsdl:arrayType="S2:WS_getdeviceproducts_php_tt_skuRow[]"/></restriction></complexContent></complexType><complexType
name="WS_getstoragedevices_php_tt_deviceRow"><sequence><element name="deviceid" nillable="true" type="xsd:int"/><element
name="deviceName" nillable="true" type="xsd:string"/></sequence></complexType><complexType
name="ArrayOfWS_getstoragedevices_php_tt_deviceRow"><complexContent><restriction base="soapenc:Array"><attribute
ref="soapenc:arrayType"
wsdl:arrayType="S2:WS_getstoragedevices_php_tt_deviceRow[]"/></restriction></complexContent></complexType></schema>
</wsdl:types>
<wsdl:message name="ws_reports_WS_printerror_php">
<wsdl:part name="daStartDate" type="xsd:dateTime"/>
<wsdl:part name="daEndDate" type="xsd:dateTime"/>
<wsdl:part name="sessionId" type="xsd:string"/>
<wsdl:part name="sequenceid" type="xsd:string"/>
<wsdl:part name="lang" type="xsd:int"/>
<wsdl:part name="productnum" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="ws_reports_WS_getdeviceproducts_php">
<wsdl:part name="devID" type="xsd:int"/>
</wsdl:message>
<wsdl:message name="ws_reports_WS_testconnectionResponse">
<wsdl:part name="o_IOK" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="ws_reports_WS_geterror_phpResponse">
</wsdl:message>
<wsdl:message name="ws_reports_WS_getstoragemap_php">
<wsdl:part name="sessionId" type="xsd:string"/>
<wsdl:part name="mapDeviceID" type="xsd:int"/>
<wsdl:part name="mapProductID" type="xsd:string"/>
<wsdl:part name="lang" type="xsd:int"/>
<wsdl:part name="productnum" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="ws_reportsConnect">
<wsdl:part name="userId" type="xsd:string"/>
<wsdl:part name="password" type="xsd:string"/>
<wsdl:part name="appServerInfo" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="ws_reports_WS_getdeviceproducts_phpResponse">
<wsdl:part name="tt_sku" type="S2:ArrayOfWS_getdeviceproducts_php_tt_skuRow"/>
</wsdl:message>
<wsdl:message name="ws_reports_WS_printstoragecont_php">
<wsdl:part name="sequenceid" type="xsd:string"/>
<wsdl:part name="sessionId" type="xsd:string"/>
<wsdl:part name="lang" type="xsd:int"/>
<wsdl:part name="productnum" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="ws_reportsConnectResponse">
</wsdl:message>
<wsdl:message name="ws_reports_WS_getstoragedevices_phpResponse">
<wsdl:part name="tt_device" type="S2:ArrayOfWS_getstoragedevices_php_tt_deviceRow"/>
</wsdl:message>
<wsdl:message name="ws_reportsID">
<wsdl:part name="ws_reportsID" type="S2:ws_reportsID"/>
</wsdl:message>
<wsdl:message name="ws_reports_WS_geterror_php">
<wsdl:part name="daStartDate" type="xsd:dateTime"/>
<wsdl:part name="daEndDate" type="xsd:dateTime"/>
<wsdl:part name="sessionId" type="xsd:string"/>
<wsdl:part name="sequenceid" type="xsd:string"/>

```

```

<wsdl:part name="lang" type="xsd:int"/>
<wsdl:part name="productnum" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="ws_reports_WS_getstorage_phpResponse">
</wsdl:message>
<wsdl:message name="ws_reports_WS_getstoragedevices_php">
</wsdl:message>
<wsdl:message name="ws_reportsRelease">
</wsdl:message>
<wsdl:message name="ws_reports_WS_printstoragemap_phpResponse">
</wsdl:message>
<wsdl:message name="ws_reports_WS_printstoragecont_phpResponse">
</wsdl:message>
<wsdl:message name="ws_reports_WS_printererror_phpResponse">
</wsdl:message>
<wsdl:message name="ws_reports_WS_getstorage_php">
<wsdl:part name="sessionid" type="xsd:string"/>
<wsdl:part name="lang" type="xsd:int"/>
<wsdl:part name="productnum" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="ws_reports_WS_printstoragemap_php">
<wsdl:part name="sequenceid" type="xsd:string"/>
<wsdl:part name="i_iDeviceID" type="xsd:int"/>
<wsdl:part name="i_iSkuld" type="xsd:int"/>
<wsdl:part name="sessionid" type="xsd:string"/>
<wsdl:part name="lang" type="xsd:int"/>
<wsdl:part name="productnum" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="FaultDetailMessage">
<wsdl:part name="FaultDetail" type="SI:FaultDetail"/>
</wsdl:message>
<wsdl:message name="ws_reportsReleaseResponse">
</wsdl:message>
<wsdl:message name="ws_reports_WS_testconnection">
</wsdl:message>
<wsdl:message name="ws_reports_WS_getstoragemap_phpResponse">
</wsdl:message>
<wsdl:portType name="ws_reportsObj">
<wsdl:operation name="Connect_ws_reports" parameterOrder="userId password appServerInfo">
<wsdl:input message="tns:ws_reportsConnect"/>
<wsdl:output message="tns:ws_reportsConnectResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="Release_ws_reports">
<wsdl:input message="tns:ws_reportsRelease"/>
<wsdl:output message="tns:ws_reportsReleaseResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="WS_getdeviceproducts_php" parameterOrder="tt_sku devID">
<wsdl:input message="tns:ws_reports_WS_getdeviceproducts_php"/>
<wsdl:output message="tns:ws_reports_WS_getdeviceproducts_phpResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="WS_geterror_php" parameterOrder="daStartDate daEndDate sessionid sequenceid lang productnum">
<wsdl:input message="tns:ws_reports_WS_geterror_php"/>
<wsdl:output message="tns:ws_reports_WS_geterror_phpResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="WS_getstorage_php" parameterOrder="sessionid lang productnum">
<wsdl:input message="tns:ws_reports_WS_getstorage_php"/>
<wsdl:output message="tns:ws_reports_WS_getstorage_phpResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="WS_getstoragedevices_php" parameterOrder="tt_device">
<wsdl:input message="tns:ws_reports_WS_getstoragedevices_php"/>
<wsdl:output message="tns:ws_reports_WS_getstoragedevices_phpResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="WS_getstoragemap_php" parameterOrder="sessionid mapDeviceID mapProductID lang productnum">
<wsdl:input message="tns:ws_reports_WS_getstoragemap_php"/>
<wsdl:output message="tns:ws_reports_WS_getstoragemap_phpResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="WS_printererror_php" parameterOrder="daStartDate daEndDate sessionid sequenceid lang productnum">

```

```

<wsdl:input message="tns:ws_reports_WS_printererror_php"/>
<wsdl:output message="tns:ws_reports_WS_printererror_phpResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="WS_printstoragecont_php" parameterOrder="sequenceid sessionid lang productnum">
<wsdl:input message="tns:ws_reports_WS_printstoragecont_php"/>
<wsdl:output message="tns:ws_reports_WS_printstoragecont_phpResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="WS_printstoragemap_php" parameterOrder="sequenceid i_iDeviceID i_iSkuID sessionid lang productnum">
<wsdl:input message="tns:ws_reports_WS_printstoragemap_php"/>
<wsdl:output message="tns:ws_reports_WS_printstoragemap_phpResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
<wsdl:operation name="WS_testconnection" parameterOrder="o_IOK">
<wsdl:input message="tns:ws_reports_WS_testconnection"/>
<wsdl:output message="tns:ws_reports_WS_testconnectionResponse"/>
<wsdl:fault name="ws_reportsFault" message="tns:FaultDetailMessage"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ws_reportsObj" type="tns:ws_reportsObj">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="Connect_ws_reports">
<soap:operation soapAction="" style="rpc"/>
<wsdl:input>
<soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
</wsdl:input>
<wsdl:output>
<soap:header message="tns:ws_reportsID" part="ws_reportsID" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:cimcorp-com:ws_reports">
</soap:header>
<soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
</wsdl:output>
<wsdl:fault name="ws_reportsFault">
<soap:fault name="ws_reportsFault" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/92600/wsa1"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="Release_ws_reports">
<soap:operation soapAction="" style="rpc"/>
<wsdl:input>
<soap:header message="tns:ws_reportsID" part="ws_reportsID" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:cimcorp-com:ws_reports" wsdl:required="true">
</soap:header>
<soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
</wsdl:input>
<wsdl:output>
<soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
</wsdl:output>
<wsdl:fault name="ws_reportsFault">
<soap:fault name="ws_reportsFault" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/92600/wsa1"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="WS_getdeviceproducts_php">
<soap:operation soapAction="" style="rpc"/>
<wsdl:input>
<soap:header message="tns:ws_reportsID" part="ws_reportsID" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:cimcorp-com:ws_reports" wsdl:required="true">
</soap:header>
<soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
</wsdl:input>
<wsdl:output>
<soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
</wsdl:output>
<wsdl:fault name="ws_reportsFault">
<soap:fault name="ws_reportsFault" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/92600/wsa1"/>
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="WS_geterror_php">
<soap:operation soapAction="" style="rpc"/>
<wsdl:input>

```



```

    <wsdl:fault name="ws_reportsFault">
      <soap:fault name="ws_reportsFault" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/92600/wsa1"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="WS_printstoragecont_php">
    <soap:operation soapAction="" style="rpc"/>
    <wsdl:input>
      <soap:header message="tns:ws_reportsID" part="ws_reportsID" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:cimcorp-com:ws_reports" wsdl:required="true">
        </soap:header>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
      </wsdl:input>
      <wsdl:output>
        <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
      </wsdl:output>
      <wsdl:fault name="ws_reportsFault">
        <soap:fault name="ws_reportsFault" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/92600/wsa1"/>
      </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="WS_printstoragemap_php">
      <soap:operation soapAction="" style="rpc"/>
      <wsdl:input>
        <soap:header message="tns:ws_reportsID" part="ws_reportsID" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:cimcorp-com:ws_reports" wsdl:required="true">
          </soap:header>
          <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
        </wsdl:output>
        <wsdl:fault name="ws_reportsFault">
          <soap:fault name="ws_reportsFault" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/92600/wsa1"/>
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="WS_testconnection">
        <soap:operation soapAction="" style="rpc"/>
        <wsdl:input>
          <soap:header message="tns:ws_reportsID" part="ws_reportsID" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:cimcorp-com:ws_reports" wsdl:required="true">
            </soap:header>
            <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
          </wsdl:input>
          <wsdl:output>
            <soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:cimcorp-com:ws_reports"/>
          </wsdl:output>
          <wsdl:fault name="ws_reportsFault">
            <soap:fault name="ws_reportsFault" use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/92600/wsa1"/>
          </wsdl:fault>
        </wsdl:operation>
      </wsdl:binding>
      <wsdl:service name="ws_reportsService">
        <wsdl:port name="ws_reportsObj" binding="tns:ws_reportsObj">
          <wsdl:documentation></wsdl:documentation>
          <soap:address location="http://localhost:8080/92600/wsa1"/>
        </wsdl:port>
      </wsdl:service>
    </wsdl:definitions>

```