

EVTEK-ammattikorkeakoulu
Mediatekniikan koulutusohjelma

Hannu Peltola
**Palvelusuuntautunut arkkitehtuuri ja xml-
verkkopalvelut**

Insinööritö 12.4.2007

Ohjaaja: toimitusjohtaja Jukka Kuikanvirta

Ohjaava opettaja: yliopettaja Kari T. Aaltonen

EVTEK-AMMATTIKORKEAKOULU INSINÖÖRITYÖN
TIIVISTELMÄ

Tekijä Otsikko	Hannu Peltola Palvelusuuntaunut arkkitehtuuri ja xml-verkkopalvelut
Sivumäärä Aika	56 sivua 12.4.2007
Koulutusohjelma	mediatekniikka
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	toimitusjohtaja Jukka Kuikanvirta yliopettaja Kari T. Aaltonen
<p>Insinööriyössä selvitettiin palvelusuuntautuneen arkkitehtuurin ja verkkopalveluiden käsitteitä.</p> <p>Yritysten hajautetut ohjelmistot ja perinteiset kahdenväliset palveluratkaisut ovat jäykkiä taipumaan nopeisiin liiketoiminnan muutoksiin. Uudenlaisella arkkitehtuuriajattelulla pyritään sopeuttamaan vanhat jo olemassa olevat palvelut helposti hallittaviksi ja muokattaviksi osiksi kokonaisuutta. Samalla rinnalle rakennettavat uudet verkkopalvelut käyttävät avoimia valmistajista riippumattomia standardeja. Näin taataan yhteensopivuus ja toisaalta riippumattomuus muiden toimijoiden ja yritysten palveluja tarjoavien tai pyytävien järjestelmien kanssa.</p> <p>Työssä rakennettiin tilaajan uusille verkkosivuille asiakaspalvelua ja ylläpitoa parantavia toimintoja käyttäen valmiita moduuleja. Niiden avulla säästettiin aikaa ja saatiin monipuolisia toimintoja integroitua nopeasti osaksi sivujen tekoprosessia.</p> <p>Verkkosivuilla käytettiin xml-hakupalvelurajapintaa, jonka kautta suoritetaan hakuja laajasta lääketieteen artikkelitietokannasta. Rajapintamäärittelyn pohjalta tehtiin PHP-kielellä ohjelma, joka muuttaa xml-muotoiset hakutulokset linkkilistaksi käyttäen xsl-muunnosta. Tämä toiminto yhdistettiin sivujen kysymysvastaustietokannan hakutoimintoon. Tuloksena oli sivuston laajaa kohderyhmää paremmin palveleva kokonaisuus.</p>	
Hakusanat	verkkopalvelu, palvelusuuntaunut arkkitehtuuri, xml, soa, web services

Author Title	Hannu Peltola Service-oriented architecture and XML-web services
Number of Pages Date	56 12 April 2007
Degree Programme	Media Technology
Degree	Bachelor of Engineering
Instructor Supervisor	Jukka Kuikanvirta, Managing Director Kari T. Aaltonen, Principal Lecturer
<p>The final thesis clarifies the concepts of service-oriented architecture and web services. Traditional server-client and point-to-point solutions and distributed architectures are not flexible enough to react to fast developing business processes. A new architectural model integrates legacy services into an easily manageable form, that can communicate with new, vendor neutral services. The primary target in general is to create services that are reusable and autonomic units, yet loosely coupled. Open standards and protocols give all vendors and parties an equal forum to develop composable services that can be discovered and used by others.</p> <p>Part of the thesis was the construction of a customer's web site. It included using free open source modules. They enabled integrating services into the web site, that would otherwise have been out of the time and resource frame.</p> <p>Using a web service, it was possible to add a search function to the pages, that explores vast medical article databases. The database search yields an XML document, that is transformed by an XSL style definition to be viewed by a browser. This is done with PHP functions. Along with the pages' own search facility, an extended search of an outside database adds considerable value to the services offered to the pages' users.</p>	
Keywords	web service, service-oriented architecture, xml, soa

Termit

palvelusuuntautunut arkkitehtuuri	(engl. Service Oriented Architecture, SOA) ohjelmistoarkkitehtuurin suunnittelutapa
verkkopalvelu	(engl. Web Services) yrityksen tms. tahon tuottama verkon ja xml-viestien välityksellä toimiva palvelu, esim. laskutus- tai tilauspalvelu
xml	(engl. Extented Markup Language) rakenteellinen tiedon kuvauskieli
wSDL	(engl. Web Service Definition Language) verkkopalvelun kuvauskieli
soap	protokolla, jonka välityksellä palveluiden välinen viestiliikenne kulkee
dtd	(engl. Document Type Definition) xml-dokumentin tyyppimäärittelytiedosto
xml-skeema	(engl. XML-Schema) xml-dokumentin rakenneosien tyyppimäärittelytiedosto
xsl	(engl. Extensible Stylesheet Language) tyylitiedosto, jonka avulla xml-dokumentti voidaan muuttaa esim. selaimella kasottavaan muotoon
uddi	(engl. Universal Description, Discovery and Integration) määrittelee standardina rekisterin rakenteen, jonka avulla voi löytää siihen kirjautuneet verkkopalvelut, niiden tuottajat ja sisällön
php	Internet-ohjelmointikieli

Sisällys

Tiivistelmä

Abstract

Termit

1 Johdanto	7
2 Ohjelmistoarkkitehtuurit	9
2.1 Palvelusuuntautunut arkkitehtuuri.....	9
2.2 Ohjelmistoarkkitehtuuri ja yrityksen arkkitehtuuri.....	9
3 Palvelusuuntautunut arkkitehtuuri	12
3.1 Ohjelmarajapinta ja käyttöliittymä.....	12
3.2 Palvelu käsitteenä.....	12
3.3 Palveluiden suhteet.....	13
3.4 Palveluiden välinen viestintä.....	13
3.5 Palveluiden suunnittelu.....	14
3.6 Palvelusuuntautuneen arkkitehtuurin käsitteet.....	15
4 Palvelusuuntautuneen arkkitehtuurin yleiset periaatteet	19
4.1 Määritelmä.....	19
4.2 Kierrätettävyys.....	19
4.3 Palvelusopimus.....	21
4.4 Löyhä yhteys.....	22
4.5 Palvelun toimintalogiikan kapseloiminen.....	23
4.6 Palveluiden yhdistettävyys.....	24
4.7 Palveluiden autonomia.....	25
4.8 Tilattomuus	26
4.9 Palvelun löydettävyys.....	27
5 Xml ja verkkopalvelut	28
5.1 Rakenteisen tiedon määrittely.....	28
5.2 Xml-dokumentin rakenne.....	29
5.3 Tyypimäärittelyt.....	30
6 Verkkopalvelut	32
6.1 Teknologiakehys.....	32

6.2	Verkkopalvelun kuvauskieli	33
6.3	Palvelun kuvauksen mainostaminen ja löytäminen.....	34
6.4	Palveluiden kommunikointi.....	35
6.5	Verkkopalvelut ja palvelusuuntautunut arkkitehtuuri.....	36
6.6	Esimerkkejä palvelusuuntautuneen arkkitehtuurin toteutuksista	38
7	Modulaarinen Internet-sivuston rakentaminen.....	40
7.1	Suunnittelu.....	40
7.2	Rakenne ja sisältö.....	41
7.3	Ulkoasu.....	43
7.4	Tekninen toteutus.....	44
8	Verkkopalvelun liittäminen sivuille.....	47
8.1	E-utilities-rajapinta	47
8.2	Hakupalvelun toteutus.....	48
9	Yhteenveto.....	53
Lähteet		55

1 Johdanto

Tietotekniikka on ratkaissut monet yritysten logistiset ja hallinnolliset ongelmat. Sen avulla hallitaan laajoja tietomääriä ja pystytään nopeasti yhdistämään ja laskemaan tietoja useista eri lähteistä. Yritysten järjestelmiä on rakennettu ja kehitetty vuosikymmeniä, ja vanhoja on kadonnut uusien tekniikoiden myötä. Silti edelleenkin on käytössä verrattain vanhoja ohjelmistoja, jotka saattavat olla vastuussa yrityksen keskeisistä toiminnoista. Uusia ohjelmistoja varten räätälöidään sopivia muunto-ohjelmia tai palikoita, joiden avulla vanhojen järjestelmien toiminnot voidaan yhdistää. Tuloksena on hyvinkin monimutkaisia infrastruktuureita, joiden päivitys, muokkaus ja yhdistäminen vaativat kalliita, pitkäkestoisia ja asiantuntijoita vaativia projekteja.

Internet on mahdollistanut yritysten palveluiden leviämisen maailmanlaajuisesti. Tämä koskee sekä yrityksen sisäisiä tietoverkkoja että yhteyksiä muihin yrityksiin. Verkkopalvelustandardit muodostivat teknisen kehyksen tiedon muotoilulle ja siirrolle. On syntynyt mahdollisuus rakentaa verkkoratkaisuja, joiden avulla yritykset pystyvät standardoidusti kommunikoidaan ja hyödyntämään toistensa tarjoamia palveluita. Aikaisemmin yritysten välistä viestintää varten oli yleensä toteutettu kahdenvälinen ratkaisu, joka edellytti tiivistä sitoutumista. Muutokset järjestelmissä olivat hankalia toteuttaa, sillä järjestely saattoi edellyttää toimia myöskin vastapuolelta. Esimerkkinä tällaisesta on laskutus ja tilausten tekeminen yrityksen ja alihankkijan välillä. Yritysten välille on luotu automaattinen laskutus- ja tilausjärjestelmä, jonka avulla suoritetaan tilaukset ja alihankkija voi helposti laskuttaa asiakastaan.

Palvelusuuntautunut arkkitehtuuri pyrkii muuttamaan olemassa olevat palvelut yhteensopiviksi ja itsenäisiksi yksiköiksi. Niihin on mahdollista liittää tai ne ovat liitettävissä toisiin palveluihin omassa tai muissa yrityksissä. Vanhoja rakenteita ei ole syytä purkaa uuden tieltä, eikä se aina olisi edes taloudellisestikaan mahdollista. Esimerkiksi tilaus- ja laskutusjärjestelmät muokataan sellaisiksi, että minkä tahansa muunkin yrityksen verkkopalvelu pystyy niitä hyödyntämään. Näin saavutetaan löyhä yhteys palveluiden välille.

Insinööriyön teknisenä kohteena on Sexpo-säätiön Internet-palveluiden uusiminen. Sexpo-säätiö kouluttaa, neuvoo, konsultoi ja tuottaa julkaisuja ja koulutusmateriaalia. Säätiöllä on pari- ja seksuaaliterapiatoimintaa, kirjasto, erilaisia projekteja ja seksuaalieettinen lautakunta. Verkkosivujen uudistamisen tarkoitus on verkossa olevien palveluiden kehittäminen ja saattaminen mahdollisimman monen käyttäjäryhmän ulottuville. Säätiö toimii julkisella rahoituksella, ja Internetin hyödyntäminen monipuolisesti tuo palvelut tasapuolisesti kaikille.

Sexpo-säätiön verkkosivustolla on runsaasti ammattilaisten tuottamaa tietoa. Lisäarvona asiakkaille sivujen hakupalveluun yhdistetään Duodecim Oy:n xml-verkkopalveluna tuottama haku Terveyskirjastosta. Se toteutetaan Duodecimin E-utilities-rajapinnan avulla. E-utilities sisältää palvelun käytön määrittelyn ja opastuksen haun rakentamiselle.

2 Ohjelmistoarkkitehtuurit

2.1 Palvelusuuntautunut arkkitehtuuri

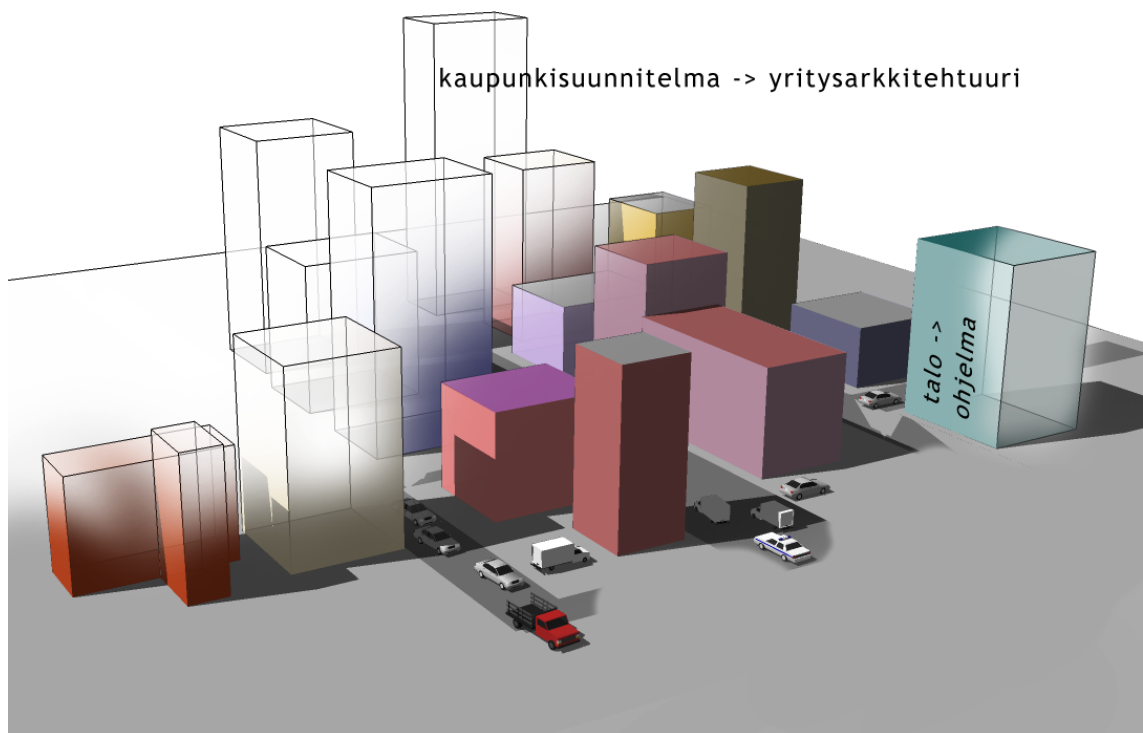
Palvelusuuntautunut arkkitehtuuri (Service Oriented Architecture, SOA) on yleisen määritelmän mukaan hajautettu ongelman tai palvelun ratkaisu. Tällä tarkoitetaan laajaa ohjelmistokokonaisuutta, joka koostuu pienistä ja helposti hallittavista erillisistä ohjelmista. Jokainen ratkaisun elementti huolehtii omasta tietystä osa-alueestaan. Arkkitehtuurilla voidaan ratkaista erityyppisiä yrityksen toimintalogistiikan ongelmia. Se ei ole pelkästään joukko teknisiä määritelmiä tai standardeja, vaan nimenomaan tapa ajatella ja jäsentää yrityksen tuottamien ja tarvitsemien palveluiden yhdistäminen tekniseen infrastruktuuriin. (1, s. 32.)

Ajatuksena hajautettu, modulaarinen ohjelmistoarkkitehtuuri ei ole uusi. Esimerkkinä voisi olla yritysten välinen yhteys, jonka kautta välitetään tietoa varastosaldoista tai hoidetaan laskutusta. Ratkaisut ovat saattaneet johtaa hyvinkin tiiviiseen ohjelmalliseen sitoutumiseen toiseen yritykseen. Yritysten välistä liikennettä varten on räätälöity yksilöllisiä ratkaisuja, palveluita (services), jotka hoitavat esimerkiksi Internetin kautta niille määritettyjä tehtäviä. Palvelusuuntautuneen arkkitehtuurin keskeisenä periaatteena on suunnitella ja toteuttaa yritysten tarjoamat palvelut siten, että rajapinta ulospäin toimii, vaikka ohjelmaa muuteltaisiin tai kehiteltäisiin. Näin mahdollistetaan yritysten palveluiden itsenäisyys toimia ja kehittyä toisista yrityksistä ja palveluista riippumatta, mutta ei kuitenkaan eristäytyneenä. Edellytyksenä tämän toteutumiselle on tiettyjen periaatteiden ja standardien mukaan toimiminen. Palvelukeskeinen arkkitehtuuri esittää nämä toimintaperiaatteet. (1, s. 33.)

2.2 Ohjelmistoarkkitehtuuri ja yrityksen arkkitehtuuri

Tekninen ohjelmistoarkkitehtuuri on ollut olemassa yhtä kauan kuin automaattinen, tietokoneilla suoritettu tietojenkäsittelykin. Alkuaikojen yksinkertaisten, suoraviivaisten sovellusten suunnittelussa ei juurikaan arkkitehtuurisia näkökohtia tarvinnut ottaa

huomioon. Ohjelmistojen laajenemisen ja monimutkaistumisen myötä syntyi tarve suunnitella ja standardoida teknisiä ratkaisuja. Se loi pohjan ohjelmistoarkkitehtuurin synnylle. Yrityksessä tai organisaatiossa voi olla monia eri ohjelmistoalustoja ja ohjelmistoja, joilla hoidetaan logistiikkaa, kirjanpitoa, henkilöstö- ja asiakashallintaa ja sisäisiä sekä mahdollisesti ulospäin tarjottuja palveluja. Näistä jokaisella on oma sovelluskohtainen arkkitehtuurinsa ja määrityksensä. Sovellustason arkkitehtuurin ensisijaisena tehtävänä on luonnollisesti toimia sovelluksen toteutuskarttana, mutta palveluilla myös yrityksen pitkän tähtäimen tietotekniikkastrategian työkaluna. Yrityksen arkkitehtuuri (enterprise architecture) kontrolloi ja ylläpitää korkean tason suunnitelmaa kaikkien ohjelmistojen ja palveluiden kokonaisuudesta. Sitä voisi verrata kaupunkisuunnitelmaan, kun ohjelmistoarkkitehtuuri on yksittäisen korttelin tai talon suunnitelma (kuva 1). (1, s. 86.)



Kuva 1. Yritysarkkitehtuuri ja ohjelmistoarkkitehtuuri kohdentuvat eri tasoille

Ohjelmistoarkkitehtuuri on joukko määritelmiä, jotka kuvailevat ohjelmiston osien teknisiä rakenteita, rajoituksia, komponenttien ominaisuuksia ja rajapintoja sekä niiden toimintaa järjestelmän osana. Arkkitehtuuria voidaan pitää järjestelmän ylemmän tason rakennuspiirrustuksena. (4, s. 56.)

Ohjelmistoarkkitehtuurilla on erilaisia määritelmiä. ANSI / IEEE -standardi määrittelee ohjelmistoarkkitehtuurin olevan järjestelmän perusta ja pohja, joka määrittelee sen rakenteelliset komponentit, niiden väliset suhteet sekä toimintaympäristön ja kehityksen sekä hallinnoinnin periaatteet. Määritelmä on tarkoituksellisesti moneen sopiva, sillä se pyrkii selventämään yleisiä ohjelmistojärjestelmien rakenne-elementtejä. Nämä voivat olla niin fyysisiä ohjelman osia ja niiden toiminnallisia linkkejä kuin periaatteellisia organisaation rakenteita. (16.)

Leikkisämmän määritelmän mukaan ohjelmistoarkkitehtuuri on termi, jota suuri joukko ihmisiä yrittää määritellä huonolla tuloksella. Siinä on kaksi päätekijää: toinen on ohjelmistojärjestelmän purkaminen pieniin osiin ja toinen on päätökset, joita on vaikea muuttaa jälkikäteen. (17, s. 12.)

3 Palvelusuuntautunut arkkitehtuuri

3.1 Ohjelmarajapinta ja käyttöliittymä

Palvelusuuntautunut arkkitehtuuri koostuu teknisesti loppukäyttäjän rajapinnasta eli käyttöliittymästä, palveluista, palveluiden tietovarastoista sekä palveluiden käyttämistä väylistä (4, s. 58).

Käyttöliittymät toimivat rajapintana ohjelmiston toimintojen ja ominaisuuksien ja loppukäyttäjän välillä. Käyttöliittymien välityksellä suoritetaan kaikki yrityksen tietojärjestelmien toiminnot. Yksinkertaisimmillaan kysymyksessä voi olla www-pohjainen sovellus tai asiakassovellus (client), joka on suoraan kosketuksessa käyttäjän kanssa. Rajapinnat voivat toki toimia myös käyttäjälle näkymättömissä, esimerkiksi eräajo-ohjelmina. Käyttöliittymä on kuitenkin aina prosessin käynnistäjä ja tulosten vastaanottaja. (4, s. 59.)

3.2 Palvelu käsitteenä

Yrityksen automaatio-sovellus voi koostua esimerkiksi tietyistä prosesseista ja sen hallinnasta. Prosessi määrittelee sovelluksen toiminnot, jotka tapahtuvat halutussa järjestyksessä sovittujen sääntöjen ja olosuhteiden mukaan. Kun sovellus toteutetaan palveluina, kukin palvelu voi sisältää yksittäisen vaiheen tai useasta vaiheesta koostuvan prosessin osan. Palvelu voi myös sisältää kokonaisen prosessin, joka koostuu pienemmistä palveluista. Näin palvelun suorittama logiikka ja sen laajuus voivat vaihdella suurestikin. (1, s. 35.)

Palvelu rakentuu tiedon ja toimintojen logiikasta, käyttöliittymistä sekä määritelmistä. Määritelmä sisältää epämuodollisen selvityksen palvelun tarkoituksesta, toiminnoista ja käytöstä. Sen muoto voi vaihdella riippuen palvelun tyypistä. Osana palvelun määritelmää voi olla muodollinen, esimerkiksi wsdl:ään (Web Service Definition Language) perustuva rajapinnan määritelmä. Se mahdollistaa riippumattomuuden

ohjelmointikielen, verkkoprotokollan ja ajoympäristön suhteen. Jokainen palvelu kuitenkin tarvitsee jonkinlaisen määritelmän tai teknisen selvityksen käytön rajapinnasta, vaikka muodollista wsdl-tyyppistä määrittystä ei olisikaan saatavilla. (4, s. 59.)

Palvelun toiminnallisuus välitetään loppukäyttäjälle verkon kautta käyttöliittymän välityksellä. Toteutus tuottaa vaaditun toiminnallisuuden ja siihen liittyvän tiedon määritelmän mukaisesti. Se koostuu ohjelmista, tietokannoista ja konfigurointimääritelmistä. Jokainen palvelu on tarkoin määritelty toiminto, joka on tai voi olla osa suurempaa yrityksen tuottamaa, loppukäyttäjälle suunnattua palvelua. Asiakkaan tai loppukäyttäjän näkökulmasta palvelu on kuin musta laatikko: suljettu kokonaisuus, joka suorittaa sille määritetyn tehtävän osana laajempaa kokonaisuutta. (4, s. 60.)

3.3 Palveluiden suhteet

Palvelusuuntautuneen arkkitehtuurin periaatteiden mukaan palvelut voivat käyttää muita palveluita ja ohjelmia. Voidakseen ymmärtää ja hyödyntää toisiaan on palveluiden oltava tietoisia toisistaan. Tämä toteutetaan käyttämällä palvelun kuvausta (service description). Yksinkertaisimmillaan palvelun kuvaus sisältää tiedon palvelun nimestä ja sijainnista sekä tiedon rakenteen vaatimuksista. Tällaista palvelun kuvausta käyttävää yhteyttä palveluiden välillä kutsutaan 'löyhäksi yhteydeksi' (loosely coupled). (1, s. 35.)

3.4 Palveluiden välinen viestintä

Voidakseen tuottaa haluttuja tuloksia ohjelmien on kyettävä kommunikoimaan ja vaihtamaan tietoja keskenään. Tähän tarvitaan yhteys, väylä, joka säilyttää löyhän yhteyden palveluiden välillä. Yksi kommunikointimenetelmä on viestien lähettäminen. Kun palvelu lähettää viestin, se menettää samalla kontrollin siihen. Siksi viestien ja palveluidenkin on oltava itsenäisiä voidakseen toimia vastaanottavassa ohjelmassa toivotulla tavalla. Näitä periaatteita on noudatettu aikaisemminkin hajautetussa

arkkitehtuurissa. Palvelusuuntautuneisuus eroaa kuitenkin siinä, miten nämä kolme ydinosa, palvelut, kuvaukset ja viestintä, on suunniteltu ja toteutettu. (1, s. 36.)

Oleellista kommunikointiväylän toiminnassa on erilaisten tekniikoiden hyväksyminen. Yrityksissä käytetään erilaisia palvelinalustoja, käyttöjärjestelmiä, ohjelmointitekniikoita ja -kieliä, ohjelmia sekä protokollia. Väylän on kyettävä tukemaan kaikkia variaatioita ja yhdistämään niitä käyttävien yritysten tuottamia palveluita. Väylän on myös tuettava synkronista ja asynkronista viestintää. Muita viestintään tarkoitettua väylän tuottamia palveluita ovat kirjautumisen hallinta, tietoturva sekä viestien mahdollinen muunto. (4, s. 65.)

3.5 Palveluiden suunnittelu

Palvelusuuntautuneessa arkkitehtuurissa palveluiden suunnittelussa tulee huomioida tiettyjä keskeisiä periaatteita. Näitä ovat löyhä yhteys, palvelusopimus, itsenäisyys, sisäisen logiikan suojaus, uudelleenkäytön mahdollisuus, yhdistettävyyden, tilattomuus ja löydettävyyden. Löyhä yhteys tarkoittaa palveluiden välisen suhteen tiedostamista ilman riippuvuutta. Palvelusopimusten avulla palvelut pystyvät kommunikoimaan toistensa kanssa yhteisten kuvausten tai vastaavien dokumenttien perusteella. Itsenäisyys takaa palveluille kontrollin sisältämäänsä toimintalogiikkaan ja tämän logiikan piilottamiseen ulkopuolisilta. Ohjelmat tulee suunnitella siten, että niitä tai niiden osapalveluita voidaan käyttää useampaan tarkoitukseen ja kierrättää. Samoin palveluita tulee voida koordinoita ja yhdistää laajemmiksi kokonaisuuksiksi. Tilattomuudella tarkoitetaan ohjelman sisältämään toimintoon liittyvän informaation minimoimista. Palvelut pitää myös suunnitella siten, että ne ovat löydettävissä ja käytettävissä olemassa olevien mekanismien avulla. (1, s. 37.)

Näiden rakennuselementtien ja perussuunnitteluperiaatteiden muodostaman perusarkkitehtuurin lisäksi tarvitaan sopiva alusta, joka mahdollistaa palvelusuuntautuneiden sovellusten ja automaatioiden toteutuksen. Vaikka palvelusuuntautuneen arkkitehtuurin periaatteet eivät teoriana olekaan sidottu

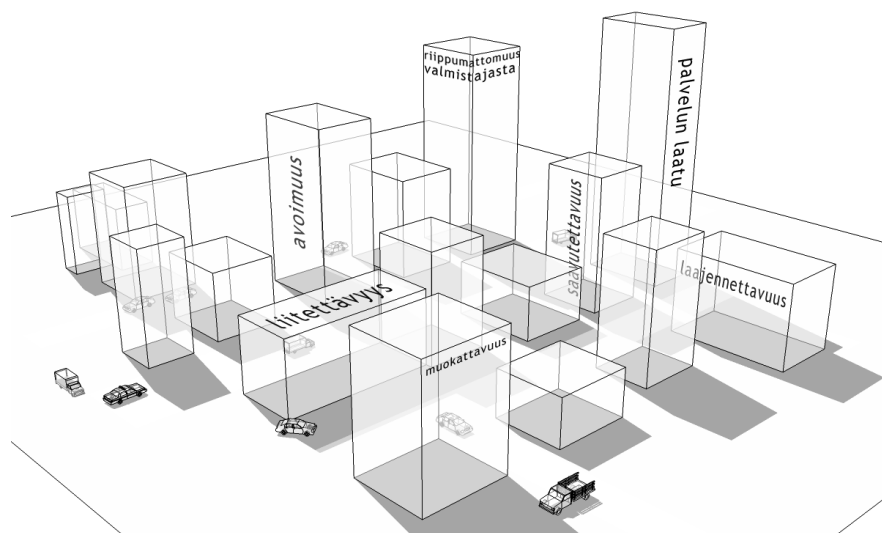
mihinkään tiettyyn valmistajaan tai tekniikkaan, ovat Web Services -standardit tuoneet erittäin hyvin soveltuvan alustan niiden toteuttamiselle. (1, s. 37.)

3.6 Palvelusuuntautuneen arkkitehtuurin käsitteet

Tekniikan nopea kehitys ja teollisuuden kehityssuunnat ovat muokanneet käytännössä palvelusuuntautuneen arkkitehtuurin sisältöä. Alkuperäiset periaatteet ovat säilyneet, mutta monia on laajennettu ja uusia tehokkaita mahdollisuuksia on kehittynyt vanhojen rinnalle. Tätä kutsutaan laajennetuksi palvelusuuntautuneeksi arkkitehtuuriksi (contemporary SOA). (1, s. 54.)

Alun perin palvelusuuntautunut arkkitehtuuri tarkoitti käsitteenä nimenomaisesti ohjelmiston tai palvelun arkkitehtuuria. Sittemmin käsite on laajentunut tarkoittamaan sovellusalustoja, jotka koostuvat Web Services -tekniikasta ja palvelusuuntautuneisuuden periaatteilla toteutetuista ratkaisuista sekä yrityksen eri toiminta-alueita ja prosesseja yhdistävistä ohjelmistoista. (1, s. 54.)

Laajennettu palvelusuuntautunut arkkitehtuuri edustaa avointa, laajennettavaa, liitettävää ja muokattavaa arkkitehtuuria, joka koostuu verkkopalveluiden avulla toteutetuista, itsenäisistä, palvelun laatua parantavista (QoS), valmistajista riippumattomista, yhteentoimivista sekä saavutettavista palveluista (kuva 2). Se muodostaa käsitteistön ja teorian toimintalogiikan ja tekniikan välille yhdistäen ne löyhästi toisiinsa. (1, s. 54.)



Kuva 2. Palvelusuuntautuneen arkkitehtuurin keskeisiä periaatteita.

Avoimuus

Verkkopalveluiden keskeinen ominaisuus on tiedon liikkuminen käyttäen avoimia standardeja. Viesti tai tieto siirretään palvelulta toiselle maailmanlaajuisesti hyväksytyjä ja standardoituja väyliä pitkin. Myös itse siirrettävä informaatio on standardoidussa muodossa ja sisältää kaiken tarpeellisen tiedon, jota vastaanottava palvelu tarvitsee. (Tämä toteutetaan käyttäen soap:a, wsdl:ää, xml:ää ja xml-skeemoja.) Ainoa, mikä vastaanottavan palvelun kannalta on oleellista, on toisen palvelun löytäminen ja kuvaus palvelun rajapinnasta. Palvelukeskeinen arkkitehtuuri tukee ja vahvistaa riippumatonta tiedosiirtoa ja rajoittaa ohjelmistotekniikan ja -alustan käytön ainoastaan palvelun sisältämiin toimintoihin. (1, s. 43.)

Laajennettavuus

Palvelun laajennettavuus tai ominaisuuksien ja toimintojen lisääminen aloittamatta suunnittelua ja rakentamista alusta on osa palvelusuuntautuneen arkkitehtuurin periaatteita. Parhaimmillaan se voidaan toteuttaa hajottamatta olemassa olevaa käyttöliittymää tai sotkematta muita ohjelman osia lisäämällä uusi toiminto 'lennossa' tai aiheuttamalla vain lyhyt toimintakatko. (1, s. 48.)

Liitettävyys

Palvelusuuntautuneeseen arkkitehtuuriin siirtyminen ei edellytä vanhan hylkäämistä. Se mahdollistaa erilaisten järjestelmien liittämisen tarjoamalla standardoidun tiedonsiirtotavan ja kielen yhdistämisen perinteisiin, suljetumpiin ohjelmistoihin. Käytännössä eri aikakausien ohjelmistojen ja järjestelmien mukauttaminen SOA:n vaatimusten mukaisiksi palveluiksi edellyttää sopivien väliohjelmien käyttöä, mutta etuna saavutetaan universaali liitettävyys muiden palveluiden kesken. (1, s. 50.)

Palvelun laatu

Toteutetun palvelun laatu koostuu toimintojen luotettavuudesta ja turvallisuudesta. Tehtävät on kyettävä suorittamaan turvaten sisällön koskemattomuus ja oikeellisuus vastaanottajalle. Ohjelman on toimitettava tieto perille luotettavasti. Häiriötapauksissa ilmoitetaan tapahtuneesta ja käynnistetään poikkeustilanteita varten suunnitellut toimenpiteet. Laitteiston kapasiteetin on oltava riittävä tuottamaan häiriöttömästi halutut palvelut huomioiden mahdolliset ruuhkat ja palvelupyyntöjen aiheuttama kuormitus. (1, s. 42.)

Muokattavuus

Palvelusuuntautuneen arkkitehtuurin ydin on sen muokattavuudessa. Tämä koskee niin yksittäisiä rakennusosia eli palveluita kuin kokonaisuuttakin. Tavoitteena on yrityksen toimintojen automatisointi mahdollisimman joustavaksi ja itsenäisistä elementeistä koostuvaksi hallituksi kokonaisuudeksi. Yksittäinen tarjottu palvelu voidaan pilkkoa sarjaksi pienempiä toimintoja, joista jokainen tuottaa osansa. Palveluiden ja arkkitehtuurin muokattavuutta on tukenut viestityyppien ja -väylien määritysten ja standardien syntyminen ja niiden käyttö. (1, s. 43.)

Riippumattomuus valmistajista

Yritysten ohjelmistot ja tietojärjestelmät voivat olla hyvin erilaisilla alustoilla toimivia ja erilaisilla ohjelmointikielillä toteutettuja. Palvelukeskeinen arkkitehtuuri mahdollistaa standardoitujen menetelmien avulla tietojen käytön ja siirron niin yrityksen sisällä kuin ulospäinkin. Huolimatta käyttäjien järjestelmien erilaisuuksista on mahdollista ilman suuria investointeja hyödyntää olemassa olevat järjestelmät ja silti tuottaa palveluita, joita muut voivat käyttää. Verkkopalveluiden toteuttaminen vanhoja ohjelmia käyttäen on yleistynyt erilaisten muunnosohjelmien (middleware) kehityksen myötä. Tämä mahdollistaa siirtymisen verkkopalveluiden tuottajaksi ilman kalliita henkilöstö- ja ohjelmistoresursseja. (1, s. 43.)

Saavutettavuus

Vaikka ensimmäisen sukupolven verkkopalvelut sisälsivätkin mahdollisuuden palveluiden rekisteröimiseen keskitetysti, ratkaisut toteutettiin useimmiten kahdenvälisinä (point-to-point). Syinä olivat palveluiden vähäisyys ja käytetty arkkitehtuuri. Palveluiden saavutettavuus ei ollut keskeinen kriteeri suunnittelussa. Palvelukeskeinen arkkitehtuuri tukee ja kannustaa ilmoittamaan ja löytämään palveluita yrityksen sisällä ja tiedottamaan niistä ulospäin rekisterin tai luettelon muodossa. (1, s. 44.)

4 Palvelusuuntautuneen arkkitehtuurin yleiset periaatteet

4.1 Määritelmä

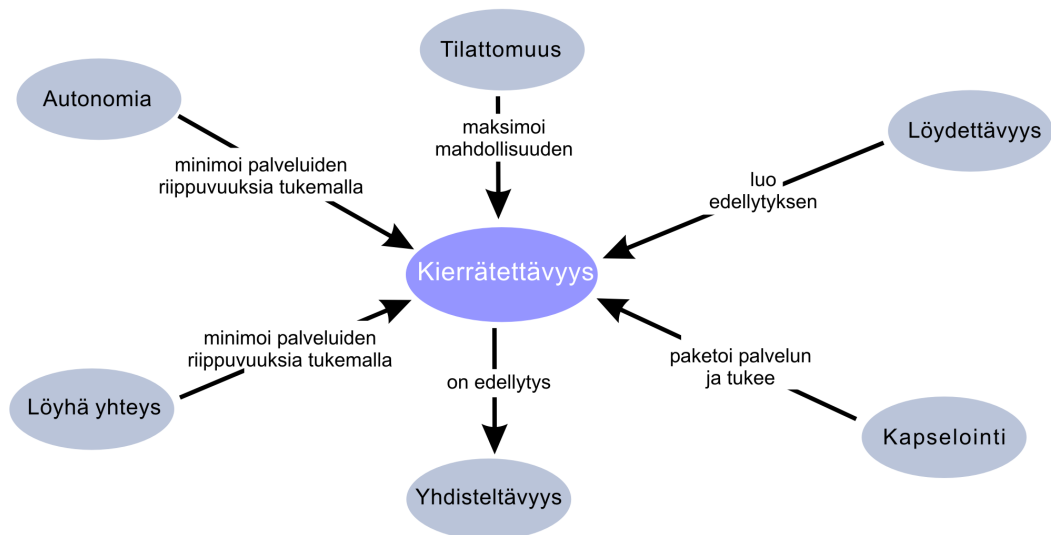
Palvelusuuntautuneelle arkkitehtuurille ei ole olemassa mitään yksittäistä, absoluuttista määritelmää tai standardia, joka olisi kaiken suunnittelun perusta. SOA:n juurista on monia näkökulmia riippuen määrittelijän lähtökohdista. Julkisen alan laitosten ja virastojen, ohjelmistovalmistajien ja konsulttiyritysten näkemykset arkkitehtuurin periaatteista ja tavoitteista saattavat vaihdella. Palvelusuuntautuneisuuden periaatteen sanotaan syntyneen ohjelmistosuunnittelun ongelmien hajauttamisen teoriasta (separaton of concerns). Teorian mukaan on järkevää pilkkoa suuri ongelmakokonaisuus sarjaksi pienempiä osaongelmia. Jokaisen osaongelman ratkaiseva sovellus liittyy toisiin vastaaviin sovelluksiin muodostaen yhdessä toimivan ratkaisun alkuperäiseen suurempaan ongelmaan. Tätä periaatetta toteutetaan erilaisilla ohjelmointialustoilla, kuten olio-ohjelmoinnissa tai modulaarisessa ohjelmoinnissa. Sovellukset koostuvat olioista, objekteista, luokista ja moduuleista. (1, s. 290.)

Palvelusuuntautuneisuuden periaatteet luovat ongelmien hajauttamisen teorian pohjalta mallin, jonka avulla käsitteistö ja ominaisuudet ovat mahdollisia toteuttaa. Nämä periaatteet, vaikka eivät siis olekaan virallisia, muodostavat yleisesti käytetyn linjauksen palvelusuuntautuneen arkkitehtuurin suunnitteluun ja toteutukseen. Arkkitehtuurin periaatteita ovat kierrätettävyys, muodollinen sopimus, löyhä yhteys, ohjelman toiminnallisuuden kapselointi, yhdistettävyys, autonomisuus, tilattomuus ja löydettävyys. (1, s. 290.)

4.2 Kierrätettävyys

Kun palvelu sisältää toimintoja, jotka ovat hyödyllisiä usemmalle kuin vain yhdelle vastaanottajalle, sitä voidaan pitää kierrätettävänä palveluna. Suunniteltaessa ja luotaessa uusia palveluita palvelusuuntautuneen arkkitehtuurin periaatteilla pyritään toteuttamaan ratkaisuja, jotka ovat mahdollisimman helposti käytettävissä monille

yrityksille. Tämä mahdollistaa ohjelmien kierrätettävyyden myöhemmin myös ennaltasuunnittelemissa tarkoituksiin. Palvelusuuntautuneen arkkitehtuurin muista periaatteista kierrätettävyyttä tukevat palveluiden autonomia, tilattomuus, määritelmä, löydettävyys ja löyhä yhteys (kuva 3). (1, s. 312.)

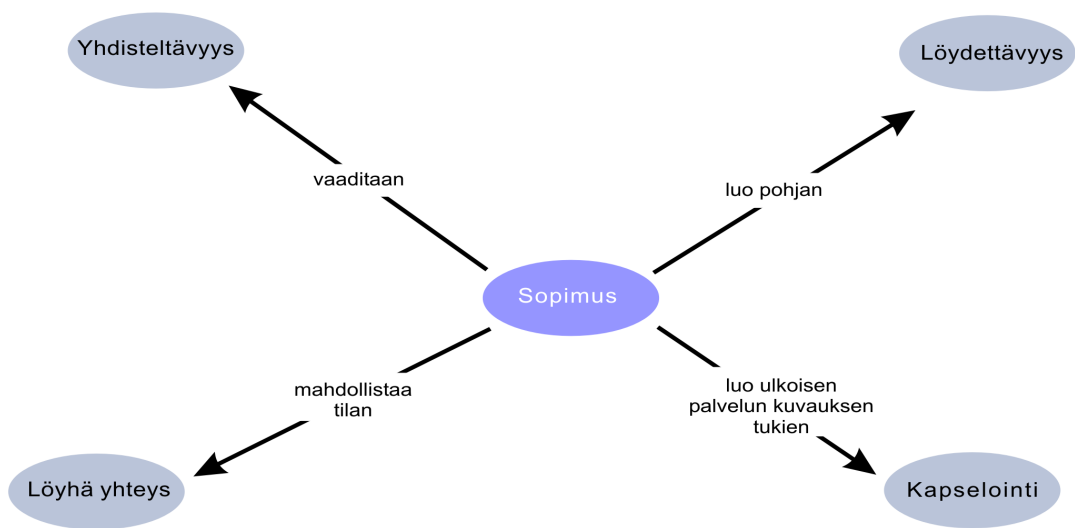


Kuva 3. Kierrätettävyyden suhde muihin palvelusuuntautuneen arkkitehtuurin periaatteisiin (1, s. 313).

Palvelun autonomia luo toteutusympäristön, joka tukee kierrätettävyyttä, koska palvelu on itsenäinen ja hallitsee omia toimintojaan. Mitä vähemmän ohjelma on riippuvainen muista, sitä laajemmat mahdollisuudet on sen kierrättämiseen. Tilattomuus tukee kierrätettävyyttä maksimoimalla palvelun saatavuuden ja tyypillisesti edistää yleistä palvelun suunnittelutapaa, joka siirtää tiedon prosessoinnin palvelun ulkopuolelle. Ohjelman kapselointi mustaksi laatikoksi helpottaa palvelun käyttöä useissa yhteyksissä, sillä palvelun käyttäjä ei pääse sisäisiin prosesseihin käsiksi. Edellytyksenä kierrätettävyydelle on palvelun löydettävyys. Löyhä yhteys palveluiden välillä vapauttaa ne suorista yhteyksistä toistensa kanssa. Tämä osaltaan helpottaa palveluiden kierrätettävyyttä.

4.3 Palvelusopimus

Palvelusopimus edustaa palvelun metatietoja. Se standardoi tavan ja ehdot, jotka täyttämällä vastaanottaja kykenee käyttämään palvelua. Sopimusta voidaan pitää palvelusuuntautuneen arkkitehtuurin kulmakivenä. (1, s. 295.) Kuva 4 osoittaa, miten palvelusopimus liittyy muihin arkkitehtuurin periaatteisiin.



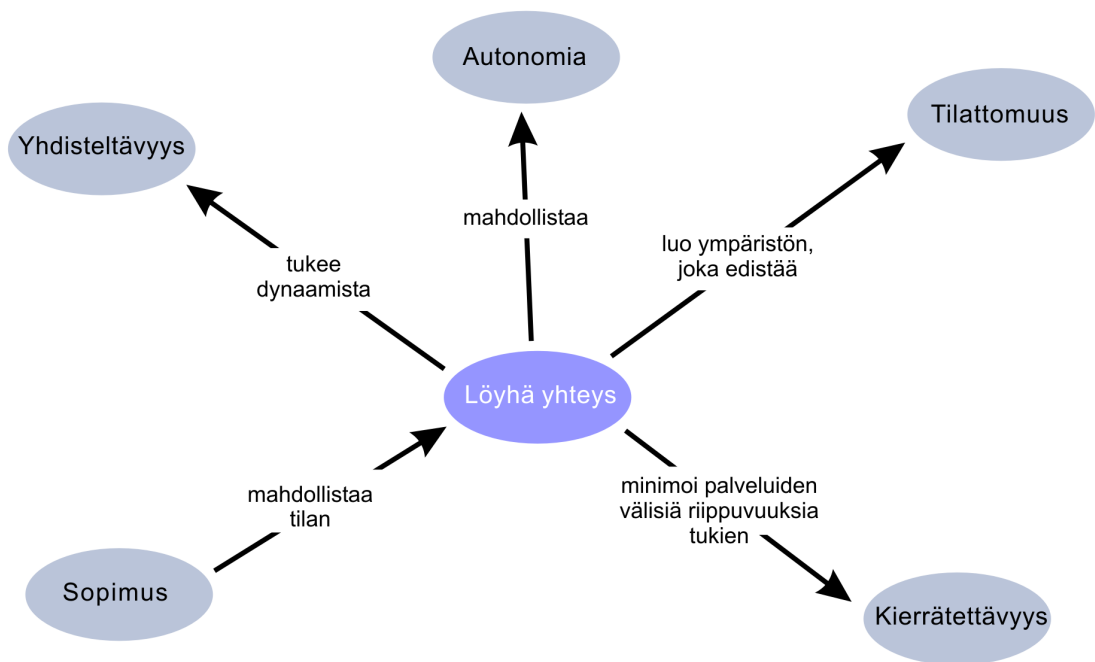
Kuva 4. Palvelusopimuksen suhde muihin palvelusuuntautuneen arkkitehtuurin periaatteisiin (1, s. 314).

Sopimus luo pohjan palvelun löydettävyydelle. Se taas on edellytyksenä palveluiden yhdistämiselle. Palvelun kuvaus esitetään osana sopimusta, ja se on tilaajalle ja käyttäjälle näkyvä metatieto. Muu rakenteeseen, prosessointiin ja toteutukseen liittyvä tieto on piilossa tämän sopimuksen sisällä tukien palvelun kapselointia. Ohjelmien yhdisteltävyyteen sopimus liittyy epäsuorasti antaen tietoa palvelun kuvauksesta.

4.4 Löyhä yhteys

Palveluiden välinen löyhä yhteys tukee niiden keskinäisten yhteyksien ja palvelun tarjoajan ja vastaanottajan välistä itsenäisyyttä ja riippumattomuutta.

Palvelusuuntautuneen arkkitehtuurin tärkeinä periaatteina ovat itsenäisyys ja riippumattomuus muista, mitä tukee myös vaatimus sisäisten ja ulkoisten yhteyksien löyhästä sidonnaisuudesta (kuva 5). (1, s. 315.)

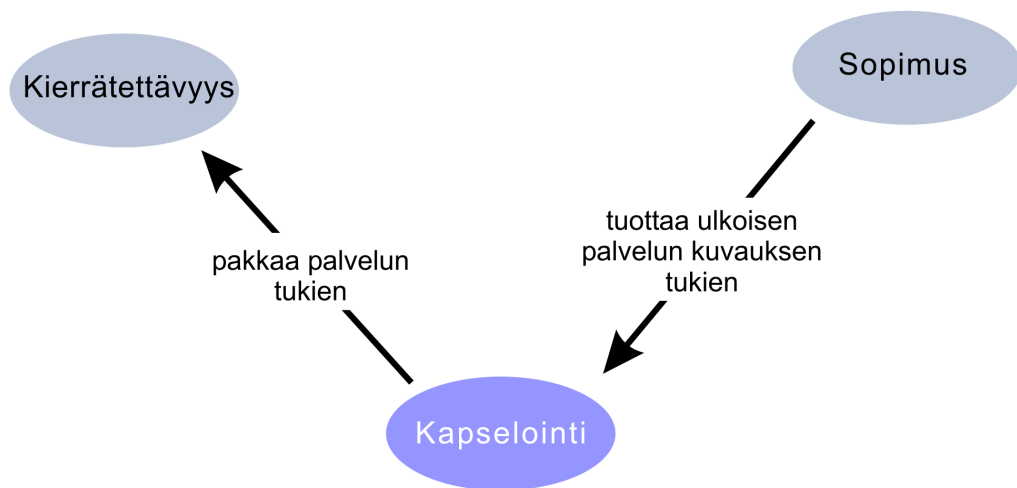


Kuva 5. Palveluiden välisten suhteiden löyhän yhteyden suhde muihin palvelusuuntautuneen arkkitehtuurin periaatteisiin (1, s. 316).

Palveluiden löyhä yhteys vapauttaa ne riippuvuuksista ja tukee osaltaan kierrätettävyttä. Erityisesti dynaamisten palveluiden yhdistelemisessä itsenäisyys on merkittävä ominaisuus. Se luo ympäristön, joka edistää ohjelman tilattomuutta ja autonomiaa.

4.5 Palvelun toimintalogiikan kapseloiminen

Itsenäisten palveluiden käyttäminen sovellusten rakennuspalikoina mahdollistaa niiden toteutuksen siten, että monimutkaisetkin tiedon käsittelyprosessit voidaan piilottaa käyttöliittymän taakse piiloon loppukäyttäjältä. Palvelusopimus, joka kertoo vastaanottajalle palvelun toiminta- ja yhteystavan, ei paljasta tarkempia yksityiskohtia teknisistä rakenteista ja logiikasta (kuva 6). (1, s. 316.)



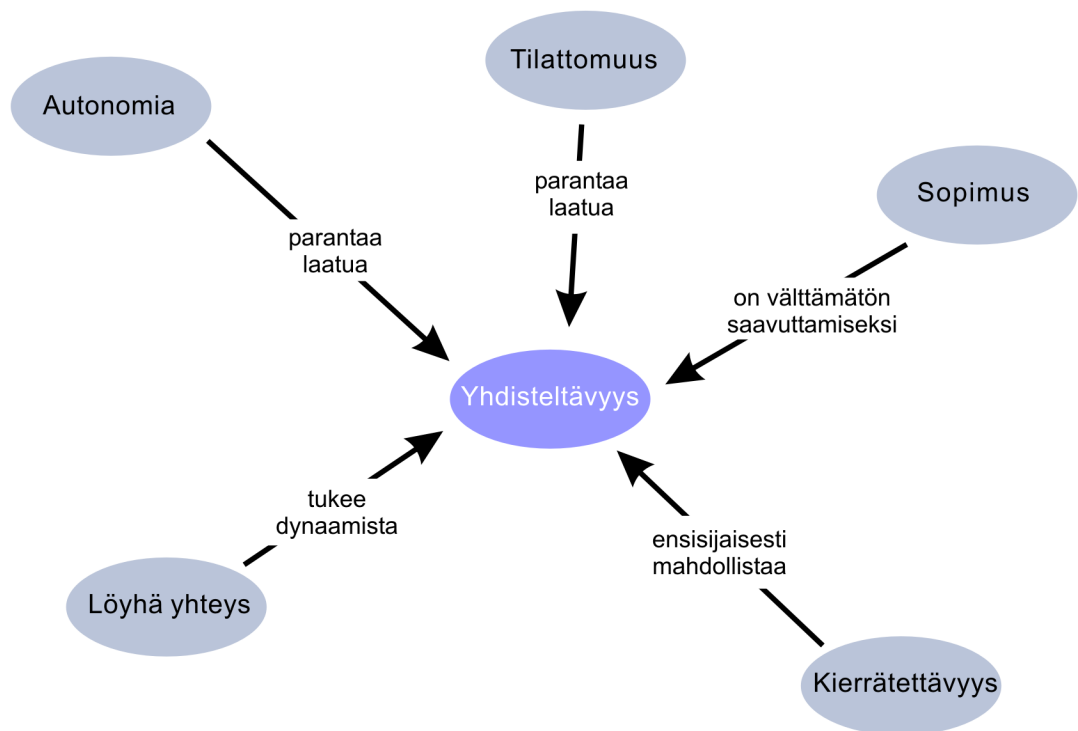
Kuva 6. Palvelun toimintojen kapseloinnin suhde muihin palvelusuuntautuneen arkkitehtuurin periaatteisiin (1, s. 317).

Palvelusopimus on virallinen ulkoinen palvelun kuvaus, joka julkaistaan tilaajille. Se tukee osaltaan palvelun kapselointia eli 'musta laatikko' -ajattelua. Mikäli ohjelma on sen tyyppinen, että se on käytettävissä moneen tarkoitukseen, kapseloinnin avulla mahdollistetaan kierrätettävyttä.

4.6 Palveluiden yhdistettävyys

Keskeinen periaate palvelusuuntautuneen arkkitehtuurin suunnittelussa on ohjelmien toteuttaminen siten, että muiden palveluiden on mahdollista käyttää niitä osana omaa sovellusalaansa. Palvelu voi siis rakentua muista palveluista, jotka osaltaan käyttävät toisia ohjelmia hyväkseen. Kuten kuva 7 osoittaa, lähes kaikki palvelusuuntautuneen arkkitehtuurin periaatteet tukevat omalta osaltaan palveluiden yhdistettävyttä.

(1, s. 317.)



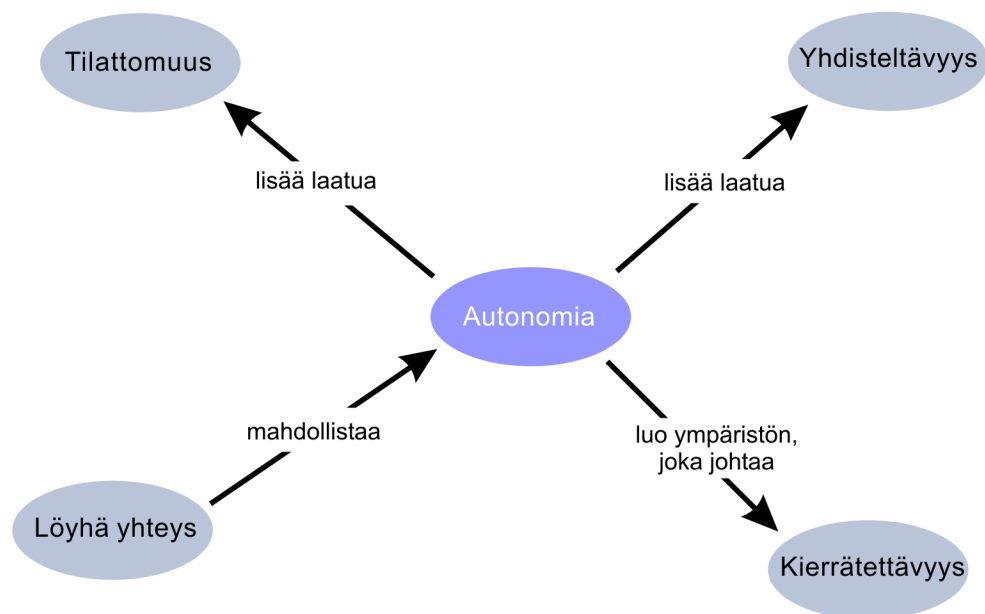
Kuva 7. Palveluiden yhdistettävyyden suhde muihin palvelusuuntautuneen arkkitehtuurin periaatteisiin (1, s. 318).

Palvelun kierrätettävyys mahdollistaa sen yhdistämisen muihin palveluihin. Se tarkoittaa joko palveluryhmän muodostamista useista palveluista tai erilaisten tilaajien käyttöä. Palveluiden välinen löyhä riippuvuussuhde luo edellytykset dynaamisten palvelukokonaisuuksien rakentumiselle. Tilattomuuden avulla voidaan vähentää kitkaa

koostettujen ohjelmien väliltä. Muodollinen palvelusopimus on välttämätön, jotta palvelut ymmärtävät toisiaan ja pystyvät hyödyntämään toistensa toimintoja.

4.7 Palveluiden autonomia

Palveluiden itsenäisyys toisiinsa nähden on tärkeä palvelusuuntautuneen arkkitehtuurin ominaisuus. Periaate koskee etenkin palvelun sisältämien toimintojen itsenäistä toteutusympäristöä ilman ulkopuolisten mahdollisuutta vaikuttaa niihin. Palveluita voitaisiin verrata lentokoneen mustaan laatikkoon, joka toimii itsenäisenä nauhoittimena koneen tapahtumista ja kykenee suoriutumaan tehtävästään koneen tuhouduttuakin. Ohjelmien tulisi olla itsenäisiä ja toisistaan riippumattomia. Niiden lähettämät viestit sisältävät kaiken tarvittavan tiedon, jota vastaanottava palvelu tarvitsee pystyäkseen hyödyntämään pyydettyä tietoa. Viestit ovat autonomisia yksiköitä, joita siirretään standardoituja väyliä pitkin toisten palveluiden käytettäväksi (kuva 8). (1, s. 318.)



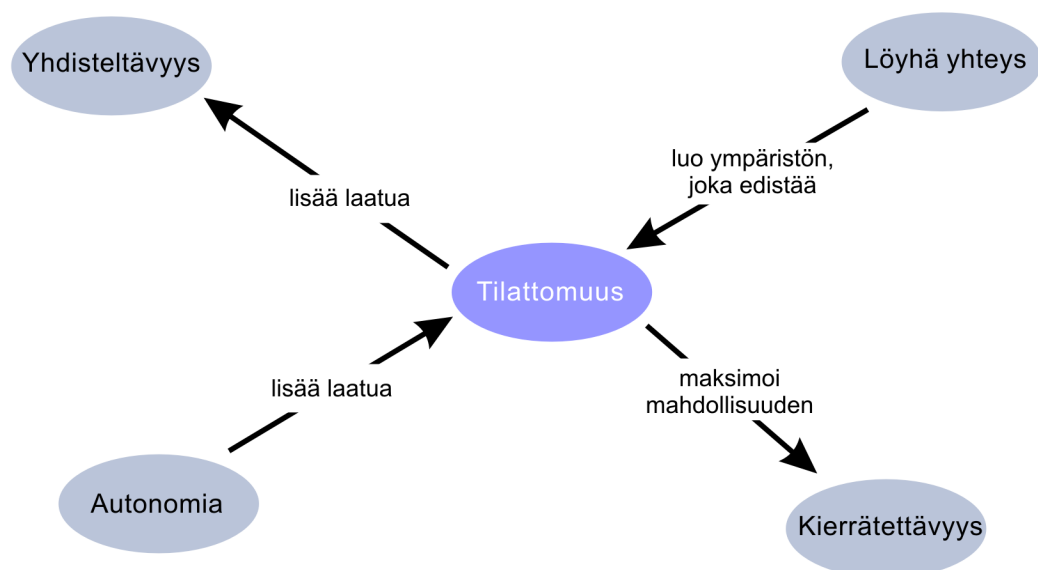
Kuva 8. Palveluiden autonomian suhde muihin palvelusuuntautuneen arkkitehtuurin periaatteisiin (1, s. 319).

Palveluiden itsenäinen toiminta on mahdollista ainostaan silloin, kun niiden välillä ei ole tiukkoja siteitä. Löyhä yhteys on perusedellytys palvelun autonomialle.

Riippumattomuus muiden palveluiden toiminnasta tukee osaltaan tilattomuuden, yhdisteltävyyden ja kierrätettävyyden periaatteita palvelusuuntautuneessa arkkitehtuurissa. (1, s.318.)

4.8 Tilattomuus

Ohjelman käsitellessä sille määriteltyä prosessia se on tilassa, jossa muut palvelupyynnöt eivät pääse läpi. Tämän tilan ajallinen kesto tulisi minimoida ohjelman suunnittelussa. Se on mahdollista toteuttaa sijoittamalla tilatieto palvelun ulkopuolelle dokumenttityyppisiin viesteihin, joiden otsake sisältää tietoa tilasta. Palveluiden autonomia, löyhä liitettävyys, muokattavuus sekä kierrätettävyys tukevat osaltaan tilattomuutta palvelusuuntautuneessa arkkitehtuurissa (kuva 9). (1, s. 307.)



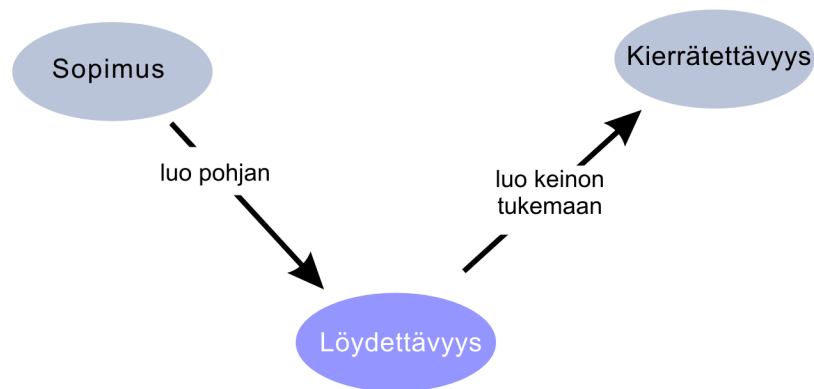
Kuva 9. Palveluiden tilattomuuden suhde muihin palvelusuuntautuneen arkkitehtuurin periaatteisiin (1, s. 320).

Palvelun itsenäisyys ja riippumattomuus muista palveluista mahdollistaa sille oman toimintaympäristönsä kontrollin. Näin on helpompi rakentaa tilattomuutta tukeva palvelu, jossa toimintalogiikka on täysin räätälöity ja tilan hallinta on siirretty palvelun ulkopuolelle. Palveluiden välinen löyhä yhteys, joka perustuu viestien välityksellä

tapahtuvaan kommunikointiin, mahdollistaa palveluiden itsenäisyyden ja tilattomuuden. Tieto palvelun tilasta välitetään viestien avulla.

4.9 Palvelun löydettävyys

Yrityksen tarjoamien palveluiden tulisi olla niitä käyttäville uusille vastaanottajille helposti löydettävissä. Tämä edellyttää arkkitehtuurin suunnittelemista siten, että tarjotut palvelut on rekisteröity käytettäviksi ja löydettäviksi samoilla periaatteilla ja samasta paikasta (kuva 10). (1, s. 320.)



Kuva 10. Palveluiden tilattomuuden suhde muihin palvelusuuntautuneen arkkitehtuurin periaatteisiin (1, s. 321).

Palvelun tilaajat etsivät palvelusopimusta selvittääkseen ja löytääkseen oikean palvelun. Hyvin toteutettu palvelun kuvaus vaikuttaa suoraan ja muodostaa pohjan sen löydettävyydelle. Palvelu, joka on tarkoitettu kierrätettäväksi, halutaan luonnollisesti tuoda mahdollisimman hyvin esille, jolloin sen löydettävyys ja kuvaus tukevat tätä ominaisuutta.

5 Xml ja verkkopalvelut

5.1 Rakenteisen tiedon määrittely

Ensimmäisen sukupolven verkkoteknologia keskittyi Internet-sivujen ja lomakkeiden väliseen kommunikaatioon. Ne eivät kuitenkaan sopineet suurten tietomassojen siirtelyyn vaan lähinnä muodostamaan visuaalisen ja suhteellisen staattisen käyttöliittymäympäristön, joka liikkui verkossa. Xml eli Extensible Markup Language toi mahdollisuuden muodostaa rakenteeltaan yleistä tietoa, jota voi siirtää Internetiä käyttäen koneelta toiselle ja joka on helposti myös ihmisen luettavissa selkokielisenä tekstinä. (3, s. 466.)

Näiden tekniikoiden juuret ovat 1970-luvulla IBM:ssä työskennelleiden Charles Godfarbin, Ed Mosherin ja Ray Lorien kehittämän SGML:n (Standard Generalized Markup Language) jälkeläisiä. Siitä tuli maailmanlaajuisen kehitystyön tuloksena standardi vuonna 1986. Perus-Internet-sivuja vaativampien verkkosovellusten toteutuskielenä oli SGML. Kielen suurimpana ongelmana oli sen monimutkaisuus. Vuonna 1996 joukko asiantuntijoita lähti työstämään kevyempää versiota kielestä. Tavoitteena oli karsia vuosien käyttökokemuksen myötä turhiksi ja hankaliksi osoittautuneet ominaisuudet ja laatia käyttäjäystävällisempi kieli. Tuloksena oli helmikuussa 1998 julkaistu xml 1.0, joka osoittautui heti menestykseksi. Oli syntynyt kieli, joka oli helppo omaksua ja muokattavissa. (7, s. 8.)

Seuraavaksi kehityksessä olivat vuorossa nimiavaruudet, joiden avulla voitiin välttää sekaannuksia samannimisiä elementtejä käytettäessä. Xml-dokumenttien katseluun selaimessa kehitettiin Extensible Stylesheet Language (xsl), joka jakautui kahteen osaan: XSL Transformations (xslt) ja XSL Formatting Objects (xsl-fo). Näiden avulla xml-dokumentti voidaan muuntaa toiseksi tai Internet-sivulla katseltavaksi. (7, s. 9.)

5.2 Xml-dokumentin rakenne

Xml-tiedostomuodon hyöty perustuu sen ympärille rakentuneiden sovellusten ja rinnakkaistekniikoiden muodostamaan kokonaisuuteen. Tiedosto pitää tulkita ennen tietojen hyödyntämistä. Ns. jäsenin eli parseri poimii yksiulotteisesta tekstistä esiin elementit ja muodostaa niistä sisäkkäisen puumallin. Yleensä samalla suoritetaan dokumentin oikeamuotoisuuden tarkistus, validointi. Jäsentimelle voi riittää tiedoston oikeamuotoisuus, eli sovitun standardin mukainen kirjoitustapa. (6, s. 50.)

Xml-dokumentti on aina tekstiä, ei koskaan binäärimuotoinen. Siten se voidaan avata millä tahansa tekstidokumentteja ymmärtävällä ohjelmalla. Esimerkkinä voisi olla puhelinluettelo, johon kirjataan henkilön nimi ja puhelinnumerot (esimerkki 1).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<puhelinluettelo>
  <henkilö>
    <nimi>Jussi Juupajoelta</nimi>
    <puhelinno tyyppe="matka">050-1234567</puhelinno>
    <puhelinno tyyppe="koti">09-8765432</puhelinno>
  </henkilö>
</puhelinluettelo>
```

Esimerkki 1. Puhelinluettelon xml-dokumentin rakenne.

Dokumentti koostuu ensimmäisen rivin yleisestä määritteestä sekä elementeistä. Tiedoston ns. juuritason elementti on <puhelinluettelo>, jota voi olla vain yksi. Toinen merkittävä ominaisuus on elementtien sisäkkäisyys. Henkilöelementin sisällä on nimi- ja kaksi puhelinnumeroelementtiä. Elementti koostuu alku- ja loppumerkinnästä eli tagista. Merkinnät muistuttavat HTML-kieltä, mutta erona on mahdollisuus uusien, itse keksittyjen merkintöjen tekemiseen. Alkumerkintä, joka aloittaa elementin, tulee aina sulkea loppumerkinnällä, jossa on lisänä kenoviiva (<tag>teksti</tag>). Xml-kielessä on merkitystä kirjainten koolla. <Henkilö> ja <henkilö> ovat eri merkinnät, ja ne tulee lopettaa aina samalla tavalla kirjoitetulla merkinnällä. (7, s. 16.)

5.3 Tyypimäärittelyt

Xml-dokumentin sisällön rakenne voidaan määrittellä hyvinkin tarkasti. Muodollinen dokumenttityypin määrittely, dtd (Document Type Definition), kertoo tiedoston elementtien ja niihin kuuluvien määreiden väliset suhteet. Se on tekstitiedosto, jolla on oma merkintätapansa (esimerkki 2). Esimerkissä 2 tehdään dtd:tä käyttäen puhelinluetteloesimerkin tyypimäärittely.

```
1  <!ELEMENT puhelinluettelo (henkilö)+>
2  <!ELEMENT henkilö (nimi, puhelinnro*)>
3  <!ELEMENT nimi (#PCDATA)>
4  <!ELEMENT puhelinnro (#PCDATA)>
5  <!ATTLIST puhelinnro tyyppi CDATA #REQUIRED>
```

Esimerkki 2. Puhelinluettelon dokumenttityypin määrittelytiedosto, dtd.

Määrittelytiedoston ensimmäisellä rivillä kerrotaan, että dokumentissa on puhelinluetteloelementtejä. Suluissa oleva määre 'henkilö' ja plus-merkki kertovat, että puhelinluettelon sisältönä voi olla yksi tai useampia henkilöelementtejä.

Toinen rivi määrittelee <henkilö> -elementin sisällä esiintyvien elementtien nimet: 'nimi' ja 'puhelinnro'. Tähti tarkoittaa, että henkilöelementin sisällä voi olla nolla tai useampia puhelinnumeroelementtejä. Riveillä 3 ja 4 määritellään nimi- ja puhelinnro-elementtien tyypit. #PCDATA (Parsed Character Data) tarkoittaa tekstimuotoista sisältöä. ATTLIST (rivi 5) kohdentaa puhelinnro-elementille lisämääreen, attribuutin, nimeltään tyyppi. Tyyppi on puhelinnumeroon liitettävä pakollinen määre, joka on muodoltaan tekstiä (CDATA). Tyypeiksi on esimerkkitiedostossa määritelty 'koti' tai 'matka'.

Kehittyneempi, monipuolisempi ja käytännössä ainut verkkopalveluiden käytössä oleva tapa ilmoittaa dokumentin sisällön rakenne on xml-skeema. Skeema-tyyppimäärittely mahdollistaa nimiavaruuksien käytön. Lisäksi on mahdollista antaa tarkka datatyyppi elementeille ja määreille (esim. string, int tai boolean), omien datatyyppien määrittäminen sekä lapsielementtien lukumäärän ilmoittaminen lukuarvona. (12.)

Esimerkin 3 skeemassa määritetään ensiksi käytettävä nimiavaruus. Toisella rivillä 'puhelinluettelo' nimetään complexType -tyyppiseksi elementiksi. Se tarkoittaa, että elementti voi sisältää muita elementtejä ja määreitä eli attribuutteja.

```
1 <xsd:schema xmlns:xsd=http://www.w3.org/2001/XMLSchema>
2   <xsd:complexType name="puhelinluettelo">
3     <xsd:sequence>
4       <xsd:element name="nimi" type="xsd:string"/>
5       <xsd:element name="puhelinnro" type="xsd:string"/>
6       <xsd:attribute name="tyyppi" type="xsd:string" use="required"/>
7     </xsd:sequence>
8   </xsd:complexType>
```

Esimerkki 3. Puhelinluettelon pelkistetty skeema-tyyppimäärittely.

Puhelinluettelon yksi yksikkö on nimitieto, ja sen sisältö on määritelty sequence-tagien sisällä. Riveillä 4 ja 5 yksikön tiedoiksi on määritelty nimi ja puhelinnro. Puhelinnro sisältää lisäksi tyyppi-nimisen attribuutin, joka on pakollinen jokaisen yksikön kohdalla. Tietojen muodoksi on määritelty string, eli tekstimuoto.

Xml-tiedosto ei välttämättä tarvitse seurakseen tyyppimäärittelyä ollakseen toimiva. Mikäli se on rakennettu käyttäen oikeaa kielioppia, tiedosto on ns. oikein muodostettu (engl. well formed). Kun dokumentti on sekä tyyppimäärittelynsä että xml-kieliopin mukainen, se on validi (engl. valid).

6 Verkkopalvelut

6.1 Teknologiakehys

Verkkopalveluiden teknologiakehys koostuu standardoiduista käsitteistä, palvelun kuvauksista, viesteistä, kommunikaatiosopimuksista, jotka pohjautuvat wsdl:ään, viestin kulkuun liittyvistä määritteistä (soap), palveluiden kuvauksen rekisteröinnistä ja esilletuonnista (uddi). (1, s. 111.)

Verkkopalvelu voi ottaa erilaisia rooleja riippuen yhteydestä, missä sitä käytetään. Se voi esimerkiksi olla tapahtuman käynnistäjä, välittäjä tai vastaaja. Palvelua ei siten voi nimetä pelkästään asiakkaaksi tai palvelimeksi, vaan rooliaan muuttavaksi ohjelmaksi, joka sopeutuu kulloiseenkin toimintaympäristöön. Palvelu saattaa vaihtaa rooliaan yksittäisen toiminnan aikana kerran tai useammin, mikä on palvelusuuntautuneelle arkkitehtuurille tyypillinen lähestymistapa. (1, s. 114.)

Verkkopalvelu toimii palvelun tarjoajana (service provider) silloin, kun esimerkiksi toinen ohjelma on ulkopuolelta pyytänyt sen sisältämää toimintalogiikkaa tuottamaan tiettyjä palveluita, kuten vaikka paikkavaraustietoa. Lähettäessään palvelupyynnön toiselle ohjelmalle tai etsiessään sopivaa palveluntarjoajaa käyttäen palvelukuvauksia verkkopalvelu toimii tapahtuman käynnistäjän roolissa (service requestor).

Verkkopalveluiden tapa välittää viestejä toisille ohjelmille poikkeaa perinteisestä pisteestä pisteeseen -kommunikointikanavasta. Viesti saattaa kulkea monen välittäjän kautta ennen lopullista päämääräänsä. Välittäjäpalveluita on kahdenlaisia: passiivisia ja aktiivisia. (1, s. 118.)

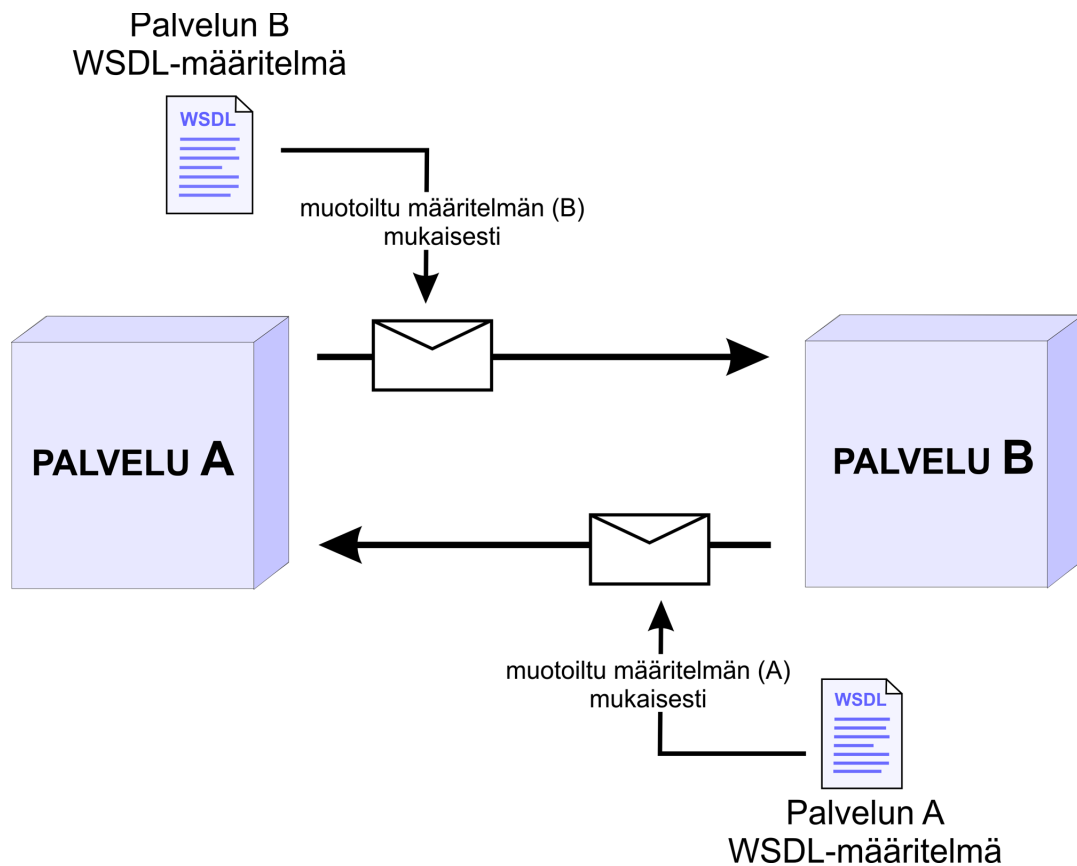
Passiivinen välittäjäpalvelu ei muokkaa sisältöä, vaan toimii ainoastaan viestin siirtäjänä paikasta toiseen. Se saattaa käyttää soap-viestin otsakkeessa oleva tietoa oikean reitin määrittelyyn. Aktiivisen välittäjäpalvelun ensisijainen tehtävä on passiivisen tavoin siirtää viesti eteenpäin kohti määränpäättä. Ennen eteenpäin lähettämistä aktiivinen

välittäjäpalvelu voi muokata viestin sisältöä. Tyypillisesti muokkauksen kohteeksi joutuvat viestin otsakkeet. (1, s. 121.)

6.2 Verkkopalvelun kuvauskieli

Voidakseen toimia löyhässä yhteydessä keskenään ohjelmien on annettava toisilleen tietoa viestien rakenteesta, palvelun toimintalogiikasta ja oikeasta tavasta kommunikoida. Verkkopalvelun sisältö, sijainti ja toiminnot kuvataan xml-rakenteisella dokumentilla, jota yleisesti kutsutaan wsdl-määritelmäksi. Kielen ensimmäinen toimiva versio julkaistiin joulukuussa 2001. Tällä hetkellä (02/2007) työskennetään wsdl 2.0:aa. (8.)

Määritelmässä kuvataan täsmällisesti, kuinka tapahtuman käynnistävän palvelun tulee muotoilla pyyntöviesti ja mikä on palvelun fyysinen osoite (kuva 11).



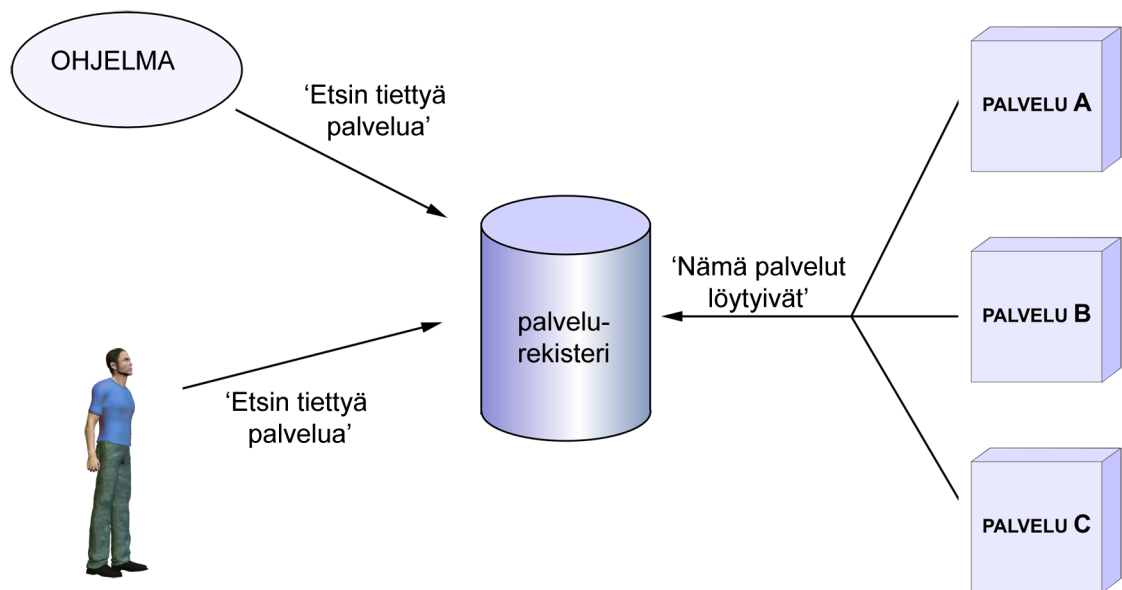
Kuva 11. Verkkopalvelun kuvauskieli (wsdl) määrittää viestille muodon, jonka toinen palvelu ymmärtää (1, s. 131).

Voidakseen kommunikoida palvelun B kanssa A:n on saatava tieto B:n sijainnista sekä viestin muodosta, jota B ymmärtää. Kun palvelu A on muokannut lähtevän viestin, B vastaanottaa ja käsittelee sen sekä muotoilee vastauksen A:n wsdl-määritelmän mukaisesti. Tämä ratkaisu mahdollistaa verkkopalveluiden löyhän yhteyden, sillä palvelut eivät ole riippuvaisia toistensa sisäisestä rakenteesta tai muodosta, vaan pystyvät kommunikoimaan wsdl-määritysten avulla toistensa kanssa.

6.3 Palvelun kuvauksen mainostaminen ja löytäminen

Voidakseen hyödyntää toisiaan on palveluiden voitava kommunikoida keskenään. Tämä on mahdollista palvelun kuvauksen avulla. Kun palveluiden määrä kasvaa organisaatioiden sisällä ja ulkopuolella, jonkinlainen listaus tai esilletuontimahdollisuus on välttämätön. Se voidaan toteuttaa esimerkiksi keskusrekisterin tai -hakemiston avulla. Niiden avulla on mahdollista paikallistaa uusimmat palvelunkuvaukset ja löytää uusia, tietyt kriteerit täyttäviä verkkopalveluita. Tämä otettiin huomioon jo ensimmäisiä verkkopalvelustandardeja laadittaessa. (1, s. 139.)

Yleinen palvelun kuvaus, löytäminen ja soveltaminen, uddi (Universal Description, Discovery and Integration), määrittelee standardina rekisterin rakenteen, jonka avulla voi löytää siihen kirjautuneet verkkopalvelut, niiden tuottajat ja sisällön (kuva 12). Rekisterin avulla palveluntarjoajat voivat myös mainostaa palveluitaan. Käyttäjät taas saavat tietoa eri palveluntarjoajista sekä palveluista ja löytävät tarvittavat palvelunkuvaukset. (10.)



Kuva 12. Palvelurekisteri mahdollistaa palveluntarjoajille palveluiden esilletuomisen ja käyttäjille niiden löytymisen (1, s. 139).

Yleisiin rekistereihin voivat rekisteröityä kaikki yritykset ja organisaatiot. Tällaisia rekistereitä löytyy suurilta yrityksiltä, kuten Microsoft, IBM ja SAP. Yksityinen rekisteri on organisaation sisällä ylläpidetty keskusrekisteri kaikista omista, ostetuista ja käyttöönotetuista palveluista. (1, s. 139.)

6.4 Palveluiden kommunikointi

Kaikkien palveluiden välinen kommunikointi perustuu viesteihin. Tämä on mahdollista vain jos viestit ja kommunikointitavat ovat standardoituja. Palvelukeskeinen arkkitehtuuri painottaa toiminta- ja ohjelmalogiikan sijoittamista viesteihin, jolloin niiden luominen ohjelman toimesta on yksi keskeisiä periaatteita. Viestintärakenteen on kokonaisuudessaan oltava joustava ja laajennettava. Verkkopalveluiden siirtoprotokollaksi valittiin aluksi soap (Simple Object Access Protocol tai Service Oriented Architecture Protocol), joka on hyväksytty yleiseksi standardiksi. Alkuperäisen julkaisun jälkeen soap:iin on lisätty kehittyneempiä viestirakenteita, jotka tukevat hajautettuja järjestelmiä ja palvelusuuntautunutta arkkitehtuuria. Protokollan nimi on säilynyt ennallaan, vaikka kirjaimet eivät enää suoraan edusta edellä mainittuja sanayhdistelmiä. (1, s. 140.)

Soap-viestin rakenne

Palvelusuuntautuneen arkkitehtuurin periaatteiden mukaan pyritään luomaan itsenäisiä ja älykkäitä viestejä palveluiden välille. Jokainen soap-viesti on pakattu kuoreen (envelope), joka sisältää kaikki viestin osat. Näitä ovat otsikko- ja runko-osat (header ja body) (esimerkki 4).

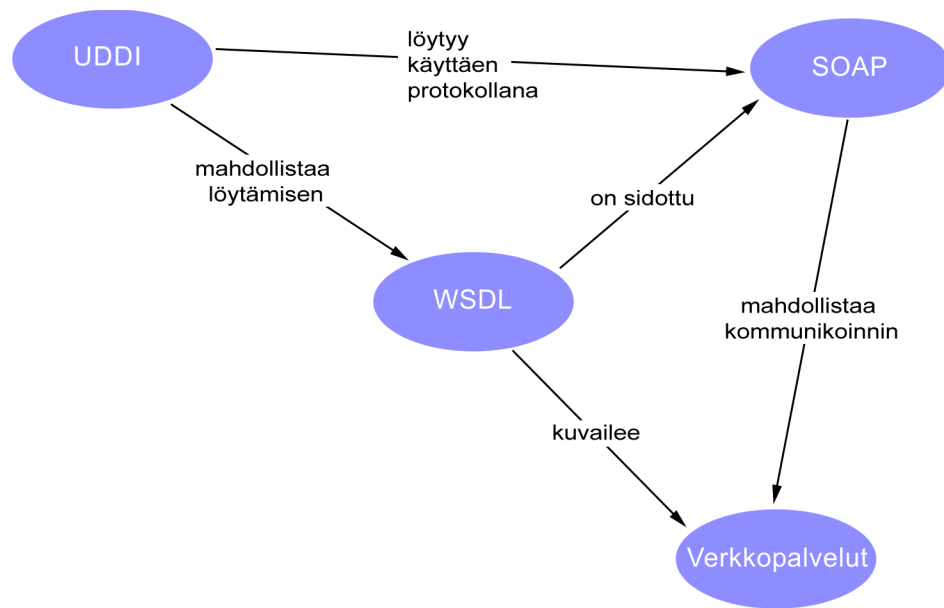
```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    ...
  </env:Header>
  <env:Body>
    ...
  </env:Body>
</env:Envelope>
```

Esimerkki 4. Soap-viestin rakenne yksinkertaistettuna.

Viestien itsenäisyys toteutetaan otsikko-osan lohkoilla. Ne voivat sisältää tietoa viestisisällön prosessoinnista, suojauksesta, siirtämisestä, työnkulusta ym. ohjeista. Tämä rakenne mahdollistaa ohjelmien uudelleenkäytön, laajennettavuuden ja yhteensopivuuden. Soap mahdollistaa myös otsikkotietojen merkitsemisen vaihtoehtoisiksi, jolloin ne voidaan joissain tapauksissa jättää huomioimatta. Viestin runko-osa sisältää vastaanottajalle tarkoitetun viestin elementit sekä mahdollisen virheilmoitustiedon, jolla tiedotetaan virhetilanteesta tapahtuneesta. (3, s. 78.)

6.5 Verkkopalvelut ja palvelusuuntautunut arkkitehtuuri

Verkkopalvelut on konseptina hyväksytty laajasti laitevalmistajien ja palveluiden sekä ohjelmistojen valmistajien puolelta. Edellä mainitut standardit ovat mahdollistaneet palvelusuuntautuneen arkkitehtuurin kehittymisen. Kuva 13 havainnollistaa verkkopalvelustandardien väliset suhteet.



Kuva 13. Ensimmäisen sukupolven verkkopalvelustandardien väliset suhteet (1, s. 482).

Verkkopalveluiden dynaamisen löytämisen mahdollistava standardi ja palvelun kuvauskieli ovat sidottuja kommunikointistandardiin. Näiden avulla verkkopalveluiden on mahdollista muodostaa verkostoja ja toimia itsenäisesti. Ajattelutapa sopii hyvin palvelukeskeiseen arkkitehtuuriin, jonka ydinteemana on yrityksen tuottamien palveluiden ja tietoteknisten rakenteiden suunnittelu mahdollisimman riippumattomiksi toisista ohjelmista.

Verkkopalveluita ja palvelusuuntaunutta arkkitehtuuria on arvosteltu muotivirtauksiksi, jotka ajallaan unohtuvat uusien keksintöjen myötä. Kyseessä ei kuitenkaan ole pelkästään tekninen hype, vaan ylemmän tason ajattelumalli verkkopalveluiden suunnittelusta ja toteutuksesta. Tarkoituksena on kehittää jo olemassa olevat sekä uudet verkkopalvelut sellaisiksi, että niiden päivittäminen on nopeaa ja edullista sekä liittäminen toisiinsa mahdollisimman automaattista.

6.6 Esimerkkejä palvelusuuntautuneen arkkitehtuurin toteutuksista

Katsaus suurten palvelusuuntautuneita arkkitehtuuria tarjoavien yritysten Internet-sivuille luo mielikuvan erityyppisistä ongelmista ja niiden ratkaisuista, joita on käytetty. IBM, BEA, Oracle ja Software AG ovat maailman johtavia tietotekniikan ratkaisuja tuottavia yrityksiä, ja kaikilla on referenssejä toteutetuista asiakastapauksista. IBM:n asiakkaista on poimittu terveydenhuoltoalan yritys Yhdysvalloista ja BEA:lta palveluita on hankkinut Suomen Asiakastieto Oy.

Aurora Health Care

Aurora Health Care on yli 25 000 ihmistä työllistävä, terveydenhuoltopalveluja tuottava ja liikevoittoa tavoittelematon yritys Wisconsinissa, Yhdysvalloissa. Sillä on 13 sairaalaa ja yli sata klinikkaa sekä saman verran muita toimipisteitä ympäri osavaltiota. Haasteena oli yhdistää ja tuoda esille yli tuhannen eri järjestelmän sisältämä tieto, tehostaa sisäistä tiedonkulkua ja asiakaspalvelua, yksinkertaistaa ja nopeuttaa yrityksen johdolle toimitettavan materiaalin tuottamista sekä asiakkaiden itsepalvelutyökalujen luominen. (14.)

Alkutilanteessa yritykselle arvokas tieto oli hajautettuna ja saavuttamattomissa eri järjestelmissä. Tavoitteena oli Internetin kaltainen informaatioympäristö, jossa tieto on jatkuvasti saatavilla yksinkertaistetun tekniikan avulla. Tämä oli toteutettava pehmeällä siirtymisellä ja mukauttamisella ilman olemassa olevien järjestelmien alasajoa. Avointen standardien tuki mahdollisti nykyisten ohjelmistojen mukauttamisen palvelukeskeisen arkkitehtuurin standardeihin käyttäen IBM WebSphere -ohjelmistoa. (14.)

Tuloksena työntekijät löytävät sairaaloiden, klinikoiden ja apteekkien tiedot kaikissa toimipisteissä samanlaisen verkkopohjaisen portaalin avulla. Tämä on vapauttanut resursseja asiakaspalveluun, ja kaikilla on selkeä käsitys, mistä tietoa voi etsiä. Henkilöstöhallinta on kehittynyt, ja esimerkiksi palkkatiedot löytyvät verkosta entisen tulostuksen ja postittamisen sijaan. Asiakkaille ollaan kehittämässä palveluportaalia, jonka kautta he voivat varata aikoja ja pääsevät katsomaan omia potilastietojaan.

Palvelusuuntautuneen arkkitehtuurin soveltaminen on siirtänyt voimavaroja teknisten prosessien pohtimisesta liiketoiminnan prosessien hallintaan ja kehittämiseen. (14.)

BEA ja Suomen Asiakastieto Oy

Suomen Asiakastieto Oy on maan johtavia yritys- ja luottotietoyrityksiä. Sen palvelut kattavat kaikki liikesuhteen vaiheet suoraan verkossa. Asiakastiedon yritystietokanta on laajin Suomessa. Palvelusuuntautuneen arkkitehtuurin avulla haluttiin kehittää joustavampi avoin ympäristö tukemaan luottotietokyselyitä. Lisäksi yrityksen sisäistä informaation kulkua ja palveluiden luotettavuutta, löydettävyyttä sekä muokattavuutta haluttiin parantaa. Ratkaisuna käytettiin BEA:n AquaLogic Service Bus -ohjelmistoa, jonka avulla rakennettiin integrointiväylä. Väylän avulla lukuisat käytössä olevat ohjelmat voidaan yhdistää luottotietopalveluiksi. Palveluiden muokkaaminen, yhdistäminen ja hienosäätäminen onnistuu toteutetun arkkitehtuurin ansiosta nopeasti ja aikaisempaa yksinkertaisemmin. (13.)

7 Modulaarinen Internet-sivuston rakentaminen

7.1 Suunnittelu

Sexpo-säätiön verkkosivujen uusimisen lähtökohtana oli vanhojen sivujen sekavuus, hankala päivitettävyyys ja kertyneen tiedon hajanaisuus. Lisäksi haluttiin uusia toimintoja ja palveluita asiakkaille, henkilökunnalle ja yhteistyötahoille. Suunnittelu aloitettiin kartoittamalla tarpeet ja toiveet.

Verkkosivuprojektien onnistumisen kannalta on välttämätöntä, että asiakkaalla on alusta asti selkeä kuva siitä, miten edetään ja missä järjestyksessä. Projekti alkaa aina tapaamisella, jossa kartoitetaan yleisellä tasolla taustoja ja olemassa olevien sivujen luonnetta. Kokouksessa käydään läpi toiveita ja mielikuvia uusien verkkosivujen sisällöstä, palveluista ja toiminnoista. Samalla tarjoutuu mahdollisuus tuoda esille omat toimintavat ja projektin etenemiseen vaikuttavat seikat sekä tutustua asiakkaaseen.

Aloituskokouksen jälkeen on mahdollista kirjoittaa synopsis, jossa hahmotellaan lyhyesti uusien verkkosivujen tavoitteet, rakennekaavio ja rajat. On tärkeää rajata projektiin kuuluvat asiat selkeästi heti alusta, sillä asiakkaalle saattaa myöhemmin tulla uusia ideoita ja ajatuksia, joiden toteuttaminen vaatisi palaamista takaisin ja aiheuttaisi suunnittelematonta lisätyötä. Rajauksen myötä asiakkaalle kerrotaan myöhäisten muutosten aiheuttavan aikataulun lykkäytymistä ja lisäkustannuksia. Synopsis on myös asiakirja, joka kokoaa paperille molempien osapuolien käsitykset siitä, mitä ollaan tekemässä ja miten. Asiakas on oman alueensa asiantuntija ja sivujen tuottaja omansa, joten yhteisen kielen ja ymmärryksen löytäminen heti projektin alussa on toimintafilosofian mukaan edellytys onnistumiselle.

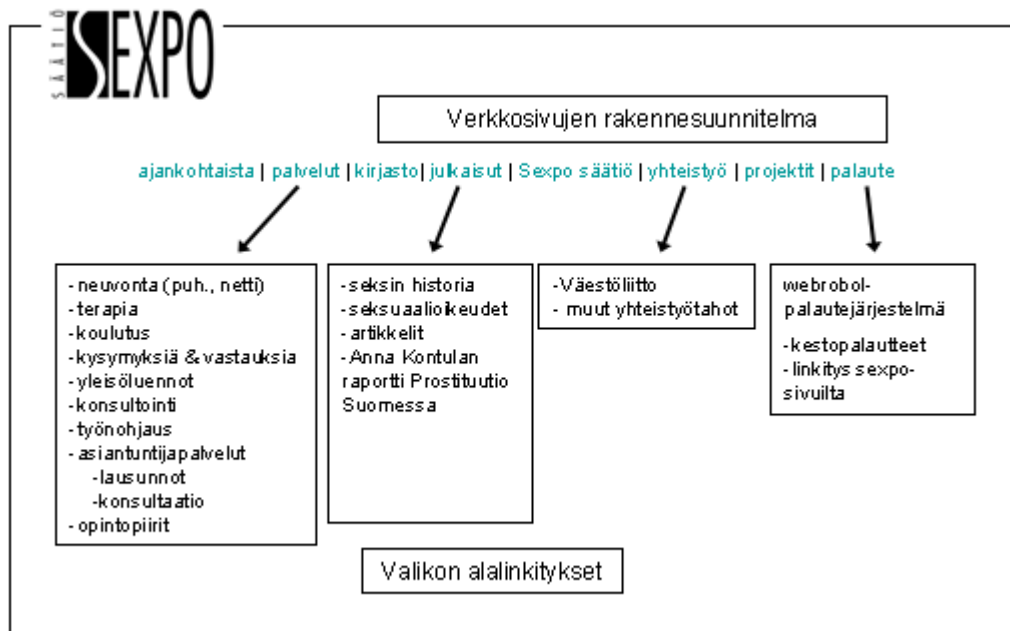
Toinen keskeiseksi osoittautunut seikka on asiakkaan sitouttaminen omalta puoleltaan projektiin. Tämä tarkoittaa vähintään yhden henkilön nimeämistä vastuu- ja yhteyshenkilöksi, jolla on valtuudet tehdä päätöksiä sivujen suhteen. Hän toimii linkkinä ja välittää asiakkaan toiveet ja päätökset toteuttajalle. Lisäksi asiakkaan

puolelta varataan riittävästi aika- ja henkilöresursseja sivujen sisällön tuottamiseen. Verkkosivuston teknisellä toteuttajalla ei millään voi olla tietoa asiakkaan asiantuntijuuteen ja liiketoimintaan liittyvistä asioista. Sisällön tuottaminen ja vastuu tiedon oikeellisuudesta ja toimitettavan materiaalin tekijänoikeuksista on asiakkaalla.

Synopsiksen hyväksymisen jälkeen tehdään tuotantosopimus, jossa sovitaan tarkasti projektin laajuus, vastuuhenkilöt, hinta, maksaminen, aikataulu, rajaus ja vastuut. Sopimuksessa mainitaan selvästi myös asiakkaan vastuu materiaalin oikeellisuudesta ja toimittamisesta aikataulun mukaisesti.

7.2 Rakenne ja sisältö

Sexpo.fi-sivujen suunnittelu käynnistettiin kartoittamalla nykyiset sivut ja laatimalla niiden pohjalta sisältökartta pelkistetyssä muodossa (kuva 14). Se auttaa hahmottamaan asioiden jaottelua ja ryhmittelyä. Samalla tulevat esille käytännössä hyväksi tai huonoksi havaitut rakenteelliset asiat.



Kuva 14. Sexpo-säätiön verkkopalvelun rakennesuunnitelma

Rakennesuunnitelman voi heti alussa laatia muistuttamaan verkkosivun valikkoa, jolloin se havainnollistaa tulevaa navigointirakennetta. Päävalikon sisältö pyritään pitämään riittävän selkeänä ja ytimekkäänä, jotta sivuilla kävijän on helppo löytää etsimänsä. Ongelmaksi on osoittautunut pääotsikoiden muokkaaminen kuvaaviksi, lyhyiksi ja vastaamaan alalinkkien sisältöä. Aluksi on hyvä liikkua hyvin yleisellä tasolla ja siirtyä suunnittelun edetessä pienempiin yksityiskohtiin.

Olellainen osa säätiön verkkopalveluita on kysymykset ja vastaukset -palsta. Siellä on mahdollista kysyä asiantuntijoilta kaikkea mahdollista seksiin ja seksuaalisuuteen liittyvää. Vastaukset löytyvät verkkosivuilta. Kysymyksiä ja vastauksia on kirjattu verkkoon vuodesta 2000 alkaen, ja niitä on pitkälle toista tuhatta. Kysymysten muuttaminen helposti selattaviksi oli merkittävä haaste suunnittelussa.

Sexpo-säätiön verkkosivuille tulee erillinen oppimisympäristö, joka on tarkoitettu lyhyiden ja pitkien koulutusten materiaalin ja opiskelijoiden hallintaan. Tekniseksi alustaksi valittiin EVTEKissäkin käytössä ollut ja hyväksi havaittu Moodle. Sen kirjautumispalvelua tullaan käyttämään sivuston käyttäjähallinnossa. Moodlen työkaluja hyödyntäen voidaan määritellä henkilöiden käyttöoikeustasot, jotka ovat voimassa koko sivustolla.

Säätiöllä on toiminnassa pieni kirjasto, josta esimerkiksi alan ammattilaisten on mahdollista lainata harvinaisempaa ammattikirjallisuutta. Nykytilanteessa lainaus on täysin manuaalista ja perustuu sähköpostipyyntöihin. Lainajista ei ole rekisteriä puhumattakaan automatisoidusta laina-aikojen kontrollista. Kirjastoa varten suunniteltiin käytettävän PHPMyLibrary-nimistä ohjelmistoa. Sen avulla voidaan hallinnoida kirjakokoelmaa ja lainauksia.

Sivuston ylläpito on tarkoitus uudistuksen jälkeen hoitaa säätiön henkilökunnan toimesta. Heillä on kohtuullisen vähäinen kokemus verkkosivujen ylläpidosta, joten tekstimuutosten ja ajankohtaisten asioiden muokkaamisen on oltava helppoa. Sivujen muokkaamista varten valittiin FCKeditor, joka on verkkosivuille integroitavissa oleva editori.

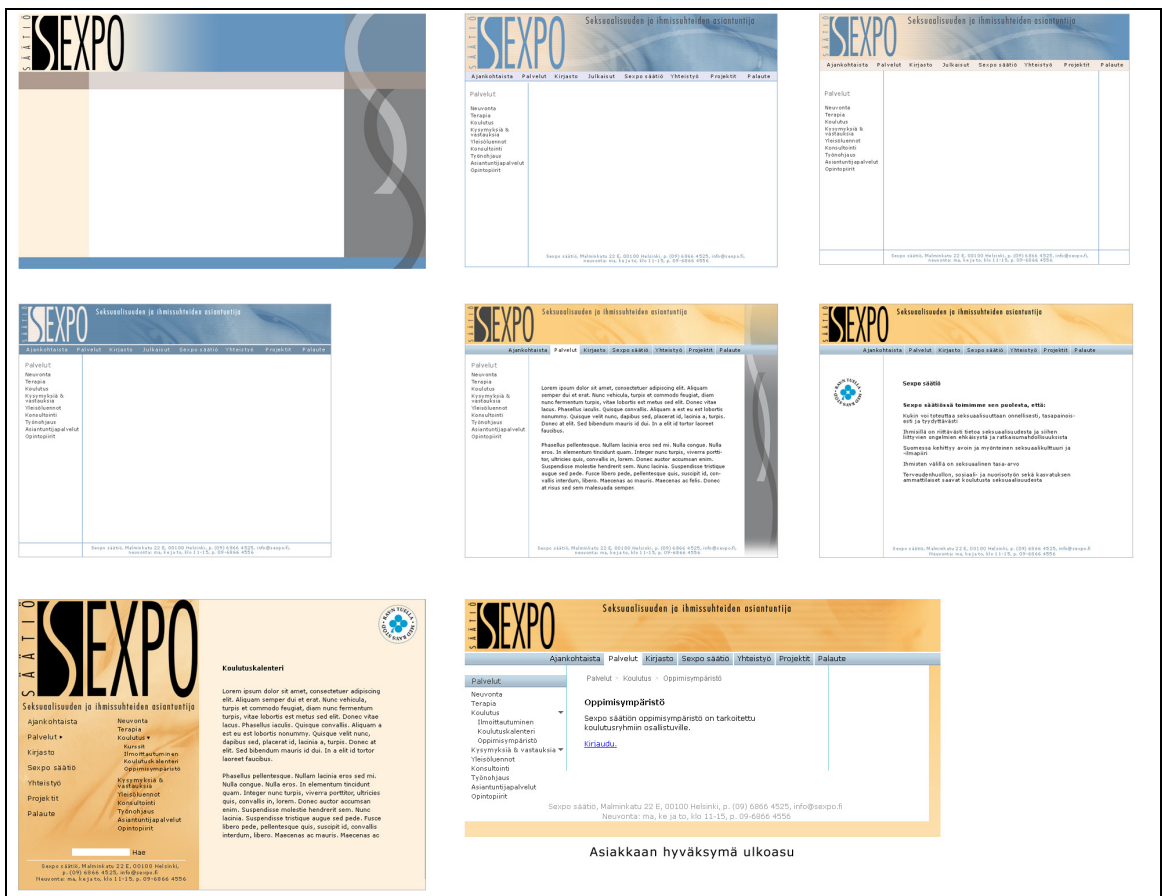
Näiden moduulien avulla voidaan melko yksinkertaisesti lisätä sivuille palveluita ja toimintoja, joiden toteuttaminen alusta asti koodaamalla olisi kohtuuton vaatimus aikataulullisesti ja taloudellisesti.

Suunnittelun alkuvaiheessa painotettiin heti eri käyttäjäryhmiä, joille sivusto on tarkoitettu. Esteettömyys asetettiin alusta alkaen teknisen toteutuksen tavoitteeksi. Sivujen on toimittava esimerkiksi näkövammaisten käyttämällä sokeainkirjoituslukijoilla ja puhesyntetisaattoreilla. Tätä varten kävin Näkövammaisten Keskusliitossa tutustumassa ja keskustelemassa vaatimuksista ja toiveista, joita näkövammaisilla on verkkosivujen suhteen. Ennen lopullista julkaisua on tarkoitus, että verkkomaailmaan perehtynyt, erilaisia apuvälineitä käyttävä näkövammaisen henkilö testaa sivut.

Suunnitelmat yhdistettiin käsikirjoitukseen, joka sisältää kaiken sivustolle tulevan materiaalin ja toimii toteutuksen ohjeena.

7.3 Ulkoasu

Kun Sexpo-säätiön vastuhenkilö oli hyväksynyt sivujen rakenteen ja sisällöllisen suunnitelman, aloitettiin ulkoasun suunnittelu. Usein yrityksillä on jo olemassa graafiset ohjeet, joissa määritellään käytettävät värit, logot ja tekstityypit. Sexpon tapauksessa ei ollut selkeitä ohjeita värien tai tekstien suhteen. Kuvien osalta toivottiin hillittyä ja abstraktia lähestymistapaa. Kaikkiaan kuvia ei tulisi käyttämään paljoa, vaan sivujen sisältö perustuu lähinnä asiapitoiseen tekstiin. Suunnittelun ja päätöksenteon pohjaksi laadittiin ehdotuksia, joissa etsittiin sopivaa värimaailmaa sekä ratkaisua navigoinnin pohjaksi (kuva 15).



Kuva 15. Sexpo.fi-sivuston ulkoasuehdotuksia.

Väriyhdistelmien hakemisessa käytettiin toimivaksi havaittua keinoa käyttää luontokuvia, joista poimimalla saa yhteensopivia värejä ja sävyjä. Sexpon värien alkuperänä oli kuva auringonlaskusta. Navigointi suunniteltiin toimivaksi hierarkkisesti siten, että päävalikko on jatkuvasti esillä ja alavalikkojen sisältö vaihtuu vasemmassa palkissa. Sisältöosan yläreunaan tulee aktiivinen murupolku, joka kertoo kuljetun polun ja jonka avulla voi siirtyä edellisille sivuille. Yksinkertaisen ulkoasun on tarkoitus helpottaa haettujen asioiden löytymistä ja luoda uskottava ympäristö asiantuntijana toimivan organisaatiolle.

7.4 Tekninen toteutus

Toimivalla organisaatiolla ei ole varaa pitää verkkosivuja pitkiä aikoja poissa käytössä. Sivujen uusimisen aiheuttaman katkon minimoiseksi avasimme omalle domainille

'varjosivuston', jonka sisältö ja tekniikka tulee olemaan identtinen uusien sivujen kanssa. Tämä helpottaa testaamista, ja sen avulla voidaan näyttää asiakkaalle kulloinkin projektin tila. Kun asiakas on hyväksynyt suojatulle alueelle tehdyn sivuston, kaikki tietokannat ja tiedostot voidaan siirtää nopealla aikataululla asiakkaan oman domainin alle.

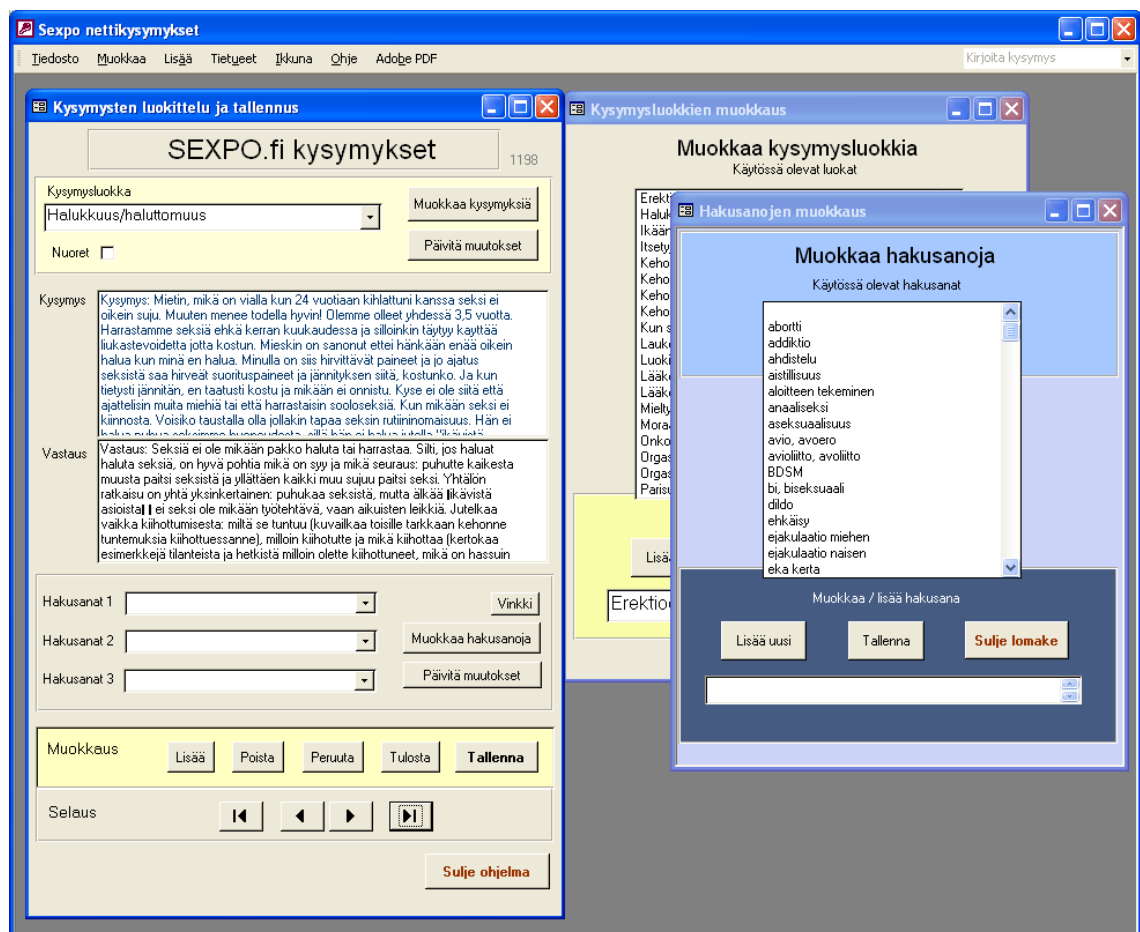
Tietokannan rakenne

Käyttäjien oikeuksien hallinnassa käytetään Moodlen tietokantaa. Siksi Moodle asennettiin palvelimelle ensin, jotta tietokantaan saadaan kaikki tarvittavat taulut. Koko sivusto tulee olemaan samassa tietokannassa. Moodlen taulut nimetään automaattisesti mdl-etuliitteellä, joten ne on helppo erottaa muista tauluista.

Kysymykset ja vastaukset vaativat omat taulunsa, kuten sisältösivut ja valikkokin. Kaikki näytettävät sivut muodostetaan tietokannassa olevasta tiedosta, jolloin niiden ulkoasua on helppo muokata ja muuttaa css-tyyleillä esimerkiksi näkövammaisten käyttämää lukulaitetta tai tulostusta varten sopivaksi.

Kysymykset ja vastaukset

Vanhoja kysymys-vastauspareja on nykyisillä sivuilla valtava määrä. Ne on jaoteltu vain vanhimpien osalta aihepiireihin. Vuoden 2002 jälkeen kysymyksiä on laitettu sivuille niiden tulojärjestyksessä, joten mitään selkeää jaottelua ei ole. Uusilla sivuilla kysymysten katselu ja haku aloitetaan valitsemalla pudotusvalikosta aihealue. Tämä helpottaa kysymysten löytämistä ja vähentää samojen kysymysten toistumista. Ratkaisuksi kysymysten siirtämiseen ja loogiseen jaotteluun tehtiin Access-tietokanta, johon asiakkaan puolelta siirrettiin ja ryhmiteltiin verkossa jo olevat kysymykset (kuva 16).



Kuva 16. Vanhat kysymykset ja vastaukset luokiteltiin ja siirrettiin tietokantaan Access-ohjelman avulla.

Tietokantaan saatiin kaikkiaan lähes 1 200 kysymystä vastauksineen luokiteltuina. Tämä vaati huomattavan aika- ja henkilöresurssien käytön. Tietokannasta oli helppo siirtää tiedot mysql-tietokantaan verkkopalvelimelle ja rakentaa sopivat kyselyt ja ulkoasu.

Sivuston 'luusto' on jo rakennettu testidomainiin. Kaikki linkit ja valikot toimivat, samoin Access-kannassa olleet kysymykset vastauksineen löytyvät. Moodle on asennettu ja odottaa käyttöä. Tilanne tätä kirjoitettaessa on se, että asiakas ei ole toimittanut kaikkia sisältöjä, joten projektin aikataulu on venähtänyt sen vuoksi. Tämä on ennakoitu jo tuotantosopimuksessa maininnalla asiakkaan vastuusta aikataulun viivästymiseen, mikäli asiakas ei kykene toimittamaan materiaaleja sovitussa ajassa.

8 Verkkopalvelun liittäminen sivuille

8.1 E-utilities-rajapinta

Kustannus Oy Duodecim vastaa terveyskirjasto.fi-verkkosivuista ja tarjoaa ulkopuolisille mahdollisuutta liittää omille sivuilleen hakupalvelu mm. Terveyskirjaston tietokannasta. Kyseessä on yleisölle suunnattu tietokanta, joka sisältää artikkeleita Duodecimin kustantamista lääketieteellisistä kirjoista ja lehdistä. Lisäksi samaa rajapintaa käyttäen voi tehdä hakuja suljetuista tietokannoista, kuten lääkärin, hammaslääketieteen tai EKG-tietokannoista edellyttäen, että käyttäjällä on oikeudet niiden käyttöön.

Tietokantojen käyttöä varten on yksinkertainen hakurajapintamäärittely nimeltään Duodecim E-utilities. Se koostuu kolmesta proseduurista: search, info ja show. Näitä voidaan kutsua Internetin kautta http-protokollaa käyttäen. Rajapinnan käytön seurannan helpottamiseksi hakuparametreissa on mukana toimijakohtainen tunniste. Sen avulla voidaan seurata ja tilastoida palvelun käyttäjien hakuja. Sexpo-säätiölle haettiin oma käyttäjätunniste, joka liitetään automaattisesti jokaiseen hakuun.

Osa Duodecimin tarjoamista ohjelmista tarvitsee erillisen käyttöoikeuden.

Käyttäjätunnistus voidaan toteuttaa ip-tunnistuksella, jolloin määritetystä ip-osoitteesta tulevat hakupyynnöt hyväksytään. Muissa tapauksissa tunnistus tehdään useimmiten vasta, kun käyttäjä avaa artikkelin.

Search

Search-proseduuri suorittaa haun määritettyyn ohjelmaan ja tarvitsee parametreina hakutermit. Hakutermi annetaan vapaamuotoisesti, kuten se kirjoitetaan hakukenttäänkin. Kysely palauttaa tulokset xml-tiedostona, jossa osumat on valmiiksi järjestetty ensin tietokannan ja sitten ohjelman arvioidun tärkeyden mukaan. Termin myötä palvelimelle on lähetettävä pakollisina parametreina 'p_db', ohjelma, josta haku suoritetaan, ja 'p_term', vapaamutoiset hakutermit. Esimerkiksi jos haetaan

Terveyskirjastosta hakusanoilla lapsi ja kuume, hakupyyntö olisi muotoa

```
'http://www.terveysportti.fi/eutil/search?p_db=tk&p_term=lapsi+kuume'
```

(15.)

Show

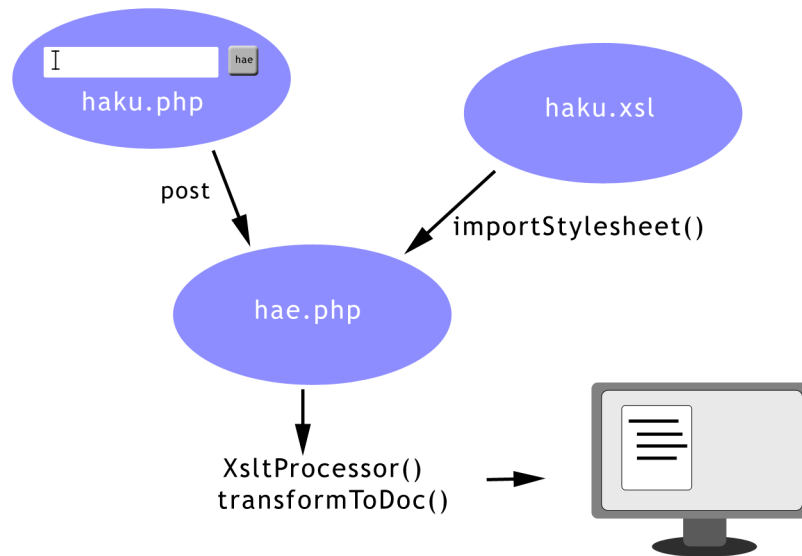
Show-proseduuri ohjaa pyynnön suoraan valitun ohjelman käyttöliittymään. Sen tuloksia on mielekästä tarkastella vain www-selaimella. Proseduuria käytetään halutun artikkelin näyttämiseen, hakuun tai selaukseen kohdeohjelman omassa käyttöliittymässä. Parametreja on käytettävissä kuusi, ja niistä käytetty ohjelma (p_db) ja artikkelin tunnus (p_id) ovat yleisimmin käytetyt. (15.)

Info

Info-proseduuri näyttää valittuja tietoja artikkelista tai kohdeohjelmasta. Tärkein annettava parametri on artikkelin tunnus (p_id). Toisena pakollisena on ohjelma, josta tietoa haetaan (p_db). Info palauttaa tuloksen xml-dokumenttina. (15.)

8.2 Hakupalvelun toteutus

Haku Terveyskirjaston tietokannasta toteutetaan käyttäen php-kieltä ja sen funktioita. Erillinen xsl-tiedosto huolehtii tulosten ulkoasusta (kuva 17). Tätä kirjoitettaessa ei Sexpo-säätiön sivujen hakutoiminto, johon Terveyskirjaston haku yhdistetään, ollut vielä kokonaan valmis. Tässä käytetyt koodit ovat toimivasta prototyypistä, joka tullaan sellaisenaan liittämään valmiiden sivujen hakutoimintoon (kuva 18).



Kuva 17. Terveyskirjaston tietokannasta tehtävän haun rakenne ja tiedostot.

Haku.php-tiedosto sisältää ainoastaan syöttöikkunan, jonka kautta hakusana lähetetään post-protokollalla hae.php-tiedostoon (esimerkki 5).

```

<form action="hae.php" method="post">
Hakusana: <input name="hae" type="text" size="20">
<input name="submit" type="submit" value="Hae">
</form>
  
```

Esimerkki 5. Tekstikentän koodi 'haku.php'-tiedostossa.

Hakusana otetaan hae.php-tiedostossa vastaan ja sijoitetaan muuttujaan. Tietojen prosessoinnissa käytetään PHP 5.0:n XSLTProcessor-funktion metodeja. Hakutulosten muokkaaminen tapahtuu erillisen haku.xsl-tiedoston määreiden avulla (esimerkki 6).

```
1 <?php
2 /* lataa xml- ja tyylitiedostot domdokumentteina */
3 $path_xsl="haku.xsl";
4 $xml_obj = new DOMDocument();
5 $xsl_obj =new DOMDocument();

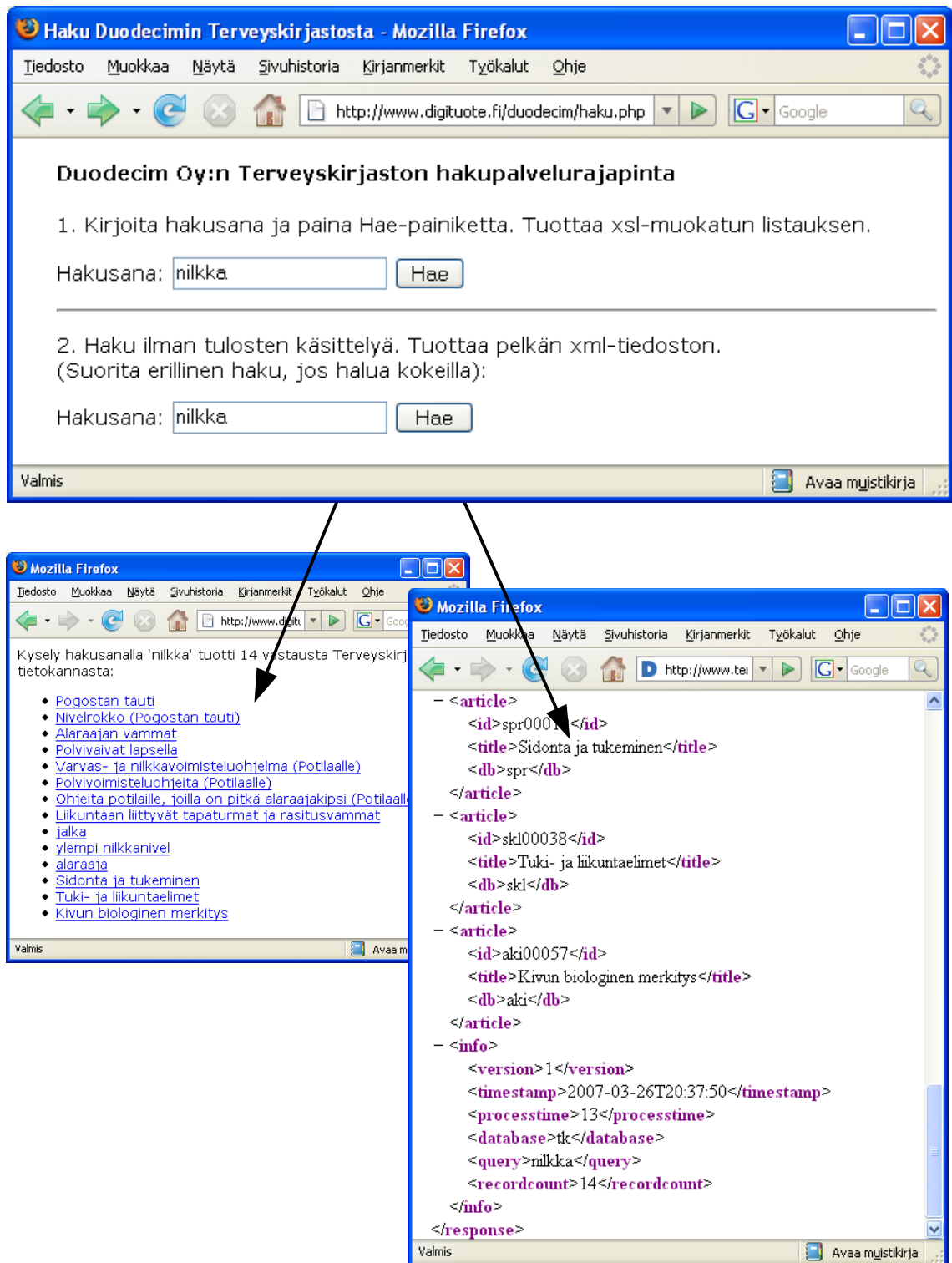
6 $xml_obj->loadXML($path_xml);
7 $xsl_obj->load($path_xsl);

8 /* luo prosessorin ja tuo tyylitiedoston */
9 $proc = new XsltProcessor();
10 $xsl_obj = $proc->importStylesheet($xsl_obj);

11 /* muokkaa xml-tiedoston ruudulle*/
12 $newdom = $proc->transformToDoc($xml_obj);
13 echo($newdom->saveXML());
14 ?>
```

Esimerkki 6. 'hae.php'-tiedoston osa, jossa xml-hakutulokset muokataan ruudulle.

ImportStylesheet()-metodilla tuodaan xsl-tiedoston muotoilumääritteet käytettäviksi. Tämän jälkeen käytetään transformToDoc()-metodia, joka muuttaa lähdetiedoston DOMDocument-muotoon, liittää siihen xsl-tiedoston muotoilut ja tulostaa lopputuloksen näytölle.



Kuva 18. Duodecim Oy:n xml-verkkopalvelun rajapintaa käyttävä hakutoiminto, joka tuottaa joko muokatun tai muokkaamattoman listauksen hakutuloksista.

Duodecimin Terveyskirjaston tuottama verkkopalvelurajapinta on minkä tahansa http-protokollaa ja xml-dokumenteja ymmärtävän sovelluksen käytettävissä. PHP 5.0:n

myötä palvelun käyttöönotto Internet-sivuilla on suhteellisen yksinkertaista valmiiden funktioiden ansiosta. Asiakkaan ei tarvitse tietää Terveyskirjaston tietokannoissa ja ohjelmistoissa tapahtuvista muutoksista niin kauan, kuin rajapintamäärittelyssä mainitut ehdot täyttyvät. Sexpo-säätiön sivuilta tehtävä haku muodostaa löyhän yhteyden Duodecimin ohjelmaan, joka toimii ulospäin mustan laatikon tavoin. Tätä periaatetta noudattaen voivat yritykset tuottaa verkkopalveluita toistensa käyttöön ja säilyttää itsenäisyytensä ja riippumattomuutensa omien palveluiden suhteen. Asiakkaat ja yhteistyökumppanit saavat palvelut ulottuvilleen, eikä niiden tarvitse muokata omia järjestelmiään mahdollisten muutosten tapahtuessa. Palvelusuuntaunut arkkitehtuuri on verkkopalveluiden ylemmän tason suunnittelustrategiana joustava ja huomioi olemassa olevat tietojärjestelmät ja toteuttaa uudet palvelut mahdollisimman joustaviksi niin asiakkaan kuin palveluita tuottavan yrityksenkin kannalta.

9 Yhteenveto

Insinööriyön alkuperäisenä aiheena oli modulaarisen verkkosivuston suunnittelu ja kehittäminen. Aiheen kohdentaminen palvelusuuntauneeseen arkkitehtuuriin ja verkkopalveluihin laajensi näkökulmaa korkeammalle, osin abstraktimmalle tasolle. Verkkopalvelut ovat yleisesti suurten yritysten tuottamia palveluita tai resursseja, jotka hyödyntävät valtavia tietovarastoja ja liittävät toisten yritysten palveluita niihin. Käytännön osassa otettiin Duodecimin verkkopalvelu käyttöön sanahaun muodossa. Edellytyksenä oli E-utilities-rajapintadokumentin sekä käyttäjätunnuksen saaminen. Palvelu ei siis ole automaattisesti löydettävissä uddi-palveluntunnistuksen avulla, mutta tämällytyypiselle palvelulle se ei olisi välttämätöntäkään.

Sexpo-säätiön Internet-sivut toimivat verkkopalvelun käyttäjänä. Duodecim tarjoaa Sitran tukemana Terveyskirjaston tietokantaa kaikkien käyttöön ilmaiseksi vuoden 2007 loppuun asti. Kun palvelu muuttuu mahdollisesti maksulliseksi, asiakastunnuksen käyttö hakujen yhteydessä mahdollistaa asiakkaiden laskutuksen helposti. Asiakkaalta tulleiden hakupyynnöiden pohjalta voidaan palveluita kohdentaa ja esimerkiksi arvioida laskutusperusteita.

Työn aiheen laajuudesta kertoo osaltaan palvelusuuntaunutta arkkitehtuuria markkinoivien yritysten koko. Suomessa esimerkiksi Fujitsu Services markkinoi arkkitehtuuria yhdessä saksalaisen Software AG:n kanssa, joka on yksi suurista arkkitehtuuria toteuttavista yrityksistä. Samoin muut maailman suurimmat tietotalot ovat kehittäneet omia työkalujaan mahdollisimman helppoon siirtymiseen vanhasta asiakas-palvelinajatusmallista dynaamisempaan ja muokkautuvampaan palvelusuuntauneeseen arkkitehtuuriin.

Sexpo-säätiön sivut toteutettiin osittain käyttäen moduuleja. Ratkaisu nopeutti ja mahdollisti erilaisten palveluiden sijoittamisen sivuille. Kyseessä ei kuitenkaan ollut verkkopalvelu luvussa 6 mainitussa merkityksessä, vaan verkossa oleville sivuille toteutettu ohjelmallinen ratkaisu, joka tuo lisäarvoa palvelemalla säätiön henkilökuntaa ja asiakkaita. Varsinainen kosketuspinta verkkopalveluihin on Terveyskirjastoon liitetty

hakutoiminto. Se käyttää http-protokollaa ja noutaa tietoa sovitun rajapinnan kautta. Palvelun tuottamat tiedot välitetään kysyjälle verkon välityksellä xml-muodossa. Vastaus muotoillaan standardien mukaisella tavalla käyttäen xsl-tiedostoa ja näytetään prosessoinnin jälkeen loppukäyttäjän ruudulla selkokieლისenä luettelona saaduista hakutuloksista.

Testausvaiheen perusteella (tilanne 03/2007) voidaan todeta, että moduulien käyttö osana sivuston toteutusta oli onnistunut ratkaisu. Toimivan oppimisympäristön, kirjastosovelluksen ja helpon päivitystyökalun kehittäminen olisi ollut ylivoimainen vaatimus annettuihin resursseihin suhteutettuna. Sexpo-säätiö sai selkeästi hyötyä valittujen työkalujen käytöstä, ja sivujen käyttäjille voidaan tarjota aikaisempaa parempia palveluita. Duodecim Oy:n Terveyskirjaston xml-hakupalvelun yhdistäminen sivujen omaan hakuun toi lisäarvoa sivujen käyttäjille laajan lääketieteellisen artikkelitietokannan kautta.

Lähteet

1. Earl, Thomas. Service-Oriented Architecture; Concepts, Technology and Design. Upper Saddle River, NJ: Prentice Hall, 2005.
2. Kraftzig, Dirk; Banke, Karl; Slama, Dirk. Enterprise SOA, Service Oriented Architecture Best Practices. Upper Saddle River, NJ: Prentice Hall, 2006.
3. Earl, Thomas. Service-Oriented Architecture, A Field Guide to integrating XML and Web Services. Upper Saddle River, NJ: Prentice Hall, 2005.
4. Möller, Andreas; Schwartzbach, Michael. An Introduction to XML and Web Technologies. Boston, MA: Addison Wesley, 2006.
5. Nykänen, Ossi. XML. Jyväskylä: Docendo, 2004.
6. Harold, Elliotte Rusty & Means, W. Scott. XML in a Nutshell. Sebastopol, CA: O'reilly, 2001.
7. Introduction to WSDL. (WWW-dokumentti.) W3Schools.
<http://www.w3schools.com/wSDL/wSDL_intro.asp>. Luettu 21.2.2007.
8. Web Services Definitions – UDDI. (WWW-dokumentti.) Searchwebservice.
<http://searchwebservice.techtarget.com/sDefinition/0,,sid26_gci508228,00.html>. Luettu 24.2.2007.
9. Universal Description, Discovery, and Integration (UDDI). (WWW-dokumentti.) Coverpages.org. <<http://xml.coverpages.org/uddi.html>>. Luettu 25.2.2007.
10. UDDI Services. (WWW-dokumentti.) Microsoft.
<<http://www.microsoft.com/windowsserver2003/technologies/idm/uddi/default.aspx>>. Luettu 25.2.2007.
11. Nimiavaruudet. (WWW-dokumentti.) W3C-organisaatio.
<www.w3c.tut.fi/training/2003/0219-xml/nimiavar-tyyppi.html>. Luettu 10.3.2007.
12. Customer Case Study. (WWW-dokumentti.) BEA.
<http://www.bea.com/content/news_events/white_papers/BEA_Asiakastieto.pdf> Luettu 11.3.2007.
13. Case Studies. (WWW-dokumentti.) IBM. <http://www-306.ibm.com/software/success/cssdb.nsf/CS/STRD-6YNLGN?OpenDocument&Site=db2software&cty=en_us>. Luettu 10.3.2007.
14. Landgren, Juuso. E-utilities rajapintamäärittely. Helsinki: Duodecim Oy, 2006.

15. Recommended Practice for Architectural Description of Software-Intensive Systems. (WWW-dokumentti.) ANSI/IEEE Std 1471-2000.
<http://www.sei.cmu.edu/architecture/published_definitions.html#Bibliographic>.
Luettu 28.10.2006.
16. Fowler, Martin. Patterns of Enterprise Application Architecture. Boston, MA: Addison-Wesley, 2002.