

XML Lanso-projektissa

Maiju Virtanen
maiju@cc.jyu.fi
20.08.2002
Tampere

Sisällys

Johdanto	3
XML:n perusteita ja etuja	3
XML:n ongelmia	3
Metatieto	4
XML:n rakenteisuuden hyödyntäminen erilaisissa käyttöliittymissä	5
Erilaisten käyttöliittymien toteutus	5
Järjestelmien integrointi	6
Yhteenveto	9
Lähteet	10
LIITE: Visioita ja esimerkkejä järjestelmien välisestä tiedonsiirrosta	11
A-netin ja Nautaneuvoksen yhteiskäyttö	11
Tiivis yhteistyö viranomaisten kanssa	15

Johdanto

Lanso-projektissa (langattomat tietoverkot ja sovellukset maaseudun toimialojen tukena) on kyse tietoverkkojen, erityisesti langattomien tietoverkkojen käyttöönotosta, Internetin soveltamisesta ja sovellusten kehittämisestä elintarvike- ja maatalousalalla. Tietoa liikkuu eri toimijoiden välillä paljon ja tiedonvälitys täytyy toimia tehokkaasti ja joustavasti. Projektissa selvitetään langattomien tietoliikennetekniikoiden turvallisuutta, käytettävyyttä, laatua ja kapasiteettia sekä pitkän ja lyhyen matkan yhteyksiä. Uusia tekniikoita testataan Alajärvelle rakennettavissa langattomissa yhteyksissä. Sovelluksia täytyisi saada myös yhdistettyä ja mahdollisesti kehittää kokonaan uusia. Langattomuus mahdollistaa tietojen päivittämisen suoraan tapahtumapaikalta ja esimerkiksi tietoa viljellystä pellon alueesta on mahdollista kerätä suoraan pellolla kiertävästä traktorista. Tila voidaan muuttaa HotSpot-alueeksi, jolloin tietojen käsittely onnistuu langattomasti mistä tahansa tilan osasta. Tiedon keruu ja välitys tehostuvat maatilalla näin huomattavasti.

Tässä kirjoituksessa tarkastellaan XML:ää ja mahdollisuuksia hyötyä siitä Lanso-projektissa.

XML:n perusteita ja etuja

XML:n käyttö toisi tähän maatalousprojektiin monenlaisia etuja. Ensinnäkin, XML-metakieli on avoin ja hyvin dokumentoitu suositus. Se on toteutettu Unicode-merkistöllä, joten sillä on mahdollisuus esittää kaikenkielisiä dokumentteja. Tosin kaikki ohjelmat, kuten selaimet, eivät vielä tue kaikkia merkkejä. Yksi suurimmista XML:n eduista on se, että se on tekstipohjaisuutensa ansiosta laitteisto- ja käyttöjärjestelmäriippumaton. Kun HTML:n tagit ovat määrättyjä, voi vastaavat XML:n elementit määritellä aina tapauskohtaisesti (vrt. tietokannan sanaston määrittely) ja siksi sanasto on aina laajennettavissa. XML:ssä on tarkemmat säännöt kuin HTML:ssä, mikä voidaan myös laskea sen haittapuoleksi, mutta XML-standardi määrittelee tarkistuskeinot (DTD, Schema), joiden avulla on mahdollista tuottaa virheettömät dokumentit selaimille. XML erottaa tiedon sisällön tiedon esittämisestä, joten tiedon sisältö voi muuttua ilman, että esitystapaa tarvitsee muuttaa ja toisinpäin.

XML:n ongelmia

IT-alan lehdissä on monenlaisia arvioita XML:n käytettävyydestä ja tulevaisuudesta. Joskus XML:ää mainostetaan ratkaisemaan lähes kaikki sähköisen asioinnin ongelmat ja jotkut taas pitävät

XML:ää vain nopeasti ohimenevänä muoti-ilmionä. (JUNA-julkaisu, 2001) XML:llä on omat vahvuutensa, mutta on syytä tarkastella myös sen heikkouksia.

Ensinnäkin XML on uusi tekniikka ja sen takia XML standardiperhe on osittain vielä kypsyvätön. Itse XML-määrittely on vakaa ja hyvin tuettu eri ohjelmistotoimittajien tuotteissa, mutta XML-perheen liitännäisstandardit ovat monelta osin vielä kesken. Myös XML:ää tukevat sovelluskehitys- ja muut työvälineet (esimerkiksi selaimet) ovat vasta kehittymässä. Osaajista, joilla olisi riittävästi kokemusta ja ammattitaitoa arvioida XML:n sovellettavuutta, on pulaa. (JUNA-julkaisu, 2001)

Yhteiset sanastot ovat vaatimuksena järjestelmien yhteentoimintaan, mutta niiden määrittely koetaan työlääksi. Koska XML antaa mahdollisuuden jokaiselle omien sanastojen määrittelyyn, sanastot hajaantuvat helposti erilaisiksi. Koska XML-dokumentit ovat tekstipohjaisuutensa takia niin suuria, niiden käsittely on resursseja kuluttavaa. Lisäksi on vaarana, että ajattelu on liian tekniikkakeskeistä ja XML:stä tulee itseisarvo. XML:ää kuitenkin pitäisi käyttää siinä, mihin se sopii, eli tällä hetkellä lähinnä tiedonvaihdon standardiksi Internet-alustoissa. XML soveltuu ennen kaikkea rajapintojen standardointiin. Siihen liittyen esitysmuodoksi tiedolle XML sopii myös, sillä se tukee *monikanavaisuutta* eli sitä, että eri käyttäjäryhmille esitetään tieto eri muodossa. (JUNA-julkaisu, 2001)

XML:n ongelmiin on kuitenkin ratkaisuja Lanso-projektissa. Omien sanastojen määrittelymahdollisuus ja XML:n tekstipohjaisuus, joka tekee siitä ihmiselle ymmärrettävää luettavaa, tuovat niin paljon etuja XML:n käyttöön, että edellä esitetyt ongelmat kompensoituvat hyödyillä. Vaikka liitännäisstandardit ovatkin kesken, voidaan itse XML:ää pitää luotettavana standardina. Ja jos kerran XML:ää sanotaan muoti-ilmioiksi, on mitä luultavammin myös kehittäjillä mielenkiintoa sen opetteluun ja osaajia tulee koko ajan lisää.

Metatieto

Metatieto on kuvailevaa tietoa itse tiedosta. Itse asiassa XML:n elementitkin ovat eräänlaista metatietoa ja siksi XML:ää voidaan sanoa metakieleksi. Esimerkiksi ilmaisussa <nimi>Maiju Virtanen</nimi> on nimi-elementti kuvailua varsinaisesta tiedosta Maiju Virtanen. Metatiedosta ja XML:n hierarkisesta rakenteesta on hyötyä hauissa ja tiedon löytymisessä.

XML:n rakenteisuuden hyödyntäminen erilaisissa käyttöliittymissä

Lanso-projektin hankesuunnitelmassa sanotaan (hankesuunnitelma luku 3.3.): ”Tietoturvallisuus ja tiedon jako oikeassa muodossa käyttäjäryhmien mukaan on kokonaisjärjestelmän vaativimpia tavoitteita.” Järjestelmään siis tarvitaan monenlaisia käyttöliittymiä eri käyttäjäryhmille. Näitä käyttäjäryhmiä ovat esimerkiksi tilanomistajat, eläinlääkärit ja autoilijat, joten on luonnollista ajatella, että käyttäjien tiedontarve ei ole samanlainen. XML on rakenteinen ja hierarkkinen kieli ja sen ansiosta voidaan elementtikohtaisesti määritellä, mitä tietoa eri käyttäjille näytetään. Esimerkiksi kuvitellut <pelto>-elementit ja niiden <viljelty_alue>-lapselementit näytetään maatalousyrittäjille, ja <maito>-elementin sisäiset <kuljetus>-elementit maidon kuljettajille. Näin saadaan myös käyttöoikeudet toteutettua tiettyihin elementteihin eli solmuihin. Voidaan päästä myös yksittäisen käyttäjän tasolle vietyyn käyttäjähallintaan vaikkapa roolitettujen ja tiettyyn asiayhteyteen liittyvien XML-web-lomakkeiden avulla. Tällainen käyttäjähallinta XSL:n avulla tosin on aika monimutkaista ja ylläpitäjältä vaaditaan hyvää ohjelmointitaitoa. Joissain tapauksissa XSL:n käytön yhteydessä voi tulla myös suorituskykyongelmia.

Erilaisten käyttöliittymien toteutus

Koska maaseudun järjestelmään täytyy päästä käsiksi myös mobiililaitteilla, XML sopii tarkoitukseen erityisen hyvin. XML erottaa tiedon rakenteen, sisällön ja esitystavan ja täten tiedon uudelleenkäyttö on mahdollista.

Tieto voidaan esittää *monikanavaisesti* erilaisissa laitteissa ja täten päätelaitteiden ominaiset sanomat, yhteys- ja turvakäytännöt voidaan ottaa huomioon. Tärkeimpiä huomioon otettavia päätelaitteita ja kanavia ovat sähköposti, web-selain, matkapuhelin, sovellukset ja uudet langattomat päätelaitteet. (Valtiovarainministeriön suositus, 2001)

Samaa sisältöä siis voidaan käyttää useissa käyttötilanteissa, vain esitystapa on erilainen. Esitystavan määräävät XSL-tyylisivut, joilla XML:stä voidaan XSLT-prosessin kautta generoida esimerkiksi HTML:ää tai vaikkapa WML:ää WAP:iin. XSL-tyylisivuilla määrätään, otetaanko kaikki, vai vain osa XML-dokumentista mukaan ja näin saadaan eri käyttäjille erilaiset näkymät tietoon. Eri näkymien käyttäjäkohtaisuutta tehostavat vielä metatiedot eli kuvailevat tiedot itse tiedosta, joiden avulla saadaan samasta tiedosta lohkottua erilaisia näkökulmia.

Järjestelmien integrointi

Järjestelmien integrointi on tärkein tämän kirjoituksen aihe. Eri järjestelmien integrointi on tärkeää, koska eri osapuolet tarvitsevat tiedon reaaliaikaisesti ja tieto kannattaa syöttää vain kerran. Maatalouden useat eri järjestelmät saattavat tarvita samaa tietoa, eikä ole järkeä tallentaa tietoa moneen eri paikkaan. Toisaalta taas vanhoissa järjestelmissä on niin paljon tärkeää tietoa ja toisaalta niihin on investoituakin niin paljon, että niitäkään ei voida poistaa. (Westerling, 2000) On siis yhdistettävä vanhoja toisiinsa ja uusiin. Ongelmana integroinnissa on, että järjestelmät voivat olla hyvinkin erilaisia alustoiltaan ja formaateiltaan.

Merkittävä XML:n sovellusalue on yritysten välinen tiedonsiirto eli EDI (Electronic Data Interchange). Perinteiset EDI-järjestelmät ovat tarjonneet luotettavan ja turvallisen tiedonsiirtotavan, mutta ne ovat olleet kalliita ja ei-julkisiin verkkoihin rakennettuja. Kun Internet ja Web ovat kehittyneet, EDI-tyyppinen tiedonsiirto on mahdollistunut julkisessa verkossa. Tällöin XML on syrjäyttämässä sanomamuotona perinteiset EDI-standardit. XML on edullisempi tapa tiedonsiirrossa ja se tarjoaa monipuolisemmat käyttömahdollisuudet siirrettävälle tiedolle. (Valtiovarainministeriön suositus, 2001)

Integrointiin tarvitaan joka tapauksessa yhteinen kieli, jota kaikki järjestelmät voivat käsitellä. Koska XML:n elementit voidaan määrittellä joustavasti itse, se tarjoaa sopivan löyhän liitettävyyden. Järjestelmät kuitenkin vaihtuvat ja muuttuvat jälkeensä. XML:n oleellisia ”kikkoja” integroinnissa ovat XML-skeemat (XML Schema) tai vanhemmat ja yksinkertaisemmat DTD:t (Document Type Definition), jotka takaavat vakioidut sanomamuodot. (Westerling, 2000) Koska XML ei määrittele käytettyjä elementtejä HTML:n tapaan valmiiksi, käytännössä organisaatioiden pitää sopia yhteinen XML-sanasto, jotta tiedonvälitys eri järjestelmien välillä onnistuisi. Kun sanasto on sovittu, tarvitaan joka järjestelmään oma sovitin, joka muuttaa tiedon XML-muotoon.

XML-skeemat ovat laajennuksia DTD:lle. DTD määrittää listan sallituista elementeistä ja attribuuteista sekä elementtien välisen hierarkian. Skeemat määrittävät samoja asioita kuin DTD:t, mutta niitä on helpompi laajentaa (syntaksi samanlainen kuin XML:ssä), helpompi tarkistaa (validoida) koneellisesti ja niihin sisältyy nimiavaruustuki (namespace), joten samassa dokumentissa voidaan käyttää eri nimiavaruuksissa määriteltyjä (samannimisiäkin) elementtejä. Skeeman avulla elementtien ja attribuuttien tietotyypit voidaan määrittää tarkemmin ja tällainen

skeeman ilmaisuvoimaisuudella on suuri merkitys erityisesti tiedonsiirrossa, kun halutaan kuvata esimerkiksi tietokantojen tietoja XML:llä. (Manninen, 2002) Todennäköisesti DTD:t ovat kuitenkin vielä laajemmassa käytössä, koska skeemat ovat monimutkaisempia käyttää ja ohjelmistojen tuki on vielä vähäinen.

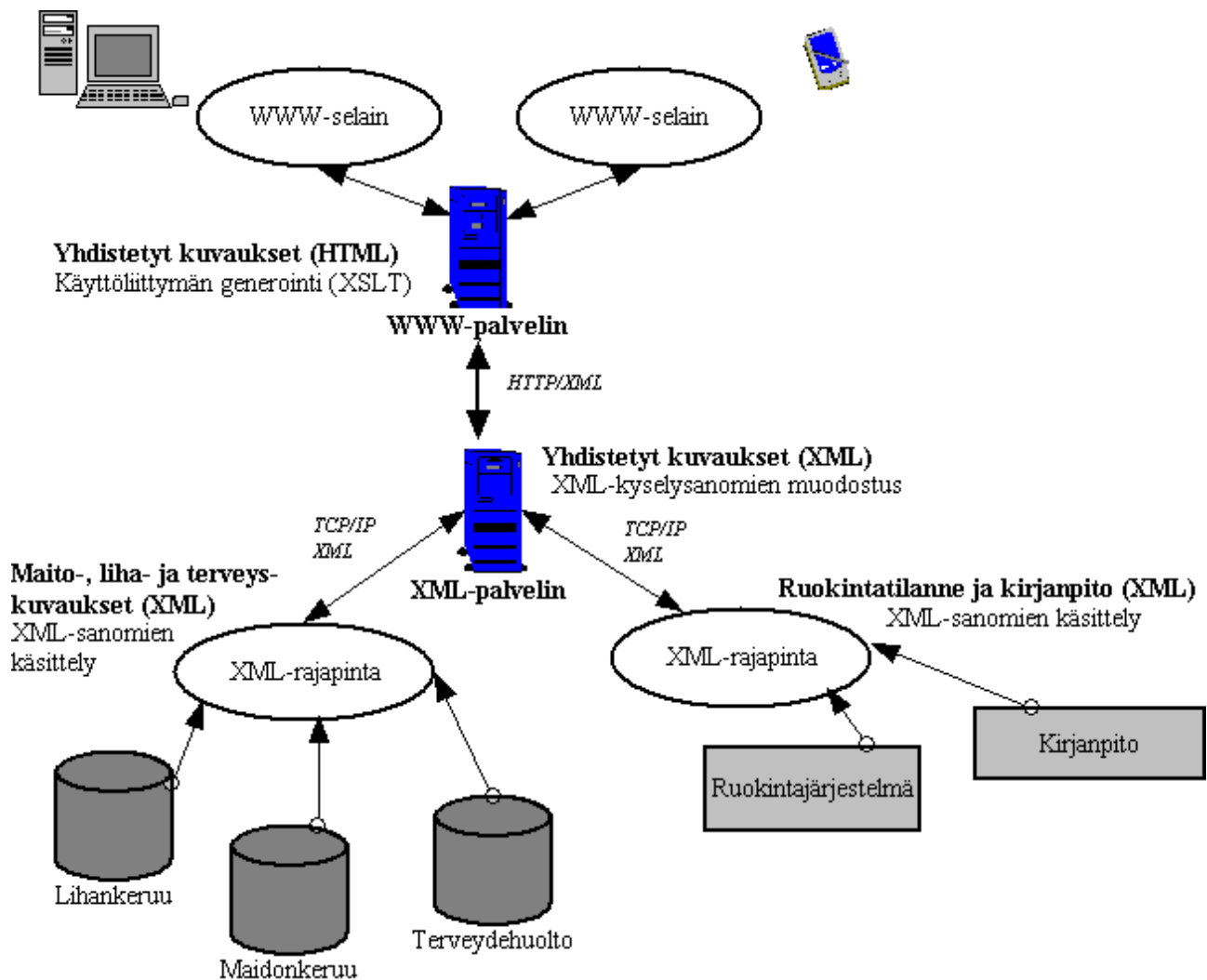
XML soveltuu 3-tasomallin (käyttöliittymä, liiketoimintalogiikka, tietokanta) kaikkiin kerroksiin. (Westerling, 2000) XML ja HTTP ovat tällä hetkellä ehkä käytetyimpiä protokollia WWW-pohjaisten sovellusten integroimisessa. XML-tietueita välitetään käyttäen HTTP-protokollaa ja tällöin ratkaisu voidaan toteuttaa usein myös sovelluksissa, jotka eivät ole WWW-pohjaisia. (Elisa Internet Oy)

JUNA-julkaisun (2001) mukaan web-palveluiden perustan muodostavat kolme XML-määrittystä: SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) ja UDDI (Universal Description, Discovery and Integration). SOAP on oleellinen osa tiedonsiirrossa ja toimii XML-pohjaisena sovellusten välisenä kommunikointimekanismina. SOAP määrittelee 1) ”kirjekuoren” sanomien kuljettamiseen, 2) säännöt, joilla sovellustason oliot muunnetaan XML-muotoon ja 3) mekanismin etäproseduurikutsujen (RPC, Remote Procedure Call) muodostamiseen. SOAP on riippumaton yhteyskäytännöstä ja mahdollisia viestin kuljetuskäytäntöjä ovat TCP/IP, HTTP ja SMTP, joista yleisin on tällä hetkellä HTTP. Web-palvelut tarjoavat SOAP-rajapinnan, joten niitä voidaan kutsua ja ne palauttavat vastauksen SOAP-sanomin. WSDL kuvaa web-palvelun rajapinnan (yhteyskäytäntö, sanomien parametrit ym.). UDDI-hakemistoteknologia on vasta kehittymässä. Se määrittää web-pohjaisen rajapinnan hakemistoon, jonne WSDL:llä tehdyt kuvaukset voidaan rekisteröidä.

Seuraavassa kuvassa 1 on mukailtu Helanderin (1998) sekä JUNA-julkaisussa (2001) olleita kuvia. Siinä kuvataan, kuinka viiden eri tietolähteen tiedot voitaisiin yhdistää ja kuinka ne esitettäisiin käyttäjälle HTML-sivuina. Järjestelmien todelliseen integrointitarpeeseen ei oteta vielä tässä vaiheessa kantaa. Järjestelmä hakee maito-, liha- ja terveystietokannat kolmesta relaatiotietokannasta. Lihan- ja maidonkeruulla tarkoitetaan maidon ja lihan kuljetuksia ja niihin liittyviä yksityiskohtia. Lisäksi haetaan ruokinta- ja kirjanpitojärjestelmästä sen hetkiset tiedot. Tiedot yhdistetään XML-palvelimessa, sen jälkeen ne muunnetaan HTML-sivuiksi ja jaetaan WWW-palvelimen kautta. (Helander, 1998)

Kuvan 1 WWW-palvelin tarjoaa käyttäjälle www-käyttöliittymän, josta on pääsy kaikkiin viiteen yhdistettyyn järjestelmään. Käyttöliittymää voidaan mahdollisesti mukailtuna käyttää sekä PC:ltä

että mobiililaitteesta. Käyttäjää varten on HTML-lomake, jonka kautta hän voi ilmaista, mistä aiheesta hän on kiinnostunut. Erityisiä haku-ehdoja lomakkeelle voi olla esimerkiksi päivämäärä. XML-palvelin analysoi lomakkeella annetut tiedot ja käyttäjän haluamat tiedot XML-rajapintojen avulla. XML-palvelin voi keskustella XML-rajapintojen kanssa esimerkiksi HTTP:n tai JAVA RMI:n välityksellä. (Helander, 1998) WWW- ja XML-palvelimet voivat toki sijaita samassakin fyysisessä palvelimessa.



Kuva 1. XML maatalouden järjestelmien integroinnissa.

Rajapinta tietovarastoihin ja operatiivisiin sovelluksiin voidaan rakentaa suhteellisen pienellä vaivalla. Rajapinnassa on esimerkiksi pieni XML-generaattori ja hakukone, joka suorittaa SQL-

kyselyjä relaatiotietokantaan tai käyttää operatiivisen sovelluksen ulkoisia rajapintoja hakeakseen halutut tiedot. Hakujen tuloksista muodostetaan yksi XML-dokumentti, joka välitetään takaisin XML-palvelimelle. XML-dokumenttien muodostuksessa ja jäsentämisessä (parsimisessa) voidaan käyttää XSL:ää tai DOM-rajapintaa. (Helander, 1998) Eri tietokantojen ja sovellusten termistöistä täytyy sopia yhteinen XML-sanasto, jota tiedonsiirrossa sovellusten välillä voidaan käyttää.

Kun XML-palvelin on saanut vastaukset, se yhdistää eri rajapintojen kautta saadut XML-dokumentit yhdeksi HTML-dokumentiksi. Tämä tapahtuu XSL-tyylisivujen avulla ja lopullinen dokumentti välitetään käyttäjälle WWW-palvelimen kautta. (Helander, 1998)

Yhteenveto

Tässä selvityksessä on todettu XML:n sopivan Lanso-projektin käyttöliittymiin sekä erityisesti projektin eri osapuolten järjestelmien yhteiseksi rajapinnaksi. XML on laite- ja käyttöjärjestelmäriippumaton ja sen rakenteisuuden ansiosta eri käyttäjäryhmille voidaan joustavasti muodostaa kullekin sopivat käyttöliittymät. XML erottaa tiedon sisällön ja esitystavan, joten samaa sisältöä voidaan näyttää erilaisissa laitteissa - sekä PC:ssä että matkapuhelimissa - monikanavaisesti.

Järjestelmien integrointi takaa eri järjestelmien yhteiskäytön. Yhteiskäyttöä varten organisaatioiden pitää sopia yhteinen XML-sanasto, jonka jälkeen joka järjestelmään tarvitaan oma sovitin, joka muuttaa tiedot XML-muotoon. Sovitin voidaan toteuttaa esimerkiksi DOM-ohjelmointirajapinnan avulla. Sanasto määritellään DTD:n tai XML Scheman avulla. XSL-tyylisivujen avulla XML voidaan muuntaa HTML-sivuiksi ja saada näin käyttöliittymät, joita www-selain pystyy lukea.

Lähteet

Elisa Internet Oy. Kronodoc lisätoiminnallisuudet, [online], [viitattu 03.07.2002]. Saatavilla www-muodossa <http://www.kolumbus.com/Tuotteet_ja_palvelut/Yritykseen/ASP-palvelut/Kronodoc/Lisaominaisuudet/lisaominaisuudet.html>.

Helander T. 1998. XML:n käyttäminen sovellusten välisessä tiedonvaihdossa. Essee teknilliselle korkeakoululle. [viitattu 04.07.2002]. Saatavilla www-muodossa <http://www.tml.hut.fi/Studies/Tik-110.300/1998/Essays/xml_2.html>.

Julkisten verkkopalveluiden kehittäminen ja XML. 2001. JUNA-julkaisu 5/2001. [online], [viitattu 04.07.2002]. Saatavilla www-muodossa <http://194.89.205.3/suom/juna/julkaisut/XML_juna.pdf>.

Lanso-projektin hankesuunnitelma. 2002.

Latva L. 2000. Johdatus WAP-tekniikkaan ja WML-kieleen, [online], [viitattu 08.07.2002]. Saatavilla www-muodossa <<http://www.hut.fi/~llatva/wap-tekniikka.pdf>>.

Manninen K. 2002. Elektronisten dokumenttien standardit –kurssi: XML-skeemat. PowerPoint-esitys, [online], [viitattu 03.07.2002]. Saatavilla www-muodossa <<http://www.infotech.jyu.fi/~kamannin/eds2002/weppisivut/skeemat/xml-skeemat.ppt>>

Valtiovarainministeriö, Hallinnon kehittämisosasto. Valtion tietotekniikan rajapintasuosituksia. Valtiovarainministeriön työryhmämuistioita 27/2001. [viitattu 03.07.2002]. Saatavilla www-muodossa <http://www.vn.fi/vm/julkaisut/tyoryhmamuistiot/pdf/tr27_2001.pdf>.

Westerling J. 2000. XML eBusiness-ratkaisuissa [online], [viitattu 3.7.2002]. Saatavilla www-muodossa <<http://xmlfin.ecraft.fi/archive/xml2000/Westerling.pdf>>.

LIITE: Visioita ja esimerkkejä järjestelmien välisestä tiedonsiirrosta

Tässä liitteessä on kehitelty ideoita siitä, minkälaista tietoa Lanso-projektiin liittyvien sovellusten välillä olisi mahdollisesti järkevää siirtää. Todellisista tarpeista on tietysti keskusteltava sovellusten käyttäjien kanssa, mutta näiden konkreettisten esimerkkien on tarkoitus selventää suunniteltua tiedon siirtämisen tapaa ja antaa lisäpontta uusien ideoiden syntymiseen.

A-netin ja Nautaneuvoksen yhteiskäyttö

Lanso-projektiin liittyvät järjestelmät A-Netti-teurasilmoituspalvelu ja Suonentiedon Nautaneuvosruokintasuunnittelusovellus toimivat tuottajan apuna karjan hoidossa. Ne koskevat käytännössä samoja asioita, joten niiden yhteiskäyttö ja integraatio olisi järkevää. Seuraavissa kahdessa kappaleessa esitellään molemmat sovellukset lyhyesti.

“Nautaneuvos-ohjelmalla on helppo tehdä karjalle aina ajan tasalla olevan ruokintasuunnitelman. Kun rehuerä vaihtuu ja siitä saa analyysituloksen, ruokintasuunnitelmaa voi tarkentaa välittömästi. Oikein suunnitellulla ruokinnalla saavutetaan tehokas ja taloudellinen tuotanto.” (http://www.suonentieto.fi/frame_maa.htm)

”A-Netti on Internetissä toimiva sähköinen tuottajapalvelu, jossa tuottaja on itse suoraan yhteydessä A-tuottajien hankintajärjestelmään. A-Netissä tuottaja voi ilmoittaa teuras- ja välityseläimensä mihin vuorokauden aikaan tahansa. Myös eläinten noutoajat ja teurastuksen jälkeen punnitus- ja luokitustiedot on heti saatavissa A-netin kautta.” (Lanso-hankesuunnitelma, 2002)

Kuvitellaan, että tilan eläimistä pidetään kirjaa Nautaneuvoksessa. Jokaiselle eläimelle voidaan pitää erillistä tapahtumakirjanpitoa. Kun tuottaja haluaa ilmoittaa teuraseläimensä, hänen täytyy käyttää toista järjestelmää eli A-nettiä. Tuottajan olisi kuitenkin helpompi ja nopeampi tehdä tapahtumakirjanpito ja ilmoitukset *samalla käyttöliittymällä internetissä*. Käytännössä siis tuottaja voisi ilmoittaa teuraseläimensä Nautaneuvoksesta käsin ja tiedot siirtyisivät suoraan A-tuottajille. Näin järjestelmään pääsisi koska tahansa mistä tahansa ja eläimen tarkkailu ja tietojen syöttö toimisi nopeammin ja tehokkaammin samasta paikasta. Mobiililaitteella tapahtumat voitaisiin kirjata suoraan paikan päältä.

XML sopisi näiden järjestelmien integrointiin. Käytännössä nautaneuvokseen tallennetut tiedot täytyisi saada muutettua XML-muotoon. Tässä tapauksessa Nautaneuvoksen tiedoista haettaisiin ja kerättäisiin A-netin tarvitsemat tiedot (esim. eläimen ikä, paino, tilan nimi jne.) sopivaksi XML-dokumentiksi, joka välitettäisiin sitten A-tuottajien järjestelmään. Eläimen tiedot muutetaan XML-dokumentiksi XML-rajapinnan avulla (kts. kuva 1).

Seuraavassa on kokeiltu Jdom-ohjelmointirajapintaa (Javalle optimoitu DOM) XML-dokumenttien muokkaamisessa. Ajatuksena on selventää toteutuksen sitä vaihetta, kun Nautaneuvoksen tiedot on jo muutettu XML-dokumentiksi ja niistä pitää saada koottua sopiva dokumentti tiedonsiirtoon A-nettiä varten. Esimerkki on hyvin yksinkertainen, mutta siitä näkyy selvästi, mitä etuja XML:n rakenteisuudesta on, kun eri osapuolten tiedontarve on erilainen. Tulodokumenttiin poimitaan JDOMNauta.java-ohjelmalla ainoastaan A-netin tarvitsemat tiedot, eikä esimerkiksi eläimen tapahtumakirjanpitoa (yleiskunto ym.) koskevia tietoja, jotka olisivat A-netille turhia.

Dokumentti Nautaneuvoksessa: NAUTA.XML

```
<nautaneuvos>
  <tila>
    <tilanro>55</tilanro>
    <tilinro>123123-54321</tilinro>
  </tila>
  <lehma>
    <nimi>maikki</nimi>
    <numero>1</numero>
    <paino>150</paino>
    <syntymaika>140501</syntymaika>
    <teurastuspvm>010802</teurastuspvm>
    <tilitys>
      <pvm></pvm>
      <summa></summa>
    </tilitys>
    <ruokintasuunnitelma></ruokintasuunnitelma>
    <koelypsy>
      <tulos></tulos>
      <pvm></pvm>
    </koelypsy>
    <yleiskunto>hyva</yleiskunto>
    <erikoistapahtuma>
      <selite>sairaana</selite>
      <pvm>22.7.2002</pvm>
    </erikoistapahtuma>
  </lehma>
  <sonni></sonni>
  <hieho></hieho>
</nautaneuvos>
```

Java-ohjelma, jolla NAUTA.XML-dokumentista muodostetaan A-netille sopiva tulosedokumentti:
 JDOMNAUTA.JAVA

```
import java.io.*;
import org.jdom.*;
import org.jdom.input.SAXBuilder;
import org.jdom.output.*;
import org.xml.sax.*;
import java.util.*;

public class JDOMNauta {

public JDOMNauta(){
}

private static String url = "http://www.cc.jyu.fi/~maiju/xml/nauta.xml";

public static void main(String[] args) {

//XML-TIEDOSTON LUKEMINEN
SAXBuilder builder = new SAXBuilder();

try{
Document doc = builder.build(url); //throws JDOMException
Element juuri = doc.getRootElement();

//TESTATAAN löytyykö lehma-elementti
if(juuri.getChild("lehma") != null){
System.out.println("Taalla on lehmia!\n");
}

//UUDEN DOKUMENTIN LUOMINEN

File tulos = new File("tulos.xml");
Element root = new Element("teuras");
Document uusidok = new Document(root);

//ELEMENTTIEN LUOMINEN DOKUMENTTIIN

Element tila = new Element("tila");
root.addContent(tila);

String tilanumero = juuri.getChild("tila").getChild("tilanro").getText(); //haetaan tilanumero-arvo
Element tilanumeroel = new Element("tilanro");
tilanumeroel.setText(tilanumero);
tila.addContent(tilanumeroel);

String tilinro = juuri.getChild("tila").getChild("tilinro").getText();
Element tilinroel = new Element("tilinro");
tilinroel.setText(tilinro);
tila.addContent(tilinroel);

Element lehma = new Element("lehma");
root.addContent(lehma);

String nimi = juuri.getChild("lehma").getChild("nimi").getText(); //haetaan nimi
Element nimiel = new Element("nimi");
```

```

nimiel.setText(nimi);
lehma.addContent(nimiel);

String numero = juuri.getChild("lehma").getChild("numero").getText(); //haetaan numero-arvo
Element numeroel = new Element("nro");
numeroel.setText(numero);
lehma.addContent(numeroel);

Element painoel = new Element("paino");
String paino = juuri.getChild("lehma").getChild("paino").getText();
painoel.setText(paino);
lehma.addContent(painoel);

Element syntyel = new Element("syntymaika");
String synty = juuri.getChild("lehma").getChild("syntymaika").getText();
syntyel.setText(synty);
lehma.addContent(syntyel);

Element pvmel = new Element("teurastuspvm");
String pvm = juuri.getChild("lehma").getChild("teurastuspvm").getText();
pvmel.setText(pvm);
lehma.addContent(pvmel);

//KIRJOITETAAN TULOS UUTEEN TIEDOSTOON

try{
XMLOutputter sarjallistaja = new XMLOutputter(); // muutetaan olio tavuvirraksi (sarjallistetaan)

//muotoillaan hierarkian mukaan:
sarjallistaja.setIndent(" ");
sarjallistaja.setNewlines(true);

FileWriter writer = new FileWriter(tulos);
sarjallistaja.output(uusidok, writer);
writer.close();

}catch(IOException e){
    System.out.println("Ongelmia tiedostoon kirjoittamisessa.");
    System.err.println(e);
}

}catch(JDOMException je){
    System.out.println("Tiedosto ei ole hyvin muodostettu.\n");
    System.out.println(je.getMessage());
}
}
}
}

```

Tulosdokumentti, joka lähetetään A-netille:

TULOS.XML

```

<?xml version="1.0" encoding="UTF-8"?>
<teuras>
<tila>
    <tilanro>55</tilanro>
    <tilinro>123123-54321</tilinro>
</tila>

```

```
<lehma>  
<nimi>maikki</nimi>  
<nro>1</nro>  
<paino>150</paino>  
<syntymaika>140501</syntymaika>  
<teurastuspvm>010802</teurastuspvm>  
</lehma>  
</teuras>
```

Tiivis yhteistyö viranomaisten kanssa

Lanson hankesuunnitelmassa viitataan yhteistyöhön viranomaisten kanssa. Yhteistyön viranomaisten kanssa kuuluisi olla joustavaa ja helppoa. Suonentiedon Agrineuvoksen kautta voidaan käsitellä EU-tukihakemuksia. Tuottajalle olisi kuitenkin helpompaa, jos tukihakemuksen voisi *lähettää suoraan internetin kautta* viranomaisille. Tämä on idea, jonka toteutuminen vaatisi paljon suunnittelua ja yhteistyötä, mutta joka tekisi kommunikoinnista viranomaisten kanssa tiiviimmän, vuorovaikutteisemmän ja ennen kaikkea nopeamman.

Tällä hetkellä Agrineuvoksen avulla pystytään ennakoimaan peltoviljelytuet ja laskemaan kullekin parhaat tukivaihtoehdot. EU-tukihakemuksen voi myös tulostaa, mikä tietenkin nopeuttaa viljelijän työtä. Tukihakemus täytyy kuitenkin postittaa eteenpäin. Verrattuna siihen, että lomakkeen pystyisi lähettämään suoraan internetissä, säästettäisiin paperi- ja postikuluja, lähettäminen olisi vaivattomampaa ja lähettämisen nopeus parantaisi ja tehostaisi viranomaisten ja viljelijöiden kommunikoinnin laatua.