XML -TOIMINTOJEN KEHITTÄMINEN SOLTEQ MERXISSÄ

Niilo Ruotsalainen

Tietojenkäsittelyn koulutusohjelma  
Huhtikuu 2005
Työn ohjaaja: Jyrki Vehmas

T A M P E R E  2 0 0 5
Tiivistelmä


Tutkintotyön toimeksiantaja on ohjelmistoja ja muita IT-palveluita tuottava yritys Solteq Oyj. Solteq Merx on toimeksiantajan kehittämä toiminnanohjausjärjestelmä tukkukaupan ja vähittäiskaupan toimialoille. Järjestelmä toimii IBM iSeries suurtietokone laitealustalla.

Tavoitteena tässä työssä on kehittää Solteq Merxiin valmius vastaanottaa ja muodostaa XML -muotoisia tietoja. Tähän tavoitteeseen pyritään työssä esiteltävällä sovelluskokonaisuudella.


Sisällysluettelo

1 JOHDANTO ................................................................................................................................. 5

2 TAUSTAA TUTKINTOTYÖLLE ..................................................................................................... 6
   2.01 TAVOITTEET JA TOIMEKSIANTAJA .................................................................................. 6
   2.02 TUTKINTOTYÖN OSA-ALUEET .......................................................................................... 7
   2.03 SOLTEQ OYJ ...................................................................................................................... 7
   2.04 SOLTEQ MERX-TOIMINNANOHJASIÅRJESTELMÄ ................................................................. 8
   2.05 SOLTEQ JA XML ............................................................................................................... 9

3 TÄRKEÄT KÄSITTEITÄ .................................................................................................................. 9
   3.01 IBM iSERIES-LAITEALUSTA ............................................................................................ 9
   3.02 OS/400 KÄYTTÖJÄRJESTELMÄ ........................................................................................ 9
   3.03 INTEGRATED LANGUAGE ENVIRONMENT .................................................................. 9
   3.04 ILE RPG -OHJELMOINTIKIeli ........................................................................................ 10
   3.05 XML .................................................................................................................................. 11
   3.06 XML-PARSERI ................................................................................................................ 14
   3.07 XML TOOLKIT FOR iSERIES ......................................................................................... 15
   3.08 EDI/OVT ........................................................................................................................ 15

4 EDI/OVT LIITTYMÄT SOLTEQ MERXISSÄ ............................................................................... 17

5 XML - TIEDONSIIRTOMUOTOINA ............................................................................................. 18
   5.01 OMINAISUUDET ............................................................................................................... 19
      5.01.1 Tiedon ymmärtettävyys ............................................................................................. 19
      5.01.2 Tiedon rakenne ........................................................................................................ 20
      5.01.3 Tiedon siirtäminen ................................................................................................... 20
      5.01.4 Tiedon käsittely ........................................................................................................ 20
   5.02 LIITTYMÄN RAKENTAMINEN ......................................................................................... 21
      5.02.1 Järjestelmävaatimukset ............................................................................................ 22
      5.02.2 Liittymän suunnittelu ............................................................................................... 22
      5.02.3 Liittymän toteutus .................................................................................................... 23
      5.02.4 Yhteenveto XML-tiedonsiirroista ........................................................................... 25

6 XML JA EDI -STANDARDIT ........................................................................................................ 25
   6.01 XML VS. EDI STANDARDIT .............................................................................................. 26
      6.01.1 Sähköisen kaupankäynnin standardit ....................................................................... 26
      6.01.2 Tiedon esittystavat ja käytettävyys ......................................................................... 26
      6.01.3 Tiedon käsittely ........................................................................................................ 27
      6.01.4 Yhteenveto XML:n ja EDI-standardien vertailusta .................................................. 28
   6.02 XML/EDI TULEVAISUUDEN RATKAISU? ................................................................. 29

7 XML -TOIMINNOT SOLTEQ MERXISSÄ ............................................................................... 30
   7.01 LAHHTOTILANTEEN SELVITYS .................................................................................. 31
   7.02 XML - TOIMINTOJEN KEHITYS .................................................................................. 31
   7.03 XML – TOIMINTOJEN TULEVAISUUS ................................................................. 33

8 XML -RAJAPINTA SOLTEQ MERXIIN ................................................................................... 33
   8.01 VAATIMUSMÄÄRITTELY .................................................................................................. 34
      8.01.1 XML-tiedoston muodostaminen .............................................................................. 34
      8.01.2 XML-tiedoston purkaminen ................................................................................... 35
      8.01.3 Kokonaisuuden ylläpidettävyys .............................................................................. 35
      8.01.4 Uusienn XML-liittymien rakentaminen rajapintasovelluksen avulla ....................... 36
   8.02 TEKNINEN SUUNNITTELU ............................................................................................ 36
      8.02.1 Kokonaisuuden jakaminen toiminnalliseen moduleihin ........................................ 36
      8.02.2 XML Toolkitin toimintojen käyttäminen ............................................................... 36
      8.02.3 Tietojen lukeminen ja kirjoittaminen tietokantaan ............................................... 37
8.03 TOTEUTUKSEN KUVAUS ................................................................................................................... 38
  8.03.1 XML -dokumentin muodostus .................................................................................................. 38
  8.03.2 XML -dokumentin purku ......................................................................................................... 39
8.04 XML -LIITTYMÄN RAKENTAMINEN TOTEUTUKSEN AVULLA ............................................................... 41
  8.04.1 XML -sanoman lähetys ........................................................................................................ 41
  8.04.2 XML -sanoman vastaanotto ................................................................................................. 42
9 LOPPUTULOS JA ARVIOINTI ................................................................................................................. 44
  9.01 LOPPUTULOS ................................................................................................................................ 44
  9.02 LOPPUTULOKSEN ARVIOINTI .................................................................................................. 44
LÄHTEET.................................................................................................................................................... 46
1 Johdanto

Nykyaikeisen tietojenkäsittelyn tärkeimpiä ja kasvavimpia toimintoja ovat tietojen automaattiset siirrot järjestelmästä toiseen. Perinteisesti tämä tarkoittaa lähettävää järjestelmässä muodostettua konekielistä sanomaa, joka siirretään vastaanottavaan järjestelmään, jossa sanoma puretaan järjestelmän tietokantaan. Tänä päivänä XML -teknologia tarjoaa tähän joustavamman ja edullisemman vaihtoehdon.

XML -tiedonsiirtoon perustuvien järjestelmien välisien liittymien määrä ja kysyntä on kokoajan kasvussa. Siksi toiminnanohjausjärjestelmiä toimittavien yritysten on perehdyttävä tähän teknologiaan ja pyrittävä tarjoamaan omat ratkaisunsa näihin toimintoihin.


Työn lopputuloksena esitellään sovelluskoonaisuus, jonka avulla XML -tiedonsiirtojen toiminnallisuudet toteutetaan käytössä olevilla työkaluilla. Lisäksi käydään läpi käytetyt ohjelmointitavat ja kerrotaan kuinka näistä komponenteista muodostetaan toimiva tiedonsiirto.
2 Taustaa tutkintotyölle

Tiedonsiirrot Yritykset pyrkivät tehostamaan toimintojaan ja säästämään kustannuksia tekemällä yhteistyötä. Yritysten välisen verkottumisen edellytyksen on tietojen vaihto. Tänä päivänä tietoverkkojen siirtoneuvojen kasvu mahdollistaa suurten tietomääräisten ja monimuotoisten sisältöjen siirtämisen sähköisesti organisaatioiden välillä. Sähköisen viestinnän määrän ja laadun kasvessa vaaditaan yritysten tietojärjestelmiltä kykyä vastaanottaa ja lähettää monenlaisia ja muotoisia tietoja. (Arto Merterniemi 1999.)


2.01 Tavoitteet ja toimeksiantaja


Tavoitteet Saavuttaakseen toimeksiantajan tavoitteen on työssä rakennettava sovellus tai sovelluskoenaisuus, jossa on huomioitu uudelleen käyttävyys Solteq Merxin ja muiden järjestelmien välissä tiedonsiirroissa. Toteutuksen vaatimuksena on siis monikäyttöisyys, eli sen on pystyttää käsittelemään usein useita tiedonsiirroissa käytettyä XML -tiedostomuodot. Ratkaisussa on myös otettava huomioon Solteq Merxin liitteistön ja käyttöjärjestelmäympäristön erikoisvaatimukset.

Jotta edellä mainitusta toteutuksesta saadaan paras hyöty irti, on työssä vertattava XML:n ominaisuuksia tiedonsiirtoon järjestelmässä pääasiassa käytetyn tiedonsiirtonäyteon ominaisuuksiin. Vertailun tarkoituksena ei ole pelkästään XML:n etujen kartoitua, vaan yhtäaikaista testausohjelman henkikoulu. Tältä osin tutkintotyön tavoitteena on selvittää missä XML:n käytöstä saadaan selkeää hyötyä ja missä sitä ylipääätään voidaan käyttää. Tämä selvitys tehdään omasta osionaan tutkin-

Toteutusosa Toteutusosa esittelee tutkintotyön tavoitteissa määritellyn XML:ää käsittelevän sovelluskokonaisuuden. Tässä osassa käydään läpi toteutuksen vaatimusmäärittely, toteutuksen suunnittelu ja esitellään ratkaisu.

2.03 Solteq Oyj


Tuotteet ja palvelut Solteq tarjoaa asiakkailleen ohjelmistoja, palveluja ja laitteita. Omaa ratkaisutarjontaa täydennetään yhteistyökumppaneiden ohjelmistotoimilla ja teknologialla. Tärkeimmät kumppanit ovat alojensa markkinajohtajat SAP ja IBM. ”Solteqin, SAP:n ja IBM:n yhteistyö tarjoaa kotimaisille keskisasuille ja kasvaville yrityksille ratkaisukokonaisuuden, jossa hyödynnetään SAP:n ja IBM:n kansainvälistä järjestelmiä- ja teknologiaosaa sekä Solteqin toimialaosaa ja asiantunteista palallisilla markkinoilla.” (Solteq Oyj:n Internet sivut 2004.)

Toimialat Solteqin asiakaskunta on toimintavuosien saatossa kasvanut teknisen tukikaupan ja autokaupan aloilta myös metsäteollisuuteen, asiakaskohtaisiin ratkaisuihin ja vähittäiskauppaan. Muutamia Solteqin asia-
Solteqin tavoitteena on olla mukana kehittämässä asiakkaan liiketoimintaa toimittamalla kokonaisvaltaisesti kaikkia IT-palveluita, toimialan tai asiakaskohtaisen tarpeen mukaan. Pyrkimyksenä on luoda pitkäaikaisia ja kumppanuutta korostavia asiakassuhteita.

Toimialaratkaisut

Palvellakseen juuri tietyn kohderyhmän asiakaskuntaa on Solteqin toiminta jaettu toimialakohtaisiin kokonaisratkaisuihin, asiakaskohtaiseen järjestelmäkehitykseen sekä kauppasvetoisille myymälöille ja erikoistavarakaupoille (Point-of-sale) suunnattuihin myymäläratkaisuihin.

Toimialakohtaisia kokonaisratkaisuja Solteq tarjoaa ketjuuntoonelle kaupalle, autokaupalle ja valikoidulle teollisuussegmentille. Näillä toimialoilla Solteq tarjoaa ratkaisun kaikkiin asiakasyrityksen tiedonhallintaan kohdistuville kehitysteräpeissä. (Solteq Oyj:n Internet sivut 2004.)

2.04 Solteq Merx -toiminnanohjausjärjestelmä


Järjestelmän käyttö on rakennettu joustavaksi, jotta erityyppiset yritykset pystyvät käyttämään sitä omissa liiketoimintoissaan. Joustavuus ilmenee mm. siten, että järjestelmä palvelee moniyritys- ja monivarasto- ominaisuuksia hyödyntävät yrityksiä yhtä hyvin, kuten yhden varaston yrityksiäkin. Järjestelmästä löytyvät myös erityyppiset myyntitoiminnot, joista järjestelmä ylläpitää monipuolista tilastomateriaalia. Järjestelmän ostotoiminnot mahdollistavat manuaaliset tai automaattiset ostot. Merxin varastotoiminnot tukevat useita erilaisia toimintatapoja, varastokohtaisesti. (Solteq Merx -esite 2003.)

kasyrityksiä vapaasti poimittuna: UPM Kymmene/Schauman Wood Oy, Berner Oy, Notex Yhtiöt Oy (Vapaavalinta), Oy Starkki Ab, Nissan, Veho Group Oy Ab, Töölön Matkatoimisto Oy. (Solteqin vuosikertomus 2004.)
2.05 Solteq ja XML


3 Tärkeitä käsitteitä

3.01 IBM iSeries -laitealusta

IBM iSeries sarjan tuotteet ovat integroitujen palvelinkokonaisuuksia, joissa on useiden käyttöjärjestelmäympäristöjen yhtäaikaisen käytön tuki. IBM iSeries palvelimet ovat suunniteltu yksinkertaistamaan järjestelmäympäristöjä ja nopeuttamaan sähköisen liiketoiminnan käyttöönottoa. Erityisiä ominaisuuksia palvelimissa ovat käyttövarmuus ja halvinointi tarpeen vähäisyys. IBM iSeries palvelimet tunnettiin aikaisemmin nimellä AS/400. (IBM:n Internet sivut 1994 - 2005.)

3.02 OS/400 käyttöjärjestelmä

IBM iSeries laitealustan peruskäyttöjärjestelmä on OS/400. OS/400 muodostaa integroidun kokonaisuuden IBM DB2 tietokannan kanssa. OS/400 on suunniteltu suuria tietomääriä käsittelevien sovelluksien alustaksi. Käyttöjärjestelmän tämän hetkinen versio on VR5R3. Yhdesä iSeries palvelinten kanssa käyttöjärjestelmä muodostaa iSeries järjestelmän. (IBM:n Internet sivut 1994 - 2005.)

3.03 Integrated Language Environment

Integrated Language Environment eli ILE käsitteää joukon työkaluja ja toimintoja, jotka ovat suunniteltu parantamaan sovelluskehitystä iSeries järjestelmässä. Nämä toiminnat antavat mahdollisuuden tehdä modulaarisia sovelluksia, joissa on rinnakkaisesti suoritettavia eri kielillä ohjelmointu ja moduleita. ILE tarjoaa ohjelmointiin myös erilaisia sisään rakennettuja funktioita, jotka mahdollistavat mm. dynaamisen muistin-käsittelyn ja päivämäärien käsitteilyn.

**3.04 ILE RPG -ohjelmointikieli**


Rivien järjestys lomakepohjaisessa RPG IV ja ILERPG syntaksissa:

1. Tiedostot
2. Tiedostojen kenttien ja nimien ohjelman sisäiset muutokset
3. Muuttuja määrittelyt, tietorakenteet (ohjelman sisäiset tiedostot), taulukot ja ohjelman aliproceduuriens prototyyppit (esittelyt)
4. Parametristimistat
5. Avainlistat
6. Ohjelman logiikka (I/O)
7. Ohjelman lopetus

**Lisäksi ohjelmakoodissa lopetuksen jälkeen voi olla seuraavia toimintoja:**

1. Alirutineja
2. Tietuekäsittelyjä
3. Aliprocedureja
4. Listauskuvauksia

Ohjelmoijalle ILE RPG on yksinkertaisuutensa vuoksi helppoa opetella ja ymmärtää.

Rajoitukset


3.05 XML

Määritelmä

XML (Extensible MarkUp Language) on metakieli, eli sillä kuvataan tiedon rakennetta. Tieto eli data koostuu yleensä tekstistä tai kuvista, jotka on järjestetty tietyt periaatteet mukaiseesti. XML on väline, jolla voidaan esittää tiedon hierarkia tai kapseloida data.

Rakente


Kuva 1. XML:n syntaksi.

XML:n syntaksi:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dokumentti>
    <elementti attribuutti = "ominaisuus" > Data </Elementti>
</dokumentti>
```

Esitysmalli


Dokumenttipuun rakenne muistuttaa hyvin pitkälti oikeaa puuta:

1. Puussa on juuri; dokumenttipuussa on runko (body, root node)
2. Puussa on oksia; dokumenttipuussa on lapsielementtejä (children, child node)
3. Puussa on lehtiä; dokumenttipuussa on lehtiä (nodes), joilla ei ole enää omia lapsielementtejä.

**Esimerkki XML -dokumentista:**
```xml
<?xml version="1.0" encoding="UTF-8"?>
<JUURI>
    <RUNKO>
        <OKSA>
            <LEHTI> Sisältö1 </LEHTI>
            <LEHTI> Sisältö2 </LEHTI>
        </OKSA>
        <OKSA>
            <LEHTI> Sisältö3 </LEHTI>
            <LEHTI> Sisältö4 </LEHTI>
        </OKSA>
    </RUNKO>
</JUURI>
```

Kuva 2. Esimerkki XML -dokumentista.

**Dokumenttipuun rakenne:**

Kuva 3. Dokumenttipuun rakenne.

Tärkein piirre dokumenttipuussa on, että siinä voidaan liikkua vapaasti ja se voidaan käydä läpi viittamaalla mihin tahansa solmun lapseen, vanhempaan tai sisarukkeen. Jokainen solmu, eli dokumentin elementti, on oma olionsa, jolla on omat ominaisuudet ja metodit. (Koulutus ja konsultointipalvelu KK Mediat 2000 - 2004.)

**Rakennekuvausset** XML -dokumentin rakenne voidaan kuvata, joko dokumentin yhteydessä DTD (document type definition) -kuvausella tai ulkoisella XML
-schema tiedostolla. DTD -kuvaus voidaan esittää myös ulkoisessa tie-
dostossa. Tiedon rakenteen kuvaamista tarvitaan, jos XML -
dokumentti halutaan validoida. Validoiminen tarkoittaa sekä XML -
dokumentin rakenteen että tietosisällön tarkastamista dokumentin ku-
vauksen mukaisesti. Tästä hyödytään tiedonsiirroissa, koska esim. jär-
jestelmään saapuva XML -dokumentti voidaan näin tarkastaa ennen kää-
sittelyä. Validoinnissa esiintyvät virheet voidaan huomioida ohjelma-
koksi ja näin ollen vähennetään riskiä tietojen korruptoitumisesta
siirron aikana.

DTD -kuvaus

Yleisimmin käytetty teknikka XML -dokumenttien rakenteen kuvaa-
miseen on DTD kuvauset. DTD kuvauset sisältävät elementtien hier-
rarkian sekä elementtien nimen ja sisällön muodon.

Esimerkki DTD kuvauksesta (kuva em. XML -dokumentin rakenteen):

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT JUURI (RUNKO)>
<!ELEMENT LEHTI (#PCDATA)>
<!ELEMENT OKSA (LEHTI+)>
```

Kuva 4. Esimerkki DTD -kuvauksesta.

Ulkoisen DTD -kuvausen liittäminen XML -dokumenttiin:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE JUURI SYSTEM "DTD_tiedoston_nimi.dtd">
<JUURI> ...
```

Kuva 5. Ulkoisen DTD -kuvausen liittäminen XML -dokumenttiin.

XML -schema

XML -scheman avulla dokumenttien rakenne voidaan määritellä tar-
kemmalla tasolla. Nimiavaruuksen hyödyntäminen, muissa määrityk-
sissä käytettävien kantaluokkien rakentamisen mahdollisuus ja tieto-
tyypien erittely antavat etua varsinkin tiedonsiirrossa, joissa tietojen
eheyden säilyttäminen ja tietotyypien ymmärtäminen ovat keskeisiä
asioita. Nykypäivänä onkin suositeltavaa käyttää schemoja XML -
dokumenttien rakenteen kuvausissa. (Helsingin ammattikorkeakoulun
internet sivut 2005.)
Esimerkki schema -kuvauksesta (kuva em. XML – dokumentin rakenteen):

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="JUURI">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="RUNKO" maxOccurs="1">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="OKSA" type="xsd:string"/>
<xsd:element name="LEHTI" maxOccurs="unbounded" type="xsd:string"/>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```


```
<?xml version="1.0" encoding="UTF-8"?>
<JUURI xmln:s:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="schema_tiedoston_nimi.xsd">
   <RUNKO>...
```


### 3.06 XML -parseri


Parserin toiminta

Yksinkertaisimmillaan parseri "vain" lukee rakenteen ja muodostaa sitä dokumenttipuun, mutta käytännössä useimmat parserit tekevät myös muototarkistuksen. Useimmissa parsereissa käytetään oikeamuotoinen (well-formed), kun taas jotkut hallitsevat myös DTD:n ja vaikeuttelevän ohjelman on helppo hylätä virheellinen tiedosto, jos parserin muototarkistus ei onnistu. Myös virheen muoto saadaan helposti selville. Tämän vuoksi XML:n jäsentämiseksi syntyvät vähemmän tulkintavirheitä, mikä johtaa yhtenäisempään ja selkeämpiin toteutuksiin. (Jouni Heikniemen Internet sivut 2001.)

3.07 XML Toolkit for iSeries


3.08 EDI/OVT

Määritelmä

EDI/OVT tarkoittaa elektronista, määramuotoista ja automaattista tiedonsiirtoa yritysten sekä julkishallinnon tietojärjestelmien välillä. Sen tuomia etuja kuten; paperirutiinien vähentämistä, nopeutta ja virheettomyyttä voidaan hyödyntää kaikissa yrityksen toiminnossa. Lyhenne EDI tulee englannin kielen sanoista Electronic Data Interchange ja OVT on EDI:n suomenkielinen vastine eli organisaatioiden välinen tiedonsiirto.

OVT:llä tarkoitetaan menettelyä, jossa tietojärjestelmän A tiedoista muodostetaan määramuotoinen tietovirta, joka lähetetään sähköisenä tiedonsiirtona tietojärjestelmään B. Järjestelmässä B tietovirta pureetaan ohjelmallisesti suoraan järjestelmän tietoihin, joista tiedot ovat heti järjestelmän käytettävissä. Esimerkkejä tietovirroista ovat tilaus, lasku, laskun maksuosoitus pankkiin, hinnasto tai tuoteluettelo.

Hyödyt

Sähköisen tiedonsiirron hyötyjä ovat tiedon nopea kulku ja virheettömyyys. Vastaanottaja saa tiedon edelleen käytettävissä muodossa ja tiedonhallinta tehostuu. Automatisointi tuo suuria kustannussäästöjä. Täy-

Rakenne


Tietosisältö on se EDI:n osa-alue, jota kehitetään eniten. Tietosisältö tarkoittaa tietyn tyyppisen siirrettävän tiedon sisältöä. Esimerkiksi laskutiedoilla ja tuotetiedoilla on omia tietosisältömääriyksiä eri tarkoituksiin ja toimintatapoihin. Laskutietojen tietosisältö tarkoittaa sitä, mitä tietoja laskusta välitetään organisaatioiden välillä.

EDI:n esitystapa tarkoittaa tiedon muotoa siirron aikana. EDI pyrkii tarjoamaan yhden yhtäläisen tietojärjestelmien esitystavoista riippumattoman käytännön organisaatioiden välisiin tiedonsiirtoihin. EDI:n esitystavoista vakiintunein ja tunnetuin on EDIFACT -standardi (Electronic Data Interchange For Administration, Commerce and Transport), joka nimestään huolimatta on käytössä kaikilla liiketoiminnan alueilla. (Jarna Hakala, EDI - Electronical Data Interchange /OVT – Organisaatioiden välinen tiedonsiirto 3.11.1998.)

Standardit


Tietoliikenne

Tiedon fyysisen siirtämiseseen on tarjolla useita eri tapoja. Seuraavassa on luettelut sanomien kuljetuksessa käytettäviä vaihtoehtoja:

- X.400-sanomanvälitysjärjestelmä
- BSC/RJE (IBM ympäristössä käytetty etätöiden siirtomenettely)
- OFTP (ODETTE File Transfer Protocol)
- Useiden palveluverkkojen EDI tai aineistonsiirtopalvelut
- Lähde kaikki menettelyt, joiden avulla voidaan siirtää tiedostoja
4 EDI/OVT liittymät Solteq Merxissä

Liittymättyypit Solteq Merxiin on rakennettu, sen kehityskaaren aikana, kymmeniä erilaisia liittymiä. Osa näistä liittymistä on ns. villejä yksittäistapauksia, joissa liittymän rakenne ei perustu mihinkään EDI-standardiin. EDI-tiedonsiirroksi voidaan määritellä esim. erilaiset myynti- ja ostotilausliittymiä, jotka noudattavat jotain yleisesti tunnettua standardia liittymäkuvausta. Muita EDI-liittymiä on olemassa mm. tuotetietojen, hinnastojen, tilausvahvistusten ja laskujen siirtoon.


Esimerkiksi toimittajatiedot poimitaan tai vastaan otetaan tiedostoon, joka tietueessa on yksi 200 alfanumerista merkkiä pitkä kenttä. Kenttässä tietty positiot on varattu tietyille tiedoille, jotka määritellään liittymäkuvausksessa.

Kuva 8. Esimerkki tiedonsiirroissa käytetystä tietojen esitystavasta:

Esimerkki tiedonsiirroissa käytetystä tietojen esitystavasta:
Tietue: (Alkuposition...Loppoposition)
1...............................................................
.....200
Toimittajan numero 7 numeerista merkkiä 0 desimaalia:
1...7
Toimittajan nimi 20 alfanumerista merkkiä:
9.........29
Toimittajan nimen jatke 20 alfanumerista merkkiä:
30........50
Toimittajan postiosoite 15 alfanumerista merkkiä:
52...67 ... jne.

Tietojen käsittely


Rajoitukset

Nykyaikaisessa tietojenkäsittelyssä toimintojen vaatimukset ja määrittelyt voivat muuttua nopeassa tahdissa. Tämä aiheuttaa edellä mainitun tyyppisten liittymien ylläpidolle ongelmia, koska käytetyssä ohjelmointitekniikassa siirrettävien tietojen muuttaminen tai lisääminen edellyttää aina monimutkaisia ohjelmamuutoksia.

EDIFACT

EDIFACT -muunninta ei Solteq Merxissä ole, vaan kyseinen ominaisuus on liitetty muun järjestelmään omalla ohjelmointirajapinnallaan. Muunin tarkoittaa sovellusta, joka muuntaa tietokantatiedoston määritelmään EDIFACT -esitysmuotoon. Tämä toiminto on Solteqin yhteistyökumppanin jokaiselle yksittäiselle asiakkaalle erikseen toimittama, joten menettely aiheuttaa ylläpito-ongelmia. EDIFACT -standaari perustuvat tiedonsiirrotkin siis kuuluvat järjestelmän tiedonsiirtovalikon

Yhteenveto

Suurin ongelmia olemassa olevien liittymien kohdalla on kuitenkin niiden kirjavuus. Esimerkiksi tuotetiedolle on tehty monia erilaisia liittymäiä, jotka kaikki pohjautuvat erilaisiin sanomastandardisiin, vaikka tuotetiedot sinällään ovat aina samankaltaisia. Liittymien hallintaa kokonaisuutena ei tämän vuoksi voida tehdä. Tämä aiheuttaa kustannuksia ja resurssejä turhakäyttöä, koska joudutaan useasti keksimään pyörä uudelleen.

5 XML – tiedonsiirtomuotona

XML:ään liitetään yleensä kolme uskomusta:
1. Tiedon avoimuus ja vapaus
2. Universaali tietojenkäsittelyratkaisu
3. Teknologian määrittämä toimintamalli

1. XML itsessään ei tee tietojen sisältöä vapaaksi tai avoimeksi, sillä se on vain työkalu, jolla kuvataan tiedon rakennetta. Kuitenkin XML on vapaasti sovellettavissa oleva työkalu, joka voi lisätä tiedon avoimuutta siten, että yritys A:n tuottamat XML-dokumentit voidaan käsittellä yritys B:n järjestelmällä.

2. Jos kaikille tiedoille määritettäisiin maailmanlaajuisesti yhtenäinen standardi XML-schema, niin sen avulla voitaisiin ratkaista kaikki tiedonsiirtojen ongelmat. Kuitenkaan tämä ei ole nykypäivänä mahdollista, joten XML-tiedonsiirroissa on tiedoille usein määritettävä schema
liittymäkohtaisesti. Esimerkiksi valtakunnan tasolla on tietyille tieoille jo olemassa standardeja esitystapoja. Tästä hyvänä esimerkkinä ovat pankkiyhdystyksen laatimat FINVOICE laskustandardit, jotka perustuvat XML -schemaan.


Tietojärjestelmien välisen tiedonvaihto ja erityisesti järjestelmien integraatio ovat eräitä tietojenkäsittelyyn perinteisempiä ongelmia. Tiedonvaihdossa kaksi järjestelmää toimii yhdessä siten, että tiedon lähetäjä järjestelmässä muodostetaan siirtotiedosto, jonka vastaan ottava järjestelmä pystyy tulkitsemaan. Tässä käytetään apuna yhdessä sovituja menetelmiä, kuten tiedon rakenne ja muoto sekä käytettävä tietoliikenne menetelmä. Kuinka XML sitten soveltuu järjestelmien väliseen integraatioon sen eri osa-alueille? (Timo Helander, XML käyttäminen sovellusten välisessä tiedonvaihdossa.)

5.01 Ominaisuudet

5.01.1 Tiedon ymmärrettävyys

Järjestelmien välissä tiedonsiirrossa siirrettävän tiedon ymmärrettävyydellä on suuri merkitys. Siirrettävän tiedon tulee olla muodoltaan sellaista, että lähettävää osapuolia pystyy sitä muodostamaan ja vastaanottava osapuoli pystyy sitä käsittelemään. Ongelmia tähän tuovat laitealustojen ja ohjelmistojen eroavaisuudet, kuten Unix (Linux) – PC tai PC – iSeries ja merkistöeroavaisuudet, kuten skandinaaviset merkit ja asialaiset merkistöt.

XML:n parhaita puolia on se, että tään päivänä kaikille laitealustoille ja käyttöjärjestelmille löytyy työkalut sen käsittelemiseen. Tämän mahdollistaa se, että XML on puhdasta tekstitietoa, jota kaikki järjestelmät pystyvät tulkitsemaan.

Merkistöongelmat XML:ssä on pyrittä ratkaisemaan määrittämällä dokumenteille niissä käytettävä merkistötunnus. Käytettävä XML -parseri ratkaisee mitä merkistöjä voidaan käyttää. Käytännössä tämä ei ole aivan ongelmatonta, sillä esimerkiksi iSeriesksen XML Toolkitin ILE RPG:n XML -parseri käyttää oletuksena pohjoisamerikkalaista UTF-8 merkistöä ja tämän takia kaikki dokumentissa käytettävä tieto on muu-
tettava erikseen ländsierooppalaiseksi ISO-8859-1 merkistöksi suomalaissyntisä järjestelmissä.

5.01.2 Tiedon rakenne

Tiedon siirrhoissa käytettävän siirrotiedoston rakenne aiheuttaa usein siirroissa tapahtuvia virheitä. Siirrotiedoston eli siirrettävän tiedon rakenne täytty olla yksiselitteistä ja vakiomuotoista. Siirtojen yleissälievit virheet syntyvät usein näiden ominaisuuksien huonosta suunnittelusta tai toteutuksesta. Nykypäivän nopeasti muuttuvassa maailmassa myös siirrettävien tietojen vaatimukset voivat muuttua usein. Tämä edellyttää tiedon rakenteelta joustavuutta, koska yksittäisen tiedon lisääminen tai poistaminen ei saa aiheuttaa koko liitymän uudelleen rakentamista.


5.01.3 Tiedon siirtäminen

Tehokkaassa tiedon siirrhoissa tietoliikenteen väliyksellä tapahtuva siirto vaatii suurien tiedostojen osalla isoa siirtokapasiteettia. Tehokkainta olisi välittää tietoja järjestelmästä toiseen mahdollisimman tiiviissä muodossa.

XML -tiedostoa ei voida pitää kovin tiiviinä pakettina, koska se sisältää tieto-osan lisäksi myös tiedon rakenteen. XML -tiedoston koko voi olla itse asiassa yli puolet enemmän mitä pelkän tieto-osan koko on. XML:n käyttäminen tiedon siirrhoissa vaatiikin hyviä tietoliikennehyteyksiä. Tämä päivänä siirtotappauksien kasvavissa myös XML:n käyttömahdollisuudet ovat lisääntymässä.

5.01.4 Tiedon käsittely

Pelkkä tiedon lähettäminen ja vastaanottaminen ei vielä anna paljoa hyötyä liittymästä. Tärkeää on nimenomaan se kuinka tietoa käsitellään ja mihin sitä käytetään. Tieto pitää pystyä käsittelemään mahdollisim-


5.02 Liittymän rakentaminen

Tänä päivänä lähes kaikki järjestelmätöimittajat pystyvät rakentamaan XML-liittymiä, mutta kaikkia XML:n ominaisuuksia ei osata aina hyödyntää. Tämä johtuu siitä, että XML on vielä nuori teknologia ja XML:n käyttö tiedonsiirtotuotona asettaa liittymän suunnittelun ja toteutukseen joitakin vaatimuksia. Kuinka sitten XML:n käytöstä saa-
daan parhaiten hyödyt esille ja mitä asioita pitää huomioida liittymää rakennettaessa.

5.02.1 Järjestelmävaatimukset


XML:n käsitelly ohjelmaa taas vaatii järjestelmiltä riittävän suurta tehoa ja suurten XML -dokumenttien siirtäminen edellyttää hyviä tietoliikennepätevyksiä. Nykyään lähes kaikilla järjestelmillä on valmius muodostaa ja vastaanottaa XML -muotoista tiedon siirtoa, mutta ennen liittymän tekemisen aloittamista kannattaa tehdä selvitys kaikkien osapuolten osalta.

5.02.2 Liittymän suunnittelu


Aina ei siirrettäville tiedoille löydy sopivaa valmistaa kuvauksesta. Tällöin kuvauksella kannattaa tehdä suunnitella yhdessä liittymän toisen osapuolen kanssa ja sovittaa tarkasti kuinka menetellään, jos kuvauksesta halutaan muuttaa. Kuvauksissa kannattaa käyttää aina XML -schema määritystä, jos se on mahdollista, koska se on paljon tarkempi kuin DTD kuvaukset. Siirrotiedoston kuvauksella pitää olla riittävän tarkasti määritelty, mutta liiallisuukiinkaan ei kannata mennä. Esimerkiksi elementtien tietotyyppien
kuvaus kannattaa yleensä jättää pois, koska käsittelevät järjestelmät pystyvät tekemään tiedoille tarvittaessa tyyppimuunnoksia. Hyvä kuvaus on kompromissi kahden järjestelmän välillä, joka tarjoaa riittävästi tietoa ilman rönnyilyä ja jossa tietojen hierarkia ja kapselfointi mallintaa oikeaa elämää.

Nykään on kuitenkin olemassa paljon liittymiä, joissa siirtotiedostolle ei ole tehty minkäänlaista kuvausta, jolloin rakenteesta ja siirrettävistä tiedoista on sovittu muilla tavoin. Tätä toimintatapaa on perusteltu sillä, että näin on ennenkin toimittu perinteisissä liittymissä, joissa kuvausta ei voida tehdä. Kuvauksen puuttuminen aiheuttaa sen, että siirrettävää tiedostoa ei voida tarkastaa ennen lähetystä tai vastaanottoa ja siten menetetään yksi XML:n suurimmista edusta.

5.02.3 Liittymän toteutus

Liittymän rakentamisessa XML:n ominaisuuksien hyödyntäminen edellyttää niiden tuntemista. Mitä liittymän toteutuksessa on huomioitava, jotta lopputulos olisi hyvä?


XML -liittymissä voidaan vähentää siirrossa tapahtuvien virheiden määrää, käyttämällä parserin muototarkastusta eli validointia. Jos käyttössä on siirrotiedoston kuvaus, niin ensimmäinen askel käsittelevää ohjelmasia on tiedoston validointi, jossa käsiteltävän tiedoston rakenne ja sisältö tarkastetaan kuvausta vastaan. Tällöin siirronaikana tapahtuneet tietojen korruptoitumiset tai muodostuksessa tapahtuneet virheet saadaan selville ennen tietojen varsinaista käsitelyä. Vastaavasti siirrotiedostoa muodostettaessa validointi kannattaa tehdä viimeiseksi ennen tiedoston lähetystä. Käytettävää XML -schemaa siirtotiedoston kuvausenä ei siirrossa voi tapahtua muotovirheitä, jos lähetysprosessia on tehty validointi. XML:n uudelleen käytettävyys kannattaa huomioida tiedoston käsitteleyssä siten, että käsiteltävää tiedostoa ei välttämättä kannata heti poistaa järjestelmästä. Esimerkiksi XML-muotoinen lasku voidaan tulostaa...
ruudulle tai paperille myöhemmin XSL tai XSLT käännöksenä. Jos siirrettävän aineiston koko on suuri, kannattaa tiedosto kuitenkin tiivistää pakattuun muotoon säilytystä varten.

Suurten siirtoaineistojen kohdalla muodostettava tiedosto voidaan tehdä tiivistetyn muotoon jättämällä ns. white space, eli elementtien väliset tyhjät tilat pois. Tästä on hyötyä sillä suurin osa FTP -protokollaa hyödyntävistä tiedonsiirtosovelluksista käsittelee nämä tilat tyhjinä merkeinä, jolloin tiedoston koko kasvaa. Useimmissa parsereissa voidaan ohjata tätä ominaisuutta validoinnin yhteydessä. Parserin toiminnan kannalta tällä ei ole merkitystä, joten toimintoa kannattaa käyttää siirron nopeuden lisäämiseksi.

Kuva 9. Esimerkki XML -dokumentista, jossa white space ovat näkyvissä:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<BATCH>
  <INVOICE>
    <Lang>FI</Lang>
    <InvType Type="IN"></InvType>
    <InvNr>30</InvNr>
    <Header>LASKU/FAKTURA</Header>
    <InvDate>20031104</InvDate>
    <DueDate>20031104</DueDate>
  </INVOICE>
</BATCH>
```

Kuva 10. Esimerkki XML -dokumentista, josta white space ovat poistettu:

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<BATCH><INVOICE> <Lang>FI</Lang><InvType Type="IN"></InvType><InvNr>30</InvNr>
  <Header>LASKU/FAKTURA</Header><InvDate>20031104</InvDate><DueDate>20031104</DueDate></INVOICE></BATCH>
```

Koska XML noudattaa aina tiettyjä kielipäätöksiä, voidaan sen käsitteily varten tehdä valmiita funktioita, luokkiä tai proseduurereja riippuen käytettävästä ohjelmointikielestä. Näitä yleiskäyttöisiä toiminnal-
lisuuksia voidaan siten käyttää useissa kohteissa, jolloin saadaan lisättävä tehokkuutta ja laskettua kustannuksia liittymien rakentamisessa.


5.02.4 Yhteenveto XML -tiedonsiirroista


6 XML ja EDI -standardit


6.01 XML vs. EDI standardit


6.01.1 Sähköisen kaupankäynnin standardit


Kummallakin teknikalla toteutettu tiedonsiirto tarvitsee määritetyyn kuvausen sisäisyyden tai tiedon rakenteesta. Näitä kuvauksia on EDI -maailmassa tarjolla paljon ja ne ovat useasti yleisiä sekä vakiointeita standardeja. XML -tiedonsiirtoihin on myös tarjolla paljon erilaisia rakennuskuvausia, mutta ongelman ei on uusia, että vakiointeita ja yleisesti käytetävää standardin kuvausia ei ole olemassa läheskaan kaikille tiedoille. Tämä johtuu siitä, että esimerkiksi EDIFACT -standarimekunainen sanoma on aina rakenteeltaan vakioiden ja XML -sanoma puolestaan voi muuttaa siirrettävien tietojen ja tiedostot kuvauksen (XML-schema tai DTD) puitteessa. Tähän vaikuttaa myös se, että XML kuvaus on kehitetty alkuvaheessa ja vakiointeita käytäntöjä ei ole vielä synnytty.

6.01.2 Tiedon esitystavat ja käytettävyys

EDI -standardeissa tiedon muoto on puhtaasti tietokoneen luoma ja sitä voi käyttää ainoastaan toinen tietokone. XML -muotoista tieto voidaan siirtää koneelta koneelle, mutta tieto on myös ihmiselle ymmärrettävä.

XML:n nousemista tulevaisuudessa johtavaksi tiedonsiirtomuodoksi edesauttaa käytöllisyyden muuttuminen selainpohjaisiksi, koska verrattuna EDI-standardeihin se vähentää tiedon muokkaamistarvetta ko- neen ja käyttäjärajapinnan välillä. XML on siis monikäyttöinen tiedos- tomuoto, siinä missä esimerkiksi EDIFACT tiedon esitystapana sovel- tuu pelkästään tiedon siirtämiseen.


6.01.3 Tiedon käsittely


XML -tiedonsiirto puolestaan ei vaadi VAN -tietoverkon käyttöä, kos- ka XML -sanomat voidaan välittää julkista Internetiä tai VPN (Virtual Privat Network) -yhteyttä pitkin. XML:n käytön helppous houkuttaa
yrityksiä rakentamaan järjestelmäisä rajapinnat, jotka eivät vaadi kalliiden muunnnospalvelumien käyttöä.

6.01.4 Yhteenveto XML:n ja EDI -standardien vertailusta

<table>
<thead>
<tr>
<th>XML:n ja EDI -standardien vertailutaulukko (J. Ricker et al 2002):</th>
<th>EDI standardit:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Optimoitu helppoon ohjelmointiin ja luettavuuteen</td>
<td>1. Optimoitu kokoon (XML sanoma on kooltaan 10 kertainen verrattuna EDI sanomaan)</td>
</tr>
<tr>
<td>2. Vaati internetpalvelimen (internet palvelimen hinta voi olla 10 osa EDI -palvelimen hinnasta)</td>
<td>2. Vaati EDI palvelimen</td>
</tr>
<tr>
<td>4. Yksinkertainen ja luettava esitysmuoto</td>
<td>4. Monimuutkaiset ja vaikeasti pitävät esitysmuodot</td>
</tr>
<tr>
<td>5. Vaatii vähemmän vaativaa ohjelmointiosamista</td>
<td>5. Vaatii korkeatasoista ohjelmointiosamista</td>
</tr>
<tr>
<td>7. Esitysnuoto on myös ihmiselle ymmärrettävä</td>
<td>7. Esitysmuodon ymmärtää vain tietokone</td>
</tr>
<tr>
<td>8. Standardit kehitysvaiheessa</td>
<td>8. Vakiintuneet standardit</td>
</tr>
</tbody>
</table>

Kuva 12. XML:n ja EDI -standardien vertailutaulukko. (J. Ricker et al 2002.)


Ominaisuuksensa vuoksi XML -tiedonsiirtomuotona palvelee tällä hetkellä parhaiten web-sovelluksia ja muita reaalialaisesti tapahtuvia tiedonsiirtoja, joissa siirrettävän tiedon määrä ei ole suuri ja joissa tiedon rakenteella on suuri merkitys. Suurten tietomäärien siirtämiseen EDI -standardit ovat kuitenkin tällä hetkellä toimivin tapaa.

6.02 XML/EDI tulevaisuuden ratkaisu?

Määritelmä

Hyödyt

Rajoitukset

EDI ansaitsee omaa kritiikkinsä kalleutensa ja monimutkaisuutensa vuoksi, mutta sen kehittämiseen käytetty 30 vuotta ei ole kokonaan tur- haan heitettyä, koska XML -toimintoja voidaan rakentaa tämän kehi- tyksen jatkeeksi, sen sijaan, että kaikki entinen hylättäisiin ja lähdeket- siin puhtaalta pöydältä. XML hyötyy EDI:stä saaduista kokemuksista ja se onkin perinyt siltä paljon hyviä ominaisuuksia. Kaikki merkit viitta- sivat siihen, että XML/EDI:stä tulee luonnollinen jatke EDI:n aloitta-
malla polulla. Tästä paras viite on se, että lähes kaikki nykyaikaiset
toiminnanohjausjärjestelmät tukevat XML:n käsitteilyä ja toisaalta per-
rustuvat tunnettuihin sähköisen kaupankäynnin standardeihin. (Alan
Cook 1999. XML and EDI Lessons Learned and Baggage to Leave Be-
hind)

EDI/XML kaavio:

Kuva 13. EDI/XML kaavio.

7 XML -toiminnot Solteq Merxissä

Solteq Merx on puhasta merkkipohjainen käyttöliittymä IBM iSeries
suurtietokone ympäristössä, joka on ohjelmoitu RPG IV ja ILE RPG
kielillä. Näissä ohjelmointikielissä ei ole tukea tietyvirtatiedostojen lu-
kemiselle/kirjoittamiselle, koska ne muodostavat integroidun kokonai-
suuden DB2 tietokannan ja OS/400 käyttöjärjestelmän kanssa. Tämä ei
siis ole mikään hyvä lähtökohta XML -toimintojen kehittämiselle,
koska monille XML:n ominaisuuksille, kuten monikäyttöisyydelle, ei
ole käytöä. (Solteq Merx -esite 2003.)

XML -toimintojen tarve on kuitenkin syntynyt siitä, että järjestelmän
halutaan toimivan integroidusti myös nykyaikaisten Web -sovellusten
kanssa. XML -toiminnot tarkoittavat siis tässä tapauksessa XML -
sanoman vastaanottoa ja lähetystä. Sanoman vastaanotossa halutaan
purkaa saapuneen XML -tiedoston tiedot tietokantatiedostoon. Tieto-
kantatiedostosta tiedot voidaan jatko käsittelä järjestelmässä käytettä-
viksi tiedoiksi. Vastaavasti sanoman lähetyksessä halutaan poimia jär-
jestelmän tietokannasta tiedot haluttuun XML -formaattiin ja lähettää
tiedosto vastaanottavaan järjestelmään.

Tulevaisuudessa järjestelmä voidaan kehittää kuitenkin olemassa olev-
illa teknologioidillaakin selainpohjaiseksi, jolloin XML:n ominaisuuksia
voidaan paremmin hyödyntää. Tämä ei kuitenkaan ole olemassa mitään
pääteoksia, joten sitä ei voida pitää perusteena siirtymisessä puhtaasti
XML -tiedonsiirtoihin. Näin ollen XML -toiminnot tulevat olemaan yh-
tenä vaihtoehtona uusille tiedonsiirroille.
7.01 Lähtötilanteen selvitys


XML Toolkitin ohjelmointirajapinnan avulla tehdyissä sovelluksissa on tähän asti käytetty myös kertakäyttöistä toimintamallia, eli jokaista sovellusta kohden on ohjelmoitu uudemmat ohjelmat. XML Toolkitin ohjelmointirajapinta on kuitenkin monimutkainen kokonaisuus, joka vaatii ohjelmoijalta paljon perehtymistä kohteeseen.

(Solteq Merx 2004/2.)


7.02 XML -toimintojen kehitys

Työkalut


(Solteq Merx 2004/2. IBM:n Internet sivut 1994 - 2005.)

Ratkaisu

XML:ää käsittelevän RPG ohjelman rakenteesta.


Rajoituksset


7.03 XML – toimintojen tulevaisuus


Pitemmällä tähtäimellä iSeries maailman ollessa menossa enemmän Java teknologian ja sitä myötä web-käyttöliittymien suuntaan tuli myös jossain vaiheessa harkita XML:n käsittelevän siirtämistä Java-pohjalaiseksi. Tämän myötä saataisiin myös käyttöön niitä ominaisuuksia XML-tiedonsiirroista, joita nykyinen järjestelmän kehitystila ei salli.

8 XML-rajapinta Solteq Merxiin


Rajapinnan toteutusta ei ole tehty omana tuotekehityspjektinaan, vaan se perustuu Solteqissä meneillään oleviin asiakasprojekteihin. Tämä aiheuttaa sen, että kokonaisuudelle ei ole olemassa omaa dokumentoitua ja itsenäistä vaatimusmäärittelyä, vaan vaatimusmäärittelyt ovat syntyneet toiminnallisuuden osalta näistä asiakasprojekteista lisätynä vaatimuksella monikäyttöisyydestä.
8.01 Vaatimusmäärittely

XML -rajapinnan toteutuksen vaatimuksena olivat seuraavat toiminnallisuudet:

- XML:n käsittelyssä käytetään IBM:n XML Toolitin ohjelmointirajapintaa ILE PRG -ohjelmointikielille:
  - Käytettävä versio rajapinnasta on 5.0
- Rajapinnan avulla on pystytettävä muodostamaan XML -tiedosto:
  - XML -tiedosto muodostetaan tiedoista, jotka on poimittu tietokantatiedostoon erillisessä poimintaohjelmassa
  - Tiedot poimitaan XML:n rakenteen mukaiseen järjestykseen
  - Muodostettavan XML -tiedoston rakenteen on noudatettava poimitun aineiston rakennetta
  - Muodostettavan XML -tiedoston rakenne ja sisältö voivat muutua
  - Tarvittaessa muodostettu tiedosto pitää pystyä validoimaan, joko DTD kuvauksen tai XML -scheman avulla ennen lähetystä:
    - Validointia ei kuitenkaan aina haluta tehdä
  - Tiedoston tietojen rakenne tai sisältö ei saa muutua siirretäessä tietokannasta XML -dokumenttiin
  - Muodostettava XML -dokumentti voi sisältää suuriakin tietomääriä

- Rajapinnan avulla on pystytettävä purkamaan XML -tiedosto tietokantaan:
  - XML -tiedoston tiedot puretaan tietokantatiedostoon, jossa niiden rakenne ja järjestys eivät saa muutua
  - Purettavan XML -tiedoston rakenne ja sisältö voivat muuttua
  - Tarvittaessa vastaanotettava XML -tiedosto pitää pystyä validoimaan, joko DTD -kuvaksen ja XML -scheman avulla:
    - Validointia ei ole kuitenkaan pakollista
  - Tiedoston tietojen rakenne tai sisältö ei saa muutua siirretäessä XML -dokumentista tietokantaan.
  - Vastaanotettava tiedosto voi sisältää suuriakin tietomääriä

- Kokonaisuus pitää olla ylläpidettävissä
- Uusien XML -liittymien toteutus pitää olla helppoa
- Uusien XML -liittymien toteutuksissa ei tarvita XML Toolkit osaamista

8.01.1 XML -tiedoston muodostaminen

Vaatimusmäärittely XML -tiedoston muodostamiselle tarkoittavat sitä, että tehtävän sovelluksen on pystyttävä muodostamaan määrämuotoon poimitusta aineistosta halutun mukainen XML -tiedosto. Määrämuotoon poimittu tieto pitää olla sellaisessa muodossa, että sovellus osaa
sen perusteella muodostaa halutun rakenteen. Poimintatiedoista pitää siis ilmetä tietojen järjestys ja hierarkia.

Koska muodostettavan tiedoston rakenne määritetään poimintaohjelmassa tehtävään aineistoon, sovelluksen tulee siis noudattaa tätä rakennetta. Tätä varten on suunniteltava joustava ja toimiva muoto tälle poiminta-aineistolle.

Jos liittymässä kulkevalle XML -tiedostolle on määritelty DTD kuvaus tai XML -schema tulee sovelluksessa olla mahdollista validoida muodostettu tiedosto tätä kuvausta vastaan.

Siirrettävällä tiedolla on merkitystä vastaanottavalalle osapuolelle vain, jos tiedon eheys säilyy siirrossa. Tämä tarkoittaa sitä, että siirrettävien tietojen hierarkia ja merkitty eivät saa muuttua siirrossa.

Suuria tietomääriä käsiteltäessä on sovelluksessa huomioitava tehokkuus ja tiedoston koko. Tavoitteena on siis muodostettavan tiedoston kokoon vaikuttaa lähinnä poistamalla turhat white space ennen lähettystä.

8.01.2 XML -tiedoston purkaminen

Vaatimukset XML -tiedoston purkamiselle tarkoittavat sitä, että vastaanotettu XML -tiedoston on pystyttävä purkamaan määrämuotoiseen muotoon, joka tallennetaan tietokantatiedostoon.

Tiedot tulee purkaa sellaiseen muotoon, että niitä edelleen käsittelevä ohjelma pystyy lukemaan ne ja käyttämään edelleen järjestelmässä. Muodossa tulee siis olla sellainen rakenne, että XML -tiedoston tietojen hierarkia ei muutu.

Vastaanotettava XML -tiedosto pitää pystyä tarvittaessa validoimaan joko DTD kuvaoksen tai XML -schema avulla, ennen tietojen purkaa. Jos siirretty tiedosto ei ole valida, tietoja ei voida purkaa sillä tiedon eheyssä on silloin voinut rikkoutua.

Sovelluksen on pystyttävä vastaanottamaan myös suuria tietomääriä sisältäviä tietoja. Tämän vuoksi on sovelluksessa käytettävä tehokkaita menetelmiä XML:n käsitellyssä.

8.01.3 Kokonaisuuden ylläpidettävyys

8.01.4 Uusien XML-liittymien rakentaminen rajapintasovelluksen avulla

Toteutuksessa on huomioitava se, että tämä rajapintasovellus ei sinälla ole vielä riittävä tekijä uuden XML-liittymän rakentamiselle vaan se tarvitsee lisäksi ohjelmat jotka käsittelevät poimittua tai purettua tie- tooa määrämuotoisesta tiedostosta. Rajapintasovelluksen ja näiden ohjelmien välinen toiminta lopulta muodostaa tietojen siirron järjestelmästä järjestelmään. Määrämuotoinen tietorakenne, joka talletetaan tie- tokantatiedostoon, tulee siis olla sellainen, että sitä käsittelevät ohjelmiin voidaan helposti rakentaa logiikka, jolla tietoja käsitellään. Näissä käsittelevissä ohjelmissa ei siis enää käytetä XML Toolkitin toimintoja, vaan perustuvat tavalliseen ILE RPG ohjelmointiin.

8.02 Tekninen suunnittelu

Toteutuksen tekninen suunnittelu perustuu XML Toolkitin toimintojen käyttöön sekä tietojen poiminnassa ja purussa käytetyn määrämuotoisen tietorakenteen suunnitteluun. Lisäksi teknisessä suunnittelussa on otettu kantaa siihen, minkälaisiin toiminnallisiin moduuleihin kokonaisuus kannattaa jakaa.

8.02.1 Kokonaisuuden jakaminen toiminnallisiin moduuleihin


8.02.2 XML Toolkitin toimintojen käyttäminen

XML Toolkitin toimintojen käyttäminen vaatii seuraavia toimenpiteitä (IBM:n Internet sivut 1994 - 2005.):

- Lisätään lähdekoodiin copymemberiksi tiedosto QXML4PR500 kirjastosta QXMLDEV500, jossa on rajapinnan prototyyppit
- XML -parseriympäristön alustus
Muistinkäsittely ja XML -ympäristön lopetus:
• Ohjelmassa käytetään rajapinnan proseduurien kautta C++ palveluohjelman funktioita ja kaikissa näissä funktioissa ei ole dynaamista muistinkäsittelyä. Tämä aiheuttaa sen, että ohjelmassa on vapautettava käytetyt muistiresurssit asiaankuuluvilla proseduureilla.
• Ennen ohjelman suorituksen lopettamista on rajapinnan XML -ympäristön suoritus lopetettava tietyllä proseduurilla

Poikkeusten käsittely:
• XML Toolkitin ohjelmointirajapinnassa on kahden tyyppisiä poikkeussanomia:
  o Rajapinnan poikkeussanomat
  o Parserin poikkeussanomat
• Rajapinnan poikkeussanomat syntyvät proseduureissa mahdollisesti syntyvistä virheistä. Esimerkiksi, jos yritetään poistaa objekti, jota ei ole olemassa.
• Parserin poikkeussanomat syntyvät, kun parsittava XML -dokumentti ei ole oikeamuotoista tai se ei ole määriteltyyn kuvauksen mukaisesti validia.
• Poikkeussanomat otetaan vastaan tiettyyn, tätä tarkoitusta varten varattuun muistialueeseen. Sanoman paluukoodi määrittelee virheentyyppin.

Merkkijonojen käsittely ja merkistötunnukset:
• XML Toolkitin perusmerkistönä on UTF-8, joka ei tue eurooppalaista ISO-8859-1 merkistöstandardin mukaisia skandinaavisia merkkejä.
• XML Toolkit kuitenkin mahdollistaa merkkimuunnoksen näiden kahden välillä. Muunnoksessa käytetään tiettyä proseduurin, jolle määritetään parametreiksi haluttu merkistö. Parametrin arvona käytetään Job_CSSID vakiota, jolloin String -objektin merkistö tunnukseni tulee työn merkistötunnus
• C++ palveluohjelma käsittelee merkkijonoja String -olioina, jotena merkkijonot on muunnettava tietyllä proseduurilla NULL -pääteiseksi String -olioksi.
• XML -elementtien ja attribuuttien nimet ovat aina UNICODE tekstimuotoisia. Tähän tarkoitukseen käytettävät merkkijonot on muunnettava asiankuuluvalla proseduurilla.

8.02.3 Tietojen lukeminen ja kirjoittaminen tietokantaan
Tässä tietorakenteessa pitää olla ohjelmien toimivuuden kannalta seuraavat tiedot:

- Erännumero, jolla tunnistetaan mitkä tiedot kuuluvat yhteen XML-tiedostoon.
- Järjestys numero, joka järjestää tiedot haluttuun järjestykseen.
- Tasonumero, joka kertoo minkä elementin lapsielementti kyseisen tieto on
- Tiedon omistavan elementin nimi, joka kertoo toisella tavalla minkä elementin lapsielementti on kyseessä
- XML-elementin nimi.
- XML-elementin tyyppi, jolla erotetaan onko kyseessä element-tyyppinen tieto vai onko kyseessä jonkun elementin attribuutti
- Tieto-osa, jossa on elementin arvo

Tämä tietorakenteen tallennetaan tietokantaan, jolloin tiedoston avaimeksi muodostuu:

- Erän numero
- Järjestys numero
- Tasonumero

Tietorakenteesta muodostuu tietokantaan yksipuolinen puurakente, joka noudattelee XML-dokumentin rakennetta.

8.03 Toteutuksen kuvaus

8.03.1 XML-dokumentin muodostus

Ohjelma RCREATEXML muodostaa kohdehakemistoon halutulle nimelle XML-tiedoston, jonka tiedot ja rakenne luetaan poimintatiedostosta.

Ohjelman parametrit (input):
- Lähetyserän numero
- Kohdehakemisto
- Muodostettavan tiedoston nimi

Ohjelman parametrit (output):
- Paluu koodi

Ohjelman käyttämät tietokantatiedostot:
- FXMLDATA (poimintatiedosto):

<table>
<thead>
<tr>
<th>TIE TUE KUV AUS</th>
<th>TIEDOSTO FXMLDATA</th>
<th>TIETUE RXMLDATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIMI</td>
<td>PITUUS/DES TEKSTI</td>
<td></td>
</tr>
<tr>
<td>1 XDNUME</td>
<td>7 P 0 LÄHETYSERÄN NUMERO</td>
<td></td>
</tr>
<tr>
<td>2 XDJNRO</td>
<td>7 P 0 JÄRJESTYSNUMERO</td>
<td></td>
</tr>
</tbody>
</table>
Ohjelman toiminta:

1. Ohjelman alustukset.
2. Luetaan poimintatiedoston ensimmäinen tietue.
4. Luetaan poimintatiedoston seuraava tietue. Tutkitaan onko tietue attribuutti vai elementti.
5. A) Jos tietueen tyyppi on elementti, etsitään tietueen tasonumerolla elementtitaulukosta sen omistava elementti ja lisätään tietue sen lapsielementteihin. Lisätään uusi elementti elementtitaulukoon tasonumeron osoittaman indeksin kohdalle.
   B) Jos tietueen tyyppi on attribuutti, lisätään se edellisen lisätyn elementin attribuutiksi.
7. Tutkitaan onko XML-dokumentissa elementtejä.
8. Jos elementtejä on, tulostetaan ne tietovirtatiedostoon kolmikoksemistoon.

Kuva 15. XML-dokumentin muodostuksen toimintakaavio.

8.03.2 XML-dokumentin purku

Ohjelma RREADXML lukee annetun XML-tiedoston tiedot ja kirjoittaa ne poiminta-aineistoksi. XML-tiedoston elementtien hierarkia säilyy myös poiminta-aineistossa.

Ohjelman parametrit (input):
Purettavan tiedoston hakemisto ja nimi

**Ohjelman parametrit (output):**
- Paluukoodi

**Ohjelman käyttämät tietokantatiedostot:**
- FXMLDATA (poimintatiedosto):

<table>
<thead>
<tr>
<th>TIE TUE KUVA U S</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIE DOSTO</td>
</tr>
<tr>
<td>FXMLDATA</td>
</tr>
<tr>
<td>X D N U M E 7 P 0 LÄHETYSERÄN NUMERO</td>
</tr>
<tr>
<td>X D J N R O 7 P 0 JÄRJESTYS NUMERO</td>
</tr>
<tr>
<td>X D T A S O 2 P 0 TASON NUMERO</td>
</tr>
<tr>
<td>X D T A N I 20 A TASON NIMI</td>
</tr>
<tr>
<td>X D E L E M 50 A XML-ELEMENTTI</td>
</tr>
<tr>
<td>X D D A T A 100 A DATA-O S A</td>
</tr>
<tr>
<td>X D A T T R 10 A DATA-O S A</td>
</tr>
</tbody>
</table>

**Ohjelman toiminta:**

1. Ohjelman alustukset
2. XML -dokumentin parsiminen. Tästä syntyy DOM puutietorakene, jota käytetään tiedoston purkamisessa.
4. Puretaan DOM puurakenne rekursiivisesti solmu kerrallaan. Päättäään solmun tyyppistä suoritettavat toimenpiteet:
   A. Solmun tyyppi on XML -dokumentin juurielementti. Haetaan listaan ne DOM tietorakenteen solmut, jotka sisältävät juurielementin lapsielementit ja kutsutaan proseduuria uudelleen jokaiselle näistä solmuista.
   B. Solmun tyyppi on elementti. Haetaan elementin attribuutit ja otetaan ne arvoinen talteen. Jos solmun ensimmäinen lapsisolmu on tekstityyppinen, eli se sisältää elementin tieto-osan, otetaan muuttujinin talteen myös elementin tiedot. Jos ensimmäinen

Kuva 16. XML -dokumentin purkamisen toimintakaavio.
solmu on elementti, kirjoitetaan elementin ja sen attribuuttien tiedot poiminta-aineistoon.

8.04 XML-liittymän rakentaminen toteutuksen avulla
Kohdassa 8.03.1 ja 8.03.2 esitellyt ohjelmakomponentit mahdollistavat XML-tiedonsiirtoon perustuvien liittymien rakentamisen. Liittymän suunnittelussa ja toteutuksessa näitä komponentteja käytäen ei välttämättä tarvita vankkaa XML-teknologian tai XML-ohjelman tuntemusta, koska XML-muotoisen tiedon käsittelemässä suoritetaan komponenttien sisällä. Tekijöille riittää näin ollen perusteellinen tietämys XML:n rakenteesta, koska heidän on osattava tulkita kohdassa 8.02.3 esitellyn poiminta-aineiston rakennetta.


8.04.1 XML-sanoman lähetys

Ensimmäinen tarvittava komponentti on IBM XML Tookit, jonka suorituksen aikaisia elementtejä tarvitaan kohdassa 8.03.1 esitellyn RCREATXML moduulin suorittamisessa. XML Tookit lisenssineen on tämän vuoksi hankittava myös asiasympäristöihin, joissa XML-toimintoja käytetään.

XML:n käsittelyssä käytetään siis RCREATXML moduulia, joka lukee poimitut tiedot poiminta-aineistosta ja muodostaa niistä XML-dokumentin haluttuun ifs-hakemistoon.

Poiminta-aineiston muodostukseen tarvitaan poiminta-ohjelma, joka lukee tarvittavat tiedot tietokannasta ja kirjoittaa ne halutun XML-tiedoston rakenteen mukaisesti poimintatiedostoon.
Näistä kaikista muodostetaan yhtenäinen ajo kasaamalla ohjelma kokonaisuudeksi. Tarvittaessa tähän ajoon voidaan myös lisätä automaattinen tiedoston lähettäminen halullalla tietoliikenneprotokollalla kohdejärjestelmään.

**Kuva 17. XML -sanoman lähetyys.**

2. Luodaan FXMLDATA tiedostosta ajonäkäinen kopi, jota käytetään poiminta-aineiston säilytyspaikkana. Kopi poistuu koko- naan ajan päättymisestä, joten aineisto ei jää turhaan levylle viemän tilaa.
4. RCREATEXML moduulissa luodaan poiminta-aineistosta uusi XML -tiedosto haluttuun ifs -hakemistoon.
6. Talletetaan muodostettu ja lähetetty tiedosto tallennushakemistoon.

**8.04.2 XML -sanoman vastaanotto**

Merxiin vastaanotettavat tiedot ovat yleensä tuote- ja hinnastotietoja sekä ostolaskuja toimittajilta ja myyntiläyksia asiakkailta. Tietoja vastaanottaessa on tavoitteena saada vastaanotetut tiedot siirrettyä järjestelmään omaan tiedotkantaan, jotta niitä voidaan käyttää päivittäisessä liiketoiminnassa. Seuraavassa selvityksessä kerrotaan, kuinka tällaisen tiedonsiirron voi toteuttaa XML -muotoisen käyttäen kohdassa 8.03.2 esitetyä RREADXML moduulia. Mitä komponentteja tarvitaan ja missä järjestyksessä niitä kutsutaan?

Poiminta-aineistosta tiedot kirjoitetaan Merxin tietokantoihin poimintaohjelmassa, joka lukee puretut tiedot tiedostosta FXMLDATA. FXMLDATA tiedostossa aineistossa on edelleen XML-dokumentin rakenteen mukana, joten tämän avulla voidaan päätellä tietojen hierarkia.

Yleensä saapuvia tietoja ei Merx järjestelmässä kirjoiteta suoraan todellisuuteen tietokantoihin, vaan esimerkiksi myyntitaloussanomana tiedot kerätään vastaanottotiedostoihin, joihin sitten lisätään mm. voimassa olevat hintat ennen kuin niistä muodostetaan varsinaisia järjestelmän myyntitilausia.

![Kuva 18. XML -sanoman vastaanotto.](image)

1. Ohjelman alustuksissa liitetään ohjelmaan XML Toolkit ympäristö ja XML käsittelymoduulit.
5. Poimintaohjelmassa luetaan tiedot poiminta-aineistosta ja kirjoitetaan ne vastaaviin Merx tiedostoihin, jolloin ne ovat järjestelmän...
käytössä jatkokäsittelyä varten. Lisäksi saapunut tiedosto tallennetaan tallennushakemistoon.

9 Lopputulos ja arviointi


9.01 Lopputulos


9.02 Lopputuloksen arviointi

Solteq XML rajapinta toteuttaa sille asetetut vaatimukset ja se on myös myyty ja otettu tuotantokäyttöön muutamalle Merx järjestelmälle käyttävälle asiakkaille sekä myös yhdelle muuta iSeries ratkaisua käyttävälle yritykselle. Näin ollen tutkintotyön toteutuksen osalta voidaan sanoa, että se saavuttaa asetetut tavoitteet niin toiminnallisuuden, kuin järjestelmän arvon lisäämisen kannalta. Lisähyötyä on saatu mahdollisuus tarjota tätä ratkaisua myös muita järjestelmiä käyttävälle asiakkaille.

Tässä on kuitenkin huomioitava se, että tämä ratkaisu on koko ajan edelleen kehittymässä parempaan suuntaan varsinkin tehokkuus mie-
lessä. Uusin XML muodostusohjelman versio mahdollistaa entistä suurempien aineistojen käsittelyn ja luomisen ajoikojen pienennettyä.
Tutkintotyössä luotu pohja ja kerätty kokemukset ovat kuitenkin tärkeässä asemassa jatkokehityksen kannalta.


Lähteet

Organisaatioiden välinen tiedonsiirto – Sähköisen kaupankäynnin seminaari 30.9.1999 [online] [Viitattu 18.4.2005]
http://www.helsinki.fi/~amertani/ovt.htm

Solteq Oyj:n Internet sivut. [online] [viitattu 12.6.2004]
www.solteq.com


Solteq Merx -esite [online] [viitattu 10.6.2004]
www.solteq.com

IBM:n Internet sivut 1994 – 2005 [online] [viitattu 10.11.2004 ja 2.3.2005]
www.ibm.com


Koulutus ja konsultointipalvelu KK Mediat 2000 – 2004 Internet sivut 2004. [online] [viitattu kesäkuussa 2004]
http://www.2kmediat.com/kkmediat/www_2kmediatcom.asp

Helsingin ammattikorkeakoulun Internet sivut 2005. XML –SCHEMA eli skeema. [online] [viitattu 18.4.2005]
http://myy.helia.fi/~vanvu/xmlkurssi/schema.html

http://www.heikniemi.net/kirj/moxml.html

Tuomas Sippalan Internet sivut 2003 – 2005. EDI/OVT. [online] [viitattu 1.9.2004]
http://www.kotinet.com/tuomas.sippala/ovt.htm

Jarna Hakala, EDI - Electronical Data Interchange /OVT – Organisaatioiden välinen tiedonsiirto 3.11.1998. [online] [viitattu 5.11.2004]

Prof. Jari Multisilta 5.11.2001, Porin korkeakouluyksikkö, Mihin XML soveltuu: XML:n hyödyt ja haitat. [online] [viitattu joulukuussa 2004]


Solteq Merx 2004/2 toiminnanohjausjärjestelmä. Solteq Oyj.