

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

AUTOMATIZĒTA XML FAILU
TRANSFORMĀCIJA E-RĒĶINU
NOFORMĒŠANAS SISTĒMĀ

BAKALAURA DARBS

Autors: Andrejs Plisko

Stud. apl. Nr. ap09070

Darba vadītājs: Dr. dat. Ģirts Karnītis

RĪGA 2013

ANOTĀCIJA

Bakalaura darbā pētīti datu apmaiņas un transformācijas sistēmas, kuras tiek izmantotas e-rēķinu noformēšanas nolūkos, tiek izskatītas problēmas, kas ir saistītas ar datu transformāciju un to iespējamie risinājumi, kā arī izvērtētas transformācijas nodrošinošas sistēmas. Ikdienišķā situācijā pārveidošanai ir jānodrošina ieejas datu pārveidošanu par izejošajiem datiem, kuri savukārt būs piemēroti izmantošanai saņēmēja informatīvajā sistēmā.

Atslēgvārdi: datu transformācija, XML, E-invoice

ABSTRACT

Bachelors thesis examines data exchange and transformation systems, which are used for e-invoicing formatting purposes, examine problems which are associated with data transformations and their possible solutions, and also the systems which provide data transformation. In common situations, transformation must provide the conversion of the input data to outgoing data, which in turn will be eligible for receivers' information system.

Keywords: data transformation, XML, E-invoice

SATURA RĀDĪTĀJS

APZĪMĒJUMU SARAKSTS.....	6
1. IEVADS	7
1.1. Kas ir E-rēķini?	9
1.2. Izplatītākie E-rēķinu formāti	9
1.2.1. EDIFACT.....	9
1.2.2. Open Application Group Integration Specification (OAGIS) no OAGi grupas	10
1.2.3. ISDOC.....	10
1.2.4. UBL.....	10
2. RĒĶINU PĀRVALDĪBAS UN PIEGĀDES ĶĒDĒS UZLABOŠANAS PERSPEKTĪVAS UN IESPĒJAS	11
2.1. Rēķinu noformēšanas saistītie procesi un optimizācijas zonas	11
3. DATU TRANSFORMĀCIJAS PROBLĒMAS UN GRŪTĪBAS	17
3.1. Kardinalitāte 1:1	17
3.2. Kardinalitāte 0:1 vai 0:N.....	17
3.3. Kardinalitāte 1:0 vai N:0.....	18
3.4. Kardinalitāte N:1	18
3.5. Kardinalitāte 1:N.....	19
3.6. Citas problēmas saistītas ar datu transformēšanu.....	19
4. DATU APMAIŅAS SISTĒMAS.....	21
Bakalaura darba ietvaros tiks analizētas sistēmas, kas apmierinās sekojošas prasības:	21
4.1. TSIM	23
4.1.1. Sistēmas risinājuma apraksts.....	23
4.1.1.1. Apvienotas XML shēmas izveide.....	24
4.1.1.2. Pirmā kartēšanas piešķiršanas posma.....	25
4.1.1.3. Otrā kartēšanas piešķiršanas posma ar nepieciešamo kardinalitāti	26
4.2. Altova komplekts	29
4.2.1. FlowForce Server	29
4.2.2. MapForce Server	32
Komplekta, kurš sastāv no FlowForce Servera un MapForce Servera, cena atbilstoši izstrādātāja norādītai ir sekojoša:	35
• FlowForce Server – 200 Euro	35

4.3.	DATA INTEGRATION SUITE	36
4.3.1.	Stylus Studio	36
4.3.2.	XQuery procesors.....	38
4.3.3.	XML pārveidotāju komplektu	39
4.4.	Individuālā sistēma.....	41
4.4.1.	Ienākošās informācijas saņemšana	42
4.4.2.	Ienākošo datņu apstrāde	43
4.4.3.	Apstrādāto datņu nosūtīšana.....	45
4.4.4.	Individuālās sistēmas trūkumi	45
4.4.5.	Individuālās sistēmas izmaksas	46
4.5.	Kopsavilkums.....	47
5.	Transformācijas praksē	49
5.1.1.	TSIM transformācija	49
5.1.2.	Altova komplekta transformācija	56
5.1.3.	Data Integration Suite transformācijas	60
	SECINĀJUMI	63
	NOBEIGUMS	64
	IZMANTOTĀ LITERATŪRA UN AVOTI.....	65
	PIELIKUMI	66
1.	pielikums. Ieejas datu formāts.....	66
2.	pielikums. Izejas datu formāts.....	68
3.	pielikums. Ieejas datnes XML shēma, XSD formātā	69
4.	pielikums. Ieejas datnes XML shēma, XSD formātā	71
5.	pielikums. Stylus Studio kartēšana.....	73
6.	pielikums. Stylus Studio kartēšana XQuery kods	74

APZĪMĒJUMU SARAKSTS

EDI – Electronic data interchange (no angļu - “Elektronisku datu apmaiņa”)

CSS – Cascading Style Sheets (no angļu: — „kaskadētas stila lapas”)

XSL:FO – Extensible Stylesheet Language Formatting Objects, XML datu formatēšanas valoda, datu izvadei uz ekrāna, papīra vai citiem plašsaziņas līdzekļiem.

HTML – Hiperteksta iezīmēšanas valoda, jeb HTML (no angļu: Hyper Text Markup Language) ir iezīmēšanas valoda, kas ir izstrādāta tīmekļa lappušu un citas pārlūkprogrammā attēlojamās informācijas glabāšanai.

XSLT – Extensible Stylesheet Language Transformations, XML datu formatēšanas valoda XML dokumentu pārveidošanai citos XML dokumentos, vai citos datos, piemēram, HTML tīmekļa lapām, vai XSL formatēšana objektos, kurus pēc tam var pārvērst PDF, PostScript un PNG.

FTP - Datņu transporta protokols (no angļu: File Transfer Protocol)

1. IEVADS

Bakalaura darbā tiks apskatīta automatizēta XML datņu transformācija E-rēķinu noformēšanas sistēmās, tiks pētītas jau gatavas sistēmas, kuras ļauj to darīt, kā arī problēmas, kas ar ir saistītas ar šiem procesiem. Šāda problēmas izvēle autoram nāca prāta, jo darba vietā ar to ir tieša saskaršanās katru dienu un bija nepieciešams uzzināt, kāds rīks vislabāk būtu piemērots autora darba vietā izvirzītajiem mērķiem, vai pat iespējams izdevīgāk būtu izveidot savu datņu transformēšanas sistēmu. Šāda iespēja arī tiks apskatīta.

Autors izvēlējās šādu darbu tēmu ar nolūku atrast optimālāko variantu, cenas, efektivitātes un ērtības ziņā, pie sistēmu izvēles gan autora darba devējam, gan uzņēmējiem, kuri vēlas ieviest elektronisko rēķinu un datņu noformēšanu savā kompānijā, ar nolūku aizvietot to vai savienot ar papīra rēķinu drukāšanu. Galvenais risināmais jautājums ir: „Kura sistēma ir vislabāk piemērota izvirzīto uzdevumu risināšanai un vai ir izdevīgi veidot savu sistēmu, kura noformēs E-rēķinus XML formātos”. Darba mērķos ietilpst izpētīt piedāvājumu tirgū, izvēlēties dažus optimālākos produktus, kuru starpā būs bezmaksas risinājumi (free-ware), ja būs tādi, kuri apmierina izvirzītās prasības, un aprakstīt savas sistēmas prototipu un dot aptuvenu cenu tās izstrādei un ieviešanai. Darba rezultātā lasītājam jābūt priekšstats par to vai ir vērts izstrādāt šādu sistēmu un vai tas ir izdevīgi. Autora darba vietā ir nepieciešams ieviest tādu sistēmu, kura sekmīgi veiktu nepieciešamos uzdevumus, ar to varētu salīdzinoši ātri iemācīties darboties un lai darbībām ar to varētu norīkot cilvēku, kurš nepārzina programmēšanu labā līmenī.

E-rēķini šobrīd iekaro tirgu zibens ātrumā. Bieži saukti par e-rēķinu vai elektronisko rēķinu, to galīgais mērķis ir samazināt pašreizējās izmaksas priekš rēķinu izsniedzējiem un adresantiem par 60-70%. Papīra rēķini tiek aizstāti ar elektroniskiem rēķiniem. Privātpersonas var apskatīt tos, piemēram, uz tiešsaistes banku portālā un viegli tos apmaksāt ar pogas nospiešanu. Uzņēmumi lejupielādēt rēķinus no norēķinu portāla un spēj automātiski importēt tos savos maksājumu kontos bankās, speciāli tam paredzētos pielikumos, un arhivēt tos elektroniski saskaņā ar likumā noteiktajām prasībām.

Standarti elektroniskajiem rēķiniem atšķiras viens no otra praktiski katrā valstī un liela vienota formāta pagaidām neeksistē. Elektroniskās datu apmaiņas (EDI) standarti, piemēram, Apvienoto Nāciju EDIFACT standarts ietver ziņojumu kodējuma vadlīnijas elektroniskajiem rēķiniem.

Ir aprēķināts, ka globālā tirgus rēķinu aprīte ir ap 150 miljardu vienību. Elektronisko rēķinu daļa joprojām ir samērā maza. ASV tirgus ir līderis „e-rēķins klientam” (B2C) segmentā, taču līderis segmentā „uzņēmuma rēķins citam uzņēmumam” (B2B) un jomā „uzņēmums valstij” ir Latīņamerikas un Eiropas tirgi.

Vairāk nekā 12 miljonu koku var tikt glābti, ja ar elektroniskajiem saziņas kanāliem būtu aizstāti milzīgie Eiropas rēķinu daudzumi, kuri joprojām ir papīra formā.

Tikai 2011. gadā, aptuveni 5 miljoni Eiropas uzņēmumu un 75 miljoni patērētāju ir nosūtījuši vai saņēmuši elektroniskos rēķinus. Viņi, iespējams apmainījās ar vairāk nekā 3000000000 e-rēķiniem.

Tabulā 1.1 var redzēt cik daudz dalībnieku piedalās e-rēķinu savstarpējā apmaiņā un kā tas pieaug ar gadiem.

1.1 tabula

Elektronisko rēķinu izmantotāju pārskats [5]

Dalībnieku skaits	2008	2009	2010	2011	2012
Patērētāji	32,000,000	43,000,000	57,000,000	75,000,000	100,000,000
Korporatīvās iestādes un organizācijas	100,000,000	2,500,000	3,600,000	4,900,000	6,700,000

Savukārt tabulā 1.2 var redzēt kā pieauga e-rēķinu skaits gadu gaitā.

1.2 tabula

Elektronisko rēķinu daudzuma pārskats [5]

E-rēķinu daudzums (miljonos)	2008	2009	2010	2012	2013
Kopējais B2C, B2B, B2G & G2B	1,200	1,700	2,300	3,100	4,200

1.1.Kas ir E-rēķini?

Elektroniski rēķini vai elektronisko rēķinu apmaksa un attēlošana ir process, kad kompānija attēlo (nosūta) savus rēķinus klientiem elektroniski (parasti uz e-pastu vai e-banku) un klienti savukārt apmaksā šos rēķinu arī elektroniski caur globālo tīmekli.

Ir divi pamat veidi kuros iedala e-maksājumus:

- Tiešais veids, tas ir, maksājumi tiek veikti tieši no preces vai pakalpojuma sniedzēja mājaslapas, piemēram, kad pērkam biļetes uz kino „Forum Cinemas”.
- Caur banku, tas ir, maksājums tiek veikts caur banku saņemot elektronisko rēķinu, un banka savukārt pēc noteiktu procedūru veikšanas pārskaitīs klienta līdzekļus uz norādītās kompānijas rēķina.

1.2.Izplatītākie E-rēķinu formāti

Eiropā, kā arī visā pasaulē pastāv ļoti daudz elektronisko rēķinu formāti, kuri pagaidām nav vienoti vienā kopīgā formātā. Tālāk tiks aprakstīti tikai daži no populārākajiem formātiem Eiropas Savienībā

1.2.1. EDIFACT

Apvienoto Nāciju standarts elektroniskajiem rēķiniem ("INVOIC") ietver standarta kodus galvenes informācijas radīšanai (bieži uz visu rēķinu) un kodus, lai nosūtītu informāciju par katru no līnijas pozīcijām (preču vai pakalpojumu). "INVOIC" standartu var izmantot arī, lai pārraidītu kredīta un debeta piezīmes. [1] "IFTMCS" standarts tiek izmantots, lai pārraidītu kravas rēķinus. [2]

Eiropas Savienības tiesību akts tika pieņemts 2010.gadā kā Direktīva 2010/45/ES, lai veicinātu Elektronisko rēķinu izaugsmi visās Eiropas Savienības dalībvalstīs. Šī likumdošana spēj apmierināt dažādus PVN un starpvalstu rēķinu prasības Eiropas savienībā, papildus likumdošanai par rēķinu autentiskumu un rēķinu integritāti, kas tiek nosūtīti elektroniski. Tika aprēķināts, ka 2011. gadā vien aptuveni 5 miljoni Eiropas Savienības uzņēmumi nosūta elektroniskos rēķinus [3].

1.2.2. Open Application Group Integration Specification (OAGIS) no OAGi grupas

XML ziņojumu formāts elektroniskajiem rēķiniem tika lietots kopš XML pirmsākumiem 1998.gadā. Open Application Group Integration Specification (OAGIS) iekļauj rēķinus kopš 2001. OAG (Open Application Group) ir darba attiecībās ar ANO / CEFACT, kur OAG un tās locekļi piedalās daudzās Tehnoloģiju un Metodoloģiju specififikāciju definēšanās. OAG ietver arī atbalstu šīm Tehnoloģiju un Metodoloģiju specififikāciju OAGIS ietvaros.

1.2.3. ISDOC

ISDOC ir standarts, kas tika izstrādāts Čehijas Republikā kā universāls formāts elektroniskajiem rēķiniem. 2008 . gada 16. oktobrī, 14 uzņēmumi un Čehijas valdība parakstīja deklarāciju, lai izmantotu šo formātu gada laikā savos produktos.

1.2.4. UBL

XML ziņojumu formāts elektroniskajiem rēķiniem tika izmantota pēdējos gados ļoti plaši. Ir divi standarti, kas pašlaik tiek attīstīti. Viens no tiem ir starpnozaru rēķins, ko izstrādā Apvienoto Nāciju standartizācijas organizācija ANO / CEFACT un otrs ir UBL (Universal Business Language), kuru izsniedz OASIS (Organization for the Advancement of Structured Information Standards). Ieviešanas rēķiniem, kuru pamatā ir balstīti uz UBL, ir kopīgas, visbūtiskāk tas notika ar rēķiniem publiskajā sektorā Dānijā, tā kā tā bija pirmā valsts, kur UBL tika ieviests kā obligātais formāts saskaņā ar likumu par visiem rēķiniem publiskajā sektorā. Patlaban turpmākas ieviešanas notiek Skandināvijas valstīs kā projekta NES (North European Subset) rezultātā. Ieviešana šobrīd notiek arī Itālijā, Spānijā, Nīderlandē (UBL 2.0) [4] un arī pašā Eiropas Komisijā.

2. RĒĶINU PĀRVALDĪBAS UN PIEGĀDES ĶĒDĒS UZLABOŠANAS PERSPEKTĪVAS UN IESPĒJAS

Daudzus gadus, lielas iepirkšanas organizācijas (piemēram, mazumtirdzniecības, automobiļu un veselības aprūpes nozares) aktīvi centās mudināt savus piegādātājus pāriet uz elektroniskās informācijas apmaiņu piegādes ķēdēm. Ļoti nesen uzmanība tika pievērsta tikai lielajiem piegādātājiem. Tie ražo vislielāko apjomu produkcijas un tie bija vienīgie, kas spēja integrēt savu elektronisko vidi atbilstošā veidā un par saprātīgu cenu. Jau vairākus gadus, situācija tiek uzlabota arī mazākiem uzņēmumiem:


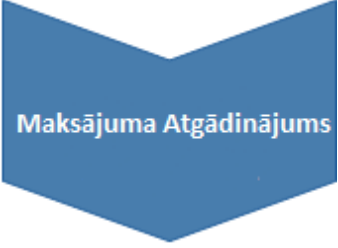
- Kopš 2004 E-rēķini ir juridiski atzīti ES dalībvalstīs un vairumā citu valstu
- Plašs klāsts risinājumu un pakalpojumu ļauj arī mazākiem uzņēmumiem apstrādātu rēķinus elektroniski
- E-rēķins ir kļuvis par augstas attīstības liecinājumu daudzās nozarēs un valstīs

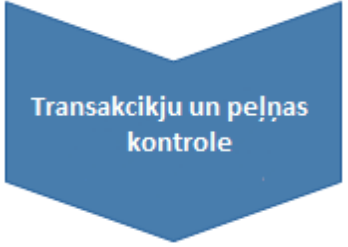

Īss autora pētījums e-rēķinu pasūtītāju starpā ir parādījis, ka katram no viņiem ir daži vai daudzi tirdzniecības partneriem, kuri jau tagad nosūta vai saņem elektronisku rēķinu papīra rēķina vietā. Agrāk mazāki klienti izturējās pret to kā "patīkami, ja būtu" variantu, taču lielākiem uzņēmumiem tas tika aprakstīts kā obligāta prasība nākotnes biznesa attiecībām. Sakarā ar paaugstinātu izmaksu un konkurences spiedienu tuvākajos gados, "patīkami, ja būtu" iespēja vairumā gadījumu kļūs par obligātu prasību ar īsu reakcijas un īstenošanas termiņu.

2.1. Rēķinu noformēšanas saistītie procesi un optimizācijas zonas



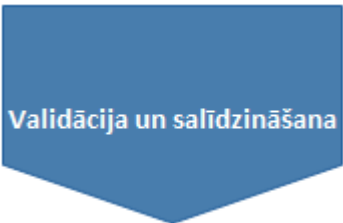
Sekojošās tabulās 2.1.1 un 2.1.2, tiks parādīts kādos procesos tiek iedalīta rēķinu piegādes ķēde, gan papīra rēķinu piegādes ķēde, gan e-rēķinu piegādes ķēde, kā arī tiks aprakstīti e-rēķinu noformēšanas sistēmu pārākums pār papīra rēķinu noformēšanas sistēmām.

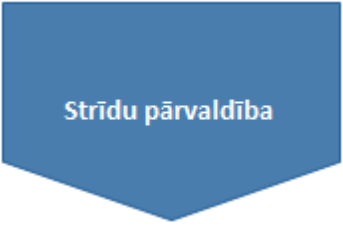
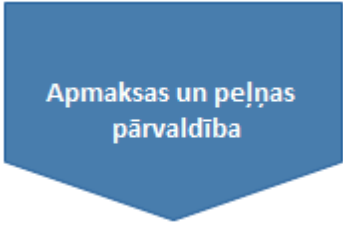
Procesi un optimizācijas zonas rēķinu izsūtītājiem [5]


Sūtītāja process	Fizisks darbs un problēmas ar papīra bāzētiem procesiem	Optimizācija ar E-rēķiniem un automatizētiem procesi
 <p>Printēšana, Likšana aploksnē, Sūtīšana</p>	<ul style="list-style-type: none"> • Augstas izmaksas • Papīram ir negatīva ietekme uz piesārņojumu • Ilgs piegādes laiks • Nav kontroles par to, vai klienti ir saņēmuši rēķinus • Klients var noraidīt Rēķinus nedēļas vēlāk, ja atslēgdati ir pazuduši no tās 	<ul style="list-style-type: none"> • Elektroniskais rēķins tiek sūtīts droši caur tīklu • Veicina uzlabošanos līdz 0,8% pēc Kioto protokola prasībām • Reālā laika piegāde ar saņemšanas / lejuplādes apstiprinājumu • Atslēgdatu validācija uzreiz pēc aizsūtīšanas
 <p>Maksājuma Atgādinājums</p>	<ul style="list-style-type: none"> • 10-15% no rēķina sastāva prasa maksājuma atgādinājumu jo saņēmējiem bieži vien ir laiktietilpīgi darba procesi un papīra rēķini apmaksājumu sistēmas 	<ul style="list-style-type: none"> • Ir samazināts, jo daudziem no klientiem elektroniskie rēķini tiek apstrādāti automātiski (ar noteiktām summām, kas ir zemākas par noteikto un kuras atbilst rīkojumam)

 <p>Transakciju un peļņas kontrole</p>	<ul style="list-style-type: none"> • Laika un līdzekļu ietilpīgi manuāli procesi • Datu kvalitātes problēmas 	<ul style="list-style-type: none"> • Automātiska naudas maksājumu nodrošināšana • Sakarā ar daudz ātrāku elektronisko atskaišu par maksājumiem veidošanu, atbildīgajiem par peļņas kontroli ir pilnīga kontrole pār visiem rēķiniem, ļaujot viņiem optimizēt peļņas vadību
 <p>Arhivēšana</p>	<ul style="list-style-type: none"> • Simtiem vai tūkstošiem mapju ar papīra rēķiniem, kuriem nepieciešamas lielas uzglabāšanas telpas • Augstas izmaksas manuālo meklēšanu 	<ul style="list-style-type: none"> • Automatizētā arhivēšana • Viegli atrodams rēķins pēc dažiem atslēgvārdiem • Ātra piekļuve pie elektroniska arhīva decentralizētā vidē • Momentāna pārbaudāmība ekrānā ar nepieredzētu integritāti un autentiskuma garantiju • Miljoniem rēķinu nepieciešama vieta tikai cietajā diskā

Procesi un optimizācijas zonas rēķinu saņēmējiem [5]

Saņēmēja process	Fizisks darbs un problēmas ar papīra bāzētiem procesiem	Optimizācija ar E-rēķiniem un automatizētiem procesi
 <p>Saņemšana</p>	<ul style="list-style-type: none"> • Pasta atvēršana • Nevēlamu pielikumi pārbaude un dzēšana • Ieejas zīmogs 	<ul style="list-style-type: none"> • Pilnīgi automatizēts
 <p>Kodifikācijas ievadīšana</p>	<ul style="list-style-type: none"> • 10% no ievadītajiem datiem pēc statistiskas ir apskatīti ar kļūdām • Aizkavēta ienākšana, periodā ar palielinātu rēķinu daudzumu • Alternatīva skenēšana atrisina tikai nelielu daļu no problēmām 	<ul style="list-style-type: none"> • Reālā laika imports, neatkarīgi no apjoma • 100% pareizi dati
 <p>Validācija un salīdzināšana</p>	<ul style="list-style-type: none"> • Nesakrītības PVN atbilstību bieži tiek konstatētas (pārāk) vēlos posmos • Līniju vienības rēķinos gluži bieži neatbilst ar rīkojumu vai līgumu noteikumiem. Manuāla saskaņošana ir ļoti 	<ul style="list-style-type: none"> • PVN sakrītība un citu galveno datu validācija var tikt izdarīta automātiski, kad saņēmējs augšupielādē E-rēķins • Līniju vienību saskaņošanas ar rīkojumu un līguma

	laikietilpīga un dārga	noteikumiem ir pilnībā automatizēta
 <p>Strīdu pārvaldība</p>	<ul style="list-style-type: none"> • Strīdu izšķiršana ar piegādātāju bieži tiek darīta īstajā brīdī pa tālruni, nestrukturētas sarakstes pa e-pastu veidā vai pa faksu • Strīdu risināšana var būt ļoti laikietilpīga 	<ul style="list-style-type: none"> • Uzlabota strīdu apstrāde un izvairīšanās no tiem • Daudzi risinājumi vai pakalpojumi var ļaut automatizētu, strukturētu un reāla laika apmaiņu ar strīda informāciju starp pircējiem un piegādātājiem
 <p>Apmaksas un peļņas pārvaldība</p>	<ul style="list-style-type: none"> • Laikietilpīgs un dārgs darba cikls uzņēmumiem, lai samaksātu • Manuāls darbs par samaksas kārtību, kas palielina kļūdu risku • Atbildīgajam par peļņas pārvaldību nav iespējas pilnīgi caurskatīt visus rēķinus, kas ir savākušies (stāv rindā) 	<ul style="list-style-type: none"> • Dati rēķinā, kas atbild par maksājumu tiek apstrādāti tieši un automātiski un tiek iekļauti apmaksas rīkojumā • Katrs ārzemju rēķins parādās uz ekrāna atbildīgajam par peļņu nekavējoties pēc tāda rēķina saņemšanas un sniedz viņam optimizētu Naudas pārvaldību (piedāvājot atlaides par maksājumiem noteiktā laikā, kopīgo darba optimizācija) • Darba cikla ietvaros uzņēmumā apmaksas ir automatizēta vai vismaz atbalsta elektronisko darbplūsmu

 <p style="text-align: center;">Arhivēšana</p>	<ul style="list-style-type: none"> • Simtiem vai tūkstošiem mapju ar papīra rēķiniem, kuriem nepieciešamas lielas uzglabāšanas telpas • Augstas izmaksas manuālo meklēšanu • Tradicionāli 6 viena rēķina kopijas atkarībā no nozares, kur ne katra ir skaidri norādīts kā "kopija" 	<ul style="list-style-type: none"> • Automatizētā arhivēšana • Viegli atrodams rēķins pēc dažiem atslēgvārdiem • Ātra piekļuve pie elektroniska arhīva decentralizētā vidē • Momentāna pārbaudāmība ekrānā ar nepieredzētu integritāti un autentiskuma garantiju • Miljoniem rēķinu nepieciešama vieta tikai cietajā diskā
---	---	---

Var redzēt, ka tabulā 2.1.1 un tabulā 2.1.2 kolonnā Sūtītāja procesi un Saņēmēja procesi atkārtojas pēdējais process, kas saucās „Arhivēšana”. Ar šo procesu autors gribēja parādīt, ka tas notiek abās pusēs, gan pie rēķina sūtītāja, gan pie rēķina saņēmēja, un principiāli process ne ar ko neatšķiras.

Autors savā bakalaura darbā izpētīs dažādas sistēmas, kas ļauj transformēt e-rēķinus XML formātos, kā izejošas datnes, tā ieejošās datnes rēķinu noformēšanas sistēmās, kā arī izpētīs vai ir izdevīgi ieviest savu noformēšanas e-rēķinu transformēšanas sistēmu, tas ir izveidot to un pēc tam ieviest ražošanā.

3. DATU TRANSFORMĀCIJAS PROBLĒMAS UN GRŪTĪBAS

Datu transformācijas grūtības ir parasti saistītas ar kardinalitāšu attiecībām, vai tā ir 1 pret 1 vai 1 pret Daudz. Neskatoties uz to, ka XML datņu formāti ir bāzēti uz elementiem, arī šeit ir iespēja sastapties ar virkni problēmu [10]. Sekojošā nodaļā tiks īsi raksturotas problēmas un to iespējamie risinājumi.

3.1.Kardinalitāte 1:1

Ir gadījumi, kad izvada vai ievada dokumenta kartēšanā ir vairāki elementi ar vienu un to pašu nosaukumu, vai tie ir elementi, kas pēc noteikta likuma dublējas pēc nosaukuma un tos ir nepieciešams apstrādāt vienu pēc otra un tāpat novietot izvada datnē mērķa elementos [10]. Iespējamais risinājums varētu būt specializētu algoritmu izveide vai izmantošana, kuri pielieto apstaigāšanu un datu pārvietošanu izvada datnes atbilstošajos elementos.

3.2.Kardinalitāte 0:1 vai 0:N

Šī ir datu transformācija, kas paredz vai nu jaunas informācijas radīšanu vai nu kādas noteiktas informācijas ievietošanu izvada datnē [10]. Šādi gadījumi var rasties, kad izvada dokumentā ir nepieciešamība ievietot datus, kuri nav minēti ieejas datnē. Iespējamie risinājumi varētu būt sekojoši:

- Datu kuri trūkst, tiek iegūti no ārējiem datu avotiem

Dati kuri neatrodas ieejas datnē varētu būt dabūti no kādām iekodētām vērtībām, vai atrastas no datu bāzes pēc speciāli ieviestiem noteikumiem.

- Dati var tikt iegūti no līdzīgiem elementiem

Ir gadījumi, kad tiek noformēta datne, kur runa iet par personām un to dzīves vietu. Ļoti bieži pēc speciālas vienošanās ar klientu dati par dzīves vietu varētu neatkārtoties pēc katras nākamās personas vārda un uzvārda dokumentā, kas nozīmētu, ka ir nepieciešams izmantot pēdējo izmantoto adresi, un ka katra nākamā persona dzīvo pēc tādas pašas adreses

- Dati var veidoties kā atvasinājums no citiem datiem vai tikt ieviesti jauni iekodēti mainīgie

Ja klientam ir nepieciešams atvasināt kādus datus no jau esošiem, to ir iespējams darīt un ievietot tos nepieciešamajos elementos, kā arī var iekodēt noteiktas vērtības kuras tiks ievietotas nepieciešamajos elementos.

- Dati var tikt summēti

Izmantojot datu transformācijas var tikt summētas pat neskaitliskas vērtības. Taču ir jāpārziņa visi noteikumi, kuri ir saistīti ar izejas datnes formātu un elementu attiecību vienam ar otru.

3.3.Kardinalitāte 1:0 vai N:0

Ir gadījumi, kad informācija ienākošajā datnē vienkārši nav nepieciešama vai to var neizmantot, jo izejas datnē tā netiek atspoguļota [10]. Risinājums šādam gadījumam ir ļoti vienkāršs: kartēšana neatspoguļo transformāciju noteiktajiem XML elementiem.

3.4.Kardinalitāte N:1

Dažos gadījumos ir nepieciešams konkatenēt vai sapludināt datus no ieejošās datnes vairākiem elementiem vienā izejošās datnes elementā [10]. Tādi gadījumi varētu rasties, kad ir nepieciešams sapludināt, piemēram, personas vārdu un uzvārdu ar atstarpi vai citu atdalītāju starp tiem. Iespējamais risinājums ir sistēmas atbalsts datu konkatenācijai vai sapludināšanai. Taču problēma ar šo kardinalitāšu apstrādi ir, ka var rasties gadījumi, kad sapludinātos datus vairs nevar sadalīt atpakaļ.

3.5.Kardinalitāte 1:N

Ir gadījumi ,kad ieejošās datnes elements satur kombinētu vai sapludinātu informāciju un šī informācija ir jāsadala un jāievieto vairākos elementos izejošajā datnē [10]. Risinājums šādai problēmai ir kartēšanas iestatīšana, kad lietotne apstrādā noteiktos XML datnes elementus un sadala tos atbilstoši noteikumiem. Taču var rasties problēma, kad šādus datus ir neiespējams sadalīt, šī problēma bija minēta augstāk. Dati varētu tikt apstrādāti kļūdaini un izejas datnes elementi var saturēt null vērtības, kuras lietotne arī var atzīt var derīgiem datiem. Izeja šajā gadījumā būtu, transformācijas lietotnes iestatīšanas tikai uz valīdo elementu apstrādi, bet šajā gadījumā ar klientu jābūt stingrai vienošanai par sūtāmo datu formātu.

3.6.Citas problēmas saistītas ar datu transformēšanu

- Ieejas datu un izejas datu formātu un tipu konflikti.

Piemēram, ieejas datos produkta mērvienība ir simbolu virkne „gabals”, bet izejas datos tam varētu atbilst mērvienības kods no klasifikatora - skaitlis 14 [10]. Šī problēma varētu būt vairāk raksturīga datu tipu transformācijas gadījumos, bet tas nav raksturīgi gadījumos, kad ir nepieciešams tikai paņemt ienākošos datus no noteikta elementa un novietot tos izejošo datu elementā vai elementos.

- Mērvienību vai izmantoto vienību konflikti.

Konflikts var rasties gadījumos, kad ieejas datos vienība nav tieši norādīta un tā tiek interpretēta un pievienota datu pārveidošanas brīdī. Tas varētu attiekties uz pārpratumiem par izmantotās naudas vienībām, garuma un svara mērvienībām starp valstīm, kuras izmanto SI sistēmu, un Angliju un ASV, u.c. [11]

- Datu neprecizitāte.

Var rasties gadījumi, kad noteikta skaitlisku datu precizitāte ir pārāk maza pie datu transformācijas, šādi gadījumi varētu rasties, piemēram, transformējot valūtas daudzumu priekš valstīm ar hiperinflāciju. Šai konkrētajā gadījumā datu tips Integer varētu kļūt pārāk mazs, lai glabātu tik lielu skaitli [9].

4. DATU APMAIŅAS SISTĒMAS

Tālāk darbā tiks aprakstītas dažas sistēmas un risinājumi, kuras nodrošina nepieciešamos procesus, kuru laikā tiek noformēts rēķins vai rēķini. Lai atrastu izdevīgāko variantu autora darba vietā, kura nodarbojas ar datu transformēšanu, ir ieviesti daži kritēriji pēc kuriem tiek analizētas sistēmas, ar mērķi atrast izdevīgāko variantu cenas ziņā un funkcionalitātes ziņā.

Bakalaura darba ietvaros tiks analizētas sistēmas, kas apmierinās sekojošas prasības:

- **Datņu automātiska saņemšana** (apstrāde sākas automātiski) - sistēmai ir stingri nepieciešams automatizēts process, kas nodrošina datņu vai datņu paku saņemšanu ar nolūku automātiski sākt apstrādāt to atbilstoši iestatījumiem. Svarīgi ir arī spēja iestatīt viedu kādā tiks saņemtas datnes, piemēram saņemot tos e-pasta veidā vai saņemot tos kādā nošifrētā FTP adresē;
- **Datņu automātiska transformācija** - sistēmai, pēc datņu saņemšanas, nekavējoties jāsāk tās transformēt un noformēt atbilstošā veidā, šī prasība ir viena no galvenajām sistēmai;
- **Atskaišu veidošana** - atskaišu veidošana ir gana svarīgs process, kam arī jānotiek automātiski pēc noteiktiem iestatījumiem, kā arī ir jābūt iespējai šīs atskaites apskatīt ērtā veidā un glabāt tās kādu laiku. Ar šī procesa palīdzību vismaz ir nepieciešams jābūt iespējai apskatīt kādas tieši datnes tika apstrādātas, kādi izvaddati tika nosūtīti un kādā laika momentā tas notika.
- **Automātiska arhivēšana** - pēc pieredzes autora darba vietā var droši teikt, ka ir ļoti maza iespēja, ka atskaites, ieejas dati un izvaddati, kuri ir vecāki par kādu noteiktu laika periodu, būs nepieciešami tuvākajā laika momentā. Tāpēc ir nepieciešams veidot speciālus arhīvus vai citas datu glabātuves priekš vecākiem datiem, ar nolūku saglabāt vairāk darba atmiņas, kura izmantojas pastāvīgi (palielināt sistēmas lietderību). Prasība nav obligāta.

Procesi, kas tiks analizēti sistēmās, ir izvēlēti balstoties uz darba pieredzi e-rēķinu apstrādes sfērā. Neskatoties uz to, ka dažādas sistēmas piedāvā lielāku iespēju klāstu, tika izvēlēti procesi, kuriem jābūt minimuma komplektācijā dotā tipa sistēmām, lai sekmīgi izpildītu savus darba uzdevumus:

- Saņemšanas veida izvēle

- Kartēšanas datnes veidošana un ievietošana sistēmā
- Datņu transformācijas noteikumu ieviešana
- Datņu nosūtīšanas veida izvēle
- Atskaišu apskatīšana
- Arhīvu apskatīšana

Pēc programmētāju aptaujas, autora darba vietā, autoram radās priekšstats par programmētāju parametriem, pēc kuriem viņi vērtē e-rēķinu noformēšanas sistēmas :

Rīku ērtība – ir svarīgi lai programmētājam vai personai, kura lietos doto sistēmu vai tās sastāvdaļu būtu ērti lietot tās rīkus, veikt nepieciešamās operācijas, kuras nodrošinās sekmīgu sistēmas darbu. Zemas ērtības gadījumā parasti rodas apstākļi, kad darbs vai nu nekustās pavisam vai aizkavējas tikai tāpēc ka nevar pietiekami ērti veikt noteiktas darbības.

Uzturamība – kartēšanas datus laiku pa laikam var būt nepieciešams mainīt vai pievienot tiem jaunu informāciju. Šis parametrs nosaka cik to ir vienkārši darīt, ja, piemēram būs nepieciešams pāriet uz jaunu sistēmas versiju.

Apgūšanas sarežģītība – parametrs raksturo cik vienkārši ir apgūt sistēmu un atcerēties ka ar to ir jādarbojas. Vai nepieciešamo zināšanu un iemaņu iegūšanai ir nepieciešama dokumentācijas caurskatīšana un uzmanīga lasīšana, vai tomēr ir nepieciešams algot speciālistus, kuri ir strādājuši ar doto sistēmu un tās rīkiem, vai ir tos veidojuši, un tie savukārt pasniegs nepieciešamo materiālu atbilstošā veidā lai tas būtu saprotams nākamajiem lietotājiem. Parametrs tiks veidots pēc lietotāju novērtējuma skalā no 1 līdz 10.

Funkcionalitāte – ir būtiski lai sistēma ļautu veikt nepieciešamās manipulācijas ar elementu vērtībām, nodrošina izejas datu sagatavošanu vēlamajā veidā. Iespējamās, ka risinājumam ir papildus iespējas, piemēram, ieejas un izejas datu shēmu veidošana un labošana, shēmu lejupielāde un imports, datņu validācija par atbilstību shēmām, u.c.

Izmaksas – ir galvenais no parametriem, pēc kuriem tiek vērtētas sistēmas, gadījumā, kad sistēmas cenu nevarēs noteikt kāda noteikta iemesla dēļ, tās cena tiks noteikta aptuveni pēc sistēmas iespējām, ierobežojumiem un sarežģītības.

4.1. TSIM

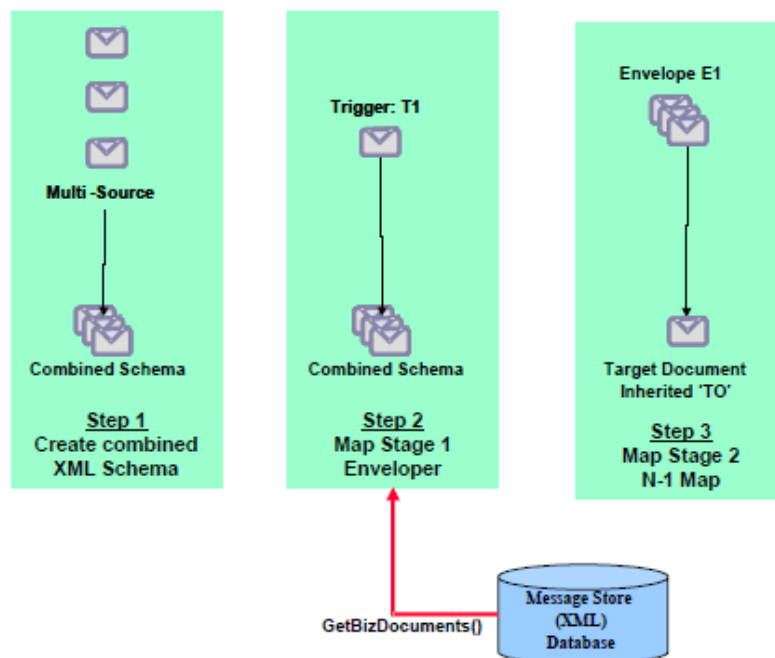
Axway kompānijas izstrādātais produkts –TradeSync Integration Manager (turpmāk tekstā TSIM), ar kuras jaunāko versiju autoram bija pieredze darbā, nodrošina stabilu darbu un atbilstību darba uzdevumiem un funkcionalitātei TSIM un tās komponentes kalpo kā centrmezgls integrācijas biznesa procesiem.

TSIM ietver sevī grafiskos rīkus, kas ļauj grafiski un koda veidā veidot atkarības starp ieejas un izejas datnēm ar kartēšanas palīdzību, noteikt izmaiņas katram ieejas vai izejas kartēšanai, ja tāda vajadzība radīsies, datņu plūsmas automatizētu saņemšanu un sūtīšanu, šīs datu plūsmas koriģēšanu visās plūsmas stadijās un atskaišu veidošanu. Visas izmaiņas un procesus, kas ir saistīti ar kartēšanas datiem, ir iespējams definēt SQL valodā vai grafiskajā vidē.

4.1.1. Sistēmas risinājuma apraksts

TSIM sistēma iekļauj sevī kartēšanas glabāšanu datu bāzē, grafiskos rīkus lai veidotu kartēšanas un saglabāt tās datu bāzē, rīkus kā noteikt datņu saņemšanas veidu un rīkus kā noteikt datņu sūtīšanas veidu, kā arī glabātuvi ar datnēm, kuras tika apstrādāti kā ieejas vai izejas 6 mēnešu laikā, un atskaites par procesiem, kuri tika iestatīti kā procesi, kurus vajag dokumentēt. Datnes, kas ir vecākas par 6 mēnešiem glabājas arhīvā, nodalītā repozitorijā, process priekš pieejas pie šīm atskaitēm aizņem vairāk laika nekā atskaišu apskatīšana, kuras ir jaunākas par 6 mēnešiem.

Funkcionālais risinājums ir sadalīta trīs funkcionālās darbībās, kas apkopotas attēlā zemāk (sk. 4.1.1.1. att.). Tas sastāv no „Apvienotas XML shēmas izveides”, „Pirmā kartēšanas piešķiršanas posma”, „Otrā kartēšanas piešķiršanas posma ar nepieciešamo kardinalitāti”.



4.1.1.1.att.Sistēmas risinājuma shēma

4.1.1.1. Apvienotas XML shēmas izveide

Lietotājam ir iespēja apvienot vairākas XML dokumentu definīcijas vienā XML formulējumā, vai izveido pilnīgi jaunu vienu formulējumu. Priekšnoteikums ir, ka visi ievades dokumenti ir izveidoti un formulēti, izmantojot W3C XML shēmu. Trešās puses XML redaktors rīks, piemēram, XML-Spy, var tikt izmantots, lai pārbaudītu izveidoto XML shēmu.

XML piemērs:

```
<?xml version="1.0" ?>

<schemaxmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ppl="http://axway.com/ns/people" xmlns:addr="http://axway.com/ns/address"
xmlns:vehicle="http://axway.com/ns/vehicle" targetNamespace="http://axway.com/ns/E1-
envelope">

<importnamespace="http://axway.com/ns/people"
schemaLocation="EnvelopeExample_people.xsd" />
```

```

<importnamespace="http://axway.com/ns/address" schemaLocation="
EnvelopeExample_address.xsd" />

<importnamespace="http://axway.com/ns/vehicle" schemaLocation="
EnvelopeExample_vehicle.xsd" />

<element name="E1">

<complexType>

<sequence>

<elementref="ppl:person" minOccurs="1" maxOccurs="1" />

<elementref="addr:address" minOccurs="1" maxOccurs="unbounded" />

<elementref="vehicle:car" minOccurs="0" maxOccurs="unbounded" />

<elementref="vehicle:moped" minOccurs="0" maxOccurs="unbounded" />

</sequence>

</complexType>

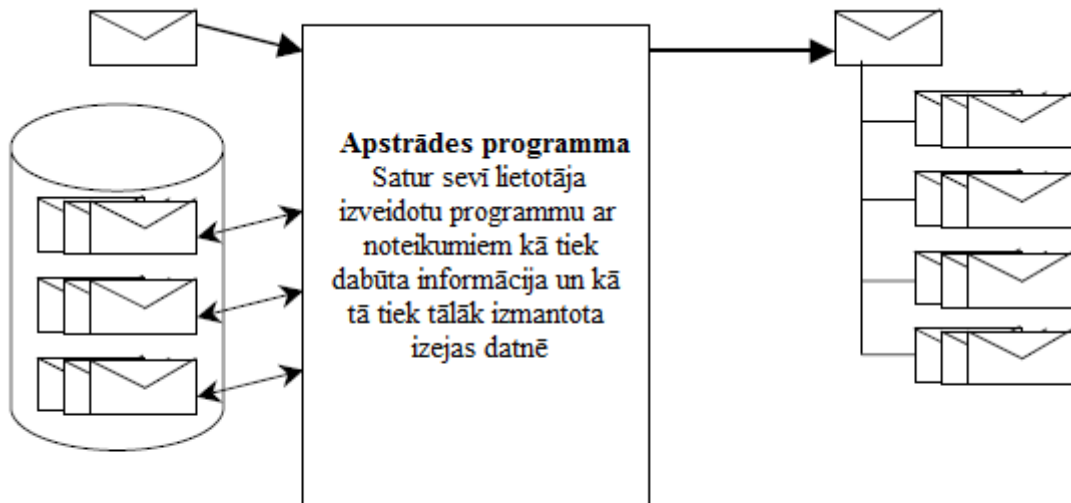
</element>

</schema>

```

4.1.1.2. Pirmā kartēšanas piešķiršanas posma

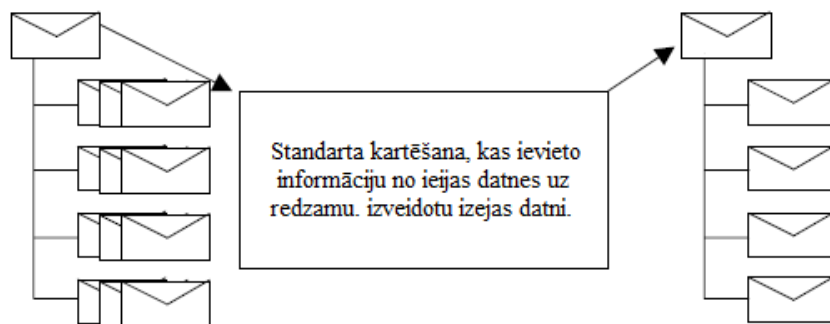
Šajā posmā lietotājs var noteikt no kurām ieejas datnes/datņu vietām ir nepieciešams ņemt informāciju, un kurās vietās šo informāciju novietot izejas datnē/datnēs.



4.1.1.2.1. att. Pirmā kartēšanas posma darba shēma

4.1.1.3. Otrā kartēšanas piešķiršanas posma ar nepieciešamo kardinalitāti

Šis solis ir parasts kartēšanas posms, kas saliks nepieciešamo informāciju jau redzamā datnē, kas izveidota iepriekšējā posmā, lai izejas dokumentu.



4.1.1.3.1. att. Otrā kartēšanas posma darba shēma

Sistēmas atbilstība izvirzītajiem noteikumiem:

Sistēma apmierina visas prasības, kas tika tai izvirzītas

Datņu automātiska saņemšana - ir iespēja iestatīt datņu saņemšanas viedu

Datņu automātiska transformācija- tiek automātiski sākts datņu apstrādāt process

Atskaišu veidošana - var apskatīt kādas datnes tika apstrādātas, kādi bija ievaddati un kādi bija izvaddati.

Automātiska arhivēšana - sistēma automātiski arhivē atskaites pēc sešiem mēnešiem. Pēc autora, kā lietotāja pieredzes šādi intervāli ir pilnīgi pietiekami.

Rīku ērtība (10/10)

TSIM sistēma iekļauj sevī kartēšanas glabāšanu datu bāzē, grafiskos rīkus, lai veidotu kartēšanas, rīkus kā noteikt datņu saņemšanas veidu un rīkus kā noteikt datņu sūtīšanas veidu, kā arī glabātuvī ar atskaitēm.

Uzturamība (8/10)

Sistēmai ir datu bāzes atbalsts un nepieciešamības gadījumā kartēšanas ir iespējams pārvietot no vecās TSIM versijas uz jaunāku, ja tāda nepieciešamība radīsies. Taču šāda iespēja ir pieejama tikai caur tiešo pieeju datu bāzei, nevis caur kādu rīku.

Apgūšanas sarežģītība (5/10)

Kaut gan sistēmas shēma nav sarežģīta, taču tās lietotāja saskarne, pēc lietotāju iepazīšanās ar to, nerada iespaidu, ka tā ir intuitīvi saprotama. Uzmanīgi lasot tās dokumentāciju var saprast par ko katra saskarne atbild, bet speciālista palīdzība nenāktu par sliktu. Šo iemeslu dēļ sistēmas apgūšanas sarežģītība tiek novērtēta 7 baļļu vērtībā.

Funkcionalitāte (9/10)

Rīks atbalsta XML un CSV formāta datnes un arī citus datu formātus. Ir iespējams patstāvīgi definēt jaunas datu struktūras.

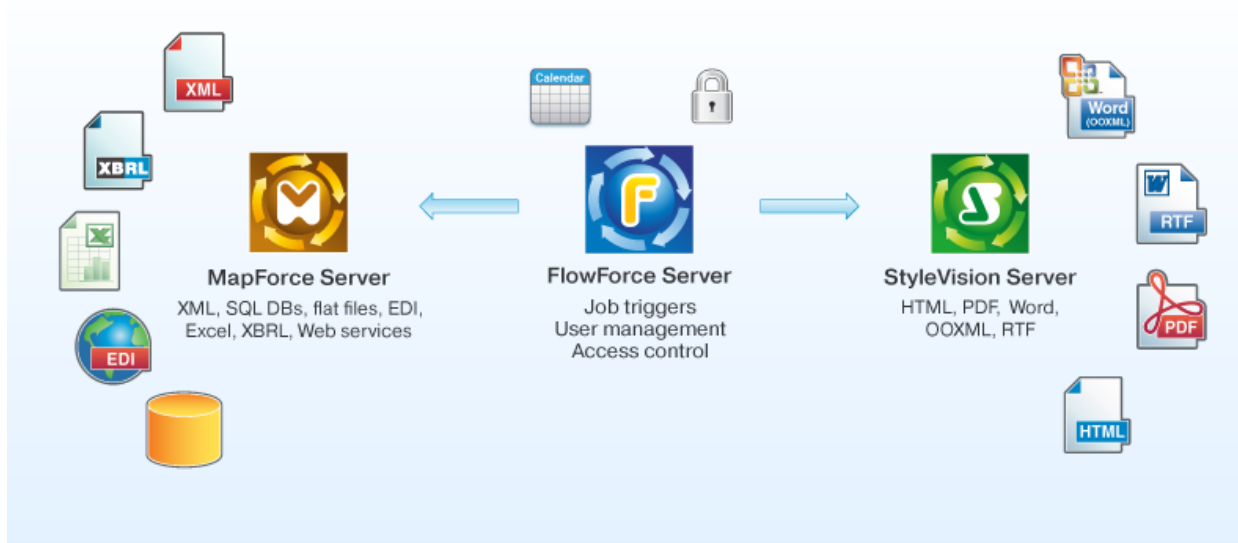
Rīkam ir iespēja nodrošināt transformācijas testu, speciālajā testa vidē, neskarot ražošanas vidi. Tas ir, sistēmai ir 2 nodalītas vides, kas ļauj aplūkot ziņojumus par kļūdām un ļauj aplūkot izejas datus.

Izmaksas

Sistēmas licence izmaksās 1500 Euro gadā, kas ir aptuveni 1058.85Ls gadā.

4.2. Altova komplekts

Kompānijas Altova piedāvātie produkti ir spējīgi nodrošināt nepieciešamo sistēmas darbu, bet ar noteikumu, ka tie tiks iegādāti vienā komplektā. Komplektā tiek iekļauts FlowForce Server un MapForce Server. Altova kompānijai ir daži rīki pēc izvēles, kas palīdz nodrošināt XML transformācijas, taču automatiskas transformācijas automatisku transformāciju ar pārējiem rīkiem palīdz nodrošināt tika MapForce Server, tāpēc autors turpmāk darbā tiks apskatīts tikai dotais rīks.



4.2.1.att. Vienkāršota FlowForce Server Sadarbības shēma

4.2.1. FlowForce Server

FlowForce Server automatizē datu transformācijas, ziņojumu un dokumentu sagatavošanu, un citus uzdevumus speciāli izdalītā serverī, virtuālajā mašīnā, vai darbstacijā, kuras ir piemērotas konkrētā projekta vajadzībām. Lietotāji nosaka vairāku soļu darbus, kas tiek uzsākti konkrētajā datumā un laikā, jaunu ieejas datu ienākšanas brīdī, vai ar HTTP komandas palīdzību. Taču FlowForce Server nodrošina tikai automatizē datu transformāciju, bet nenodrošina to reālu transformāciju. Lai nodrošinātu pašas transformācijas, zemāk būs apskatīts

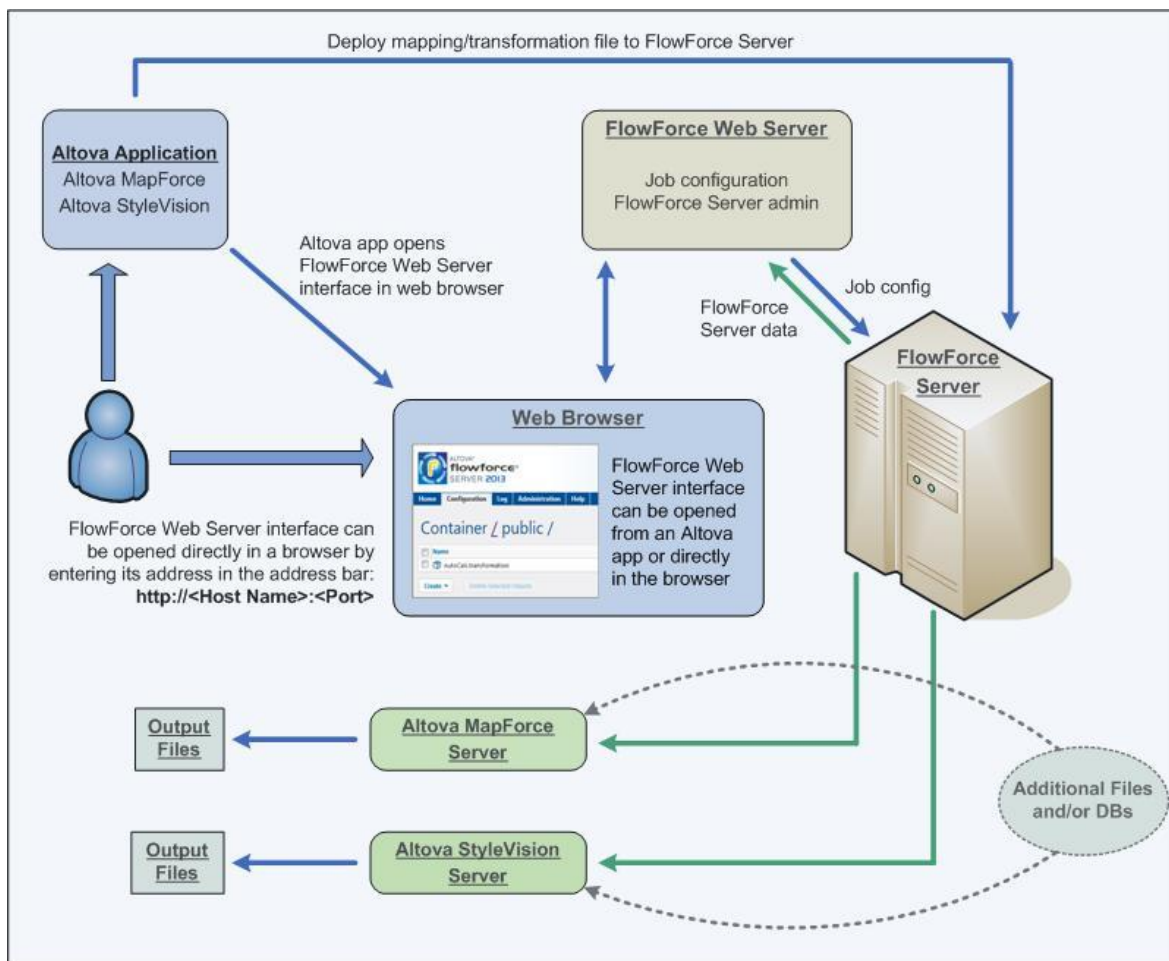
MapForce Server rīks. Rīks nodrošina atskaišu veidošanu un glabāšanu noteiktā vietā cietā diskā vai attālinātā datu glabātuvē ar iespēju noteikt periodu, kad atskaites tiks arhivētas.

FlowForce Server atbalsta trigeru definēšanu un izpildi, lai sāktu datu pārveidošanas procesu, pamatojoties uz dažādiem kritērijiem, tostarp laika trigeriem uz vienu reizi vai atkārtotai izpildei, datņu trigeri, lai sāktu darbu, kad konkrētā datne tiek mainīta, vai kad jaunā datne tiek izveidota noteiktā direktoriņā, vai nosakot attālinātos trigerus, kas palaiž procesu, kad HTTP serverī notiek konkrētas izmaiņas. FlowForce Server darbs liek MapForce Server pārveidot datus, pamatojoties uz kartēšanas datnēm, kuras augšupielādētas Altova MapForce.

The screenshot shows the 'Triggers' configuration window. It is divided into two main sections: 'Check' and 'Run'.
The 'Check' section includes a dropdown menu set to 'Content', a text field for 'of file or directory' containing 'C:\CameraGPS\hotFolder', a 'polling interval' of '30 seconds', and an 'enabled' checkbox which is checked.
The 'Run' section includes a 'Run' dropdown set to 'on days of week', a frequency of 'every 1 week(s)', and a 'Days of week' table with checkboxes for 'all', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', and 'Sun'. The 'all' checkbox is checked, and individual day checkboxes are also checked for Mon, Tue, Wed, Thu, and Fri. Below this is a 'Repeat' section with 'every 60 minutes from 08:00:00 to 20:00:00'. A 'Start' date is set to '2013-04-18'. There is an 'Expires' section with a '+' icon. The 'Time zone' is set to 'America/New_York'. An 'enabled' checkbox at the bottom of the 'Run' section is checked. At the bottom of the window are three buttons: 'new Timer', 'new Filesystem trigger', and 'new HTTP trigger'.

all	Mon	Tue	Wed	Thu	Fri	Sat	Sun
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4.2.1.1.att. FlowForce Server trigeru grafiskā saskarne



4.2.1.2.att. FlowForce Server darba shēma

ALTOVA® flowforce® Beta

Home Configuration Log Administration Help

Log View

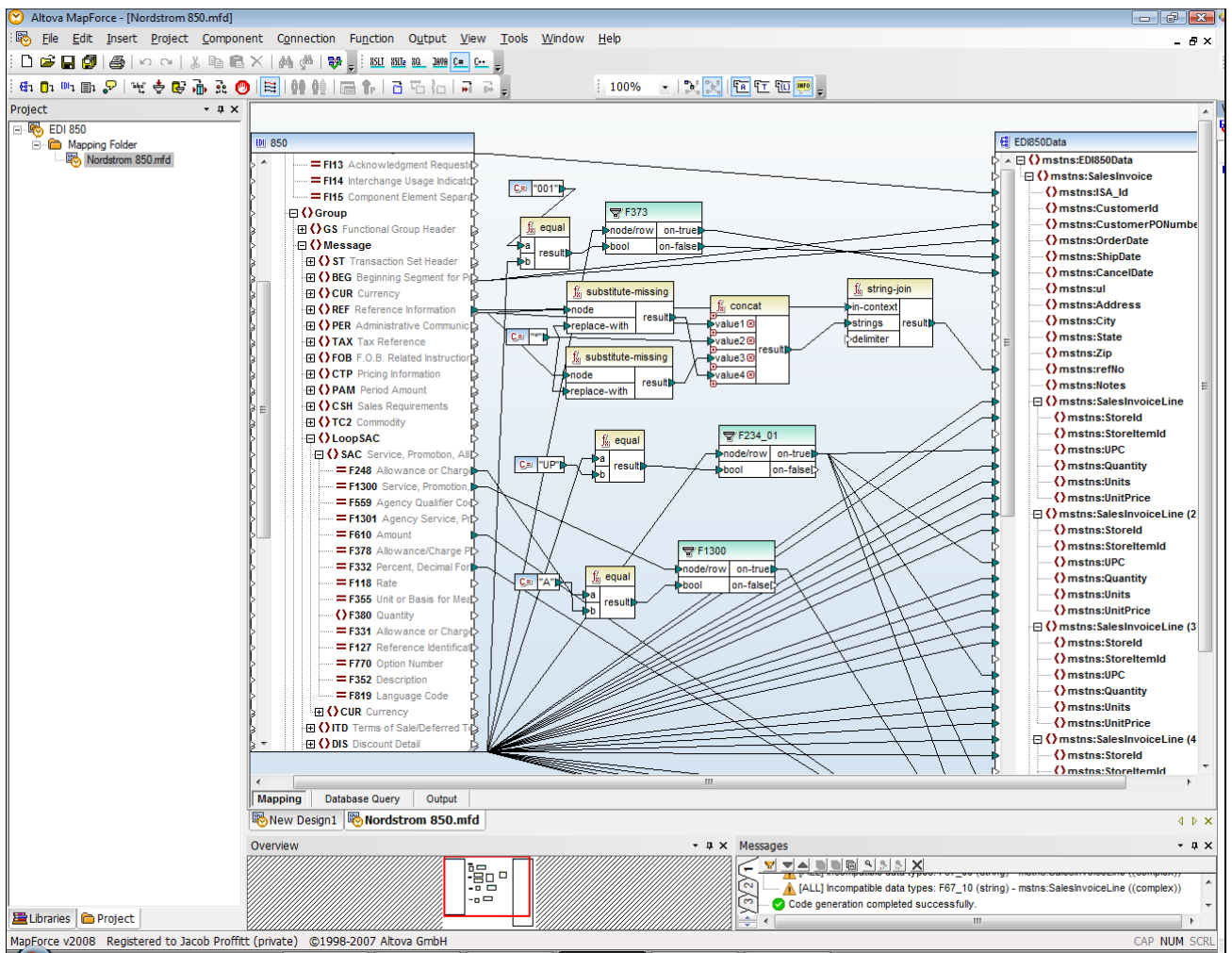
Page 1 of 19

Date	Severity	Module	Instance	Message
2013-01-30 10:56:50	INFO	flowforce	1781	Finished job execution: /public/CameraLogToGPX.job
2013-01-30 10:56:50	INFO	flowforce	1781	Step FlowForce.move completed with status: 0 more
2013-01-30 10:56:49	INFO	flowforce	1781	Executing FlowForce.move with parameters: {"Source": "C:\\Car", "Destination": "C:\\CameraGPS\\completedInput", "Overwrite":
2013-01-30 10:56:49	INFO	flowforce	1781	Step MapForce.Mapping completed with status: 0 more
2013-01-30 10:56:47	INFO	flowforce	1781	Executing MapForce.Mapping with parameters: {"Working-dir": "C:\\CameraGPS\\hotFolder\\1212030.LOG"}
2013-01-30 10:56:47	INFO	flowforce	1781	Starting job execution: /public/CameraLogToGPX.job

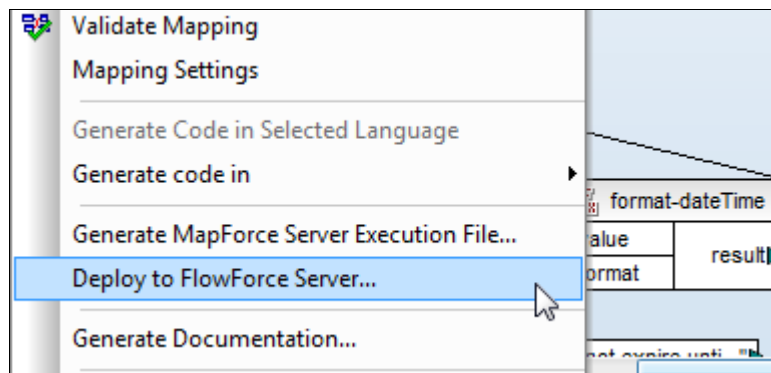
4.2.1.3. att. FlowForce Server tīmekļa saskarnes piemērs

4.2.2. MapForce Server

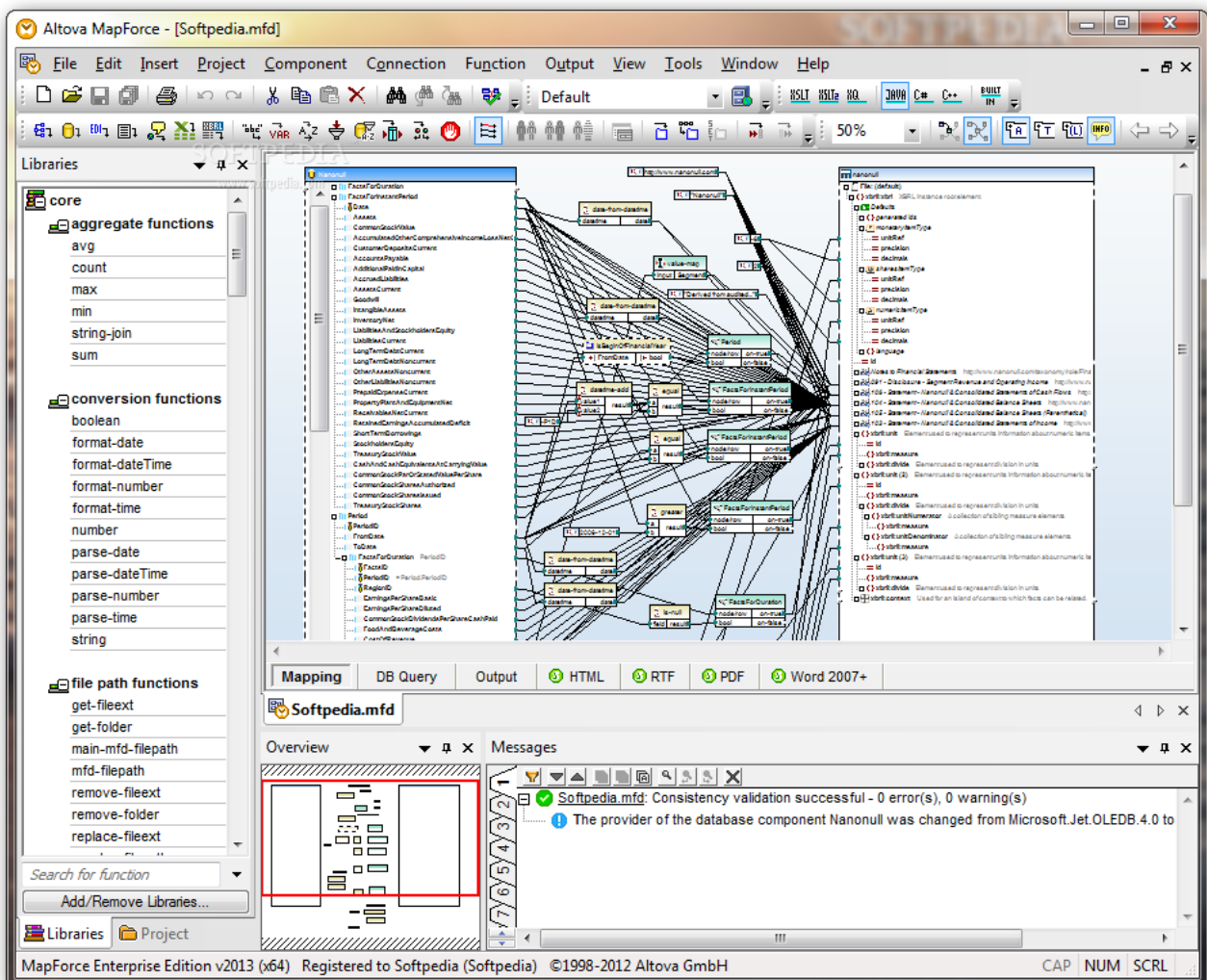
Altova MapForce servera pamatā ir iebūvētais datu transformācijas dzinējs, kas tika speciāli izstrādāts priekš MapForce un tas veic datu pārveides jebkurām XML, datu bāzēs ievietotām datnēm, EDI, Excel, XBRL un/vai Tīmekļa Servisa kombinācijām izmantojot iepriekšapstrādātas un optimizētas datu kartēšanas, kuras glabājas MapForce Servera izpildes datnēs. Iepriekš apstrāde nodrošina ātrāku veikspēju un samazinātu operatīvās atmiņas zudumus vairumam datu kartēšanu. MapForce serveris darbojas FlowForce Server vadīts, vai kā neatkarīga lietotne, kuras darbs tiek palaists no komandrindas, taču automatizētu darbu nodrošina tieši FlowForce Server. MapForce iepriekš apstrādā un optimizē datu kartēšanas, glabā tās MapForce Server izpildes datnēs priekš izpildes MapForce Servera komandrindā, un augšupielādē tās izmantošanai FlowForce Server darbiem.



4.2.2. att. Altova MapForce grafiskās saskarnes piemērs Nr.1



4.2.3. att. Kartēšanas augšupielāde FlowForce Serverim



4.2.3. att. Altova MapForce grafiskās saskarnes piemērs Nr.2

Sistēmas atbilstība izvirzītajiem noteikumiem:

Sistēma apmierina visas prasības, kas tika tai izvirzītas

Datņu automātiska saņemšana - ir iespēja iestatīt datņu saņemšanas viedu

Datņu automātiska transformācija- tiek automātiski sākts datņu apstrādāt process

Atskaišu veidošana - var apskatīt kādas datnes tika apstrādātas, kādi bija ievaddati un kādi bija izvaddati.

Automātiska arhivēšana—ir iespēja noteikt pēc kāda laika momenta tiks arhivētas atskaites.

Rīku ērtība (9/10)

Sistēma ļauj ģenerēt kartēšanas datnes, taču tās tiek veidotas specializētā nelasāmā formātā, kuru var izmantot tika Altova produkti. Tādējādi stipri mazinot tās ērtību. Kartēšanas datnes glabājas datu bāzē. Sistēma iekļauj sevī arī ērtus un saprotamus grafiskos rīkus kartēšanu sastādīšanai, kā arī ērtus rīkus servera pārvaldībai, un rīkus atskaišu veidošanai un apskatīšanai.

Uzturamība (8/10)

Sistēmai ir integrēta datu bāze un servisa atbalsts, tādējādi, ja rodas kādas problēmas ar lietotnēm, var griezties pēc palīdzības pie izstrādātājiem. Taču nav garantijas, ka kartēšanas datnes strādās uz citām lietotņu versijām. Visu transformāciju kartēšanu un funkcionalitāti ir iespējams nokompilēt tādās valodās kā JAVA, C# un C++, bet šis kods ir grūti lasāms ar lielu daudzumu iebūvēto funkciju un metožu.

Apgūšanas sarežģītība (10/10)

Sistēmai ir daudz saprotamu lietotāja saskarņu, kuras ir pietiekami saprotamas pat bez dokumentācijas lasīšanas. Saskarnes ir intuitīvi saprotamas cilvēkam, kurš saprot ar ko viņš strādā. Bet tieši savu ļoti daudzu saskarņu dēļ ir grūti uzreiz atcerēties, kur tieši atradās konkrētie iestatījumi vai iespējas.

Funkcionalitāte (9/10)

Altova komplekts izmanto ieejas un izejas datu shēmas, kuras nepieciešamības gadījumā rīks ļauj izveidot un labot, kad tas ir nepieciešams. Rīks atbalsta XML transformācijas un citu

datņu formātu transformācijas. Sistēma nodrošina datu iegūšanu tieši no izplatītākajām datu bāzu pārvaldības sistēmām, u.c. Ir iespējams lejuplādēt papildus shēmas formātiem, kuri netiek atbalstīti standarta komplektācijā. Sistēmai ir spējīga pārbaudīt vai datnes atbilst izvēlētajai shēmai.

Java koda ģenerēšanas laikā sistēma izveidoja Java projektu, kuru bija iespējams darbināt ar Eclipse vai Apache Ant rīkiem. Bet šādai koda ģenerācijai ir būtisks trūkums, tas izmanto iekodētas direktorijas un tiešsaistes uz ieejas un izejas datnēm, šī iemelsa dēļ, kods ir jālabo.

Izmaksas

Komplekta, kurš sastāv no FlowForce Servera un MapForce Servera, cena atbilstoši izstrādātāja norādītai ir sekojoša:

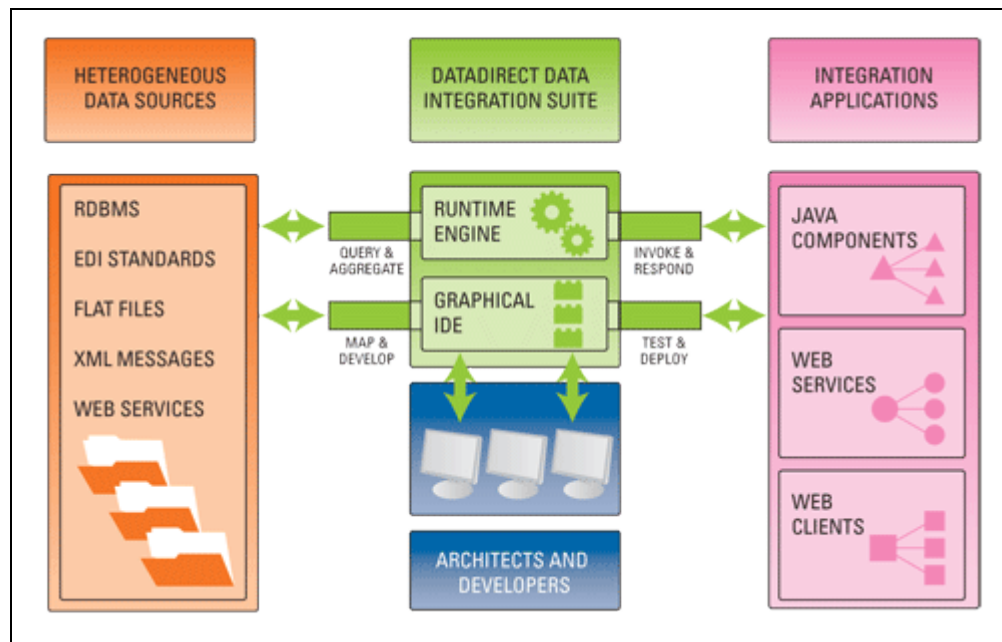
- FlowForce Server – 200 Euro
- MapForce Server – 800 Euro

Ņemot vērā, ka licence nepieciešamajiem rīkiem tiek iegādāta uz 1 gadu, FlowForce Server ir nepieciešams instalēt tikai 1 reizi un tad piekļūt attālināti, MapForce Server ir nepieciešams instalēt uz datora katram speciālistam, kurš strādās ar doto sistēmu un veidos tajā jebko. Tāpēc cena vienam speciālistam sanāk 1000 Euro gadā, tas ir apmēram 705.90Ls gadā.

4.3. DATA INTEGRATION SUITE

DataDirect kompānijas izstrādātais produkts Data Integration Suite sniedz izstrādātājiem rīkus, lai izmantot datus no gandrīz no jebkura datu avota mūsdienu XML balstītās lietotnēs. DataDirect Data Integration Suite ir trīs galvenie elementi:

- XML pieteikumu izstrādes rīks Stylus Studio,
- XQuery procesors, ar kura palīdzību lietotājam ir iespēja veidot vaicājumus (query) gan XML, gan ne-XML formāta datiem
- XML pārveidotāju komplektu.



4.3.1. att. Data Integration Suite vienkāršota darbības shēma

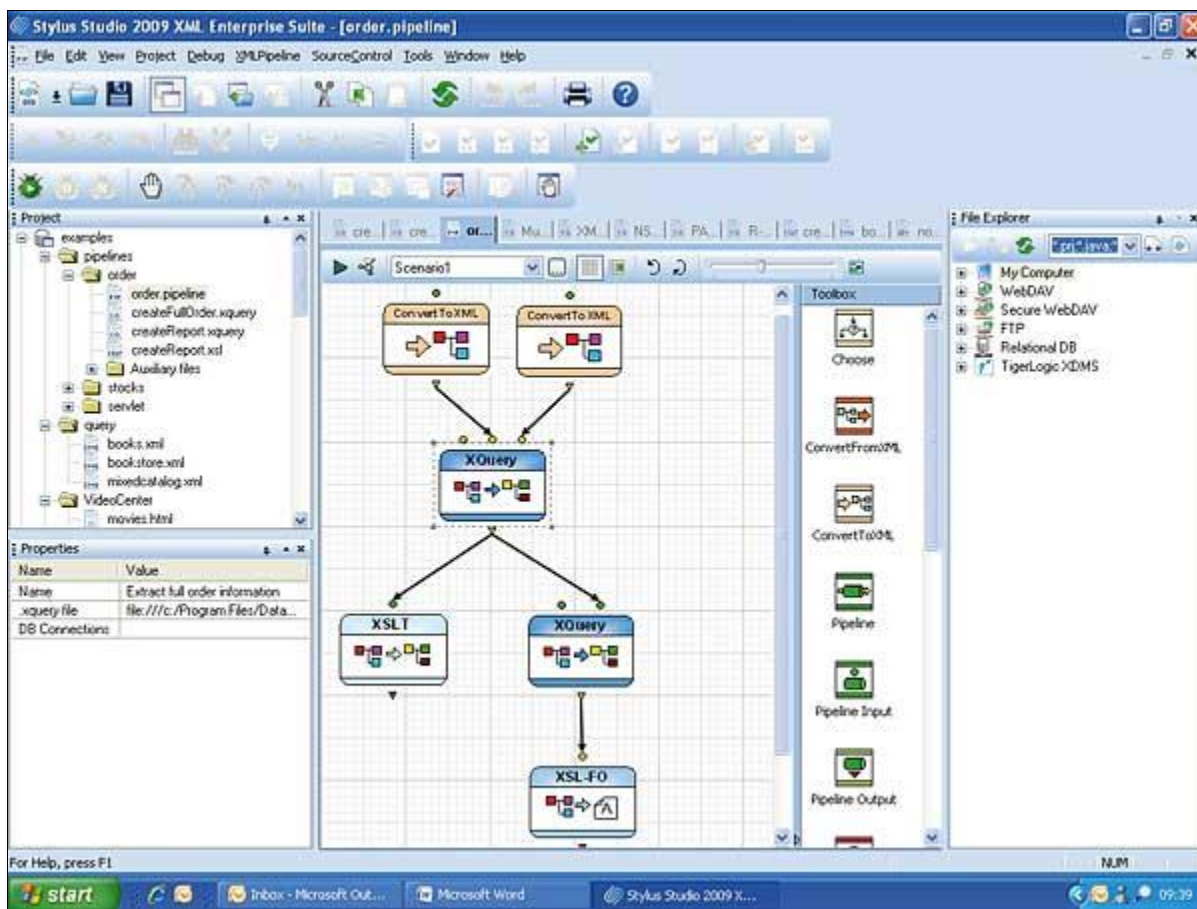
4.3.1. Stylus Studio

Stylus Studio ir grafiskais rīks, kur lietotājam ir iespēja salikt kopā XML lietotnes. Stylus Studio ir moduļi, kurus var izmantot, lai izveidotu vai rediģētu XML, XQuery, XSLT, XML Pipelines, XML ziņojumus, XML shēmas, Web servisu, Java, DTD un relāciju datu avotus. Katram modulim ir savs redaktors, un lielākā daļa atbalsta grafisko „drag and drop” tipa saskarni vai kodēšanas iespēju. Vairumā gadījumu lietotāji veido pieteikumus izmantojot Pipeline redaktoru (sk. 4.3.1.1. att.). Kā liecina nosaukums, tas ļauj jums saistīt kopā atsevišķus XML

elementus, piemēram, priekš pārveidotājiem, vaicājumiem un atskaitēm, izmantojot „cauruļvadus”, kas norāda, kā no viena elementa dati ir saistīts ar nākamo elementu.

Piemēram, lai saliktu kopā pieteikumu, kurš ņemtu datus no pasūtījumu apstrādes datu bāzē un izveidotu atskaiti, kurā būtu parādīts konkrētu produktu pārdošanas statistika, lietotājam būs jāizmanto pārveidotāju, lai pārveidotu sākotnējos datus XML formātā, XQuery, lai atrastu datus, ko vēlaties; un XSLT instrumentu, lai izveidotu HTML ziņojums.

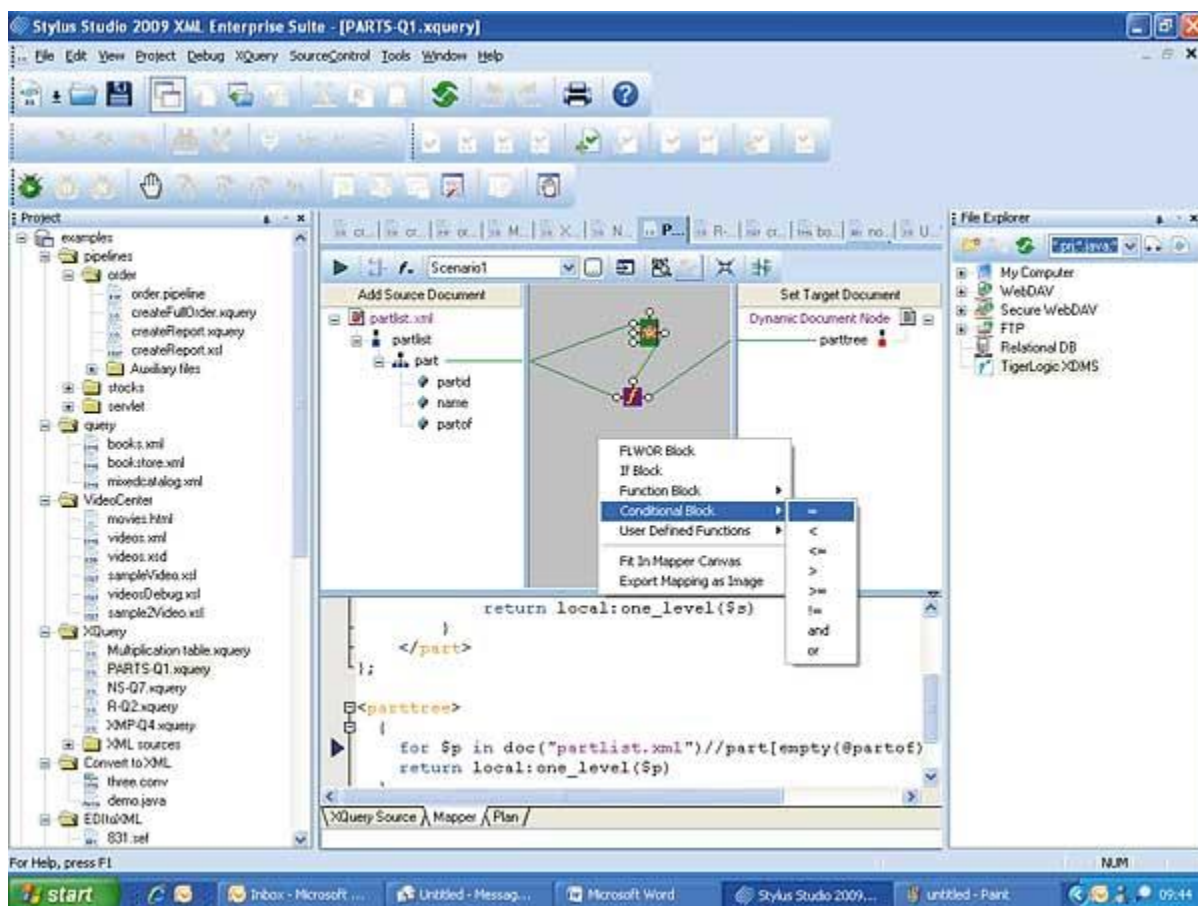
Individuālie rīki parādās kā ikonas, kuras lietotājs „velk un nomet” pareizā secībā, saistot tos ar „cauruļvadiem”. Lai gan tas izklausās vienkārši, lietotājam jābūt sapratne par vaicājumu izveidi, par to kā definēt ziņojuma izvadi, kā definēt noteikumus, lai transformētu datus.



4.3.1.1. att. XML pieteikuma izveidošana, izmantojot Pipeline redaktoru Stylus Studio rīkā

Stylus Studio izmanto grafisko kartēšanu, lai izveidotu kartēšanas vaicājumus FLWOR sintaksē, kura ir gandrīz identiska SQL vaicājumiem, tāpēc tam nevajadzētu veidot problēmas rīka lietotājam. Kartēšanas rīks nodrošina ļoti vienkāršu veidu, kā izveidot vaicājumus, un

lietotājs tajā pat laikā var ievadīt kodu tieši, ja tāda vajadzība vai vēlšanās rodas (sk. 4.3.1.2. att.). Ir iespēja veidot atskaites izmantojot HTML+CSS vai XSL:FO formātu, izmantojot vai nu XSLT, vai XQuery, lai definētu atskaites struktūru.



4.3.1.2. att. Pievienojot nosacītu bloku uz XQuery ar Kartēšanas rīku

4.3.2. XQuery procesors

XQuery procesors iekš Data Integration Suite saucās DataDirect XQuery. Tas ir tāds pats kā XQuery procesors iekš Stylus Studio rīka, tomēr šī sistēmas sastāvdaļa ir procesors, kas var iekļauties jebkurā Java programmā priekš izpildes. Nav nepieciešamības kādam citam, trešās puses, produktam, tas neizmanto serveri, un tai nav nepieciešama nekāda papildu produkta vai lietotņu servera. Jūs varat to izmantot, lai piekļūtu pie XML, relāciju datiem un citai informācijai ienākošajās datnēs un pielietotu tam vaicājumu. Relāciju atbalsts ietver tādu iespēju, kā labot vaicājumus. XQuery procesors kalpo kā rīks, kurš automatizē datņu saņemšanu, transformēšanu un nosūtīšanu, kā arī atskaišu izveidi, ja tādas tika definētas. Tās ir automatizācijas „smadzenes”.

4.3.3. XML pārveidotāju komplektu

Data Integration Suite ietilpst arī rīki, kas nodrošina transformācijas datiem, kas nav XML formātā. Taču šī darba ietvaros tas netiks apskatīts, tā kā tas neietilpst darba mērķos.

Sistēmas atbilstība izvirzītajiem noteikumiem:

Sistēma apmierina visas prasības, kas tika tai izvirzītas

Datņu automātiska saņemšana - ir iespēja iestatīt datņu saņemšanas viedu

Datņu automātiska transformācija- tiek automātiski sākts datņu apstrādāt process

Atskaišu veidošana - var apskatīt kādas datnes tika apstrādātas, kādi bija ievaddati un kādi bija izvaddati. Ir iespēja izvēlieties noteiktus datus no ievaddatiem vai izvaddatiem, vai abiem. Bet jo lielāku daudzumu datu ir nepieciešams dabūt no sistēmas, jo lielāka slodze ir uz servera.

Automātiska arhivēšana—ir iespēja noteikt pēc kāda laika momenta tiks arhivētas atskaites.

Rīku ērtība (9/10)

Vienīgā sistēma, kurai ir specializēts modulis ar plašām iespējām atskaišu veidošanai. Kā arī vienīgā sistēma, kura glabā visus datus sadalītus iekš datu bāzes, nevis kā XML datni ar visu informāciju iekš tās. Tas palielina ērtību, ja būtu nepieciešams dabūt kādu noteiktu ierakstu, bet palielina slodzi uz serveri, ja būs nepieciešams dabūt visu ievada datni vai izvada datni.

Uzturamība (9/10)

Sistēmai strādā ar savu datu bāzi, kurā glabājās visi vaicājumi, ienākošie datu un dati kurus pēc tam sūta uz norādīto adresi. Visas transformācijas ir iespējams pārveidot JAVA vai C# valodā, taču vaicājumi glabājas FLWOR sintaksē, kuru, ja būs nepieciešams, vajadzēs pārredīgēt manuāli nepieciešamajā sintaksē.

Apgūšanas sarežģītība (9/10)

Sistēma ir saprotama un tai ir daudz specializētas dokumentācijas un pamācošu video, kuru nebija nevienai iepriekš aprakstītai sistēmai, rīku lietotāja saskarne ir saprotama pietiekamā

līmenī, lai mēģinātu iziet cauri iespējām bez īpašas sagatavotības. Taču rīku procesu automatizēšanai ir nepieciešamas labas zināšanas programmēšanā.

Funkcionalitāte (8/10)

Sistēma atbalsta XML datu transformācijas un citu datņu formāta transformācijas, taču salīdzinoši ar citām sistēmām, šo formātu ir mazāk. Rīks ļauj iegūt informāciju kā no datnēm tā no relāciju datubāzēm. Sistēmā ir pašai sava skriptu izveides sintakse, kas ļauj veidot sarežģītākas transformācijas ar datiem. Vienīgais mīnus salīdzinoši ar citām sistēmām bija mazāks atbalstāmo formātu klāsts.

Izmaksas

Visam produktam kopumā autoram neizdevās uzzināt cenu, izstrādātāja nevēlēšanās dēļ paust šādu informāciju, taču izdevās uzzināt sistēmas sastāvdaļas cenu, tāpēc dotajai sistēmai cena tiks sastādīta aptuvena pēc tās iespējām.

695.00 Euro Licence priekš Stylus Studio

145.95 Euro Licence katram nākamajam lietotājam priekš Stylus Studio

XQuery līdzīgu procesoru licences cena parasti ir 800 Euro gadā.

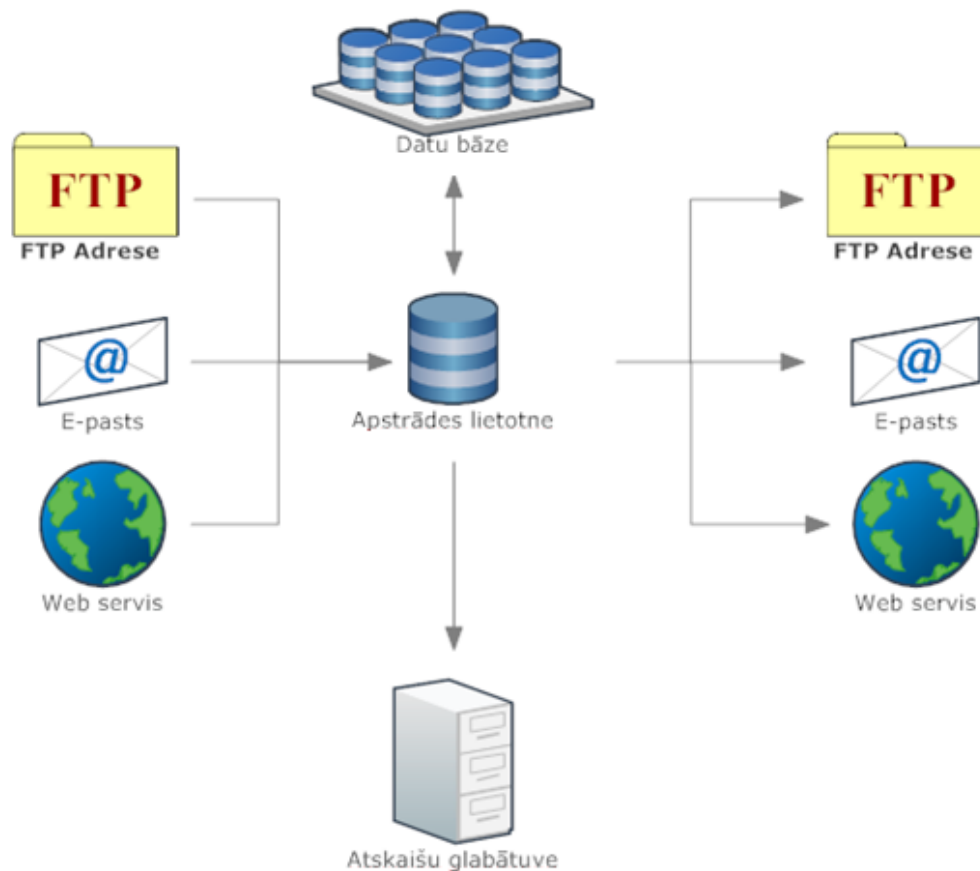
Tātad šādas sistēmas cena varētu būt apmēram 1495 Euro gadā, tas ir 1055.32 Ls gadā.

4.4. Individuālā sistēma

Tirgū pašlaik nav ļoti lielas izvēles jau gatavu sistēmu kuras nodrošinātu nepieciešamo funkcionalitāti un pie tam būtu pārdodamas kā sistēma, kuru pircējs var lietot savā uzņēmumā kā savu. Parasti tādas sistēmas tiek pasniegtas kā pakalpojums ko sniedz pārdevējs, un klients savukārt maksā par kādu noteiktu daudzumu rēķinu, datņu vai informācijas daudzumu, kas iziet cauri sistēmai.

Darba mērķis bija salīdzināt esošās sistēmas un to funkcionalitāti ar autora izveidoto prototipu un dot lasītājam priekšstatu par izmaksām, kuras veidosies sistēmas veidošanas laikā. Autors sistēmas prototipu veidos pēc pieredzes dotajā nozarē. Sistēmai, kura sekmīgi darbotos, kā jau bija minēts iepriekš, ir jā sastāv no vairākiem moduļiem, kuri būs, prototipa kontekstā, detalizētāk aprakstīti zemāk:

- Ienākošās informācijas saņemšana,
- Ienākošo datņu apstrāde,
- Apstrādāto datņu nosūtīšana,
- Atskaišu veidošana



4.4.1. att. Sistēmas darba vienkāršota shēma

4.4.1. Ienākošās informācijas saņemšana

Sistēmai ir jābūt spējīgai saņemt informāciju no vairākiem avotiem, atpazīt šo informāciju, kā pareizu un gatavu apstrādei, vai tā ir valīda. Autora pieredzē informācija var tikt saņemta no e-pasta, attālinātām FTP adresēm vai kā nosūtīta informācija no kāda WEB servisa. Tas var tikt panākts izstrādājot lietotni vai serveri, kurš automatizēs informācijas saņemšanu un nosūtīšanu tālāk lietotnēm, kuras sāks apstrādāt un transformēt datus, pēc to transformācijas beigām lietotnei vai serverim ir jānāk nosūtīt šos datus uz vienu no klienta izvēlētajām vietām.

Visupirms saņemot informāciju sistēmai ir jāpārbauda vai ienākošā XML datne ir valīda sazinoties ar datu bāzi un salīdzinot ar esošo XML struktūru vai kartēšanu (sk. 15. att. - Informācijas saņemšana), atrodot to pēc konkrētās XML versijas un identifikatora no noteikta

XML elementa, XML datnēs versija ir pats pirmais elements (angļu tag), kartēšanas identifikators var saturēties citos elementos:

```
<?xml version="1.0" ?>
```

Gadījumā, ja klients nosūta datni, kuras versija nav datu bāzē, kļūdas paziņojums tiek nosūtīts uz Atskaišu glabātuvī, un paziņojums par konkrēto kļūdu tiek nosūtīts klientam.

Autors uzver, ka individuālā sistēma dotajā prototipā ir domāta tikai XML datņu formāta apstrādei.

4.4.2. Ienākošo datņu apstrāde

Autors pēc pieredzes dotai sistēmai izmanto XSLT transformācijas valodu, jo tā ir specializēti izstrādāta priekš XML datņu transformēšanas citās XML datnēs. Izveidojot lietotni, kura pēc konkrētām informācijas no ienākošās XML datnes noteiks, kuru kartēšanu izmantot izejas datnei un kādu informāciju no ienākošās datnes būs nepieciešams ievietot elementos izejošā datnē.

Informāciju par klienta nepieciešamo izejas datni ir iespējams ievietot XML elementos. Piemērā zemāk var aplūkot konkrētu iespēju :

```
<DOCUMENT schemaxmlns="Coded::DB:ID" xmlns:ppl="Coded::DB.table.elementPeople"
xmlns:addr="Coded::DB.table.elementAddress"
xmlns:vehicle="Coded::DB.table.elementVehicle">>
```

Piemēra var redzēt, ka shēma tiek ņemta no adreses "<http://www.w3.org/2001/XMLSchema>", un tiek izmantotas trīs konkrētas shēmas izvadei:

- <DOCUMENT schemaxmlns="Coded::DB:ID"
- xmlns:ppl="Coded::DB.table.elementPeople"
xmlns:addr="Coded::DB.table.elementAddress"
- xmlns:vehicle="Coded::DB.table.elementVehicle">

Informāciju no ienākošām XML datnēm ir iespējams glabāt datu bāzē kā visu datni, vai kā saskaldītu pa elementiem informāciju. Visas datnes glabāšanai priekšrocības ir, mazāks laiks,

kad būs nepieciešams dabūt visu datni no datu bāzes, taču laiks palielināsies, kad vajadzēs dabūt konkrēta elementa vērtību no datnes. Datnes glabāšanai datu bāzē saskaldītā veidā pa elementu vērtībām ir tieši otrādas priekšrocības: veselas datnes dabūšana no datu bāzes aizņems lielāku laiku, taču konkrētu datnes elementu dabūšana no datu bāzes aizņems mazāk laika. Autors sistēmas prototipā izmantos datnes skaldīšanu, jo bieži vien, kādu iemeslu dēļ, ir nepieciešams atrast konkrētu informāciju konkrētajā datnē.

Katrai konkrētai datnei, kā jau katram, piemēram, rēķinam, ir savs numurs pēc kura to var atpazīt, kas var darboties kā primārā atslēga pēc kuras to var ātri atrast datu bāzē un līdz ar to tās konkrēto elementu:

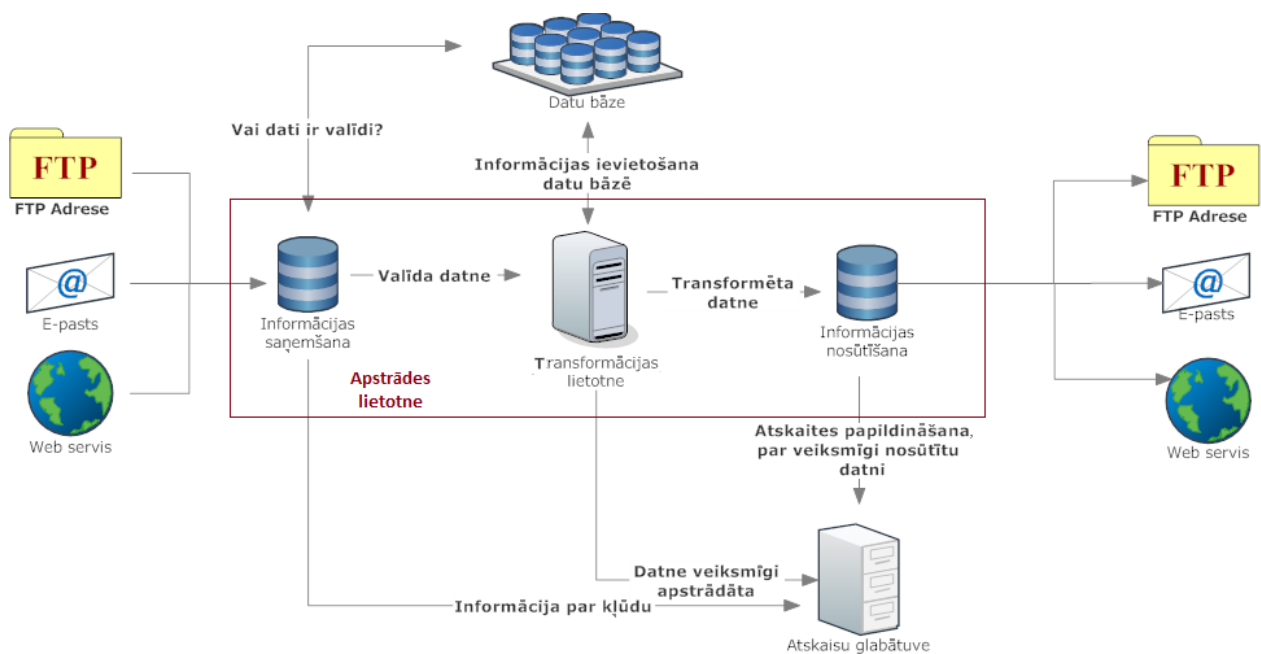
```
<?xml version="1.0" encoding="utf-8"?>

<DOCUMENT schemaxmlns="Coded::DB:ID" xmlns:ppl="Coded::DB.table.elementPeople"
xmlns:addr="Coded::DB.table.elementAddress"
xmlns:vehicle="Coded::DB.table.elementVehicle">

<SUBDOCUMENTS>

    <DOCUMENT name="B201210025113BC51.xml" type="SINGLE">
```

Izmantojot XSLT transformāciju tiek noformēta izejas datne atbilstoši visiem noteikumiem, tiek arī noformēta atskaite par veiksmīgu ievades datnes apstrādi, un veiksmīgu izvades datnes nosūtīšanu uz noteiktu vietu.



4.4.2.1. att. Apstrādes lietotnes sadarbība ar citām sistēmas daļām

4.4.3. Apstrādāto datņu nosūtīšana

Pēdējais posms ir informācijas nosūtīšanas klienta norādītajā vietā, ar šo var nodarboties tas pats modulis, kas nodarbojās ar informācijas saņemšanu un nosūtīšanu Transformācijas lietotnei. Ar svarīgu izmaiņu - konkrētā atskaite par konkrēto datni tiks papildināta ar ierakstu, ka datne tika nosūtīta veiksmīgi (sk. 15. att.).

4.4.4. Individuālās sistēmas trūkumi

Izstrādājot savu individuālo XML datņu noformēšanas sistēmu, pasūtītājam jābūt priekšstatam par to, ka tas nebūs lēts pasākums un naudu būs nepieciešams izsniegt pirms darba sākuma, vai darba gaitas laikā. Vienkāršai sistēmai, kā ir minēts prototipā, nav pieejami grafiskie rīki, kartēšanu noformējumam, un visas kartēšanas ir nepieciešams formulēt XSLT valodā un SQL vaicājumos, tas savukārt nozīmē to, ka tam vajag cilvēku ar pamat programmēšanas iemaņām un datu bāzu zināšanām, nevis parasto operatoru.

4.4.5. Individuālās sistēmas izmaksas

Gadījumā, ja personai, kura vēlas savam uzņēmuma izveidot un ieviest tādu sistēmu, ir informācijas tehnoloģiju (IT) nodaļa Latvijā, ar pieredzējušiem speciālistiem datu bāzu pārvaldībā, serveru iestatīšanā un izveidošanā, un lietotņu veidošanā, kuri zina, kā līdzīgas komponentes strādā un kā tās savienot vienā sistēmā, pēc autora pieredzes, tas varētu aizņemt dažus mēnešus: no 2-6 mēnešiem, ar noteikumu, ka pie tā strādās vismaz 1 sistēmu arhitekts, 1 sistēmu analītiķis, 1 projekta vadītājs, un 4 programmētāji un 3 testētāji. Autors ir aprēķinājis vidējās algas no vairākiem avotiem [8][9], kur piedāvā darbu IT speciālistiem.

Cenu šādai sistēmai var izrēķināt vienkārši saskaitot algas padotajiem:

- 1 sistēmu arhitekts uz 1-2 mēnešiem (tā kā sistēmu arhitekta darbs parasti nav piesiets 1 sistēmai uz visu tās izstrādes laiku uz 100%) : 1300Lsmēnesī = **1300Ls līdz 2600Ls**
- 1 sistēmu analītiķis uz 1-3 mēnešiem (tā kā sistēmu analītiķa darbs parasti nav piesiets 1 sistēmai uz visu tās izstrādes laiku uz 100%): 800Ls mēnesī = **800Ls līdz 3200Ls**
- 1 projekta vadītājs uz 2-4 mēnešiem :1100Ls mēnesī = **2200Ls līdz 4400Ls**
- 4 programmētāji uz 2-4 mēnešiem :700Ls mēnesī = **5600Ls līdz 11200Ls**
- 3 testētāji uz 2-4 mēnešiem : 500Ls mēnesī = **1000Ls līdz 2000Ls**

No šiem skaitļiem var izrēķināt, ka dotā sistēma, izstrādāta Latvijā, sava uzņēmuma IT nodaļā izmaksās **10900Ls līdz 23400Ls**. Cena ir rēķināta balstoties tikai uz darbinieku algām., tas nozīmē, ja pasūtītājs grib šādu produktu pasūtīt no kāda IT uzņēmuma, šī summa var izaugt vēl lielāka.

4.5.Kopsavilkums

Produktu novērtējumu kopsavilkums ir redzams tabulā 4.5.1.

4.5.1 tabula

Sistēmu vērtējumu kopsavilkums

	TSIM	Altova komplekts	DATA INTEGRATION SUITE	Individuālā sistēma
Rīku ērtība	10	9	9	X
Uzturamība	8	8	9	X
Apgūšanas sarežģītība	6	10	9	X
Funkcionalitāte	9	9	8	X
Izmaksas (Ls)	1059Ls gadā	706Ls gadā	1055Ls gadā	10900Ls līdz 23400Ls pavisam

Pēc autora veiktā pētījuma par doto produktu stiprajām un vājajām pusēm, gribētos piebilst, ka gatavie risinājumi, kuri tika aprakstīti darbā, veic ne tika XML datu tipa transformācijas, bet arī citu datņu formātu transformācijas, bet cenas ir pieejamas tikai par visu komplektu kopā. Savukārt individuālā sistēma, ko pasūtītājs grib izveidot savam uzņēmumam šī prototipa ietvaros ir spējīga noformēt tika XML formāta datnes.

Darbā uzskaitītās un analizētās sistēmas nav vienīgās pieejamās, kuras var izmantot datu saziņai un to transformēšanai. Taču izvēle nav stipri liela, parasti kompānijas izveido līdzīgas sistēmas un tad piedāvā tās kā servisu un naudu liek maksāt par izmantošanu gadā, tā parasti nav ļoti liela summa, un par kādu noteiktu datņu daudzumu, ko sistēma apstrādā. Tādā veidā cena var augt atkarībā no tā, cik daudz datņu lietotāja kompānija nosūtīja transformācijai gadā.

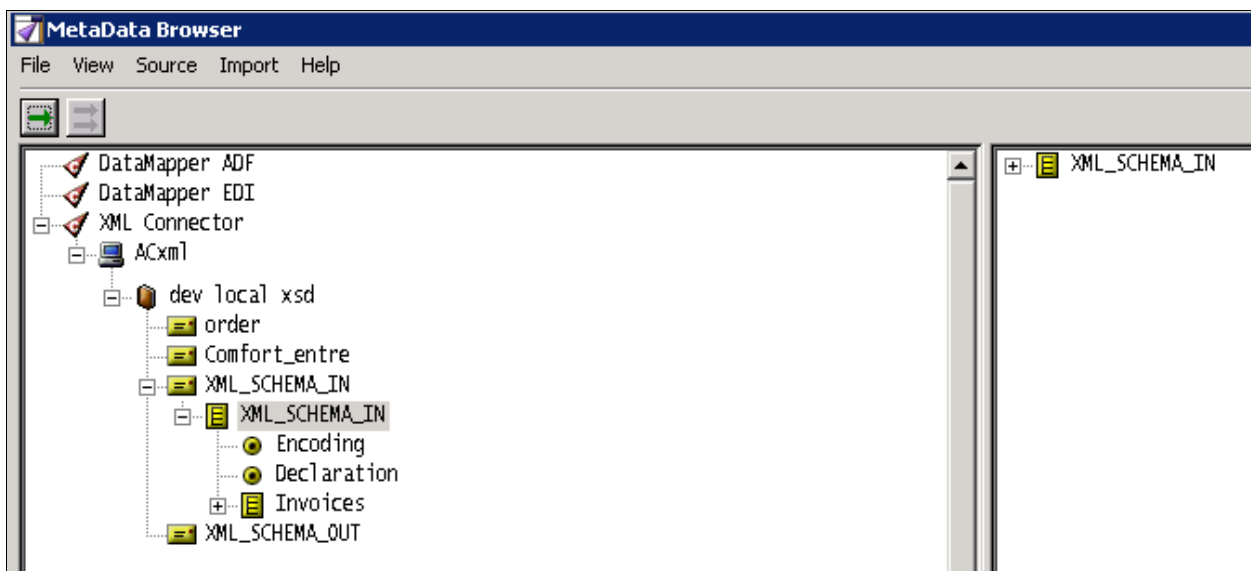
Grafiska lietotāja saskarne ir visiem augstākminētiem produktiem, un tā nepārprotami atvieglo darbu ar transformāciju veidošanu, taču var rasties gadījumi, kad grafiskā saskarne nenodrošina nepieciešamo noteikumu un likumu ieviešanu, kur var nākt palīgā kartēšanas koda labošanas rīks, tas ir, koda labošana manuāli.

5. Transformācijas praksē

Lai praksē pārbaudītu katra rīku piemērotību autora darba vietas datu transformācijai, autors veiks transformācijas ar XML datņu tipiem katrā no sistēmām. Tikš padota 1 datne ar ienākošo informāciju (sk. 1. pielikumu) un sistēmai būs jāizdod 1 datne, sistēmām tiks padotas vienas un tās pašas datnes un iznākumā jāsanāk vienādām izvada datnēm katrā sistēmā.

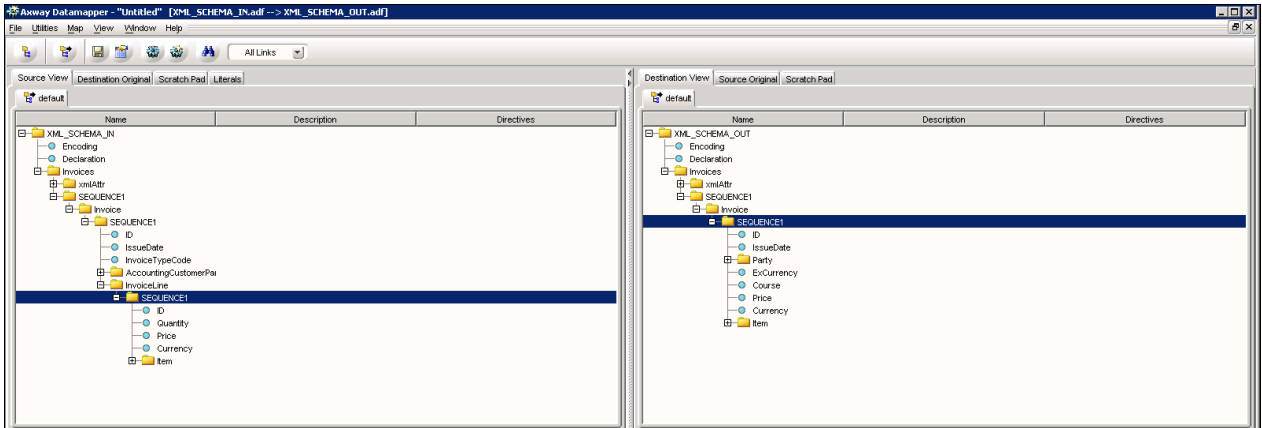
5.1.1. TSIM transformācija

TSIM rīkam nebija iekšējā rīka lai izveidotu XML shēmu, tāpēc to bija nepieciešams izveidot vai nu manuāli vai kādā citā rīkā. Autors šim nolūkam izmantoja automātisko XML shēmu veidošanas rīku, kas ir pieejams globālajā tīmeklī [12]. Pēc XML shēmas izveides, kurai ir XSD paplašinājums (3. pielikums), to ir nepieciešams augšupielādēt TSIM sistēmā (sk. 5.1.1.1. att.). Attēlā 5.1.1.1. var redzēt augšupielādēto shēmu, kura XSD formātā TSIM sistēmā, ir apskatāmas ar visiem elementiem (sk. 5.1.1.2. att.). Augšupielādēšana tiek veikta ar Metadata Browser rīka palīdzību, kurš ir TSIM sistēmas sastāvdaļa.



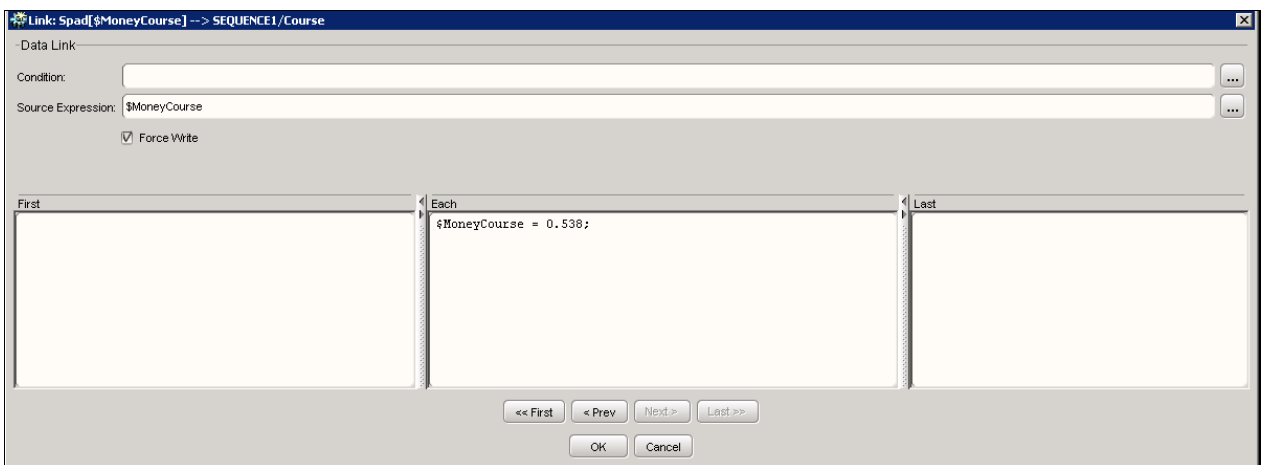
5.1.1.1. att. Augšupielādēta XML shēma TSIM sistēmā

Nākamais solis transformācijas veidošanā TSIM sistēmā ir kartēšanas izveidošana, kura notiek ar tehnikas „drag and drop” palīdzību, kur pārvelk ieejas shēmas elementu uz izejas shēmas atbilstošo elementu (sk. 5.1.1.4. att.).



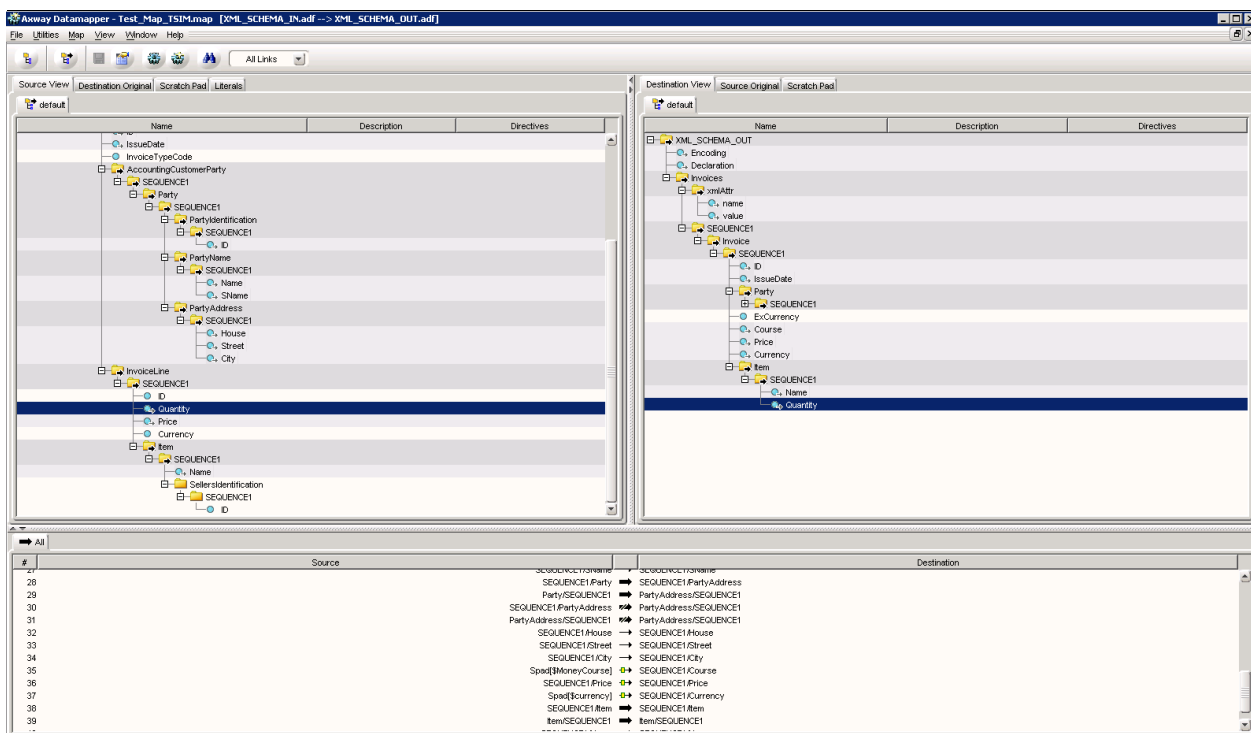
5.1.1.4. att. Kartēšanas izveidošana starp XML shēmām

Ja transformācijas ir nepieciešams izmantot kādu konstanti, tad to ievieš iekš kartēšanas procesa (sk. 5.1.1.5. att.). Šo rīka funkciju ir iespējams arī izmantot 0:1 un 0:N kardinalitāšu veidošanai, kā arī ir iespēja ņemt vērtības no datu bāzes. Šo funkciju arī ir iespējams izmantot kardinalitātes N:1 problēmu risināšanai, piemēram kombinēto aprēķinu veidošanai.



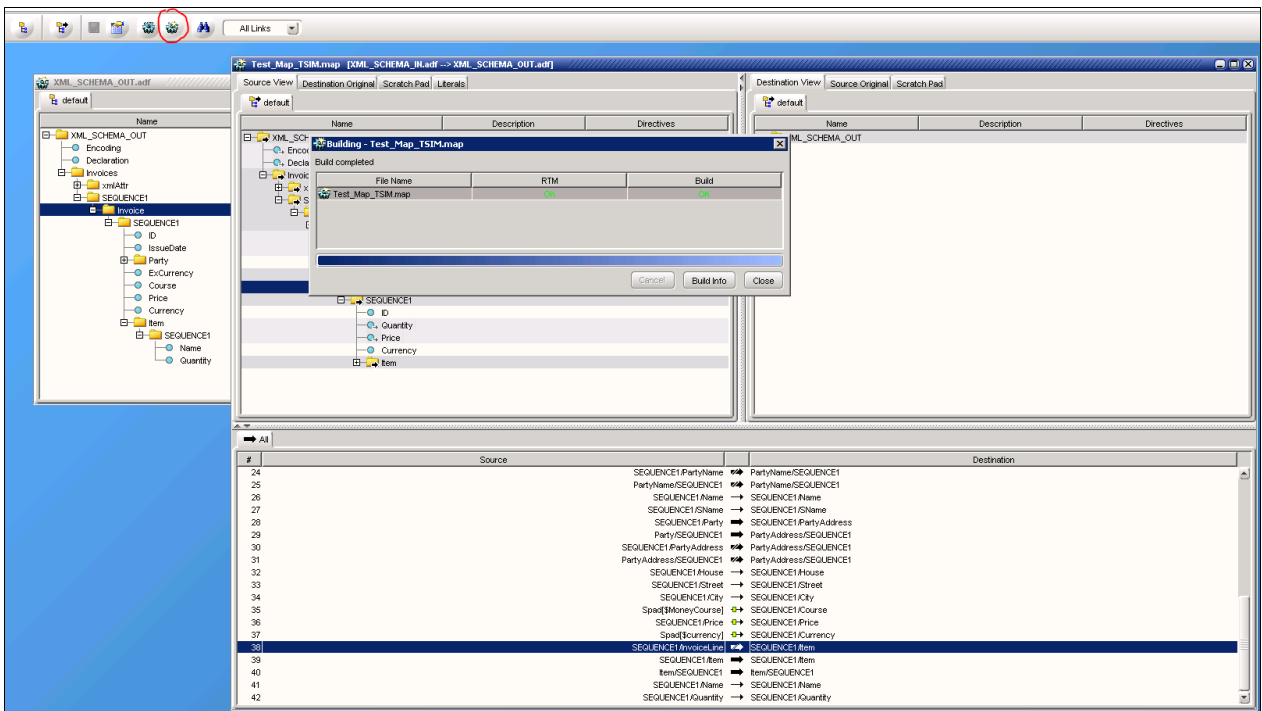
5.1.1.5. att. Ieviesta konstante TSIM kartēšanā

Attēlā 5.1.1.6. var redzēt kā izskatās nokartētas XML shēmas. Ir jāpiebilst, ka kartēšana notiek ne tikai starp elementiem, kuri ietilpst informācijas transformācijā, bet arī mātes elementi.



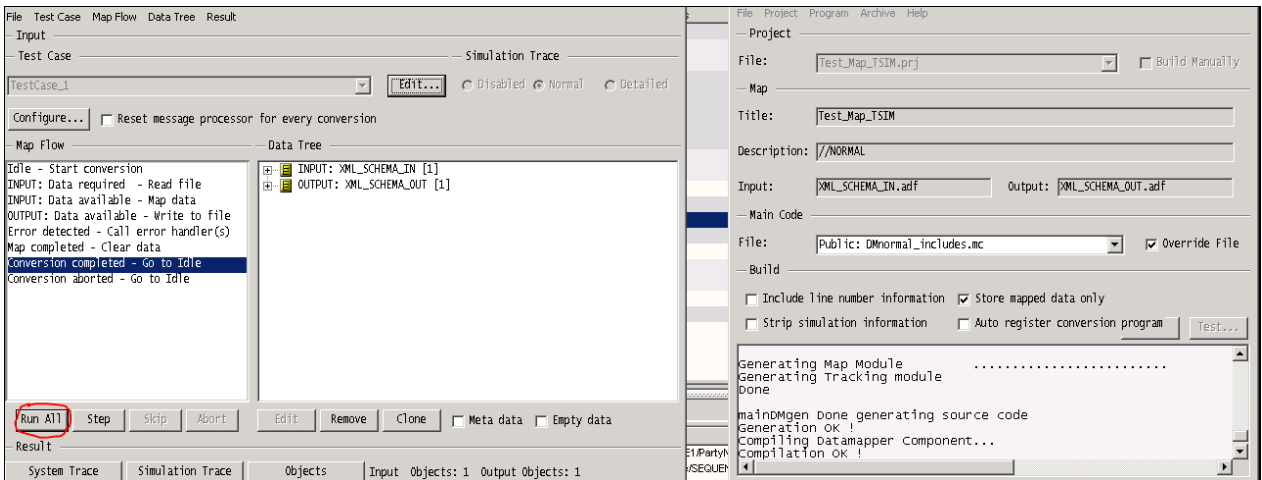
5.1.1.6. att. Pilnīgi nokartēta transformēšana

Pēc kartēšanas pabeigšanas to ir jāpārbauda, ko nodrošina speciāla operācija, kura automātiski augšupielādē kartēšanu testa vidē un notestē to (sk. 5.1.1.7. att.).

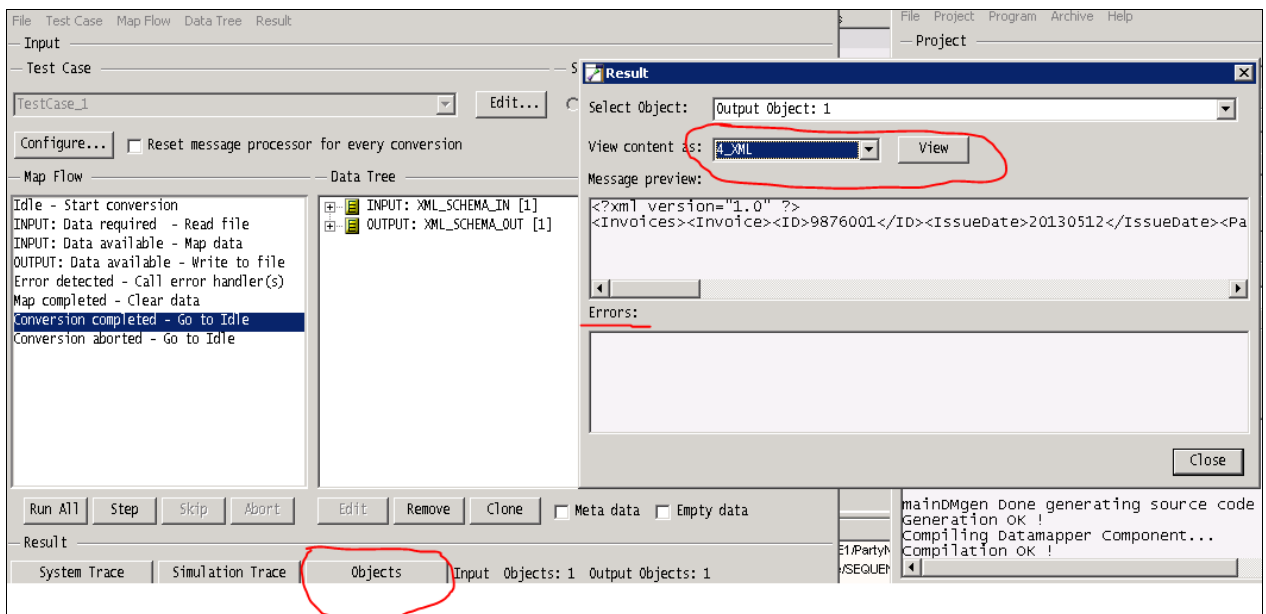


5.1.1.7. att. Kartēšanas testēšana

Pēc kartēšanas veiksmīgas testēšanas ir laiks padot sistēmai ieejas datni un sagaidīt rezultātu (sk. 5.1.1.8. att. un 5.1.1.9. att.).

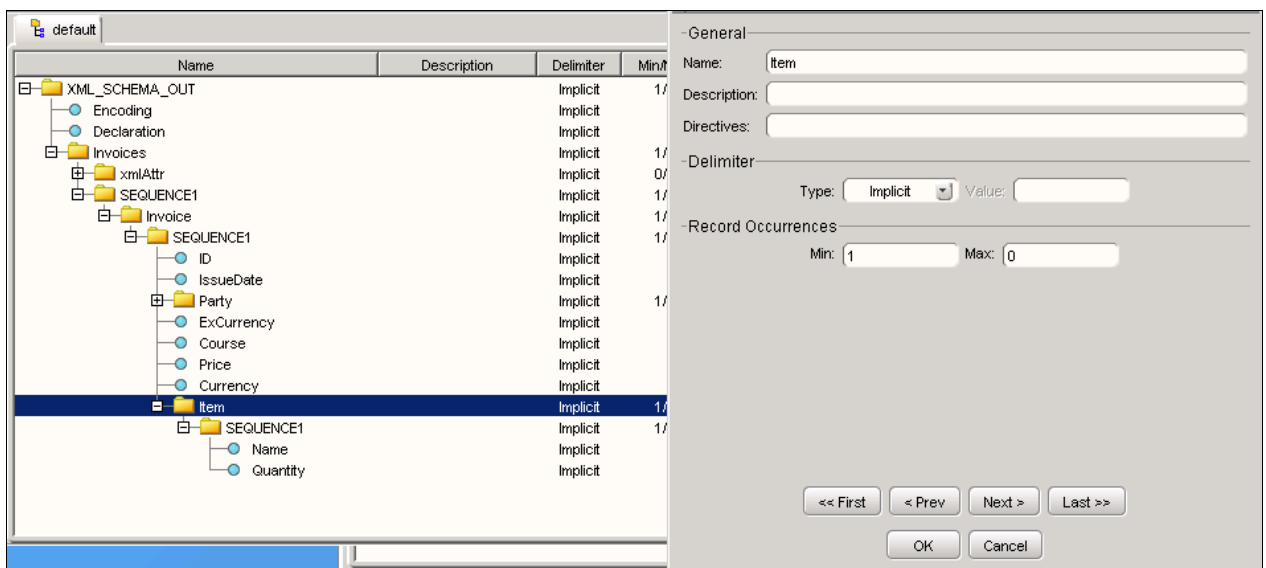


5.1.1.8. att. Ieejas datu padošana un rezultāta saņemšana Nr.1



5.1.1.9. att. Ieejas datu padošana un rezultāta saņemšana Nr.2

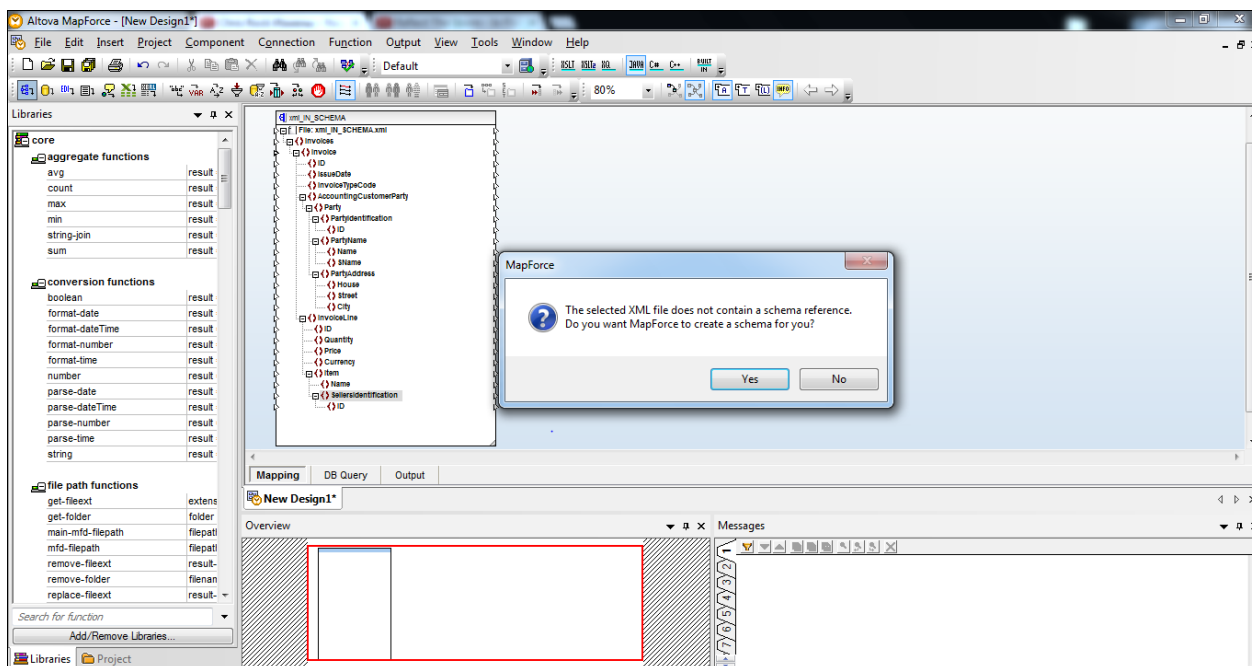
Gadījumos, kad ir nepieciešams izveidot transformācijas ar elementiem, kuriem ir vienādi nosaukumi, kā tas ir redzams 1. pielikumā elementā `<InvoiceLine>`, TSIM ir paredzēta kardinalitāšu attiecību funkcija, kura ir redzama 5.1.1.10. attēlā.



5.1.1.10. att. Kardinalitāšu attiecību funkcija

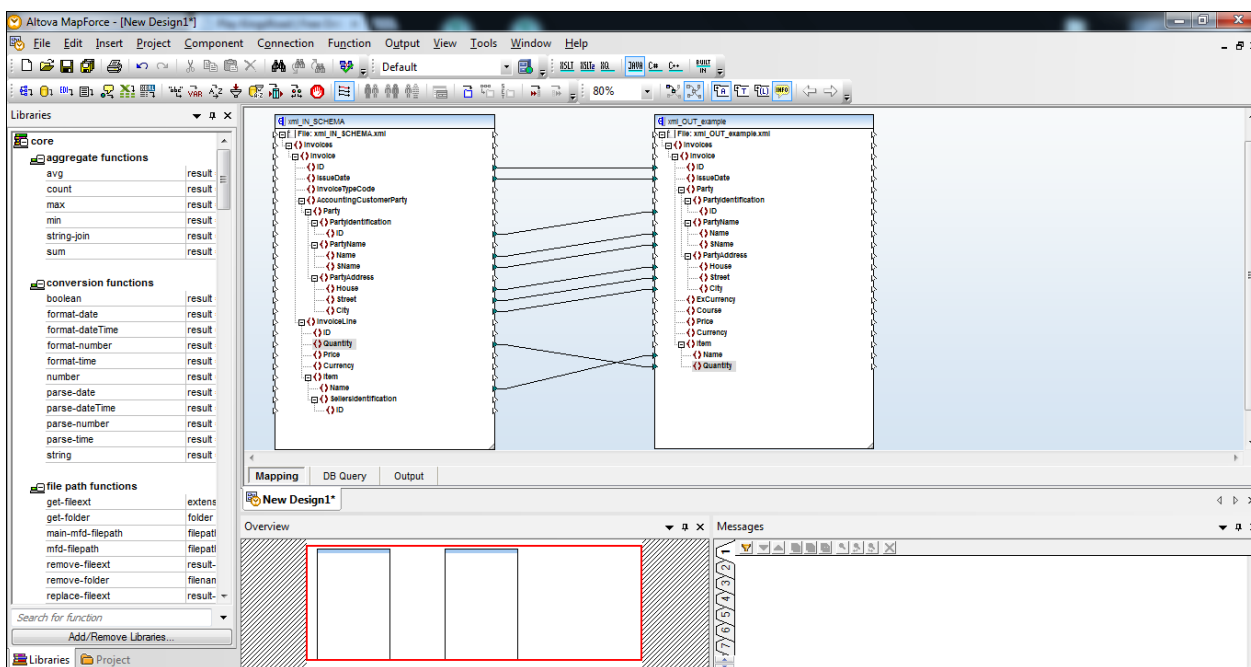
5.1.2. Altova komplekta transformācija

Altova sistēmai ir iebūvēta metode, kas izveido XML datnes shēmu automātiski. Kā tikko lietotājs grib ielādēt XML datni, lai izveidotu kartēšanu, rīks piedāvā izveidot XML shēmu (sk. 5.1.2.1. att.).



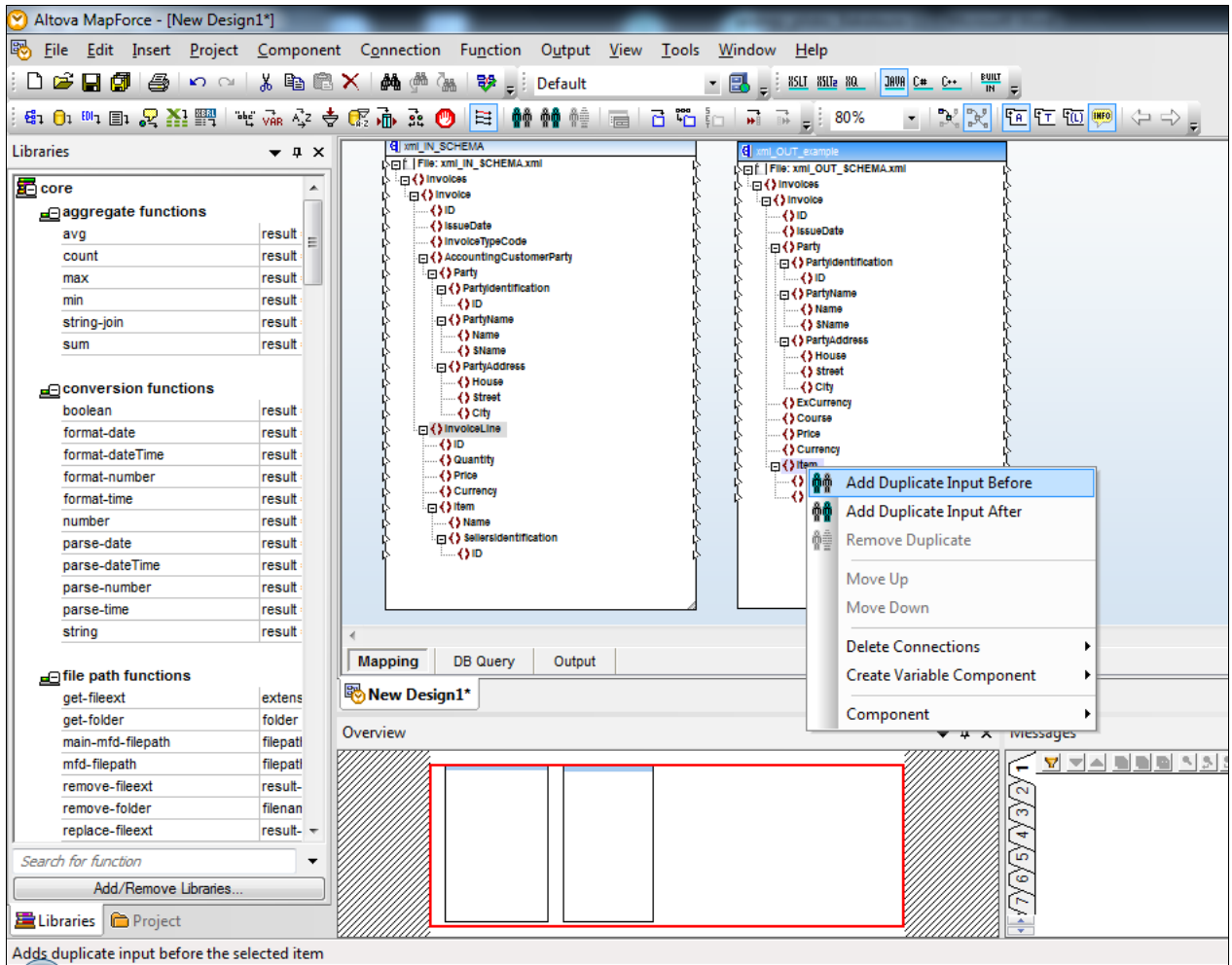
5.1.2.1. att. XML shēmas izveide

Pēc katras XML shēmas, ieejas un izejas, izveides var sākt veidot kartēšanu. Kartēšanu veido pārvelkot vajadzīgo elementu no ieejas datnes uz izejas datnes attiecīgo elementu. Tiek izmantot „drag and drop” tehnika (sk. 5.1.2.2. att.).



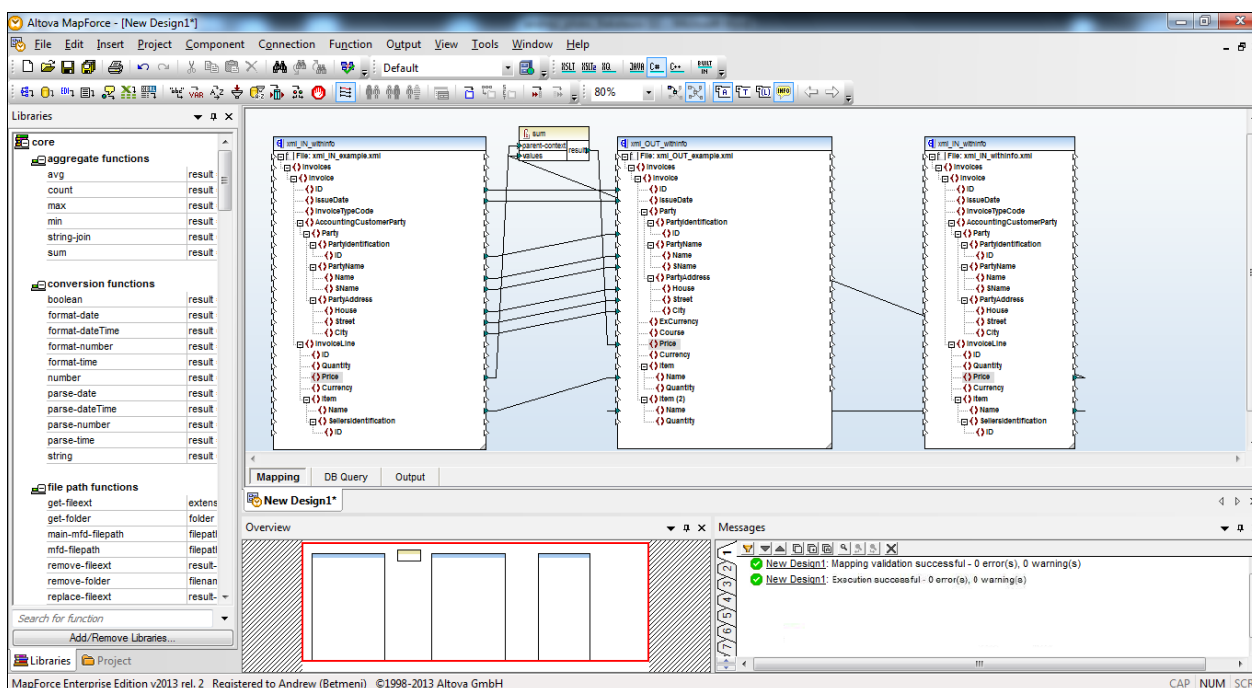
5.1.2.2. att. Kartēšanas izveide

Autors neatrada pietiekami viegla veida MapForce rīkā, lai izvairītos no problēmas ar vienādiem elementu vārdiem, bet bija atrasts veids tehniskajā dokumentācijā [13], kādā veidā to var realizēt, kaut gan tas izskatās ļoti neveikli. Ir jāizveido dublikāta elements izejas shēmā (sk. 5.1.2.3. att.), tad ir jāimportē ieejas shēma vēl vienu reizi, un dublicējošos elementus ir jānokartē dubltelementā, kuru lietotājs ir izveidojis izejas shēmā.



5.1.2.3. att. Dublikāta elementa izveide izejas shēmā

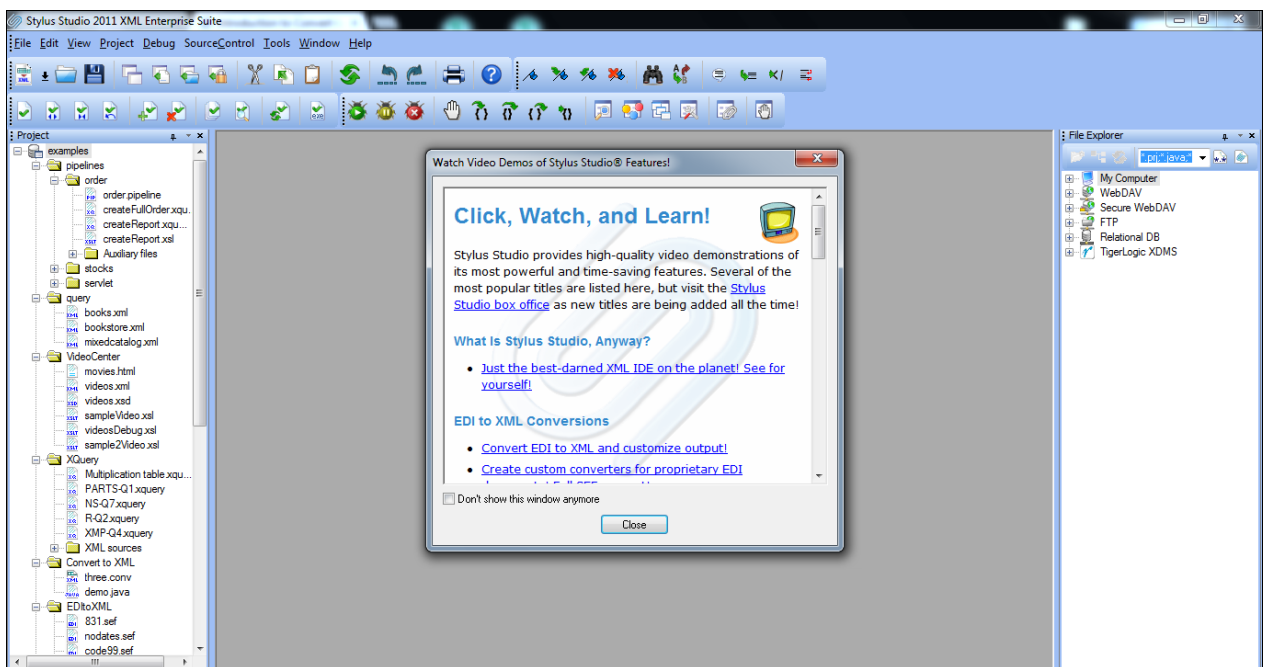
Ja ir nepieciešamība ir iespēja ieviest gan konstantes, gan plaša klāsta matemātiskos aprēķinus un datu pārveidojumus ar iebūvēto funkciju palīdzību. Pieejamās funkcijas ir pieejamas kreisajā rīku joslā, un tās ir nepieciešams tikai pārvilkt darba virsmā un pareizi savienot kartēšanu. Pēc visu noteikumu ieviešanas ir jāspiež tikai viena poga „Output”, un rīks pārbauda vai shēmas un kartēšanas ir atbilstošas uzdotajai shēmai, un vai datu transformēšana ir sanākusi veiksmīga (sk. 5.1.2.4. att.). Izvada dokuments tiek novietots norādītajā direktoriņā.



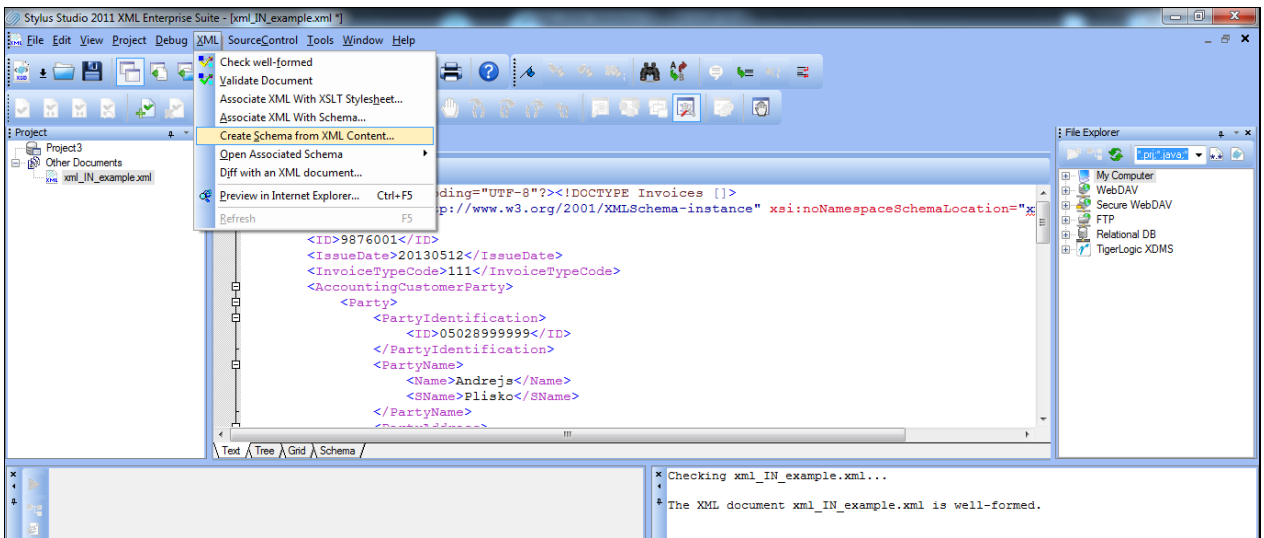
5.1.2.4. att. Kartēšanas pārbaude.

5.1.3. Data Integration Suite transformācijas

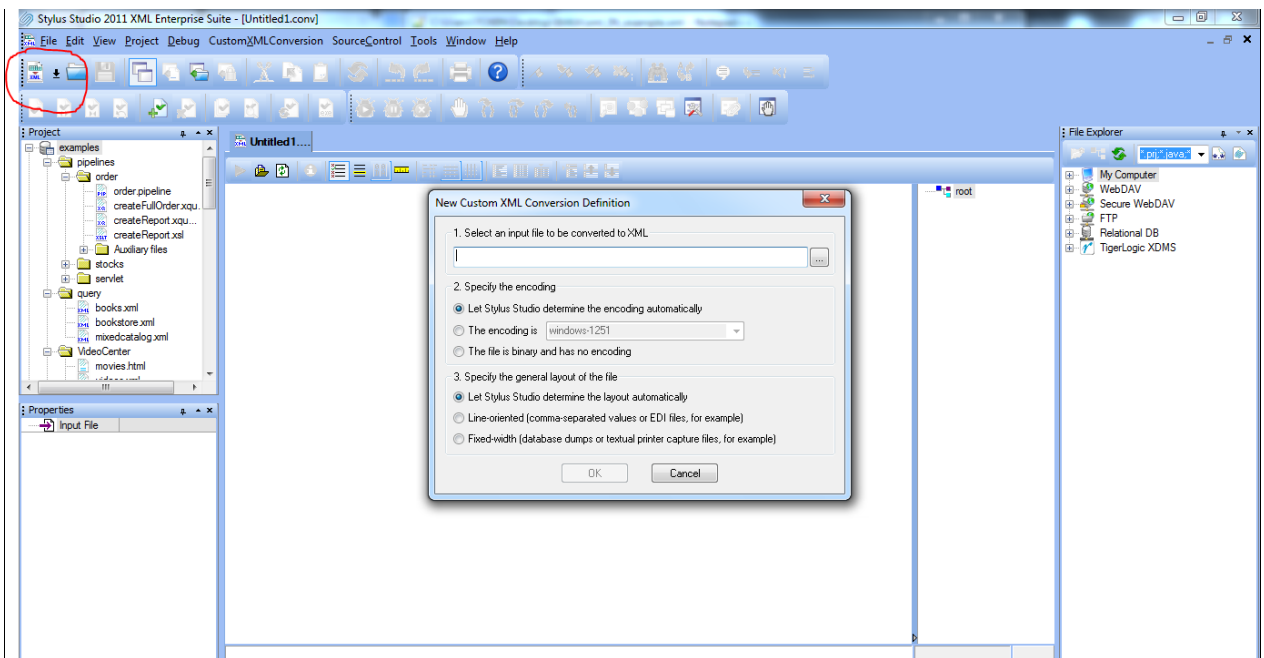
Data Integration Suite, kā jau bija minēts agrāk sastāv no vairākiem elementiem, kuros ietilps Stylus Studio ar vairākiem papildinājumiem, kuri nodrošina operācijas caur grafisko saskarni un integrētu koda korekcijas rīku. Pie ieejas Stylus Studio, rīks piedāvā plašu klāstu ar pamācībām un labākajām praksēm, kuras ir ierakstītas video pamācībās (sk. 5.1.3.1. att.), šāda iespēja nebija nevienai iepriekš aprakstītai sistēmai. Pie XML datnes ielādes rīkam ir iespēja automātiski izveidot XML shēmu atvērtaī datnei, lai turpmāk izmantotu to transformāciju veidošanā (sk. 5.1.3.2. att. un 5.1.3.3. att.).



5.1.3.1. att. Pamācību piedāvājums

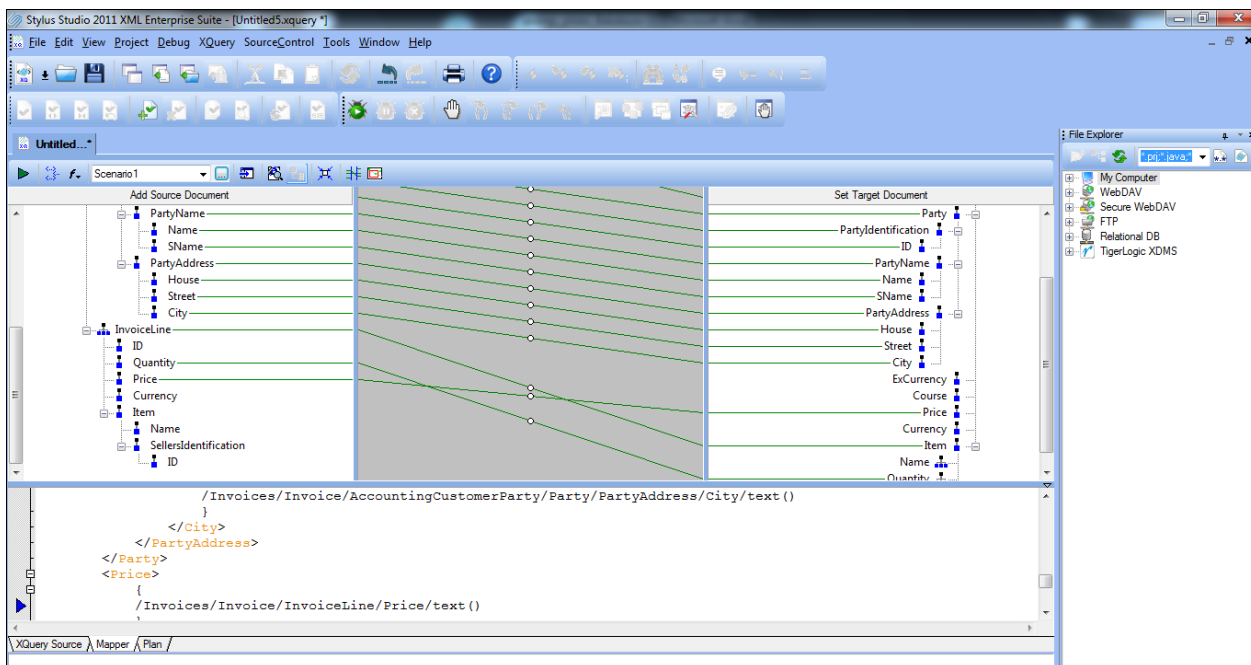


5.1.3.2. att. XML shēmas izveide Nr.1

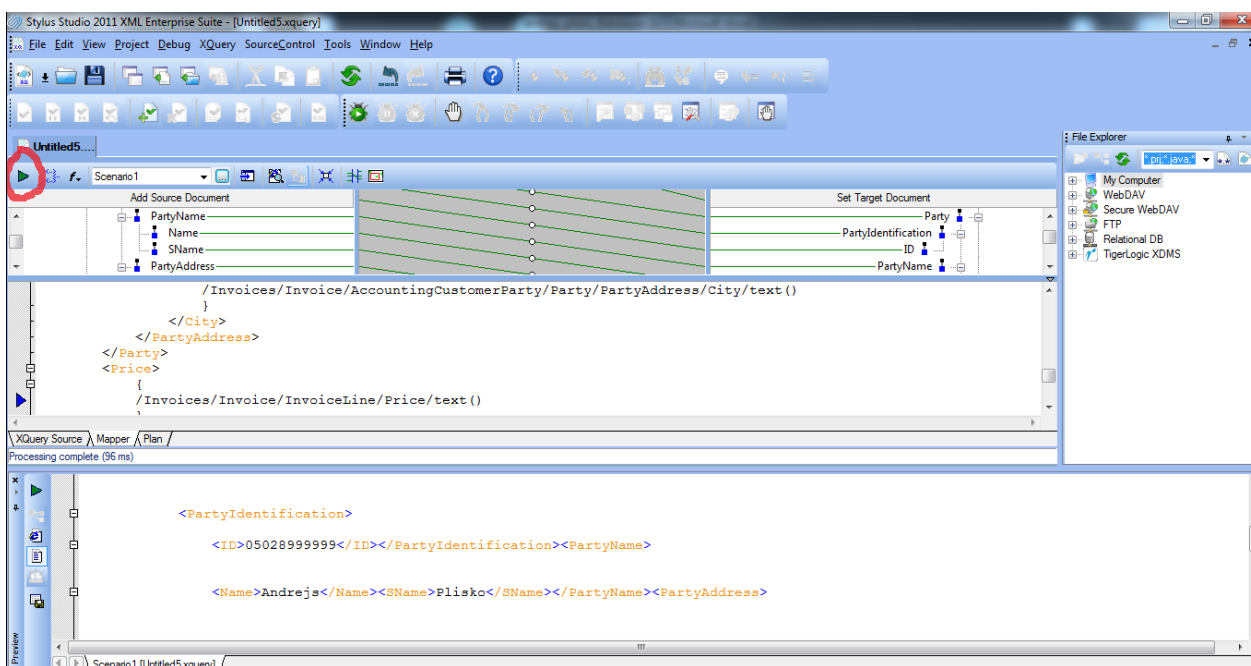


5.1.3.3. att. XML shēmas izveide Nr.2

Pēc XML shēmas izveides var sākt veidot kartēšanu. Kartēšana veidojas izvēloties ieejas XML datnes shēmu un izejas datnes XML shēmu. Pati datu transformēšana vai pārvietošana kartēšanas izveidošanā notiek vienkārši pārvelkot elementus no ieejošās datnes elementa uz attiecīgo izejošās datnes elementu (sk. 5.1.3.4. att.). Kartēšanai ir iespēja uzreiz apskatīties izvadi, kā tas redzams attēlā 5.1.3.5., un uzreiz apskatīt XQuery kodu, kurš ir apskatāms 6. pielikumā.



5.1.3.4. att. XML shēmas izveide Nr.2



5.1.3.5. att. Izvada informācijas aplūkošana

Sistēma pēc visu noteikumu ieviešanas darbojas automātiski. Ir iespēja ieviest dublikāta elementus ar speciālu funkciju, kas ir attiecināma uz ieejas datnes attiecīgiem elementiem. Ir iespēja ievietot izejas datnes elementos noklusētās vērtības vai vērtības no datu bāzes kā arī veikt

nepieciešamās datu pārveidošanas, atvasināšanas un visdažādākos aprēķinus, kā arī ieviest savus noteikumus, kurus būs nepieciešams ievadīt specializētajos pieprasījumos FLWOR skriptā.

SECINĀJUMI

Visas transformācijas izdevās sekmīgi veikt un datni, kura redzama 1. pielikumā bija iespējams sekmīgi transformēt datnē, kuru var redzēt 2. pielikumā. Neskatoties uz to, ka bija vairāki elementi ar to pašu nosaukumu, visās sistēmās bija iespējams veikt nepieciešamo datu ieguvī no pareizajiem elementiem. Visas sistēmas atbalsta konstanšu ieviešanu, kuras palīdz nodrošināt transformēšanu kardinalitātēs 0:1 un 0:N. Galvenā problēma bija ieviest noteikumu, kuri atbalstītu 1:N transformācijas, kad nepieciešama datu sadalīšana. Bet visas sistēmas tika ar to galā sekmīgi, kaut gan TSIM autoram tas likās vienkāršāk, jo autoram ir pietiekama pieredze darba vietā ar darbībām ar kodu, un TSIM vairāk atbilda koda labošanai vai pielabošanai nekā pārējās sistēmas. Gribētos piebilst, ka TSIM sistēma ir visvecākā no dotajām, un daudzi risinājumi tajā ir novecojuši, kā arī pietiekama daudzuma dokumentācijas sistēmai nav pieejama plašai publikai. Altova sistēmai un Data Integration Suite savukārt, dokumentācijas bija pietiekami daudz gan izstrādātāju mājaslapa, gan globālajā tīmeklī. Data Integration Suite kaut arī ir ļoti plaša funkcionalitāte, taču tas diezgan traucē un nav vienmēr intuitīvi skaidrs, kur meklēt nepieciešamo funkcionalitāti, tāpēc bieži ir jāgriežas pēc dokumentācijas un video pamācībām, savukārt Altova rīki ir maksimāli intuitīvi skaidri un autoram, kurš saskārās ar Altova rīkiem pirmo reizi, nebija grūtību atrast nepieciešamo funkcionalitāti. Bet TSIM sistēmai ar intuitīvo saprašanu ir lielas problēmas, tas ir diezgan vecs rīks, iespējams tieši tāpēc jaunam lietotājam to apgūt ir ļoti sarežģīti. Kaut gan Data Integration Suite ir dārgāks par Altova rīkiem, to būtu vērts iegādāties un uzstādīt, ja ir pārlicība par to, ka ir nepieciešamība par funkcionalitāti, kuras nav Altova rīkos, piemēram atbalstāmo datņu formātu automātiska transformēšana XML datnēs pēc rīka noklusētiem noteikumiem.

Izanalizējot dotos rīkus un savas sistēmas ieviešanas iespējas autors ir nolēmis, ka TSIM sistēma, kaut arī ir ļoti labi piemērota gan transformāciju koda labošanai, gan transformāciju koda rakstīšanai un ģenerēšanai, bet vienkāršajiem operatoriem ar to darboties būtu pagrūti. Taču pārējie rīki, Altova komplekts un Data Integration Suite, ir ļoti labi piemēroti tam, lai ar tiem darbotos cilvēki ar nelielām zināšanām programmēšanā. Ar savas sistēmas veidošanu var ņemt vērā tikai cenu, kura nav maza un var atmaksāties ļoti lielā laikā.

NOBEIGUMS

Strukturēto datu apmaiņai un to transformācijai ir ļoti nopietna loma pašlaik pasaulē. Pieprasījums pēc elektroniskā datu apmaiņas veida un to apjoms aug ļoti straujos tempos un jau vairs ne gadu mērogā, bet gan mēnešu mērogā. Pakalpojumi, kas ir saistīti ar elektronisko datu apmaiņu arī palielina savu tirgu kopā ar pieprasījuma pēc elektronisko datu apmaiņas pieauguma, piemēra aug IT speciālistu pieprasījums, kuri var nodrošināt nepieciešamo sistēmu darbu un funkcionalitāti.

Darba praktiskajā daļā analizētās elektronisko datu transformācijas un transporta sistēmas dod priekšstatu par konkrēto sistēmu funkcionalitāti, iespējām un cenām. Pēc dotām cenām lasītajam jārodas priekšstats par to, kura sistēma labāk piemērota lietotāja mērķu realizācijai. Rupji sakot, ja lietotājs vēlas izveidot pats savu elektronisko datu noformēšanas sistēmu, viņam jāreķinās ar to, ka tā sevi atmaksās labākajā gadījumā pēc vairāk nekā 10 gadiem. Laika aprēķini, ko izmantoja autors bija aptuveni un balstījās uz vidējo speciālistu pieredzi IT jomā. Runājot par grafiskajiem rīkiem, kuri ir iekļauti jau gatavajās sistēmās, tie ļoti atvieglo darbu ar transformāciju veidošanu un ieviešanu ražošanā, bet tos nevajadzētu uztvert vairāk nekā rīku. Visu, uz ko ir spējīgi grafiskie noformēšanas rīki, var uzrakstīt koda veidā, iespējams pat optimizētāk un īsāk nekā to dara šie rīki, taču parastajam lietotājam vai operatoram, kuram ir maz zināšanu par programmēšanu, koda rakstīšanu un noformēšanu ir nesalīdzināmi vieglāk rīkoties grafiskajā vidē.

Rakstot šo darbu, autors ir sasniedzis darba mērķi un devis priekšstatu par izdevīgāko XML formāta datņu noformēšanas sistēmas variantu. Kaut gan neizdevās atrast gatavu risinājumu, kurš nodarbosies tikai un vienīgi ar XML formāta datņu saņemšanu, transformēšanu un nosūtīšanu.

Darba laikā autors ir izanalizējis un izpētījis vairāku sistēmu funkcionalitāti un iespējas, kā arī veicis datņu transformācijas šajās sistēmās. Ir izveidots sistēmas prototips, kurš varētu būt ieviests autora darba vietā, taču izmaksas un laika patēriņš šim nolūkam ir ļoti nopietni. No darbā aplūkotajiem risinājumiem, autors uzskata, ka visizdevīgākais variants tomēr ir Altova kompānijas rīku komplekts. Tas izrādījās vissaprotamākais iestatīšanā un transformāciju veidošanā, kā arī transformāciju koda noformēšanā.

IZMANTOTĀ LITERATŪRA UN AVOTI

- [1] European Commission, Communication From The Commission To The European Parliament, The Council, The European Economic And Social Committee And The Committee Of The Regions. Reaping The Benefits Of Electronic Invoicing For Europe, COM(2010) 712 final, 2 December 2010
- [2] EDIFACTORY - The EDIFACT resource
- [3] The European Electronic Invoicing Experts
- [4] "Elektronisch factureren" (in Dutch); tiešsaiste: <http://www.rijksoverheid.nl>.
- [5] B. Koch, Billentis, E-Invoicing / E-Billing in Europe and abroad From evolution to revolution, March 21, 2012
- [6] Graphical Data Mapping, Conversion, and Integration Tool; tiešsaiste: <http://www.altova.com/mapforce.html>
- [7] Data Integration Suite; tiešsaiste: <http://www.developerfusion.com/article/84372/data-integration-suite/>
- [8] CVO Recruitment mājaslapa; tiešsaiste: http://www.cvor.lv/jaunumi/it-algas_latvija/
- [9] CV-Online mājaslapa; tiešsaiste: <http://www.cv.lv/darba-sludinajumi/informacijas-tehnologijas>
- [10] Jānis Bērziņš. Strukturētu datu transformācijas E-Komercijā. Maģistra darbs Datorzinātņu maģistra zinātniskā grāda iegūšanai, Latvijas Universitāte, Rīga, 2012
- [11] Ģ. Karnītis. Informācijas Sistēmu Integrācijas Problēmas. Promocijas darbs Datorzinātņu doktora zinātniskā grāda iegūšanai, Latvijas Universitāte, Rīga, 2004
- [12] XSD/XML Schema Generator; tiešsaiste: <http://www.freeformatter.com/xsd-generator.html>
- [13] Altova MapForce dokumentācija; tiešsaiste: http://manual.altova.com/mapforce/mapforce-basic/index.html?mffduplicating_input_items.htm

PIELIKUMI

1. pielikums. Ieejas datu formāts

```
<?xml version="1.0" encoding="UTF-8"?>
<Invoices>
  <Invoice>
    <ID>9876001</ID>
    <IssueDate>20130512</IssueDate>
    <InvoiceTypeCode>111</InvoiceTypeCode>
    <AccountingCustomerParty>
      <Party>
        <PartyIdentification>
          <ID>05028999999</ID>
        </PartyIdentification>
        <PartyName>
          <Name>Andrejs</Name>
          <SName>Plisko</SName>
        </PartyName>
        <PartyAddress>
          <House>20</House>
          <Street>Chaka</ Street >
          <City>Riga</ City >
        </PartyAddress>
      </Party>
    </AccountingCustomerParty>
    <InvoiceLine>
      <ID>1</ID>
      <Quantity>2</Quantity>
      <Price>24.58</Price>
      <Currency>LS</Currency>
      <Item>
        <Name>Teapot 079</Name>
        <SellersIdentification>
          <ID>135853</ID>
        </SellersIdentification>
      </Item>
    </InvoiceLine>
    <InvoiceLine>
      <ID>2</ID>
      <Quantity>1</Quantity>
      <Price>99.96</Price>
      <Currency>LS</Currency>
      <Item>
        <Name>Office Chair 197</Name>
        <SellersIdentification>
          <ID>234110</ID>
        </SellersIdentification>
      </Item>
    </InvoiceLine>
  </Invoice>
</Invoices>
```

```
        </Item>
    </InvoiceLine>
    <InvoiceLine>
        <ID>3</ID>
        <Quantity>3</Quantity>
        <Price>42.94</Price>
        <Currency>LS</Currency>
        <Item>
            <Name>Chair 084</Name>
            <SellersIdentification>
                <ID>234110</ID>
            </SellersIdentification>
        </Item>
    </InvoiceLine>
    <InvoiceLine>
        <ID>4</ID>
        <Quantity>1</Quantity>
        <Price>124.96</Price>
        <Currency>LS</Currency>
        <Item>
            <Name>Office desk 002</Name>
            <SellersIdentification>
                <ID>234110</ID>
            </SellersIdentification>
        </Item>
    </InvoiceLine>
</Invoice>
</Invoices>
```

2. pielikums. Izejas datu formāts

```
<?xml version="1.0" encoding="UTF-8"?>
<Invoices>
  <Invoice>
    <ID>9876001</ID>
    <IssueDate>20130612</IssueDate>
    <Party>
      <PartyIdentification>
        <ID>0502899999</ID>
      </PartyIdentification>
      <PartyName>
        <Name>Andrejs</Name>
        <SName>Plisko</SName>
      </PartyName>
      <PartyAddress>
        <House>20</House>
        <Street>Chaka</Name>
        <City>Riga</SName>
      </PartyAddress>
    </Party>
    <ExCurrency>LVL</ExCurrency>
    <Course>0.5496</Course>
    <Price>1275.73</Price>
    <Currency>USD</Currency>
    <Item>
      <Name>Teapot 079</Name>
      <Quantity>2</Quantity>
      <Name>Office Chair 197</Name>
      <Quantity>1</Quantity>
      <Name>Chair 084</Name>
      <Quantity>3</Quantity>
      <Name>Office desk 002</Name>
      <Quantity>1</Quantity>
    </Item>
  </Invoice>
</Invoices>
```

3. pielikums. Ieejas datnes XML shēma, XSD formātā

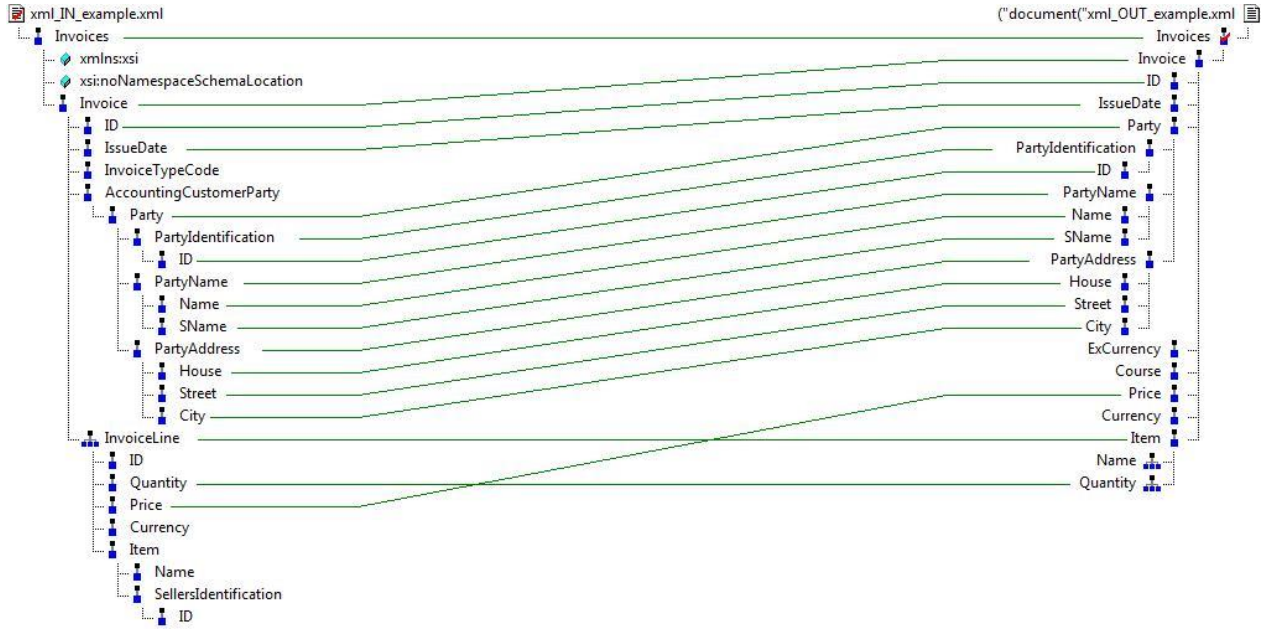
```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Invoices">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Invoice">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="ID"/>
              <xs:element type="xs:string" name="IssueDate"/>
              <xs:element type="xs:string" name="InvoiceTypeCode"/>
              <xs:element name="AccountingCustomerParty">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Party">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="PartyIdentification">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element type="xs:string" name="ID"/>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="PartyName">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element type="xs:string" name="Name"/>
                                <xs:element type="xs:string" name="SName"/>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="PartyAddress">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element type="xs:string" name="House"/>
                                <xs:element type="xs:string" name="Street"/>
                                <xs:element type="xs:string" name="City"/>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<xs:element name="InvoiceLine">
  <xs:complexType>
    <xs:sequence>
      <xs:element type="xs:string" name="ID"/>
      <xs:element type="xs:string" name="Quantity"/>
      <xs:element type="xs:string" name="Price"/>
      <xs:element type="xs:string" name="Currency"/>
      <xs:element name="Item">
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string" name="Name"/>
            <xs:element name="SellersIdentification">
              <xs:complexType>
                <xs:sequence>
                  <xs:element type="xs:string" name="ID"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

4. pielikums. Ieejas datnes XML shēma, XSD formātā

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Invoices">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Invoice">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="ID"/>
              <xs:element type="xs:string" name="IssueDate"/>
              <xs:element name="Party">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="PartyIdentification">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="ID"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="PartyName">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="Name"/>
                          <xs:element type="xs:string" name="SName"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="PartyAddress">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:string" name="House"/>
                          <xs:element type="xs:string" name="Street"/>
                          <xs:element type="xs:string" name="City"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element type="xs:string" name="ExCurrency"/>
              <xs:element type="xs:string" name="Course"/>
              <xs:element type="xs:string" name="Price"/>
              <xs:element type="xs:string" name="Currency"/>
              <xs:element name="Item">
                <xs:complexType>
```


5. pielikums. Stylus Studio kartēšana



6. pielikums. Stylus Studio kartēšans XQuery kods

```
<Invoices>
{
/Invoices/text()
}
<Invoice>
{
/Invoices/Invoice/text()
}
<ID>
{
/Invoices/Invoice/ID/text()
}
</ID>
<IssueDate>
{
/Invoices/Invoice/IssueDate/text()
}
</IssueDate>
<Party>
{
/Invoices/Invoice/AccountingCustomerParty/Party/text()
}
<PartyIdentification>
{
/Invoices/Invoice/AccountingCustomerParty/Party/PartyIdentification/text()
}
<ID>
{
/Invoices/Invoice/AccountingCustomerParty/Party/PartyIdentification/ID/text()
}
</ID>
</PartyIdentification>
<PartyName>
{
/Invoices/Invoice/AccountingCustomerParty/Party/PartyName/text()
}
<Name>
{
/Invoices/Invoice/AccountingCustomerParty/Party/PartyName/Name/text()
}
</Name>
<SName>
{
/Invoices/Invoice/AccountingCustomerParty/Party/PartyName/SName/text()
}
</SName>
</PartyName>
```

```
<PartyAddress>
  {
  /Invoices/Invoice/AccountingCustomerParty/Party/PartyAddress/text()
  }
  <House>
  {
  /Invoices/Invoice/AccountingCustomerParty/Party/PartyAddress/House/text()
  }
</House>
  <Street>
  {
  /Invoices/Invoice/AccountingCustomerParty/Party/PartyAddress/Street/text()
  }
</Street>
  <City>
  {
  /Invoices/Invoice/AccountingCustomerParty/Party/PartyAddress/City/text()
  }
</City>
</PartyAddress>
</Party>
<Price>
  {
  /Invoices/Invoice/InvoiceLine/Price/text()
  }
</Price>
<Item>
  {
  /Invoices/Invoice/InvoiceLine/text()
  }
  <Quantity>
  {
  /Invoices/Invoice/InvoiceLine/Quantity/text()
  }
</Quantity>
</Item>
</Invoice>
</Invoices>
```

DOKUMENTĀRĀ LAPA

Bakalaura darbs: AUTOMATIZĒTA XML FAILU TRANSFORMĀCIJA E-RĒĶINU
NOFORMĒŠANAS SISTĒMĀ

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Andrejs Plisko _____

(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto bakalaura darbu un atzīstu to par **piemērotu/nepiemērotu** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu bakalaura studiju programmas gala pārbaudījuma komisijas sēdē.

Darba vadītājs: _____

(Vadītāja paraksts)

Darbs iesniegts **Datorikas Fakultātē** _____.

(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: Ārija Sproģe _____

(Metodiķes paraksts)

Recenzents: Hd. Datorzinātņu habil. doktors, prof. Audris Kalniņš _____

(Recenzenta paraksts)

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____, vērtējums _____

(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____

(Sekretāra paraksts)