

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

STRUKTURĒTU DATU
TRANSFORMĀCIJAS
E-KOMERCIJĀ

MAGISTRA DARBS

Autors: Jānis Bērziņš

Stud. apl. Nr. jб10003

Darba vadītājs: Dr. dat. Ģirts Karnītis

RĪGA 2012

ANOTĀCIJA

Maģistra darbā pētīti datu apmaiņas un pārveidošanas jautājumi e-komercijas lietojumos, apzinātas ar datu transformāciju saistītās problēmas un ieskicēti iespējamie risinājumi, kā arī izvērtēti transformāciju veidošanas rīki. Vispārīgā gadījumā datu pārveidošanai jānodrošina divus mērķus, pirmkārt, ieejas un izejas datu formātiem jābūt piemērotiem izmantošanai atbilstošajā informatīvajā sistēmā un otrkārt, datiem jā saglabā semantiskā jēga, t.i., vaicājumiem ieejas un izejas datos jāsniedz saturiski saskanīgas atbildes. Grūtības rodas apstākļi, ka ieejas datu instancei var radīt vai nu daudz, vai nevienu atbilstošu izejas datu instanci. Veidojot darbu, autors iepazinās ar datu transformāciju teoriju un praktiski realizēja vairākas transformācijas ar dažādiem rīkiem, lai analizētu to piemērotību datu pārveidošanas uzdevumu risināšanā.

ABSTRACT

Master thesis title: Data exchange format transformations for e-commerce.

In this paper, an author analyzes data exchange and transformation related issues which arise in e-commerce. The author identifies and describes various problems and potential solutions. Several data transformation tools have been analyzed and their suitability was assessed. In general data transformations must ensure that input and output data is in format which is suitable for use in respective information system and queries on input and output data instances give consistent answers. Difficulties with data transformation mainly arise because for a given source instance there may be created many or none target instances which satisfy data transformation objective. The author got acquainted with data transformation theory and in practice implemented several transformations with different tools for the analysis of their suitability for data conversion tasks.

AUTOREFERĀTS

Autors ir iepazinies ar e-komercijā lietotajiem strukturēto datu formātiem, datu apmaiņas principiem, nozari regulējošajiem normatīvajiem dokumentiem, piemēram, „Council Directive 2010/45/EU” [4] un pētījumiem, piemēram, „E-invoicing 2010. European market guide” [2].

Autors ir apzinājis datu transformāciju rīkus, studējis rīkiem veltītās publikācijas un instrukcijas, apguvis pētījumā aprakstīto rīku lietošanu, izveidojis sešus datu transformāciju testa piemērus ar trijiem dažādiem rīkiem un analizējis rīku piemērotību datu pārveidošanai atbilstoši darba tēmai.

Autors ir iepazinies ar specializēto literatūru, piemēram, „Data Exchange: Getting to the Core” [7], „Data Exchange: Semantics and Query Answering” [8] un citiem avotiem, kas saistīti ar datu transformācijas teoriju un patstāvīgi apguvis teorijas pamatus.

SATURA RĀDĪTĀJS

Apzīmējumu saraksts	5
Ievads	6
1. Strukturētu datu pielietojums e-komercijā.....	9
2. Datu transformācijas teorija	14
2.1 Datu transformācijas definīcija	14
2.2 Datu pārveidošanas piemērs.....	14
2.3 Teorija un prakse.....	15
3. Datu fiziskie formāti	17
3.1 Uz pozīcijām bāzēts formāts (Position Based).....	17
3.2 Elementu nosaukumu un vērtību pāri (Valuepair).....	18
3.3 Ar atdalītāju nodalītas vērtības (dinamiska garuma vērtības, CSV).....	18
3.4 XML.....	19
3.5 EDIFACT.....	20
3.6 Formātu lietojuma un transformāciju specifika	21
4. Datu transformācijas	23
1:1	23
0:1 vai 0:N.....	23
1:0 vai N:0.....	26
N:1	26
1:N.....	27
Citas datu pārveidošanas problēmas	28
5. Transformāciju veidošanas rīki.....	30
5.1 GCI TradeXpress.....	31
5.2 Altova Mapforce	35
5.3 GMC Inspire Designer	39

5.4 Axway Datamapper Builder.....	42
5.5 Kopsavilkums.....	44
Nobeigums un secinājumi.....	46
Izmantotā literatūra un avoti	48
Pielikumi.....	50
1. pielikums. Ieejas dati uz pozīcijām bāzēts formāts	50
2. pielikums. Ieejas dati pārveidoti EDIFACT formātā	50
3. pielikums. Ieejas dati atkārtoti pārveidoti ar secīgām transformācijām TEXT > EDIFACT > TEXT.....	51
4. pielikums. Ieejas dati pārveidoti XML formātā	51
5. pielikums. XML datnes šablons izmatošanai RTE kodā.....	53
6. pielikums. Transformācija RTE programmēšanas valodā	53
7. pielikums. XML datne iegūta ar GMC Inspire Designer	54
8. pielikums. Altova Mapforce transformācijas dokumentācija	55

APZĪMĒJUMU SARAKSTS

A2A (Application to Application) – sadarbība, saite vai saskarne starp informatīvajām sistēmām

B2B (Business to Business) – komerciālās attiecības starp diviem uzņēmumiem

B2C (Business to Customer) – komerciālās attiecības starp uzņēmumu un patērētāju

CSV (comma-separated values) – ar atdalītāju nodalītu vērtību teksta datnes formāts

DTD (Document Type Definition) – XML dokumenta struktūras uzdošanas veids

E2B - XML bāzēts elektronisko dokumentu standarts

EDI (Electronic Data Interchange) – elektroniska strukturētu datu apmaiņa starp organizācijām

EDIFACT – United Nations/Electronic Data Interchange For Administration, Commerce and Transport (UN/EDIFACT) – starptautisks strukturētu datu glabāšanas un apmaiņas formāts, kurš izstrādāts ar Apvienoto Nāciju Organizācijas atbalstu.

EHF (Elektronisk handeľsformat) – XML bāzēts elektronisko dokumentu standarts Norvēģijā.

ERP (Enterprise Resource Planning) – integrēta uzņēma resursu pārvaldības un plānošanas sistēma

IDoc (Intermediate Document) - SAP informatīvās sistēmas dokumentu formāts datu apmaiņai par darījumu transakcijām

XML – Extensible Markup Language

XSD (XML Schema Document) – XML dokumenta struktūras uzdošanas veids

XSLT (Extensible Stylesheet Language Transformations) – deklaratīva XML bāzēta valoda, kuru izmanto XML dokumentu transformācijai

.

IEVADS

Maģistra darbā tiks apskatīti elektronisku datu apmaiņas formāti e-komercijas lietojumos un pētītas ar formātu pārveidošanu saistītas problēmas.

Eiropas Komisija ir izstrādājusi un pieņēmusi iniciatīvu „Digitālā programma Eiropai” (*A Digital Agenda for Europe*), kura skar e-komerciju un elektroniskos norēķinus un kuras mērķis ir panākt vienotu digitālu tirgu. Iniciatīva rosina novērst normatīvo aktu pretrunas un ierobežojumus, kas kavē masveida elektronisko rēķinu un norēķinu lietošanu [1,3].

Eiropas Savienība ir pieņēmusi virkni rekomendāciju dalībvalstīm par likumdošanas sakārtošanu un viens no programmas mērķiem – panākt e-rēķinu dominēšanu tirgū līdz 2020.gadam.

Piemēram, 2010.gada 13.jūlijā Eiropas Padome ir pieņēmusi direktīvu 2010/45/EU, kas pārskata pievienotās vērtības nodokļa (PVN) piemērošanu. Direktīva stāsies spēkā 2013.gada 1.janvārī un tā prasa ES dalībvalstīm ievērot vienlīdzīgu attieksmi pret papīra un e-rēķiniem. ES dalībvalstis vairs nedrīkstēs pievienot īpašus regulējumus un prasības, piemēram, uzstājot, ka e-rēķini, tiek parakstīti ar elektronisko parakstu vai balstīti uz specifisku EDI standartu [4].

Direktīva konstatē, ka neskatoties uz to, ka elektroniskā paraksta izmantošana sakārto e-rēķinu izmantošanu atsevišķās valstīs, tomēr dažādie standarti un likumdošanas regulējumi rada problēmas starpvalstu tirdzniecībā. Direktīva nosaka, ka elektroniskajos norēķinos jāiestrādā kontroles, kas nodrošina uzticamu liecību (audit trail) nodrošināšanu starp e-rēķinu un pakalpojumu vai preču piegādi tādā pašā veidā, kā tas tiek nodrošināts pašlaik papīra rēķiniem.

Kāda ir e-rēķinu izmantošanas prakse?

E-komercijā e-rēķini tiek izmantoti jau gadiem, neskatoties uz regulējumiem, tomēr tos izmanto tikai uzņēmumi, kas novērtē e-rēķinu izmantošanas izdevīgumu. Diemžēl vairumā gadījumu e-rēķini tiek izmantoti paralēli ar papīra rēķiniem – e-rēķini un to apstrāde starp sadarbības partneru informatīvajām sistēmām ir automatizēta, kamēr papīra rēķins tiek fiziski nosūtīts vai seko piegādei un bieži kalpo tikai kā darījuma pierādījums, jo tā izmantojums ir nošķirts no informatīvajām sistēmām.

E-rēķinu izplatība pasaulē ir redzama 1.attēlā, bet izplatība Eiropā 2.attēlā. Kā apgalvo avots [2] e-rēķinu izmantošanā dominē Eiropa un Ziemeļamerika, neskatoties uz to, ka e-rēķinu īpatsvars (e-invoicing adoption) starp visiem rēķiniem ir samērā zems tomēr ar stabilu pieguma tendenci.

Country	% of e-invoicing market	E-invoicing adoption (2008 estimate)
Europe	56%	4-15%
North America	35%	3-10%
Asia-Pacific	7%	[unknown]

Table 1: Relative size of e-invoicing market and adoption in three major global areas (Source: SWIFT)

1. att. e-rēķinu tirgus daļa un īpatsvars pasaulē¹

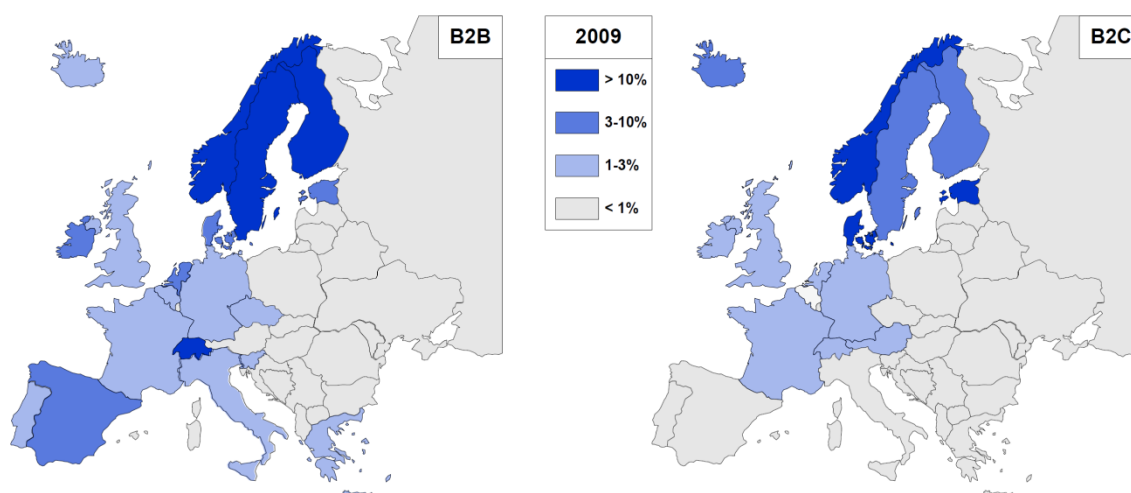


Figure 7: E-invoicing adoption in Europe (source: Billentis)

2. att. e-rēķinu izplatība Eiropā²

Igaunija B2C apgabalā ir Eiropas līderis ar 22% e-rēķinu īpatsvaru no kopējo rēķinu skaita, bet Latvija minētajā pētījumā [2] nav pieminēta.

E-rēķinu salīdzinoši zemā izplatība Latvijā, lielais tirgus potenciāls, Eiropas Savienības iniciatīvas un atbilstošā atbalsta programma pamudināja darba autoru veikt pētījumu strukturēto datu apmaiņas un datu transformācijas jomā.

Autora darbs ir saistīts maģistra darba tēmu un tā ir papildus motivācija, lai iedziļinātos datu transformācijas jautājumos, iepazītos ar teoriju, apzinātu iespējamās ar datu pārveidošanu saistītās problēmas, noskaidrotu to cēloņus un iespējamās

¹ Avots [2], atsauce uz SWIFT, SWIFT e-invoicing consultation, October 2008

² Avots [2], atsauce uz Billentis, e-invoicing market report 2009

risinājumus. Autors praksē lieto vienu no praktiskā pētījumā daļā apskatītajiem datu transformāciju veidošanas rīkiem, tomēr viņam ir šaubas par rīka atbilstību mūsdienu prasībām. Alternatīvu rīku izpēte un salīdzināšana ļaus apzināties izmatotā rīka iespējas un trūkumus.

Darba struktūra:

1.nodaļā ir aprakstīti e-komercijas darījumu procesi un elektroniskās informācijas loma tajos, minēti datu apmaiņas standarti.

2.nodaļā ir dots ieskats datu transformācijas teorijā

3.nodaļā ir aplūkoti izplatītākie e-komercijā lietotie fiziskie datu formāti, to lietošanas pamatprincipi.

4.nodaļā ir detalizēti aprakstītas ar datu pārveidošanu saistītās problēmas un sniegti iespējamie risinājumi.

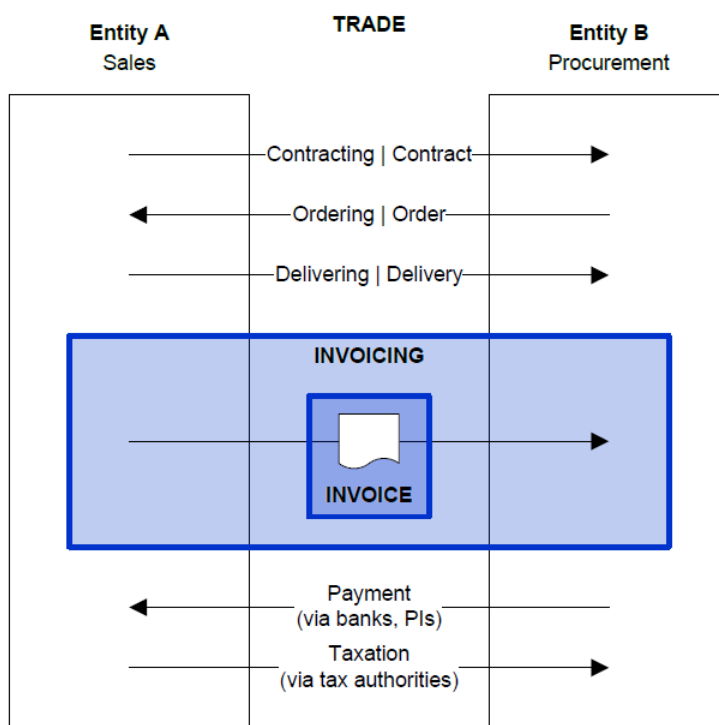
5.nodaļa satur praktiskā pētījuma rezultātus, kurā autors analizē datu transformācijas rīku piemērotību.

1. STRUKTURĒTU DATU PIELIETOJUMS E-KOMERCIJĀ

Autors darbā atsauksies un aplūkos e-rēķinus (elektroniskos rēķinus), tomēr strukturētas informācijas formāti un to transformācijas paņēmieni ir kopīgi vairumam elektronisko dokumentu, kurus lieto e-komercijā, piemēram, tie būs piemēroti arī pasūtījumam vai produktu katalogam.

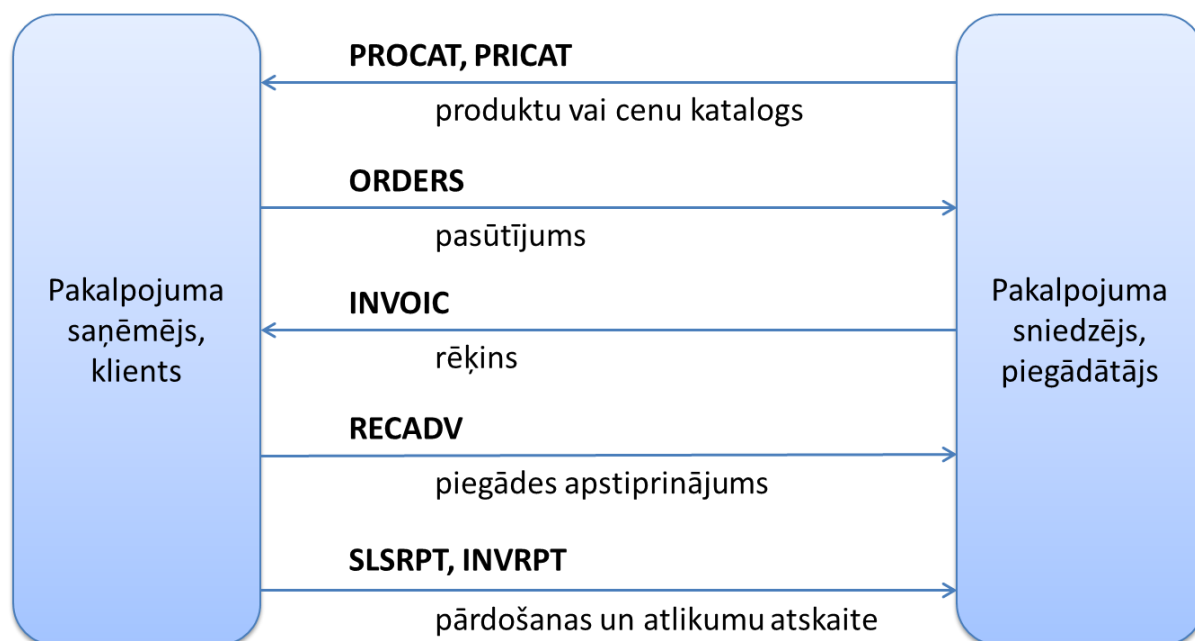
Lai pētītu datus un to transformācijas būtu jāsaprot kā tie rodas, kā un kāpēc notiek to pārveidošana un kā tie tiek pārsūtīti. Šajā nodaļā autors aplūkos komercijas darījumu procesu, elektroniskas informācijas veidus un ieguvumus no elektroniskas datu apmaiņas.

Rēķinu apstrādes aktivitātes parasti ir tikai daļa no daudz plašāka darījuma procesa, kas iekļauj pasūtījuma izveidošanu un apstiprināšanu, pasūtījuma izpildi, piegādi un maksājumu. Šo procesu no pircēja viedokļa mēdz saukt par purchase-to-pay (no pirkuma izdarīšanas brīdim līdz apmaksai), bet no pārdevēja viedokļa par order-to-cash (no pasūtījuma saņemšanas brīža līdz apmaksai). Visas šīs aktivitātes tikai kopumā apraksta komercijas procesu. Jāatzīmē, ka no darījumu procesu viedokļa rēķins nav izolēts dokuments, bet gan cieši saistīts ar citiem procesiem vai ir kāda cita procesa izejas vai ieejas dati.



3. att. Tipisks tirdzniecības process [2]

Tirdzniecības partneri var apmainīties ne tikai ar elektroniskiem rēķiniem un 4.attēlā redzami iespējamo elektronisko dokumentu veidi un plūsma. 4.attēlā dotie dokumentu veidi atbilst EDIFACT datu apmaiņas komplektiem (sk. 8. att.).



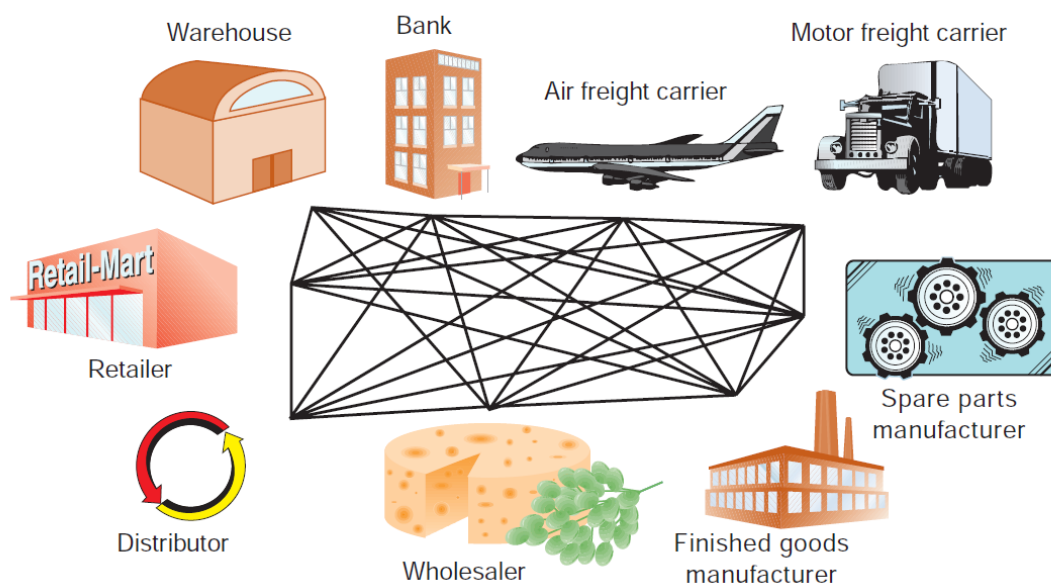
4. att. Elektronisko dokumentu veidi

Kā rodas strukturēti dati e-komercijas vajadzībām?

- 1) Dati tiek eksportēti no uzņēmuma informatīvās sistēmas un tieši vai netieši tiek nosūtīti tirdzniecības partnerim to vēlākam importam atbilstošajā informatīvajā sistēmā.
- 2) Dati tiek eksportēti no uzņēmuma informatīvas sistēmas, lai tos nosūtītu sadarbības partnerim, kas veiks rēķinu vizualizāciju un izplatīšanu (e-mail vai drukātā veidā) uzņēmuma klientiem.
- 3) Uzņēmums veic ienākošo rēķinu skenēšanu un atpazīšanu, ar vēlāku iespēju veikt datu importu informatīvajā sistēmā.

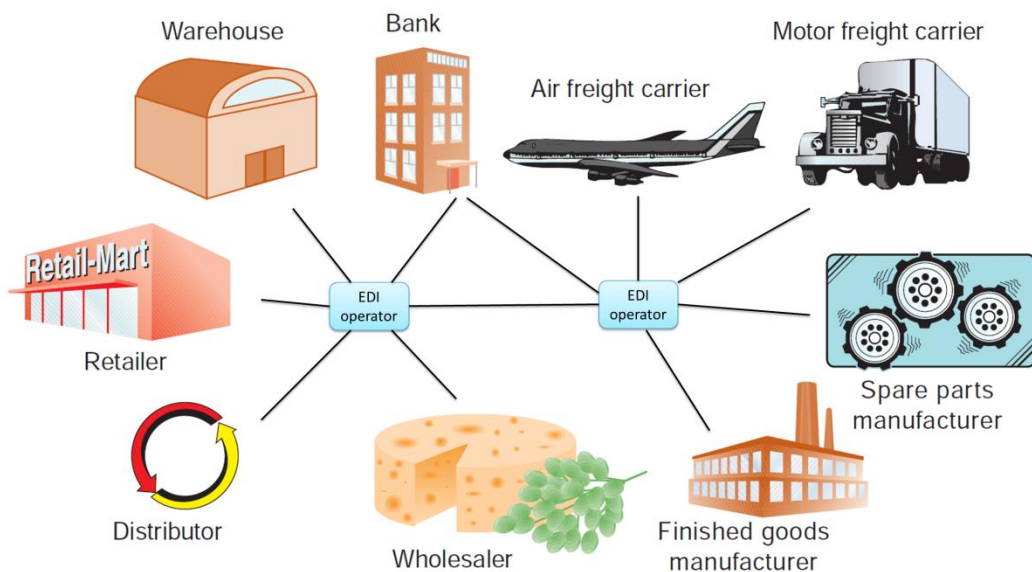
Kā notiek apmaiņa ar datiem?

Tirdzniecības partneri var apmainīties ar datiem tieši, tomēr šāda sadarbība ir viegli realizējama un pārvaldāma tikai gadījumā, ja ir iesaistīts ierobežots sadarbības partneru skaits.



5. att. Tieša elektronisku datu apmaiņa [5]

Gadījumos, kad sadarbības partneru skaits ir liels, parasti tiek izmantoti EDI operatoru pakalpojumi (sk. 6. att.). Latvijā autoram ir zināmi 3 šādi uzņēmumi - Edisoft, Telema un Itella, tomēr to varētu būt vēl vairāk. Datu pārsūtīšanas mehānisms ir sekojošs: klients vienojas par datu sūtīšanas un saņemšanas pakalpojumiem ar kādu no EDI operatoriem. Klienta sadarbības partneri elektroniskos datus saņem no tā paša vai cita operatora, ar kuru tam ir sadarbības līgums. EDI operatori vienojas par datu apmaiņu savā starpā. Tādējādi starptautiskajos darījumos datu apmaiņai starp partneriem ir būtiski un pietiekami, lai pastāv datu pārsūtīšanas iespēja starp iesaistītajiem EDI operatoriem.



6. att. Netieša elektronisku datu apmaiņa

Tirgū ir neskaitāmi daudz uzskaites, grāmatvedības, noliktavu pārvaldības sistēmu. Autoram būtu grūtības nosaukt visas Latvijā lietotās informatīvās sistēmas, bet situācija ārvalstīs nemaz nav vienkāršāka. Katrai no informatīvajām sistēmām ir sava vēsture, datu apstrādes līdzekļi un formāti, un praksē nav iespējams realizēt sadarbības scenāriju tām visām ar katru. Tas radījis nepieciešamību uzņēmējiem vienoties par datu pārsūtīšanas standartiem, kas ir ārkārtēji svarīgi, jo sevišķi starpvalstu tirdzniecībā.

Kāpēc vispār tiek veikta elektroniska datu apmaiņa (varbūt pietiktu ar e-pastu vai faksu)?

Ar neliela apjoma datiem praktiski ir iespējams apmainīties izmantojot tikai e-pasta vai faksa palīdzību, bet tas neļauj izvairīties no cilvēku kļūdām datus sagatavojot vai apstrādājot. Piemēram, iedomāsimies rēķinu par preču piegādi starp diviem uzņēmumiem uz 5 lapām ar 30 pozīcijām katrā lapā un preču kodus pozīcijās pierakstītus ar 13 simboliem. Pietiek nokļūdīties tikai vienā no $5 \times 30 \times 13 = 1950$ simboliem, lai abiem sadarbības partneriem rastos lielākas vai mazākas problēmas. Optimālā gadījuma datu pārsūtīšanai ir jābūt pilnībā automatizētai. Tas nozīmē, ka dati, kas iegūstami vienas informatīvās sistēmas eksportā, tiek pārveidoti un nosūtīti importam sadarbības partnera informatīvajā sistēmā izslēdzot cilvēka līdzdalību un novēršot kļūdīšanās iespējas.

Kādi ir ieguvumi no elektroniskas datu apmaiņas?

Elektroniska strukturētu datu apmaiņa samazina cilvēku radīto kļūdu sekas, nodrošina automatizētu datu apstrādi, atvieglo daudzveidīga sortimenta pārvaldību, samazina darījumu veikšanai nepieciešamo laiku un samazinās izmaksas, kas saistītas ar papīra rēķinu veidošanu, drukāšanu, nosūtīšanu un saņemšanu, un apstrādi.

Kāpēc būtu nepieciešams pārveidot datus?

- 1) B2B. Sadarbības partneri izmantoto dažādas informatīvās sistēmas, kuras atbalsta noteiktu formātu importa un eksporta iespējas
- 2) B2C. Klientu iespējas saņemt norēķinu informāciju no pakalpojuma sniedzēja izplatītā formātā. Piemēram, norēķinu informācija no uzņēmuma IS tiek importēta, transformēta un vajadzības gadījumā vizualizēta, piemēram PDF formātā, ar vēlāku nosūtīšanu klientiem.
- 3) Starptautiskajā tirdzniecībā. Dažādu valstu pieņemtie un popularizētie standarti, kas izraisa datu pārveidošanas nepieciešamību nosūtot datus starp valstīm.
- 4) Elektronisku dokumentu standartu daudzveidība un attīstība, jaunu populāru risinājumu kļūšana par de facto standartiem (IDoc).

Kāda ir IT nozares saikne ar elektronisko datu apmaiņu?

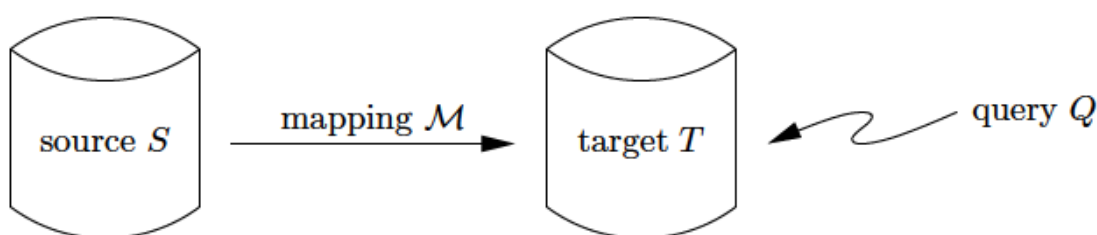
IT nozare izstrādā vai apkalpo programmatūru vai risinājumus uzņēmumiem, tāpēc speciālistiem būtu jāseko līdzī tendencēm un jābūt spējīgiem piedāvāt un realizēt risinājumus, kurus tuvākajā laikā var pieprasīt tirgus. Ja IT speciālisti nebūs gatavi vai atbilstoši sagatavoti, tad risinājumus pirksim ārpus Latvijas un iespējams, ka Latvijā izstrādātās sistēmas nebūs konkurētspējīgas vai neatbilstošas tirgus prasībām.

Manuprāt, Latvija nav pieņēmusi elektroniku rēķinu pielietošanas standartus, bet, ja nu tomēr ir, tad par to nekas publiskajā telpā nav zināms. Eiropas ziemeļu valstīs EDI standarti ir publiskoti (piemēram, Norvēģijā - <http://www.anskaffelser.no/e-handel/in-english/invoice>) un standartiem atbilstošo risinājumu izmantošana tiek stimulēta. Nedomāju, ka Latvijai vajadzētu pieņemt savu lokālu standartu, bet, ja valsts kādu no populārākajiem formātiem pasludinātu par rekomendējamu izmantošanai, tad grāmatvedības programmatūras ražotāji un uzturētāji to varētu sākt mērķtiecīgāk piedāvāt saviem klientiem. Diemžēl pašlaik tirgū dominē klientu specifiski risinājumi tādējādi, vairojot haosu.

2. DATU TRANSFORMĀCIJAS TEORIJA

2.1 Datu transformācijas definīcija

Datu transformācijas uzdevums [6,7,8] ir atrast instanci, kas atbilst mērķa shēmai (struktūrai) dotajai instancei, kas atbilst avota shēmai (struktūrai), pie dotas relāciju specifikācijas starp avota un mērķa shēmām. Mērķa instancei ir ne tikai jāattēlo ieejas dati atbilstoši mērķa shēmai, bet arī jānodrošina, ka vaicājumi (queries) avota un mērķa instancēs sniedz saskanīgas (consistent) atbildes.



7. att. Datu transformācijas uzdevums [6]:

M – kartējums, kas apraksta relācijas starp avota un mērķa shēmām,

S – avota instance, T – mērķa instance, Q – datu vaicājumi

2.2 Datu pārveidošanas piemērs

Pieņemsim, ka ir dota *ieejas* datu *instance* EDIFACT formātā (*avota shēma*), kas satur sekojošu fragmentu:

```
DTM+3:20120112:102'
```

DTM segments EDIFACT formātā (sīkāk būs aprakstīts 3.nodaļā) apraksta datumu/laiku/periodu [20]. Piemērā pirmā vērtība ir datuma tips no klasifikatora (3 = rēķina datums), otrā - datuma/laika/perioda vērtība (20120112), trešā – datuma formāts no klasifikatora (102 = CCYYMMDD).

Noteiksim XML datni kā *mērķa shēmu*, un uzskatīsim, ka rēķina datuma XPath ir Invoice/IssueDate.

Kartējumu (mapping) vienkāršoti un ar vārdiem varētu uzdot sekojoši: „Ja ieejas instance satur DTM segmentu ar klasifikatoru „3”, tad mērķa instances atbilstošajā XPath ievietojam datuma/laika/perioda vērtību, pārveidojot formātu uz CCYY-MM-DD.”

Datu transformācijas rezultāts būs **mērķa instance** - XML datne, kas saturēs, sekojošu fragmentu:

```
<Invoice>  
  <IssueDate>2012-01-12</IssueDate>  
</Invoice>
```

Acīmredzot, ka **vaicājumi** ieejas un izejas instancēs sniegs identiskas un saskanīgas atbildes, ja vien dati un atbilstošās struktūras tiks izmantotas atbilstoši definīcijai.

2.3 Teorija un prakse

Datu transformācijas mērķis ir realizēt risinājumu, kas apmierina transformācijas nosacījumus un visprecīzāk attēlo ieejas datus. Tomēr pat vienkāršā gadījumā viennozīmīgi nav iespējams atbildēt uz jautājumu vai datu transformācijas uzdevumam eksistē risinājums. Avots [6] apgalvo, ka jautājums par risinājuma eksistenci uzdotajiem kartēšanas nosacījumiem vispārējā gadījumā nav izšķirams.

Iespējams, ka risinājums datu transformācijas uzdevumam neeksistē, ja, piemēram, trūkst nosacījumiem atbilstošu ieejas datu vai arī kartēšanas nosacījumi ir nepilnīgi. No praktiskā pielietojuma viedokļa šāds apgalvojums, protams, rada vielu pārdomām, bet ne vairāk, jo, sadarbojoties ar klientu, iespējamus trūkumus var noskaidrot un novērst tādējādi pārveidojot konkrēto uzdevumu par tādu, kuram risinājums eksistē.

Datu transformācijas teorija definē terminus – universāls risinājums un universālo risinājums kodols (core of universal solution) un citus [6,7,8], kā arī pēta datu transformācijas uzdevumu risinājuma eksistences, optimāla risinājuma noskaidrošanas un ar vaicājumu atbildēšanu saistītus jautājumus. Tomēr praksē teorijas piensums ir nenozīmīgs, jo meklējot risinājumu konkrētajā klienta

specifiskajām vajadzībām datu transformācijas uzdevuma nosacījumus var saskaņoti mainīt un par labāko un optimālo risinājumu tiek uzskatīts pirmais realizētais risinājums, kurš pieņemamā laikā tiek galā ar uzdevumu.

3. DATU FIZISKIE FORMĀTI

Varētu izdalīt sekojošus fiziskos elektronisku dokumentu veidus:

Nestrukturēts elektronisks dokuments (piemēram, teksta dokuments (TXT, DOC), PDF, JPEG, TIFF, HTML un citi). Rēķins ir veidots manuāli vai automātiski informatīvajā sistēmā, atbilstoši rēķina sagatavotāja iespējām un noformēts kā elektronisks dokuments un nosūtīts elektroniski. Tomēr rēķina struktūra nav noteikta, nav būtiska vai nav saskaņota ar rēķina saņēmēju. Rēķins bez papildus apstrādes nevar tikt automātiskai apstrādāts rēķina saņēmēja informatīvajā sistēmā. Tradicionālais papīra rēķins var kļūt par nestrukturētu elektronisku rēķinu, to ieskenējot.

Strukturēts elektronisks dokuments (piemēram, EDIFACT, XML, SAP R/3 IDoc, teksta dokuments (TXT), u.c.). Rēķins tiek sagatavots ievietojot atbilstošus datus dokumentā ar noteiktu un fiksētu struktūru, kura zināma gan rēķina sūtītājam, gan saņēmējam. Atšķirībā no nestrukturētas dokumenta, rēķins var tikt apstrādāts automatizēti, jo ir zināma datu semantika.

Strukturētie dati var tikt reprezentēti sekojošos veidos:

- uz pozīcijām bāzēts formāts (Position Based) (fiksēta garuma vērtības)
- elementa nosaukuma un vērtības pāris (Valuepair)
- ar atdalītāju nodalītas vērtības (dinamiska garuma vērtības, CSV)
- minēto trīs reprezentāciju kombinācijas ļauj veidot sarežģītākus datu formātus (XML, EDIFACT, u.c.)

3.1 Uz pozīcijām bāzēts formāts (Position Based)

Lai aprakstītu dokumenta vai datu ieraksta struktūru precīzi tiek norādīta vērtību secība, to maksimālais garums un vajadzības gadījumā arī tips.

Piemēram, datne varētu saturēt sekojošus datus:

```
LINE#1    234110 Datorpele Logit M10          2    24.58
```

Datu ierakstu struktūru varētu aprakstīt sekojoši:

- Konstante, LINE# - rēķina līnija, alphanumeric, 5 simboli
- Rindas Nr., rēķina līnija kārtas skaitlis, alphanumeric, 4 simboli
- Produkta kods, alphanumeric, 7 simboli
- Produkta nosaukums, alphanumeric, 27 simboli
- Skaitis, decimal number, 5 simboli
- Cena, decimal number, 10 simboli

3.2 Elementu nosaukumu un vērtību pāri (Valuepair)

Dokuments satur ne tikai vērtības, bet arī dod datu elementu nosaukums, tāpēc struktūra ir zināma vai viegli atpazīstama.

Piemēram, datne varētu saturēt sekojošus datus:

```
LINE_NO=1  
ITEM_CODE=234110  
ITEM_DESCRIPTION=Datorpele Logit M10  
QUANTITY=2  
PRICE=24.58
```

Var, bet nav nepieciešams, uzdot datu ierakstu aprakstu, ja struktūra ir atpazīstama.

3.3 Ar atdalītāju nodalītas vērtības (dinamiska garuma vērtības, CSV)

Dokuments satur tikai vērtības, kuras atdalītas ar atdalītājsimbolu (piemēram „,” vai „;”). Ja kāda no vērtībām nav zināma vai uzdota, tā var tikt izlaista tomēr atbilstošam atdalītājam jāparādās datnē.

Datne varētu saturēt, piemēram, sekojošu informāciju:

```
1;234110;Datorpele Logit M10;2;24.58
```

Ir nepieciešams uzdot datu ierakstu aprakstu, jo bez tā datu struktūra ir tikai labākā gadījumā uzminama.

3.4 XML

Datu apmaiņā plaši tiek izmantots XML formāts. Ir izstrādāti un tiek izmantoti vairāki strukturētu datu standarti (formāti), kas veidoti uz XML bāzes un kurus atšķir tikai dažādu XML tagu lietošana un elementu kārtība.

Sekojošā tabulā doti XPath semantiski identiskiem elementiem diviem XML bāzētiem elektronisko dokumentu standartiem.

3.4.1 tabula

XML bāzēto standartu EHF un E2B struktūras salīdzināšana (fragments)

	EHF	E2B
Piegādātāja nosaukums	AccountingSupplierParty/Party/Party Name/Name	InvoiceHeader/Supplier/Name
Saņēmēja nosaukums	AccountingCustomerParty/Party/Party Name/Name	InvoiceHeader/Buyer/Name
Produkta vai pakalpojuma nosaukums	InvoiceLine/Item/Name	InvoiceDetails/BaseItemDetails/Description

No darījumu loģikas viedokļa abi līdzvērtīgas nozīmes datu elementi tiek glabāti vienā un tajā pašā fiziskā formātā - XML datnē, tomēr datu struktūra, kā tabulā redzams, ir atšķirīga.

XML formāta priekšrocības – pašaprakstošs, vieglā modificējamība [9] salīdzinot ar citiem formātiem praksē bieži vien rada negaidītas neērtības šo formātu izmantošanā. Abas šīs īpašības provocē viegli veikt nelielas izmaiņu standartos tos piemērojot specifiskajam gadījumam kā rezultātā rodas neskaitāmas un nekontrolētas XML formātu variācijas. Bieži ir nepietiekami zināt, ka datne ir, piemēram, EHF formātā, jo tas vēl nenozīmē, ka datne ir validējama un satur visus elementus atbilstoši EHF formāta definīcijai.

3.5 EDIFACT

Vēsturiski izveidojies strukturētu datu formāts, kas izstrādāts Apvienoto Nāciju Organizācijas paspārnē [16, 20]. Tas ir sekojošu formātu kombinācija – „elementu nosaukumu un vērtību pāri” un „ar atdalītāju nodalītas vērtības”.

EDIFACT datne, piemēram, varētu saturēt sekojošu fragmentu:

```
NAD+IV+92221144++SIA Velkonis'
```

Ja vien speciāli nav norādīts savādāk, tad EDIFACT standarts nosaka, ka

- + atdala datu elementus
- : atdala kompleksu elementu daļas
- ' norāda uz segmenta beigām

Augstākminētais piemērs būtu jālasa sekojoši:

- 1) NAD norāda, ka sākas segments ar adresāta informāciju (Name and Address)
- 2) IV ir pirmā elementa vērtība, kas nosaka adresāta tipu (Party Qualifier).
Atbilstoši klasifikatoram var noskaidrot, ka IV ir “rēķina saņēmējs”
- 3) 92221144 ir otrā elementa vērtība, un tas ir “adresāta identifikators”
- 4) divi secīgi atdalītāji + nozīmē, ka trešā elementa „adrese” vērtība nav uzdota
- 5) SIA Velkonis ir ceturta elementa vērtība, un tas ir “adresāta nosaukums”
- 6) ' norāda, ka adresāta segments ir beidzies

Konkrētā gadījumā datnes fragments norāda, kā „rēķina saņēmējs” ir „SIA Velkonis” ar identifikatoru „92221144”.

Detalizēta EDIFACT formāta struktūra aprakstīta dokumentācijā [16] un tā pastāvīgi tiek atjaunināta.

AUTHOR - Authorization	IFTDGN - Dangerous Goods Notification
BOPCUS - Balance of Payment Customer Transaction Report	IFTFCC - International Transport Freight Costs/Other Charges
BOPDIR - Direct Balance of Payment Declaration	IFTMAN - Arrival Notice
BOPINF - Balance of Payment Information from Customer	IFTSTA - International Multimodal Status Report Message
COARRI - Container Discharge/Loading Report	INVOIC - Invoice
COHAOR - Container Special Handling Order	INVRPT - Inventory Report
CONAPW - Advice on Pending Works	ORDCHG - Purchase Order Change Request
CONDPV - Direct Payment Valuation	ORDERS - Purchase Order
CONITT - Invitation to Tender	ORDRSP - Purchase Order Response
CONPVA - Payment Valuation	PAXLST - Passenger List
CONQVA - Quantity Valuation	PAYMUL - Multiple Payment Order
COPRAR - Container Discharge/Loading Order	PAYORD - Payment Order
COREOR - Container Release Order	PRODEX - Product Exchange Reconciliation
COSTCO - Container Stuffing/Stripping Confirmation	QUALITY - Quality Data
COSTOR - Container Stuffing/Stripping Order	QUOTES - Quote
CREADV - Credit Advice	RECADV - Receiving Advice
CUSDEC - Customs Declaration	REMADV - Remittance Advice
CUSRES - Customs Response	REQDOC - Request for Document
DEBADV - Debit Advice	REQOTE - Request for Quote
DELFOR - Delivery Schedule	SSREGW - Notification of Registration of a Worker
HANMOV - Cargo/Goods Handling and Movement	STATAC - Statement of Account
IFCSUM - Forwarding and Consolidation Summary	SUPRES - Supplier Response
IFTCCA - Forwarding/Transport Shipment Charge Calculation	

8. att. Populārākie UN/EDIFACT datu apmaiņas komplekti (transaction sets) [5]

8. attēlā redzams EDIFACT formāta izplatītākie pielietojanas komplekti, kas pilnībā nosedz e-komercijas vajadzības. Šajā darbā autors parasti izmantos e-rēķinus (INVOIC) kā piemēru, tomēr gribētu uzsvērt, ka jebkurš no komplekciem tikai apraksta avota vai mērķa datu struktūru, un datu transformācijas risinājumi ir piemērojami jebkuram no komplekciem.

3.6 Formātu lietojuma un transformāciju specifika

Datu struktūru ir samērā viegli atpazīt XML un *elementu nosaukumu un vērtību pāri* bāzētajiem formātiem, jo datnes bez vērtībām satur arī datus raksturojošu vai aprakstošu informāciju, kas ļauj identificēt vērtību lietojumu. Tomēr saprast semantiku datnēm, kas ir pozīcijām bāzētā formātā vai ar atdalītāju nodalītu vērtību formātā ir gandrīz neiespējami vispārīgā gadījumā, ja vien nav pieejama papildus informācija.

Acīmredzot, lai pārveidotu datus no viena fiziskā formāta citā tomēr ir viennozīmīgi jāsaprot gan datu fiziskā struktūra, gan biznesa loģika un vispārīgā

gadījumā nebūs iespējams veikt datu pārveidošanu ignorējot zināšanas un semantiku, jo nav iespējams viennozīmīgi definēt transformāciju starp fiziskajiem formātiem nezinot datu loģiku.

Tomēr iespējams, ka atsevišķos gadījumos un situācijās, kad tiek izmantoti plaši izmantoti un stingri definēti standarti, varētu runāt par risinājumu konstruktoru vai domēna specifisku rīku eksistenci, kas ļautu pārveidot datus no viena fiziskā formāta citā, ja ir zināma un tiek saglabāta to loģika (piemēram, ieejas un izejas dati ir e-rēķins) tikai nosakot avota un mērķa fizisko reprezentāciju, neiedziļinoties loģikā un pieņemot, ka tā atbilst standartam.

Nav vienotas pieejas un ideāla formāta datu eksportam, pārsūtīšanai un importam. Katra informatīvā sistēma datus var sagatavot tikai tai raksturīgā formāta vai ar tai raksturīgajām iezīmēm. Atšķirības datu formātus var noteikt arī izmantotās tehnoloģijas vai izstrādes līdzekļi, programmatūra, izmantotā RDBMS, u.c.

Nozīmīgas un izplatītās izstrādes, piemēram IDoc (SAP) var kļūt par de facto standartiem, kuriem tirgus dalībnieki pielāgojas.

Starptautiskajā līmenī izmantojamus un populārākos datu formātus var noteikt arī valstu vai valstu grupas pieņemtie standarti vai standartu trūkums.

4. DATU TRANSFORMĀCIJAS

1:1

Vienkāršākā un tipiskā gadījumā „līdzīgs pret līdzīgu” datu pārveidošana izmantojama, kad ieejas dati tiek pārveidoti atbilstoši mērķa shēmas struktūrai, nepieciešamas gadījumā citā fiziskajā formātā, ar nelielām vai bez modifikācijām tomēr saglabājot 1:1 kardinalitāti starp ieejas un izejas datu elementiem un atbilstošajām vērtībām.

Piemērs, rēķina saņēmēja nosaukums avota shēmā tie pārveidots par rēķina saņēmēju mērķa shēmā.

Būtiski ir pievērst uzmanību, lai kartējums starp avota un mērķa shēmām pieļauj realizēt pārveidojumus saglabājot kardinalitāti, jo mērķa shēmai var būt strikti noteikta struktūra ar tai specifisku elementu kardinalitāti, kas to nepieļauj. Var gadīties, ka elements mērķa shēmā ir ar kardinalitāti 1, bet avota shēmā ir vairāki atbilstoši vai līdzīgas nozīmes elementi, tāpēc, neskatoties uz to ka pārveidojums „līdzīgs pret līdzīgu” būtu realizējams, to nepieļauj mērķa shēma.

Piemērs, vairums datu formāti pieļauj, ka datne satur vairākus rēķinus, tomēr EHF formāts nosaka, ka datne var saturēt tikai un tieši vienu rēķinu.

Iespējamie problēmu risinājumi:

1. Iespējams, ka elementus var dublēt augstākas struktūras (vecāku elementu) līmenī, piemēram, katru rēķinu atbilstoši mērķa shēmai veidojam kā atsevišķu datni.
2. Problemātisko transformāciju var reducēt uz transformācijām 1:0 vai N:0, kuras sīkāk aprakstītas vēlāk.

0:1 vai 0:N

Datu transformācijas, kas paredz jaunas informācijas tiešu vai atvasinātu radīšanu atbilstoši mērķa shēmai. Parasti šādās transformācijās nepieciešamību nosaka tas, ka mērķa shēmā datu elements ir obligāts, tomēr ieejas datos atbilstošā elementa vai informācija nav.

Piemēram, ir zināms rēķina saņēmējs, bet nav zināma piegādes adrese; ir zināmas rēķina pozīcijas, bet nav dota rēķina summārā informācija (kopsumma), u.c.

Iespējamie problēmu risinājumi:

1. Trūkstošie dati var tikt iegūti no papildus ārējiem datu avotiem.

Tas būtu loģisks risinājums, ja dati tiek ņemti no datu bāzēm, kur relācijas starp tabulām nosaka atslēgas. Diemžēl datnes, kuras iegūtas eksportējot datus no informatīvās sistēmas, var nesaturēt atslēgas un tāpēc ir speciāli jā rūpējas un jānodrošina, lai veidojas viennozīmīga saite starp informāciju ieejas datos un papildus datu avotos.

2. Trūkstošie dati var tikt iegūti no līdzīgas nozīmes ieejas datiem.

Piemēram, ja mērķa shēmā piegādes adrese ir obligāts elements, tad varētu, saskaņojot ar klientu, ieviest kartējumu: „ja piegādes adrese ieejas instancē nav uzdota, tad izejas datu instancē piegādes adrese ir vienāda ar rēķina saņēmēja adresi”.

Šāds risinājums novērš ieejas datu trūkumu, tomēr samazinās datu uzticamību, jo nav iespējams garantēt informācijas atbilstību īstenībai.

3. Tiek ieviestas atvasinātas vai klientam specifiskas datu vērtības.

Ja datu pārveidošana tiek veidota noteikta klienta vajadzībām, tad var ieviest kartējumu, kas ievieto izejas datu instancē ar klientu saskaņotas datu elementu vērtības, neatkarīgi vai atbilstošie elementi ir vai nav ieejas datos, jo faktiski tie var netikt izmantoti.

Risinājums tiek lietots gadījumos, ja datu sūtītājs nevar vai nevēlas mainīt ieejas datu formātu un elementu vērtības, tomēr datu saņēmējs pieprasa mērķa shēmai atbilstošu un klientam specifisku informāciju.

Piemēram, ieejas instance nesatur uzņēmuma reģistrācijas numuru, bet tā ir nemainīga un var tikt viegli realizēta ar atbilstošu kartēšanas sakarību palīdzību.

Šāds risinājums nodrošina, ka mērķa instance atbilst mērķa shēmai un satur klienta specifisku informāciju, tomēr jāņem vērā un jāpievērš papildus uzmanība datu korektumam gadījumos, kad risinājums tiek adaptēts citiem klientiem.

4. Summāru datu aprēķināšana.

Ar datu transformācijas palīdzību var nodrošināt summāru datu aprēķināšanu atbilstoši mērķa shēmas prasībām gadījumos, kad ieejas dati nesatur atbilstošo informāciju vai tā neatbilst mērķa shēmas definīcijai.

Piemēram, ieejas datu instance satur rēķina pozīciju summas, bet nesatur rēķina kopējo summu, kurš ir obligāts elements mērķa shēmā.

Summāros datus varētu viegli aprēķināt datu transformācijas laikā ar atbilstošām kartēšanas sakarībām, ja vien ir zināms algoritms un ir pieejami ieejas dati, kuri nepieciešami aprēķinos. Tomēr izmantojot šādu risinājumu kļūst būtisks ne tikai datu korektuma jautājums, bet arī atbildības jautājums par aprēķināto rezultātu. Vai sadarbības partneris, kurš veicis transformāciju, ir tiesīgs un pilnvarots šādus aprēķinus veikt un kurš ir atbildīgs par iespējamām kļūdām? Ir iespējamās situācijas, kad pakalpojumu sniedzējs nevēlas uzņemties atbildību par kļūdainu aprēķinu rezultātiem, kuriem var būt daudz un dažādi iemesli, ieskaitot kļūdainus ieejas datus. Datu pārveidošanas pakalpojuma sniedzējs var uzstāt, lai datu nosūtītājs iekļauj summāros datus ieejas datus, vai arī deklarātīvi norobežoties no atbildības fiksējot to sadarbības līgumā, lai izvairīties no situācijām, kad gan datu sūtītājs, gan saņēmējs nav apmierināti ar aprēķinu rezultātu un pusēm radušies zaudējumi kā atbilstošas transformācijas rezultāts.

Ir vēl viena lieta, kas jāņem vērā gadījumos, ja summāro informācija tiek aprēķināta datu pārveidošanas laikā – pie atkārtotas vai turpmākas datu transformācijas būtu jāņem vērā datu iegūšanas mehānisms un jāparedz vai summārie dati jāsauglabā vai jāatmet atkarībā no to iegūšanā veida. Piemēram, ja summārie dati iegūti no izejas datiem un tos nav aprēķinājis starpnieks, tad to tālākā transformācija ir pieļaujama, bet pretējā gadījumā turpmāka izmantošana noliegta un tie jāaprēķina no ieejas datiem atkārtoti jau atbilstoši atbilstošās mērķa shēmas prasībām vai atmetami.

Visi augstākminētie risinājumi izsauc paliekošas izmaiņas izejas datu instancē. Lūkojoties uz risinājumiem no datu pārveidošanas teorijas viedokļa var viegli secināt, ka datu vaicājumi izejas un ieejas instancēs var sniegt dažādas atbildes vai arī vaicājumi ieejas instancē vispār var nedot atbildes uz vaicājumiem, kuri ir korekti atbildami izejas instancē. Otra izrietošā problēma – ja pieņemam, ka transformācijai M jāveido atbilstošā atgriezeniskā datu transformācija M^{-1} , tad būtiski ir izvairītos no artefaktiem, lai panāktu, ka $M^{-1}(M(S)) = S$.

1:0 vai N:0

Datu transformācijas, kas paredz informācijas atmešanu, jo tā nav pārveidojama izejas datu instancē vai neatbilst mērķa shēmai.

Piemēram, ieejas instancē identiska klienta informācija dublēta un pieejama vairākos eksemplāros atbilstoši avota shēmai, tomēr mērķa shēmā atbilstošajiem datiem ir kardinalitāte 0..1.

Iespējamie problēmu risinājumi:

1. Lai veiktu atbilstošo datu pārveidošanu pietiek ar to, ka kartējumā netiek aprakstītas atbilstošo ieejas datu pārveidošanas sakarības.

Risinājums izsauc paliekošas izmaiņas izejas datu instancē. Lūkojoties uz risinājumiem no datu pārveidošanas teorijas viedokļa var viegli izsecināt, ka datu vaicājumi izejas un ieejas instancēs var sniegt dažādas atbildes vai arī vaicājumi izejas instancē vispār var nedot atbildes uz vaicājumiem, kuri ir korekti atbildami ieejas instancē.

N:1

Datu transformācijas, kas paredz datu elementu vērtību radīšanu sapludinot vairāku ieejas datu elementu vērtības vienā.

Piemēram, mērķa shēmā klienta adrese ir uzdodama ar diviem elementiem, bet ieejas dati satur detalizētu klienta informāciju atsevišķos elementos: pasta indekss, iela, pilsēta, valsts, u.c.

Iespējamie problēmu risinājumi:

1. Lai veiktu atbilstošo datu pārveidošanu pietiek ar to, ka kartējumā tiek aprakstītas atbilstošo ieejas datu pārveidošanas sakarības.

Risinājums izsauc paliekošas izmaiņas izejas datu instancē. Lūkojoties uz risinājumiem no datu pārveidošanas teorijas viedokļa var viegli izsecināt, ka datu vaicājumi izejas un ieejas instancēs var sniegt dažādas atbildes. Otra izrietošā problēma – datu transformācija tos sapludinot var radīt vērtības, kuru atkārtota

sadalīšana var būt neiespējama un transformācijai M atgriezeniskā datu transformācija M^{-1} neeksistēs.

1:N

Datu transformācijas, kas paredz datu elementu vērtību radišanu mēģinot sadalīt ieejas datu elementa vērtību vairākās.

Iespējamie problēmu risinājumi:

1. Lai veiktu atbilstošu datu pārveidošanu pietiek ar to, ka kartējumā tiek aprakstītas atbilstošo ieejas datu pārveidošanas sakarības.
2. Datu transformācijas uzdevumam var nebūt risinājuma!

Piemēram, pieņemsim, ka ieejas dati ir iegūti sapludinot datus informatīvajā sistēmā, tomēr iespējams, ka dati ir bijuši kļūdaini vai ir sapludinātas arī *null* vērtības. Var veidoties problemātiskas situācijas, kad datu sadalīšanas algoritms nevar nodrošināt datu uzticamību, jo nav iespējams noskaidrot kļūdaino elementu un *null* elementu eksistenci ieejas datos. Kā piemēru varētu dot adreses „Rīgas 3”, „Mārupes 3”. „Rīgas” un „Mārupes” var būt kļūdainas apdzīvotās vietas, ielas vai mājas nosaukums.

Automātiska datu sadalīšana var būt neiespējama, tomēr tā varētu notikt pie noteiktiem pieņēmumiem datu pārveidošanas algoritmā. Tā, pieņemsim, varētu pieņemt, ka gadījumos, kad adrese nesatur apdzīvoto vietu, apdzīvotā vieta būs Rīga. Ar līdzīgu pieņēmumu palīdzību būtu iespējams pārveidot datus, bet to kvalitāte un ticamība ir atkarīga no mūsu pieņēmumu kvalitātes.

Datu transformācijas, kas veiktas e-komercija vajadzībām parasti nenodarbojas ar simbolu virknes „uzlabošanu”. Dati tiek apstrādāti un pārveidoti līdzīgi pret līdzīgu ar iespējamu pēcapstrādes validāciju. Ja dati validācijas rezultātā nav atzīti par derīgiem, tad atbilstoši pieraksti tiek atstāti žurnālos un datnes nosūtītas manuālai apstrādei. Ir pieļaujams, ka dati vai datne pēc manuālas labošanas tiek atkārtoti apstrādāti, bet nedrīkst pieļaut, ka datu pārveidošana ietekmē citu nesaistītu datu apstrādi.

Citas datu pārveidošanas problēmas

1. Datu elementu neviennozīmība.

Piemēram, ieejas dati ir pozīcijām bāzētā vai ar atdalītāju nodalīto vērtību formātos, un satur simbolu virkni, tomēr nav iespējams viennozīmīgi identificēt kādam avota shēmas elementam virkne atbilst.

2. Elementu nosaukumu konflikti, kad semantiski vienādi objekti tiek saukti dažādi [9].

Piemēram, ieejas dati var saturēt vienu un to pašu informāciju elementos, bet mērķa shēmā ir citi atbilstošie elementi vai to pat nav. Ilustrācijai, EDIFACT segmenti NAD+IV – „invoicee”, NAD+BY – „buyer”, NAD+AA – “party to be billed” var būt un var nebūt sinonīmi un var būt uzdoti vienlaicīgi.

3. 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.

4. Mērvienību vai izmantoto vienību konflikti. Konflikts iestājas 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. [9]

Problēmu varētu ilustrēt ar datnēm, kas dotas 1., 2. un 3. pielikumos. Ieejas datos nav norādītas naudas un produktu uzskaites vienības. Vienību „LVL” un „PCS” parādīšanās izejas datos ir tikai transformācijas izstrādātāja interpretācija, kas varētu būt aplama citā ģeogrāfiskā reģionā.

5. Datu precizitātes problēmas. Dažādās valstīs darījumu summas vietējā naudā var stipri atšķirties uz citu valstu fona, kā rezultātā jāizmanto specifiski kartēšanas nosacījumi, piemēram, valstīs ar hiperinflāciju integer tipa mainīgajā var būt par maz zīmju, lai glabātu ikdienišķu darījumu summas. Pretējs piemērs, mobilā tālruņa rēķinā minūtes tarifs dots vienībās, kas stipri mazāks par mazāko naudas nominālu valstī, tāpēc jāizmanto specifiski summāras informācijas aprēķināšanas un apaļošanas algoritmi.

6. Ieejas datu trūkums vai to neatbilstība prasībām.

Piemēram, datu saņēmējs vēlās saņemt no datu sūtītāja piegādāto produkta EAN kodu, tomēr ieejas dati kodu nesatur. Iespējams, ka datu sūtītāja informatīvā sistēma nesatur atbilstošo informāciju un to nav iespējams iegūt arī no papildus datu avotiem. Ko darīt transformējot datus?

7. Informācijas neatgriezeniskas izmaiņas.

Datu pārveidojumu rezultātā starp dažēji savietojamiem formātiem varētu rasties artefakti.

Piemēram, transformācijai M jāveido atbilstošā atgriezeniskā datu transformācija M^{-1} , bet pārveidojot datus bija veikta datu atmešana (sk. augstāk transformācijas 1:0 vai N:0). Iespējama situācija, kad datu transformācijas uzdevumam M^{-1} nav risinājuma.

Problēmu varētu ilustrēt ar datnēm, kas dotas 1., 2. un 3. pielikumā. Var pamanīt, ka 3. pielikumā dati ir ieguvuši summāro informāciju un naudas vienības, kuras ieejas datos (1. pielikums) nav.

8. Simbolu kodu tabulu (character coding un mapping) konflikti.

Piemēram, Latviešu valodas speciālie simboli kreisais apostrofs, apakšējās un augšējās pēdiņas nepazīnāti nonāk IS datubāzēs un vēlāk datnēs starptautiskā datu apmaiņā, jo valodas tastatūras (keyboard) dziņi automātiski pārveido lietotāja ievadītos simbolus apostrofs un pēdiņa par to Latviskajiem analogiem. Pārsvarā datu apmaiņā netiek izmantoti Unicode atbalstoši formāti, lai ierobežotu pārsūtītās informācijas apjomu, tāpēc šādu simbolu nekontrolēta nonākšana XML datnēs var paralizēt to apstrādi.

5. TRANSFORMĀCIJU VEIDOŠANAS RĪKI

Turpmāk darbā tiks aprakstīti vairāki rīki un risinājumi, kurus varētu izmantot datu transformāciju veidošanai. Lai izvērtētu risinājumu piemērotību datu pārveidošanas uzdevumu risināšanā autors ir izvēlējies noteikt zemāk aprakstītos kritērijus.

Funkcionalitāte. Ir būtiski, lai risinājums nodrošina darbu ar pēc iespējas plašāku klāstu e-komercijā izmantojamajiem datu formātiem un ļauj veikt nepieciešamās manipulācijas ar elementiem vērtībām, nodrošina izejas datu sagatavošanu vēlamajā formātā. Iespējamais, 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.

Lietojamība. Kritērijs raksturo to, cik viegli lietotājam būs apgūt un lietot risinājumu. Svarīgi, lai veidojot transformāciju, to varētu veikt maksimāli ērti patērējot minimālu laiku. Ja rīkiem ir izstrādes vide, tad cik ērta ir rīka saskarne.

Lietošanas efektivitāte. Tipiskā gadījumā transformācijas nav vienreizējas un tās paredzēts lietot atkārtoti vai regulāri. Kritērijs raksturo to, cik ērts ir transformācijas pielietojums, vai to var izpildīt automatiski no uzdevumu plānotāja bez saskarnes ar lietotāju, vai transformāciju var realizēt kā interpretējamu kodu vai varbūt to var kompilēt izpildāmā modulī.

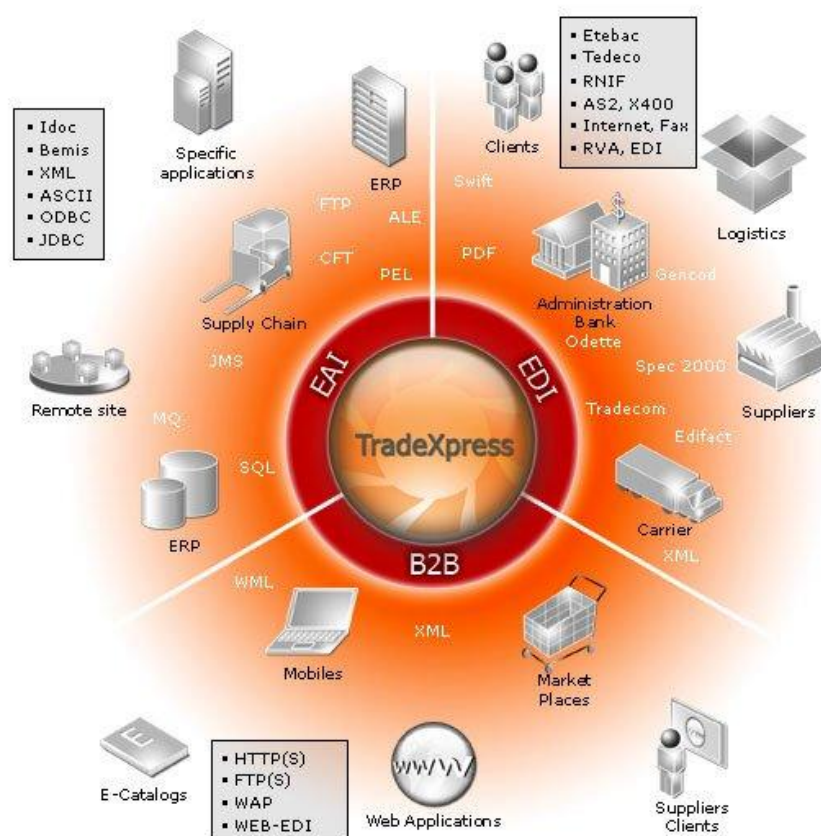
Ilgtermiņa atbalsts. Ikvienu transformāciju ilgākā termiņā var nākties gan labot, gan papildināt. Ir būtiski apzināties vai un cik ērti ir iespējams pilnveidot vai atkārtoti izmantot transformācijas, kāda informācija par transformāciju jāglabā, lai to vispār varētu atjaunināt vai atbalstīt.

Piemērotība. Kritērijs raksturo to, vai un cik lielā mērā risinājums ir piemērots vispārēju datu pārveidošanas uzdevumu un ar datu pārveidošanu saistītu problēmu risināšanai.

Izmaksas. Ne visiem risinājumiem ir zināmas precīzas izmaksas, tomēr izmaksu līmeni var mēģināt prognozēt. Jāuzsver, ka risinājumiem var būt dažādas prasības pret lietotājiem un to atbilstošo kvalifikāciju, tāpēc izmaksas jāskatās plašākā kontekstā un ilgtermiņā.

5.1 GCI TradeXpress

Francijas kompānija Generix Group piedāvā profesionālu platformu, kas nodrošina pilna servisa risinājumus datu pārsūtīšanai, pārveidošanai un sistēmu integrācijai, lai nodrošinātu B2B un A2A vajadzības. Platformu izmanto EDI operatori un IT pakalpojumu sniedzēji, kas nodarbojas ar informatīvo sistēmu integrāciju. Atbilstoši izstrādātāja tiešsaistes vietnē sniegtajai informācijai tai ir zināmas aptuveni 3000 instalācijas [10].

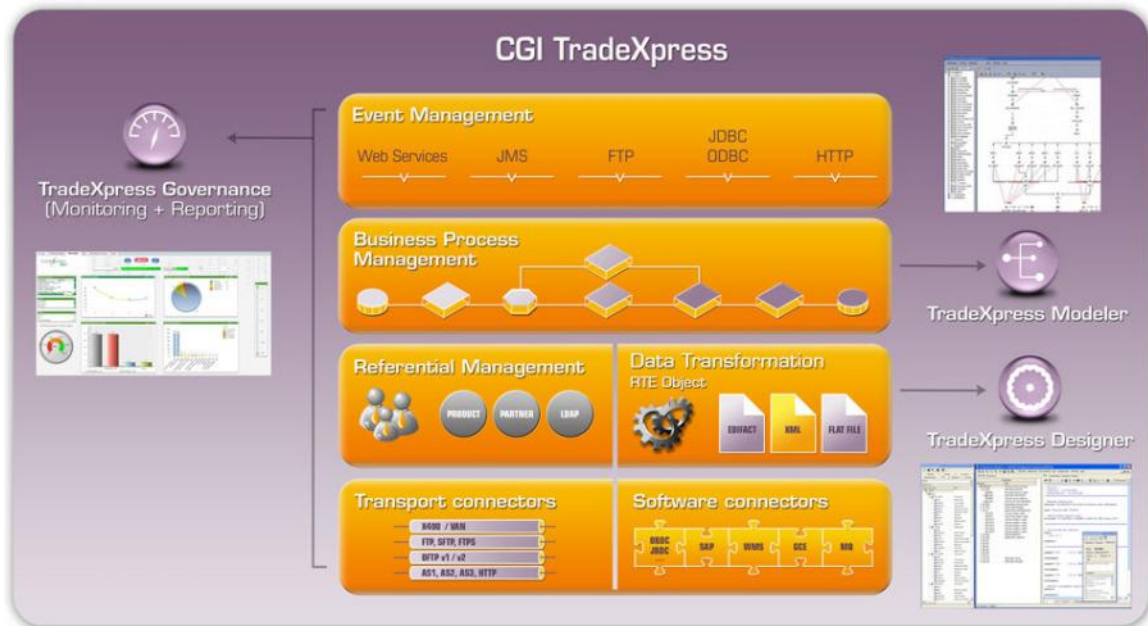


9. att. TradeXpress sistēmu un biznesa integrācijas iespējas

Risinājums satur vairākas komponentes [10, 11] (sk. 10. att.):

- TradeXpress Modeler ir komercijas procesu modelēšanas rīks, kas domāts B2B/EDI vajadzībām, bet risina arī sistēmu sadarbības jautājumus. Rīks ļauj modelēt datu plūsmu un izpildīt komercijas procesus.
- TradeXpress Monitoring ir modulis, lai uzraudzītu datu pārveidošanas automatizētos procesus un datu plūsmu.

- TradeXpress Designer ir rīks, lai grafiski modelētu datu transformācijas. Rīks ļauj ģenerēt kodu RTE programmēšanas valodā.



10. att. TradeXpress moduļi

Autoram nav pieejami risinājuma grafiskie rīki tomēr RTE programmēšanas valodas, kompilatora un datu plūsmas automatizācijas rīki ļauj gūt diezgan labu priekšstatu par risinājumu un izvērtēt tā atbilstību kritērijiem.

Izmantojot risinājumu, autors realizēja 3 datu transformācijas:

- 1) uz pozīcijām bāzēta teksta datne (1.pielikums) > EDIFACT formāta datne (2.pielikums)
- 2) EDIFACT formāta datne (2.pielikums) > uz pozīcijām bāzēta teksta datne (3.pielikums)
- 3) uz pozīcijām bāzēta teksta datne (1.pielikums) > XML datne (4.pielikums)

Lai ilustrētu trešo transformāciju darba 5.pielikumā dots XML šablons, kurš tika izmantots, lai aprakstītu izejas datu formātu, bet 6.pielikums satur izveidoto transformāciju RTE programmēšanas valodā.

Risinājuma atbilstība kritērijiem:

Funkcionalitāte 8/10

Neskatoties uz to, ka grafiskie transformācijas modelēšanas rīki autoram nebija pieejami, RTE koda rakstīšana nebija sarežģīta. Programmēšanas valoda ir speciāli piemērota strukturētu dokumentu apstrādei. Tā ļauj izmantot ieejas un izejas datu šablonus, kas atvieglo ieejas datu apstrādi un arī izejas datnes veidošanu.

Atbilstoši izstrādātāja informācijai risinājums ir pieejams Windows un Unix platformām.

Atbalstīto formātu skaits ir plašs un viegli papildināms, veidojot bibliotēkas (analoģiski C programmēšanas valodas lietošanas praksei) un datņu šablonus. Autors, realizējot vienu no transformācijām, izveidoja XML datnes šablonu un izmantoja XML apstrādes bibliotēkas funkcijas.

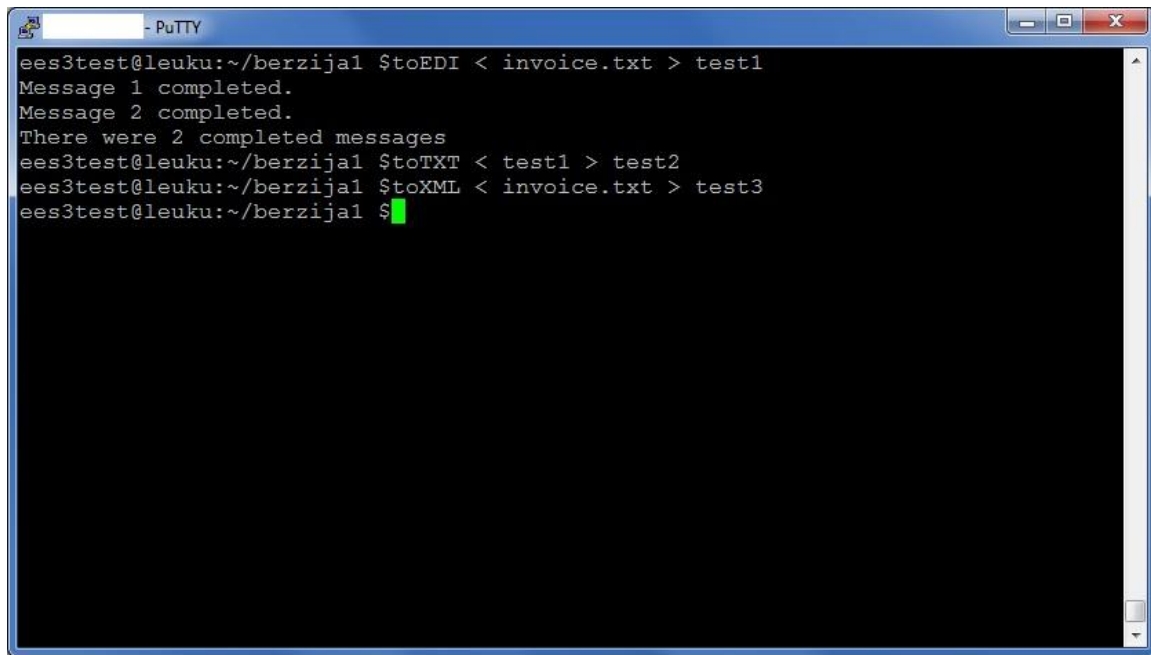
Trūkumi. Autoram nebija pieejama grafiskā vide tāpēc par risinājuma iespējām kopumā ir grūti spriest, tomēr veidojot kodu nebija iespējams uzdot XML struktūru ne ar XSD, ne DTD. Nav iespējams pārbaudīt datu atbilstību struktūrai.

Lietojamība 8/10

Autors RTE koda rakstīšanai izmantoja UNIX teksta redaktoru vi ar valodas sintakses izcelšanu, bet varētu izmantot jebkuru teksta redaktoru. Transformācijai, kas ir veidota kā teksta datne – programmas kods ir savi trūkumi, ja skatās no lietojamības viedokļa, jo ar grafisku rīku transformācijas veidot noteikti ir vieglāk. Tomēr tā kā programmēšanas valoda ir speciāli pielāgota strukturētu dokumentu apstrādei, kods ir viegli veidojams un lasāms.

Lietošanas efektivitāte 10/10

Valodā RTE kods tiek kompilēts izpildāmā modulī un transformāciju var viegli aktivizēt no komandrindas tāpēc viegli automatizēt tās lietošanu (sk 11. att.).



```
ees3test@leuku:~/berzija1 $toEDI < invoice.txt > test1
Message 1 completed.
Message 2 completed.
There were 2 completed messages
ees3test@leuku:~/berzija1 $toTXT < test1 > test2
ees3test@leuku:~/berzija1 $toXML < invoice.txt > test3
ees3test@leuku:~/berzija1 $
```

11. att. Nokompilētas transformācijas izpilde UNIX vidē

Ilgtermiņa atbalsts 10/10

Transformācija kā programmas kods ir ļoti viegli uzturama versiju kontroles sistēmas, piemēram, Apache Subversion. Izmaiņas kodā var viegli atsekot un transformāciju salīdzinoši viegli var uzlabot, papildināt ar jebkuru teksta redaktoru neskatoties uz kompilatora versiju.

Piemērotība 10/10

Risinājums ir piemērots datu pārveidošanas vajadzībām un sniedz izstrādātājam visu nepieciešamo funkcionalitāti, lai veidotu transformācijas.

Tomēr autors vēlas uzsvērt, ka transformācijas kvalitāte ir atkarīga no programmētāju kvalifikācijas un prasmēm. Autors, veidojot testa transformācijas, ar nodomu ieviesa nelielus „uzlabojumus”. Rēķins, kas dots 3.pielikumā, iegūts no 1.pielikumā dotā rēķina ar secīgu transformāciju pielietošanas palīdzību un faktiski ir piedzīvojis pārveidošanu TEXT > EDIFACT > TEXT. Atkārtotu pārveidojumu rezultātā sākotnējais rēķins ir ieguvis atribūtus, kuri tam nav bijuši, piemēram, rēķina kopsavilkums un naudas vienības. Tas lieku reizi pierāda, ka jābūt apdomīgiem ar jebkuriem pārveidojumiem, jo jebkura datu transformācija var radīt nebijušas vai pazaudēt esošas atribūtu vērtības tādējādi radot ar elementu kardinalitāti saistītās problēmas, kas aprakstītas augstāk.

Izmaksas 9/10

Diemžēl autoram nav zināmas risinājuma licences cena tomēr domājams, ka risinājums ir pilnībā funkcionāls bez būtiskiem ieguldījumiem programmētāju darba stacijās, jo ir runa par uzņēmuma licenci, ja vien netiek lietoti grafiski rīki. Risinājums neprasa speciālas prasmes no transformāciju izstrādātājiem, ja vien tiem ir pamatzināšanas programmēšanā.

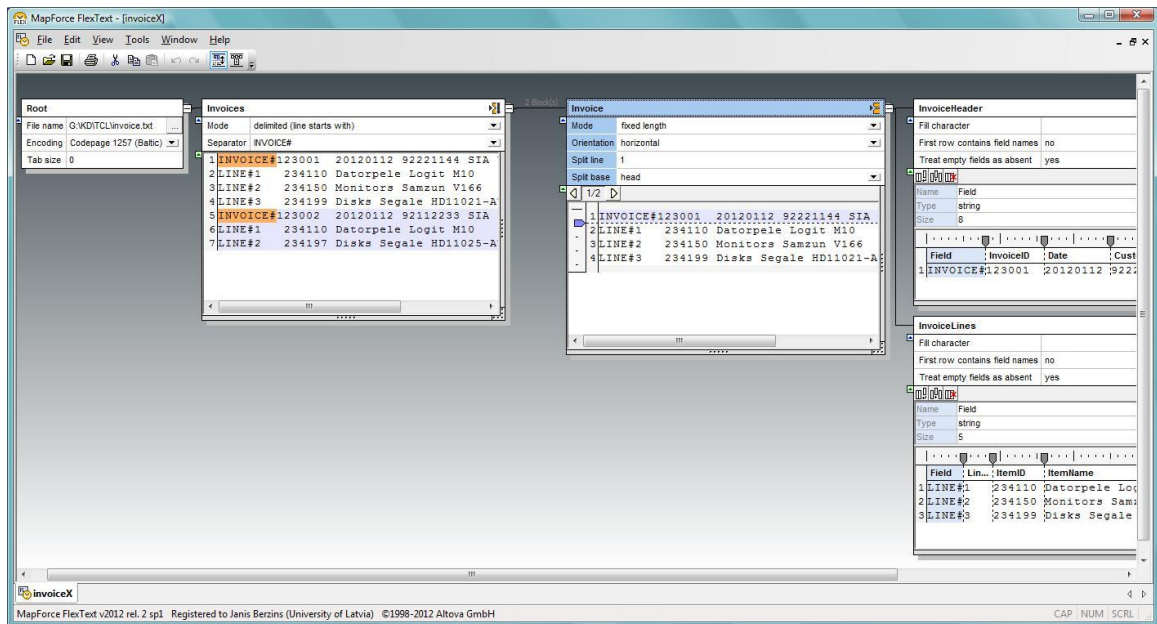
5.2 Altova Mapforce

Kompānija Altova [13] piedāvā vairākus programmatūras produktus, kas autoram šķita piemēroti datu transformāciju veidošanai. Autors apskatīja divus rīkus – XMLSpy un MapForce, tomēr secināja, ka otrais risinājums ir piemērotāks vispārīgiem uzdevumiem, tāpēc turpmāk darbā tiks detalizēti iztirzāti tikai MapForce rīks. Arī XMLSpy ļauj veidot datu transformācijas, tomēr atbalsta tikai ar XML formātu saistītās manipulācijas.

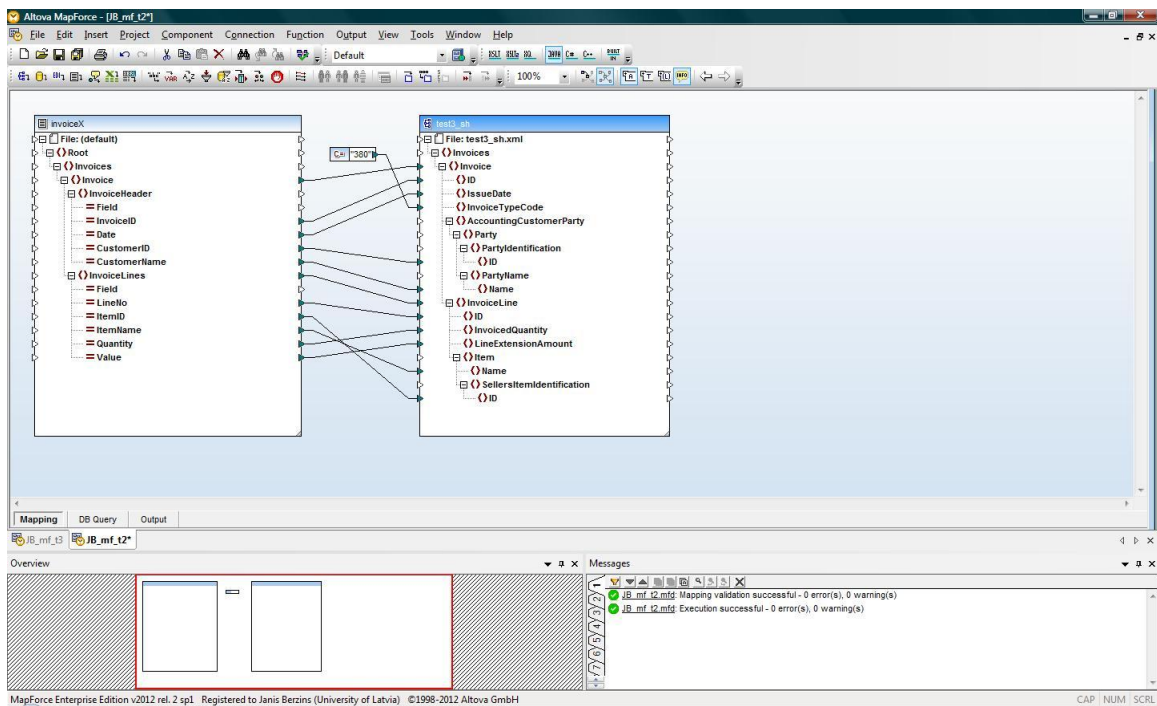
MapForce ir grafisks modelēšanas rīks, kas ļauj veidot kartēšanu starp ieejas un izejas datu shēmām, definēt neieciešamos pārveidojumus, vispārīgajā gadījumā ļauj ģenerēt transformācijas Java, C# un C++ kodu, bet gadījumā, kad tiek dati tiek pārveidoti starp XML formātiem ir iespējams ģenerēt XSLT 1.0/2.0 un XQuery kodu.

Autors pētot MapForce iespējas izveidoja 2 datu transformācijas:

- 1) uz pozīcijām bāzēta teksta datne (1.pielikums) > XML formāta datne maksimāli tuvināta 4. pielikumā dotajai (sk. 12. un 13. att., 8.pielikums).

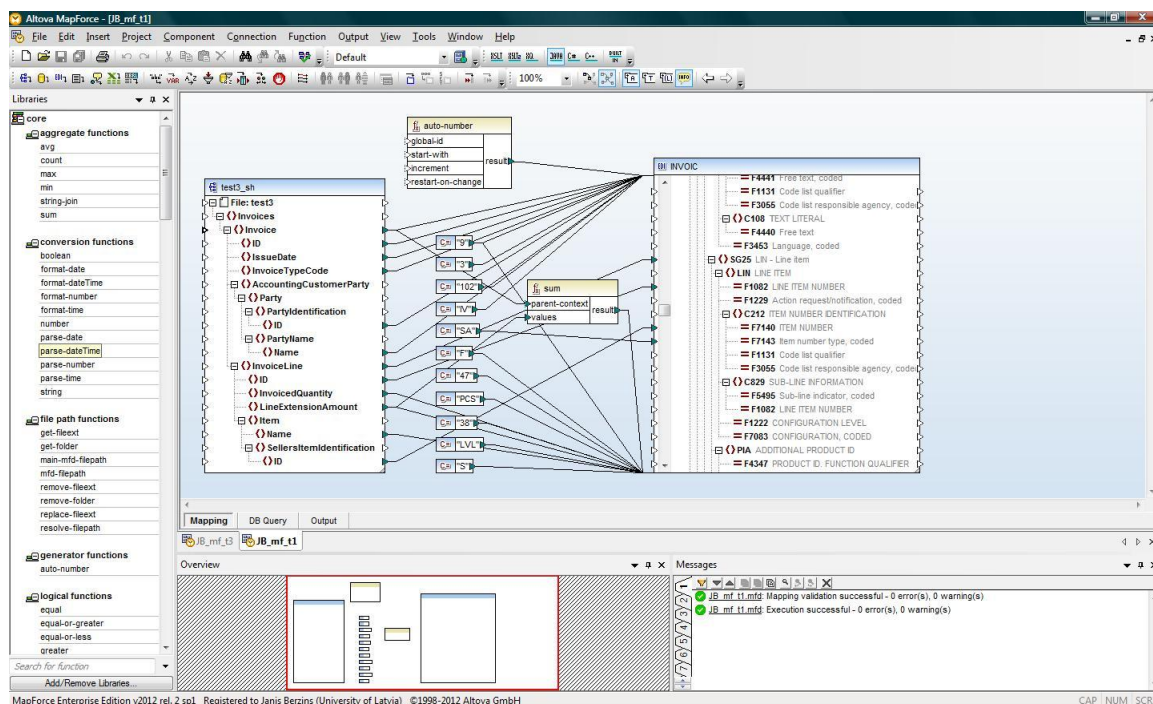


12. att. MapForce rīks. Ieejas datu shēmas veidošana 1.testa piemēram



13. att. MapForce rīks. Kartējums 1.testa piemēram

- 2) XML formāta datne (4.pielikums) > EDIFACT formāta datne maksimāli tuvināta 2.pielikumā dotajai



14. att. MapForce rīks. Kartējums 2.testa piemēram

Risinājuma atbilstība kritērijiem:

Funkcionalitāte 9/10

Veidojot datu transformācijas, MapForce izmanto ieejas un izejas datu shēmas, kuras nepieciešamības gadījumā rīks ļauj izveidot. Atbalstāmo formātu klāsts ir iespaidīgs un viegli papildināms. Rīks atbalsta XML formāta datus, EDIFACT, SAP IDoc, CSV un citus datņu formātus, nodrošina datu iegūšanu tieši no izplatītākajām datu bāzu pārvaldības sistēmām, u.c. Ir iespējams lejuplādēt papildus shēmas formātiem, kuri netiek atbalstīti standarta komplektācijā. Autors, strādājot pie testa piemēriem, izveidoja nepieciešamās XML un teksta datņu shēmas kā arī lejupielādēja un importēja atbilstošo EDIFACT formāta shēmu.

Ļoti noderīga ir rīka iespēja pārbaudīt vai datnes atbilst izvēlētajai shēmai.

Testa transformācijas veidošanas rezultātā tika ģenerēts Java kods. Rīks izveidoja pilnvērtīgu Java projektu, kurš bez izmaiņām bija gatavs kompilēšanai ar Eclipse vai Apache Ant rīkiem. Tomēr jāpiemin, ka ģenerētais kods saturēja tiešas saites uz ieejas un izejas datnēm, tāpēc kodā ir jāveic izmaiņas, lai to varētu izmantot automatizēti no komandrindas, transformācijas ieejas un izejas datņu nosaukumus padodot kā parametrus.

Trūkumi. Vienīgais trūkums varētu būt tas, ka atbilstoši izstrādātāja informācijai [13] risinājums ir pieejams tikai Windows videi.

Lietojamība 10/10

Grafiskā saskarne ir ļoti vienkārša un intuitīvi saprotama, tāpēc ļauj viegli tikt galā ar ieejas un izejas shēmu veidošanu un kartēšanas nosacījumu definēšanu. Transformācijas modelēšanā var izmantot datu starppapstrādi, rīka iebūvētās funkcijas (sk. 13.att.), piemēram, virknes apstrādei, vai summāro datu aprēķināšanai, un citas papildus iespējas. Datu transformāciju veidošana ar šo rīku prasīja vismazāk laika salīdzinot ar citiem risinājumiem.

Lietošanas efektivitāte 9/10

Būtiski, ka MapForce rīks nodrošināja kompilējama koda ģenerēšanu programmēšanas valodās Java, C# un C++, tomēr ģenerētais kods jālabo, lai to varētu lietot atkārtoti un automātiski.

Ilgtermiņa atbalsts 7/10

Neskatoties uz to, ka rīks ir ļoti ērts jaunu transformāciju veidošanā, gatavo risinājumu ilgtermiņa atbalsts varētu radīt grūtības, jo transformācijas projektu bez atbilstošās vides aplūkot vai atjaunināt nevarēs. Arī uzģenerēt jaunu transformācijas kodu bez vides nevar. Protams, ir iespējams veikt izmaiņas pa tiešo Java, C# vai C++ kodā, tomēr tas nebūs vienkārši, jo kods ir diezgan sarežģīts un satur risinājuma iekšējo tipu un funkciju realizācijas klases. Tas nozīmē, ka jāarhivē un jāuztur visas projekta komponentes: pats rīks, rīka izveidotais transformācijas projekts, ieejas un izejas datu shēmas, transformācijas kods un izpildāmie moduļi.

Autora pieredze ar testa transformāciju veidošanu ir nepietiekama, lai pārliecinātos par to, ka projekta komponentes iespējams uzglabāt un pārvaldīt versiju kontroles sistēmās tādās kā Apache Subversion.

Piemērotība 10/10

Risinājums ir piemērots datu pārveidošanas vajadzībām un sniedz izstrādātājam visu nepieciešamo funkcionalitāti, lai veidotu transformācijas.

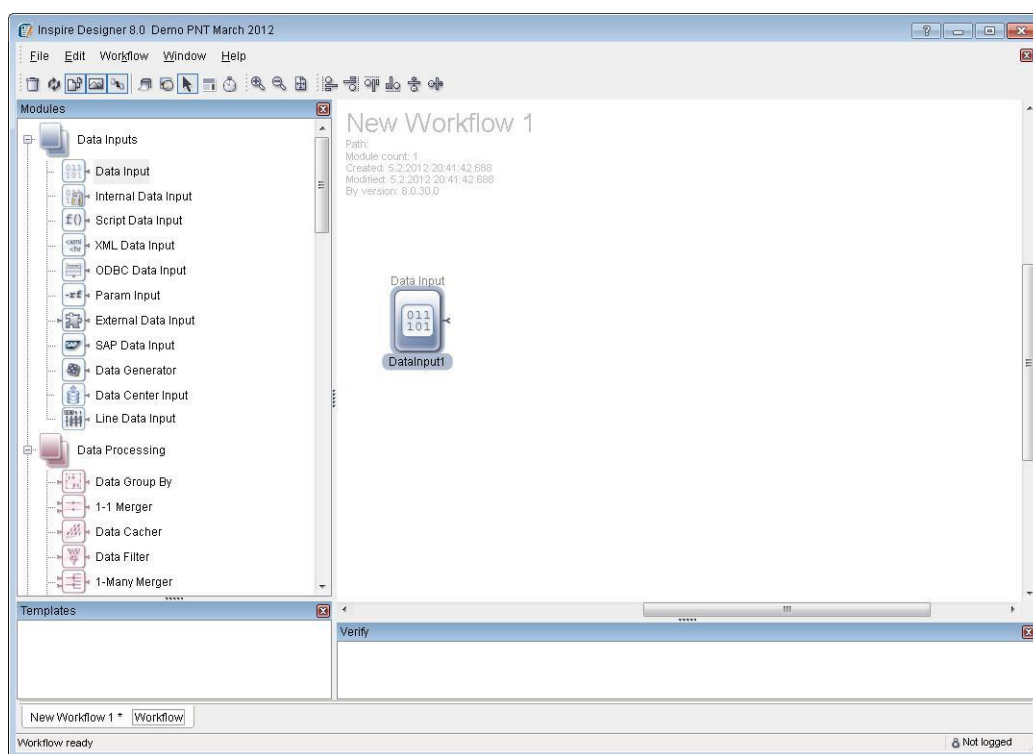
Izmaksas 7/10

Atbilstoši izstrādātāja tiešsaistes vietnē [13] pieejamajai informācijai Altova MapForce 2012 Enterprise Edition cena ir sākot ar 800 Eiro. Programmatūrai ir jābūt instalētai uz visiem uzņēmuma darbinieku datoriem, kuri izstrādā vai pat tikai aplūko transformācijas.

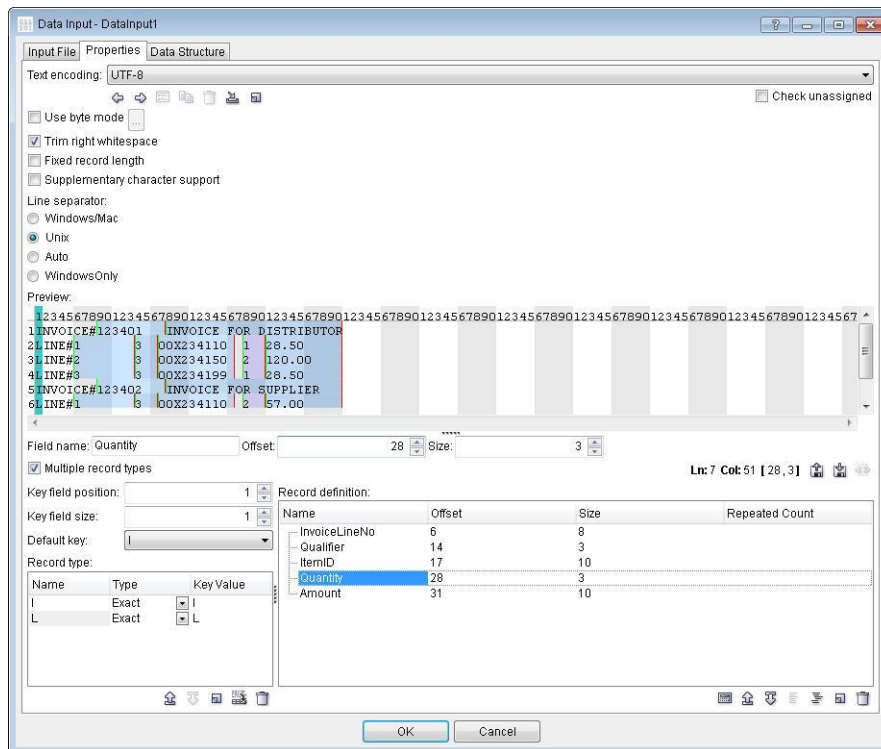
5.3 GMC Inspire Designer

Starptautiska kompānija GMC ar galveno mītni Šveicē piedāvā rīku Inspire Designer, kura pamatuzdevums ir datu vizualizācija. Tomēr rīks ļauj ar ļoti intuitīviem grafiskiem rīkiem iegūt datus no neskaitāmiem avotiem un veikt manipulācijas ar tiem. Datu iegūšana un pārveidošana realizēta ar grafiskas saskarnes palīdzību, kur ar vilkt un nomest tehnoloģijas palīdzību izvēlētie moduļi ir kombinējami datu pārveidošanas plūsmā (sk. 17. att.). Autors vēlējās pamēģināt un novērtēt vai rīks būtu piemērots datu transformācijas uzdevumu risināšanai.

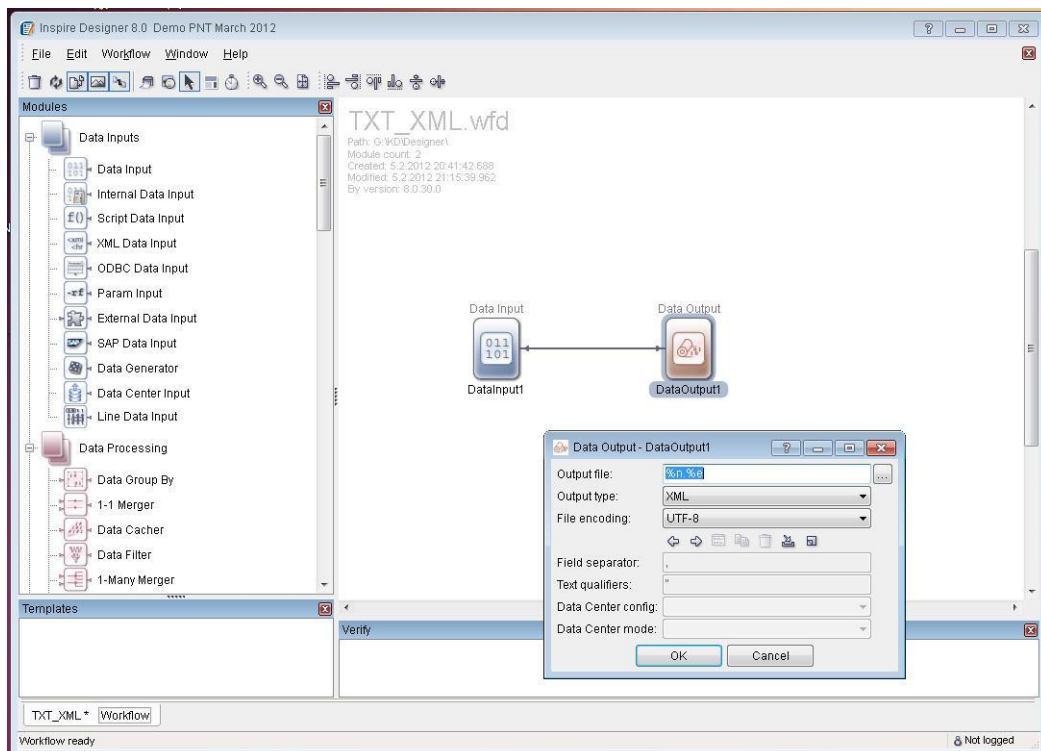
Autoram bija pieejama rīka demonstrācijas versija un tika realizēta viena datu transformācija, kurā dati no uz pozīcijām bāzēta formāta tika pārveidoti XML formātā. Transformācijas izveidošanas soļi redzami 15., 16. un 17.attēlos, bet iegūtais rezultāts 7.pielikumā.



15. att. Inspire Designer. Ieejas datu formāta izvēles un pārveidošanas moduļi.



16. att. Inspire Designer. Ieejas datu shēmas veidošana uz pozīcijām bāzētai datnei.



17. att. Inspire Designer. Datu transformācijas modelēšana.

Risinājuma atbilstība kritērijiem:

Funkcionalitāte 6/10

Rīkam ir pārdomāts un ērts ieejas datu apstrādes mehānisms, jo tas ļauj veikt ne tikai datu importu, bet arī filtrēšanu, apvienošanu un citas manipulācijas. Atbalstīto ieejas datu formātu klāsts ir diezgan plašs, tomēr mazāks, salīdzinot ar citiem apskatītajiem risinājumiem. Rīks ļauj iegūt datus no dažādas struktūras teksta un XML datnēm, SAP IDoc un arī tieši no populārākajām relāciju datubāzēm. Rīkam ir plašas datu pārveidošanas iespējas, iebūvētās funkcijas, specializēta skriptu programmēšanas valoda, kas ļauj veidot sarežģītākās manipulācijas.

Trūkumi. Atbilstoši izstrādātāja informācijai risinājums ir pieejams tikai Windows platformai. Diemžēl rīks neatbalsta ieejas un izejas datu struktūras uzdošanu ar šablonu vai XML gadījumā ne ar XSD, ne DTD palīdzību. Netiek nodrošināta datu validācija. Rīkam ir ļoti plašas iespējas manipulēt ar ieejas datiem, bet diemžēl ļoti ierobežotas iespējas, lai uzdotu izejas datu struktūru.

Lietojamība 9/10

Grafiskā saskarne ir pārdomāta un ērta, datu manipulācijas moduļus ar vilk un nomest tehnoloģijas palīdzību var viegli un ātri sakārtot transformācijā. Diemžēl tā kā rīks nav domāts transformāciju veidošanai, tam trūkst iespējas veidot kartējumu starp ieejas un izejas datu struktūrām.

Lietošanas efektivitāte 5/10

Atbilstoši izstrādātāja sniegtajai informācijai pilnā versijā risinājums tiek piedāvāts komplektā ar uzdevumu pārvaldības serveri un datu pārveidošana varētu tikt automatizēta, tomēr ar autora rīcībā esošo komplektāciju par to pārliecināties nebija iespējams. Acīmredzot risinājums neveido izpildāmu vai interpretējamu moduli, kas varētu darboties bez uzdevumu pārvaldnieka vai ārpus risinājuma vides.

Ilgtermiņa atbalsts 5/10

Ar rīku veidotā transformācija ir bināra datne, kura nav izmantojama un atjaunojama ārpus izstrādes vides. Varētu rasties grūtības arī ar versiju vai izmaiņu kontroli, tomēr kamēr vien atbilstošā rīka versija būs pieejama vai datnes formātu atbalstīs atjauninātās rīka versijās, problēmām ar ilgtermiņa atbalstu nevajadzētu rasties.

Piemērotība 2/10

Risinājums nav piemērots datu transformācijām vispārīgā gadījumā, jo neļauj uzdot izejas datu struktūru.

Izmaksas

Diemžēl informāciju par izmaksām iegūt neizdevās.

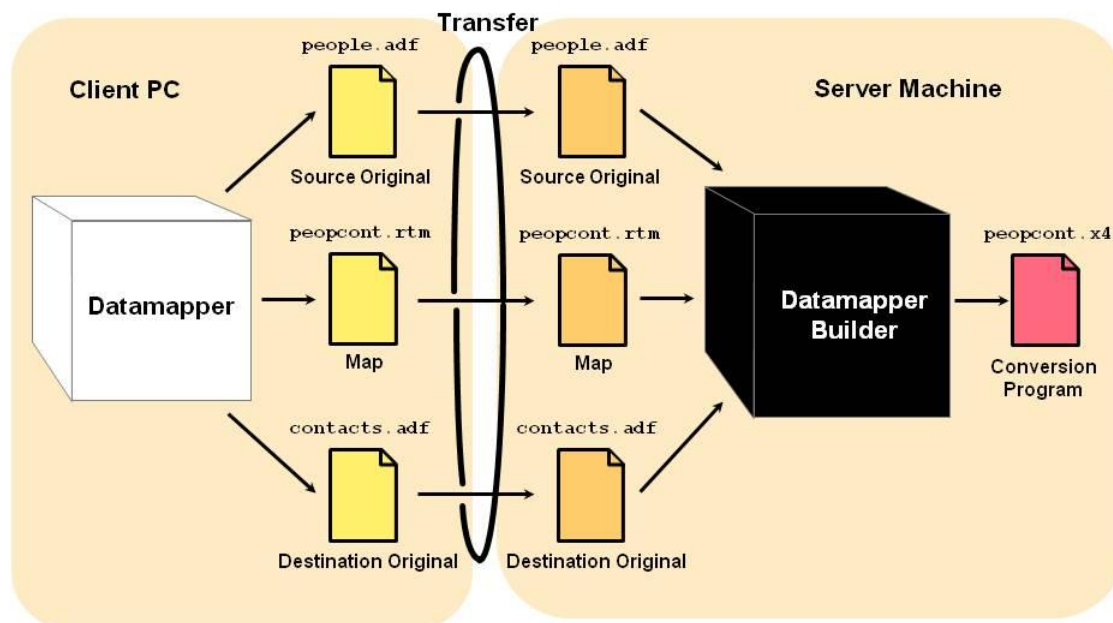
5.4 Axway Datamapper Builder

Šis ir viens no rīkiem, kuru datu transformāciju veidošanai izmanto autora darba devējs. Diemžēl autoram nav piekļuves rīkam, tāpēc šis apskats un vērtējumi ir sagatavoti no publiski pieejamās informācijas [15], rīka prezentācijām un lietotāju apmācības materiāliem.

Rīka lietotājs izmanto klienta programmatūru, lai pieslēgtos pie Datamapper Builder serverim, uz kura notiek ieejas un izejas datu struktūru aprakstīšana, tiek veidots kartējums un definētas nepieciešamās manipulācijas ar datiem. Kad transformācija ir gatava to var testēt, simulējot datu pārveidošanu un ģenerēt kompilējamu kodu specializētā programmēšanas valodā.



Datamapper Client and Builder

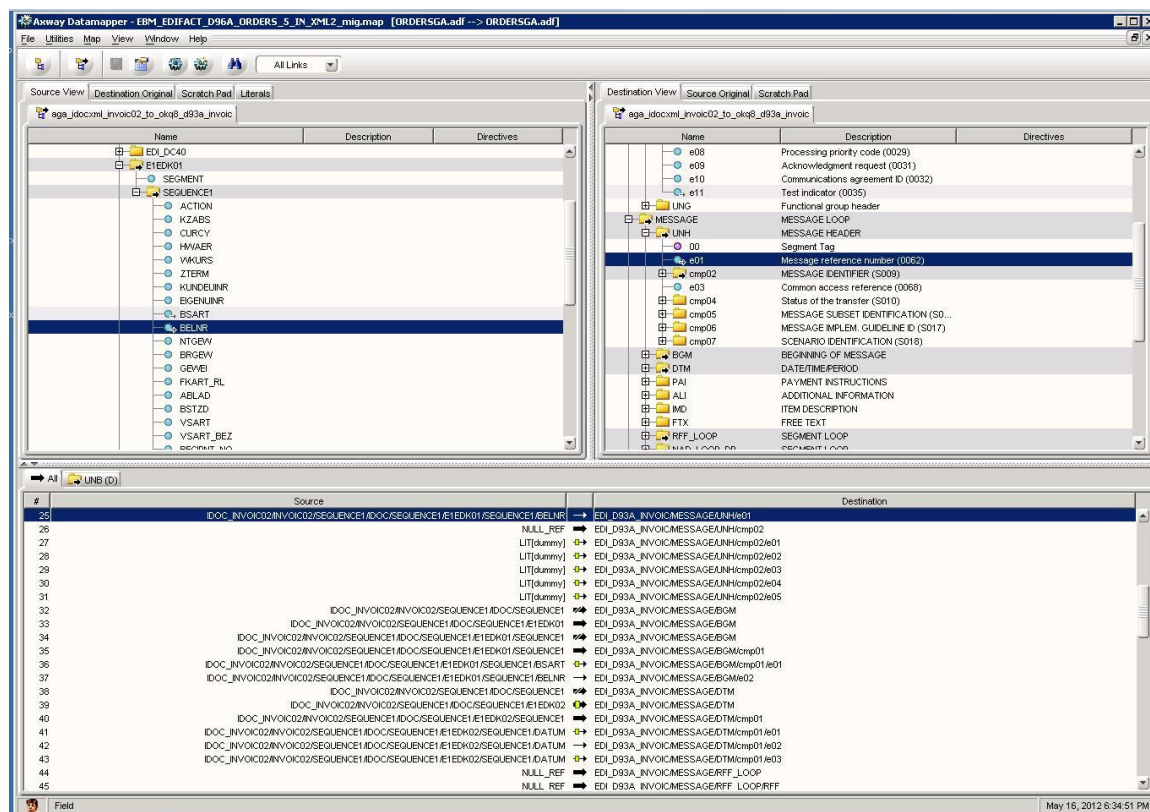


© 2006 Axway, Inc.

18. att. Axway Datamapper darbības principi

Autoram nebija iespējas patstāvīgi veidot datu transformāciju izmantojot rīku, tomēr tika iegūti esošu transformāciju projekta ekrānšāviņi, kas ļauj gūt ieskatu par

rīka iespējām. 19.attēlā var redzēt, kā starp ieejas un izejas datu struktūras elementiem (ekrāna augšpusē) tiek veidots manipulāciju kopums (ekrāna apakšdaļā).



19. att. Axway Datamapper transformācijas veidošana.

Risinājuma atbilstība kritērijiem:

Funkcionalitāte 9/10

Pilns atbalstāmo datu formātu uzskaitījums nav pieejams, tomēr zināms, ka rīks atbalsta XML, EDIFACT, SAP IDoc, CSV formāta datus un arī citus datu formātus. Ir iespējams patstāvīgi definēt jaunas datu struktūras.

Rīks nodrošina transformācijas izpildes simulāciju, kas ļauj aplūkot ziņojumus par kļūdām un ļauj aplūkot izejas datus.

Lietojamība 8/10

Grafiskā lietotāja saskarne atvieglo datu struktūras aprakstīšanu un transformācijas modelēšanu, tomēr salīdzinot ar citiem rīkiem, saskarne autoram nešķita pārdomāta un intuitīvi lietojama.

Lietošanas efektivitāte 9/10

Rīks ļauj ģenerēt, kompilēt un glabāt transformācijas kodu, serveris nodrošina automātisku transformāciju izpildi un atbilstošo uzdevumu pārvaldīšanu.

Ilgtermiņa atbalsts 8/10

Datu pārveidošanas projekts ar visām komponentēm tiek glabāts uz servera, kas nodrošina iespēju atkārtoti izmantot esošus risinājumus un atvieglo ilgtermiņa atbalstu. Tomēr jāņem vērā, ka transformācijas projekts (kartējums un struktūras) ārpus rīka vides nav izmantojami.

Piemērotība 10/10

Risinājums ir piemērots datu pārveidošanas uzdevumu risināšanai un sniedz izstrādātājam visu nepieciešamo funkcionalitāti, lai veidotu transformācijas.

Izmaksas

Diemžēl informāciju par rīka izmaksām iegūt neizdevās. Papildus izdevumi varētu rasties, lai nodrošinātu apmācību, jo rīka iespējas nav pašsaprotamas, un lietotāja darba staciju aprīkošanu un specializēta servera uzturēšana.

5.5 Kopsavilkums

Risinājumu novērtējumu kopsavilkums ir dots tabulā 5.5.1. Diemžēl izmaksas nav objektīvi izvērtējamas ierobežotas informācijas dēļ, tāpēc tās netiek iekļautas kopsavilkumā.

5.5.1 tabula

Rīku vērtējumu kopsavilkums

	Funkcionālitate	Lietojamība	Lietošanas efektivitāte	Ilgtermiņa atbalsts	Piemērotība
GCI TradeXpress	8	8	10	10	10
Altova Mapforce	9	10	9	7	10
GMC Inspire Designer	6	9	5	5	2
Axway Datamapper Builder	9	8	9	8	10

Autors vēlas uzsvērt, ka GMC Inspire Designer ir profesionāls rīks, lai veidotu un pārvaldītu uzņēmuma komunikāciju ar tā klientiem, piemēram, veidotu personalizētas vēstules, rēķinus, komerciālus piedāvājumus. To ir iespējams izmantot,

lai pārveidotu datus, tomēr tā ir drīzāk blakne nekā rīka pamatuzdevums. Rīks tika iekļauts pētījumā pateicoties perfektajai lietotāja saskarnei, datu iegūšanas un manipulācijas iespējām, tomēr rīks nevar sacensties ar specializētajiem datu transformāciju veidošanas rīkiem.

Darbā apskatītie rīki nav vienīgie, kurus var izmantot datu transformāciju veidošanai. Programmproduktu tirgū ir virkne citu risinājumu, piemēram, Syncro Soft programmatūra <oxygen/> XML Editor [17] vai Progress Software Corporation produkts Data Integration Suite [18] un citi. Autors pētījumā apskatīja risinājumus, kuri viņam bija pieejami darbavietā vai kuru demonstrācijas versijas bija iespējams salīdzinoši viegli iegūt.

Jebkurš no trijiem pētījumā apskatītajiem rīkiem TradeXpress, Mapforce un Datamapper Builder ir piemērots datu pārveidošanas uzdevumu risināšanai, tomēr katram no tiem ir savas stiprās un vājās puses. Grafiskā saskarne nepārprotami atvieglo, un paātrina datu transformāciju veidošanu, tomēr vienlaicīgi arī samazina iespējas viegli izsekot tam vai visas manipulācijas ir realizētas kā arī veikt versiju pārvaldību. Specializēta programmēšanas valoda un vienkāršs teksta redaktors ir ārkārtīgi vienkāršs un reizē efektīvs risinājums, tomēr prasa lielāku laiku transformāciju veidošanā.

NOBEIGUMS UN SECINĀJUMI

Strukturētu datu transformācijai un apmaiņai ir būtiska loma e-komercijā. Vajadzības, pielietojums un pārsūtamo datu apjoms nemitīgi pieaug. Jo lielākas iespējas darījumu partneriem ir apmainīties ar datiem elektroniski, jo lielākas ir pieprasījums pēc saistītajiem pakalpojumiem (konsultācijas, datu pārveidošana un datu pārsūtīšana). Strukturētu datu transformācija ir būtiska arī ārpus e-komercijas lietojumiem, piemēram, informatīvo sistēmu integrācijā (A2A). Iespējams, ka domājot plašāk un abstrahējoties no e-komercijas specifikas, daļa no maģistra darbā iztirzātajiem jautājumiem, problēmām un risinājumiem kā arī rīkiem ir piemērojami un izmantojami arī A2A lietojumos.

Autors uzskata, ka Latvijas IT kompānijas vēl nenovērtē un nesaredz iespējas, ko pierāda ārvalstu kompāniju dominānce elektronisko datu apmaiņas pakalpojumu sfērā. Datu apmaiņai robežu nav un tā ir starptautiska pēc savas būtības, jo starptautiska ir arī uzņēmējdarbības vide, tāpēc saprotams ir Eiropas Komisijas rosinājums ES dalībvalstīm novērst normatīvo aktu pretrunas un ierobežojumus, kas kavē masveida elektronisko rēķinu un norēķinu lietošanu. Domājams, ka Latvijas IT industrijai ir iespējas sevi apliecināt šajā laukā un piedāvāt pakalpojumus vietējā un starptautiskajā tirgū.

Darba praktiskajā daļā analizētie transformāciju veidošanas rīki ir tikai instruments prasmīga lietotāja rokās, kas labākajā gadījumā atvieglo darbu, izmantojot ērtu saskarni, nodrošina nepieciešamās manipulācijas un ļauj izvairīties no kļūdām, pateicoties datu validācijas mehānismam. Tomēr, izmantojot rīkus, iespējams pārbaudīt tikai datu atbilstību shēmai (struktūrai) vai arī simulācijas rezultātā iegūt kļūdu ziņojumus par pārveidojamo elementu tipu neatbilstību. Diemžēl, analizēt elementu vērtību atbilstību semantikai neviens no rīkiem nevar, tāpēc kļūdaina kartējuma realizācijas rezultātā iegūtie dati būs sintaktiski pareizi, bet saturiski kļūdaini. Tas nozīmē, ka transformāciju veidotājiem jāpārzina un jāņem vērā pētījuma teorētiskajā daļā apskatītās datu transformācijas problēmas un jālieto situācijai piemērotākie no risinājumiem neatkarīgi no izvēlētajā transformāciju veidošanas rīka.

Veidojot darbu, viena no autora motivācijām bija apzināt un izmēģināt alternatīvus transformāciju veidošanas rīkus, jo uzņēmumā lietotais risinājums (GCI TradeXpress) šķita primitīvs un morāli novecojis (neatbalsta XML failu struktūras

uzdošanu ne ar XSD, ne DTD palīdzību, nav validācijas mehānisma). Tomēr pētījums parādīja, ka rīks lai arī zaudē funkcionalitātē savas vienkāršības dēļ ir universālāks un izveidotās transformācijas ilgtermiņā ir vieglāk uzturamas.

No darbā aplūkotajiem rīkiem autors visaugstāk novērtēja Altova Mapforce, jo ar to bija ērti un viegli strādāt, laiks, kas nepieciešamas transformācijas izveidošanai bija visīsākais, un tas nodrošina visplašākās iespējas lietotājam. Daļa no rīka iespējām, piemēram, transformācijas XSLT vai XQuery koda ģenerācija nav aktuāla e-komercijas risinājumiem, jo ierobežo atbalstāmos datu formātus, tomēr iespējams šī un citas iespējas var noderēt informatīvo sistēmu integrācijas vai citu līdzīgu uzdevumu risināšanā.

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] J. J. Nienhuis, C. Bryant, *E-invoicing 2010. European market guide*, Euro Banking Association (EBA) and Innopay, 2010.
- [3] *Digital Agenda for Europe. Annual Progress Report 2011*. [tiešsaiste]. http://ec.europa.eu/information_society/digital-agenda/publications/index_en.htm
- [4] *Council Directive 2010/45/EU*, 13 July 2010
- [5] G. Schneider. *Electronic Commerce, 9th Edition*, Quinnipiac University, 2011
- [6] M. Arenas, P. Barcelo, L. Libkin, F. Murlak. *Relational and XML Data Exchange*, Morgan & Claypool Publishers, 2011
- [7] R. Fagin, P. G. Kolaitis, L. Popa. *Data Exchange: Getting to the Core*, ACM Transactions on Database Systems (TODS), 2005
- [8] R. Fagin, P. G. Kolaitis, R. J. Miller, L. Popa. *Data Exchange: Semantics and Query Answering*, Theoretical Computer Science 336 (2005) 89 – 124, 2005
- [9] Ģ. Karnītis. Informācijas Sistēmu Integrācijas Problēmas. Promocijas darbs Datorzinātņu doktora zinātniskā grāda iegūšanai, Latvijas Universitāte, Rīga, 2004
- [10] Generix Group vietne. *GCI TradeXpress On Premise*. [tiešsaiste]. <http://www.gci.generixgroup.com/en/gci-tradexpress/>
- [11] *Release Note - TradeXpress Enterprise 5.0.5*, Generix Group, 2008
- [12] *TradeXpress RTE Reference*, Sonera, 2001
- [13] Altova vietne. *MapForce – Graphical Data Mapping, Conversion, and Integration Tool*. [tiešsaiste]. <http://www.altova.com/mapforce.html>
- [14] GMC vietne. *GMC Inspire overview*. [tiešsaiste]. <http://www.gmc.net/en/gmc-inspire/gmc-inspire-overview/>
- [15] Axway vietne. [tiešsaiste]. <http://www.axway.com/products-solutions/b2b/b2bi-solutions>
- [16] United Nations Rules For Electronic Data Interchange For Administration, Commerce And Transport (EDIFACT). [tiešsaiste]. http://www.unece.org/trade/untdid/texts/d422_d.htm
- [17] Syncro Soft vietne. *oXygen XML Editor*. [tiešsaiste]. <http://www.oxygenxml.com>

- [18] Progress Software Corporation vietne. *Data Integration Suite*. [tiešsaiste].
http://www.datadirect.com/products/data_integration/index.html
- [19] Tiešsaistes enciklopēdija Wikipedia. [tiešsaiste].
<http://en.wikipedia.org>
- [20] EDIFACT code list. [tiešsaiste].
<http://www.stylusstudio.com/edifact/d96a/2379.htm>

PIELIKUMI

1. pielikums. Ieejas dati uz pozīcijām bāzēts formāts

INVOICE#123001	20120112	92221144	SIA Velkonis		
LINE#1	234110	Datorpele	Logit M10	2	24.58
LINE#2	234150	Monitors	Samzun V166	1	121.99
LINE#3	234199	Disks	Segale HD11021-ATA	1	45.99
INVOICE#123002	20120112	92112233	SIA Saule		
LINE#1	234110	Datorpele	Logit M10	1	12.29
LINE#2	234197	Disks	Segale HD11025-ATA	1	49.99

2. pielikums. Ieejas dati pārveidoti EDIFACT formātā

```
UNH+1+INVOIC:D:96A:UN'  
BGM+380+123001+9'  
DTM+3:20120112:102'  
NAD+IV+92221144++SIA Velkonis'  
LIN+1++234110:SA'  
IMD+F++:::Datorpele Logit M10'  
QTY+47:2:PCS'  
MOA+38:24.58:LVL'  
LIN+2++234150:SA'  
IMD+F++:::Monitors Samzun V166'  
QTY+47:1:PCS'  
MOA+38:121.99:LVL'  
LIN+3++234199:SA'  
IMD+F++:::Disks Segale HD11021-ATA'  
QTY+47:1:PCS'  
MOA+38:45.99:LVL'  
UNS+S'  
MOA+9:192.56:LVL'  
UNT+19+1'  
UNH+2+INVOIC:D:96A:UN'  
BGM+380+123002+9'  
DTM+3:20120112:102'  
NAD+IV+92112233++SIA Saule'  
LIN+1++234110:SA'  
IMD+F++:::Datorpele Logit M10'  
QTY+47:1:PCS'  
MOA+38:12.29:LVL'  
LIN+2++234197:SA'  
IMD+F++:::Disks Segale HD11025-ATA'  
QTY+47:1:PCS'  
MOA+38:49.99:LVL'  
UNS+S'  
MOA+9:62.28:LVL'  
UNT+15+2'
```

3. pielikums. Ieejas dati atkārtoti pārveidoti ar secīgām

transformācijām TEXT > EDIFACT > TEXT

```
INVOICE#123001  20120112 92221144 SIA Velkonis          (Total:
192.56 LVL)
LINE#1    234110 Datorpele Logit M10          2  24.58
LINE#2    234150 Monitors Samzun V166        1  121.99
LINE#3    234199 Disks Segale HD11021-ATA    1  45.99
INVOICE#123002  20120112 92112233 SIA Saule          (Total:
62.28 LVL)
LINE#1    234110 Datorpele Logit M10          1  12.29
LINE#2    234197 Disks Segale HD11025-ATA    1  49.99
```

4. pielikums. Ieejas dati pārveidoti XML formātā

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Invoices>
  <Invoice>
    <ID>123001</ID>
    <IssueDate>20120112</IssueDate>
    <InvoiceTypeCode>380</InvoiceTypeCode>
    <AccountingCustomerParty>
      <Party>
        <PartyIdentification>
          <ID>92221144</ID>
        </PartyIdentification>
        <PartyName>
          <Name>SIA Velkonis</Name>
        </PartyName>
      </Party>
    </AccountingCustomerParty>
    <InvoiceLine>
      <ID>1</ID>
      <InvoicedQuantity>2</InvoicedQuantity>
      <LineExtensionAmount>24.58</LineExtensionAmount>
      <Item>
        <Name>Datorpele Logit M10</Name>
        <SellersItemIdentification>
          <ID>234110</ID>
        </SellersItemIdentification>
      </Item>
    </InvoiceLine>
    <InvoiceLine>
      <ID>2</ID>
      <InvoicedQuantity>1</InvoicedQuantity>
      <LineExtensionAmount>121.99</LineExtensionAmount>
      <Item>
        <Name>Monitors Samzun V166</Name>
        <SellersItemIdentification>
          <ID>234150</ID>
        </SellersItemIdentification>
      </Item>
    </InvoiceLine>
    <InvoiceLine>
      <ID>3</ID>
      <InvoicedQuantity>1</InvoicedQuantity>
      <LineExtensionAmount>45.99</LineExtensionAmount>
      <Item>
        <Name>Disks Segale HD11021-ATA</Name>
```

```

    <SellersItemIdentification>
      <ID>234199</ID>
    </SellersItemIdentification>
  </Item>
</InvoiceLine>
</Invoice>
<Invoice>
  <ID>123002</ID>
  <IssueDate>20120112</IssueDate>
  <InvoiceTypeCode>380</InvoiceTypeCode>
  <AccountingCustomerParty>
    <Party>
      <PartyIdentification>
        <ID>92112233</ID>
      </PartyIdentification>
      <PartyName>
        <Name>SIA Saule</Name>
      </PartyName>
    </Party>
  </AccountingCustomerParty>
  <InvoiceLine>
    <ID>1</ID>
    <InvoicedQuantity>1</InvoicedQuantity>
    <LineExtensionAmount>12.29</LineExtensionAmount>
    <Item>
      <Name>Datorpele Logit M10</Name>
      <SellersItemIdentification>
        <ID>234110</ID>
      </SellersItemIdentification>
    </Item>
  </InvoiceLine>
  <InvoiceLine>
    <ID>2</ID>
    <InvoicedQuantity>1</InvoicedQuantity>
    <LineExtensionAmount>49.99</LineExtensionAmount>
    <Item>
      <Name>Disks Segale HD11025-ATA</Name>
      <SellersItemIdentification>
        <ID>234197</ID>
      </SellersItemIdentification>
    </Item>
  </InvoiceLine>
</Invoice>
</Invoices>

```

5. pielikums. XML datnes šablons izmatošanai RTE kodā

```
<?xml version="1.0" encoding="utf-8"?>
<Invoices>
  <Invoice>
    <ID/>
    <IssueDate/>
    <InvoiceTypeCode/>
    <AccountingCustomerParty>
      <Party>
        <PartyIdentification>
          <ID/>
        </PartyIdentification>
        <PartyName>
          <Name/>
        </PartyName>
      </Party>
    </AccountingCustomerParty>
    <Note/>
    <InvoiceLine>
      <ID/>
      <InvoicedQuantity/>
      <LineExtensionAmount/>
      <Item>
        <Name/>
        <SellersItemIdentification>
          <ID/>
        </SellersItemIdentification>
      </Item>
      <Price>
        <PriceAmount/>
      </Price>
    </InvoiceLine>
  </Invoice>
</Invoices>
```

6. pielikums. Transformācija RTE programmēšanas valodā

```
#include <xmlesimple.tch>
#include <xmlesimple-util.tcf>
#define XML_TEMPLATE build("my_xml_template")
begin

nfXMLInit()
nfXMLSetPrintEncoding("ISO-8859-1")

xDoc := xfXMLUtilParseTemplateFile(XML_TEMPLATE)
xRoot := xfXMLRootNode(xDoc)

if (xDoc = NULL) then
  log("Can not parse xml template: ", XML_TEMPLATE, NL)
  exit(1)
endif

endbegin

line (1:"INVOICE#")
  xInvoice := xfDupAndMark(xRoot, "Invoice")
```

```

nfXMLPathRelSet(xInvoice,"InvoiceTypeCode","380")
nfXMLPathRelSet(xInvoice,"ID", peel(pick(1,9,7)," "))
nfXMLPathRelSet(xInvoice,"IssueDate", peel(pick(1,17,8)," "))
nfXMLPathRelSet(xInvoice,"AccountingCustomerParty/Party/PartyIdentification/ID",
peel(pick(1,26,8)," "))
nfXMLPathRelSet(xInvoice,"AccountingCustomerParty/Party/PartyName/Name",
peel(pick(1,35,EOL)," "))

nLines := 1
endline

line (1:"LINE#")
xLine := xFDupAndMark(xInvoice, "InvoiceLine")

nfXMLPathRelSet(xLine,"ID", build(nLines))
nfXMLPathRelSet(xLine,"Item/Name", peel(pick (1,17,27)," "))
nfXMLPathRelSet(xLine,"Item/SellersItemIdentification/ID", peel(pick (1,10,6)," "))
nfXMLPathRelSet(xLine,"LineExtensionAmount", peel(pick (1,47,EOL)," "))
nfXMLPathRelSet(xLine,"InvoicedQuantity", peel(pick (1,44,3)," "))
nLines++
endline

end

nfXMLCleanUpGreedy(xRoot)
nfXMLDocPrint(xDoc, TRUE)
nfXMLDeinit()

endend

```

7. pielikums. XML datne iegūta ar GMC Inspire Designer

```

<?xml version="1.0" encoding="UTF-8"?>
<Array-Invoice>
  <Invoice>
    <InvoiceNo>123401</InvoiceNo>
    <InvoiceDsc>INVOICE FOR DISTRIBUTOR</InvoiceDsc>
    <Array-InvoiceLine>
      <InvoiceLine>
        <InvoiceLineNo>1</InvoiceLineNo>
        <Qualifier>3</Qualifier>
        <ItemID>00X234110</ItemID>
        <Quantity>1</Quantity>
        <Amount>28.50</Amount>
      </InvoiceLine>
      <InvoiceLine>
        <InvoiceLineNo>2</InvoiceLineNo>
        <Qualifier>3</Qualifier>
        <ItemID>00X234150</ItemID>
        <Quantity>2</Quantity>
        <Amount>120.00</Amount>
      </InvoiceLine>
      <InvoiceLine>
        <InvoiceLineNo>3</InvoiceLineNo>
        <Qualifier>3</Qualifier>
        <ItemID>00X234199</ItemID>
        <Quantity>1</Quantity>
        <Amount>28.50</Amount>
      </InvoiceLine>
    </Array-InvoiceLine>
  </Invoice>
</Array-Invoice>

```

```

</Array-InvoiceLine>
</Invoice>
<Invoice>
<InvoiceNo>123402</InvoiceNo>
<InvoiceDsc>INVOICE FOR SUPPLIER</InvoiceDsc>
<Array-InvoiceLine>
  <InvoiceLine>
    <InvoiceLineNo>1</InvoiceLineNo>
    <Qualifier>3</Qualifier>
    <ItemID>00X234110</ItemID>
    <Quantity>2</Quantity>
    <Amount>57.00</Amount>
  </InvoiceLine>
  <InvoiceLine>
    <InvoiceLineNo>2</InvoiceLineNo>
    <Qualifier>3</Qualifier>
    <ItemID>00X234150</ItemID>
    <Quantity>2</Quantity>
    <Amount>120.00</Amount>
  </InvoiceLine>
  <InvoiceLine>
    <InvoiceLineNo>3</InvoiceLineNo>
    <Qualifier>3</Qualifier>
    <ItemID>00X234199</ItemID>
    <Quantity>2</Quantity>
    <Amount>57.00</Amount>
  </InvoiceLine>
</Array-InvoiceLine>
</Invoice>
</Array-Invoice>

```

8. pielikums. Altova Mapforce transformācijas dokumentācija

Mapping **JB_mf_t2** (JB_mf_t2.mfd)

Input **invoiceX** (invoiceX.mft)

Nodes	Connections	
File: (default) Type: string		
Root Type: [0..∞]		
Root/Invoices Type: [0..∞]		
Root/Invoices/Invoice Type: [0..∞]	<i>direct</i>	Invoices/Invoice Type: restriction of xs:anyType [1..∞]
.../Invoices/Invoice/InvoiceHeader Type: [0..∞]		
.../Invoice/InvoiceHeader/Field Type: string [0..1]		
.../Invoice/InvoiceHeader/InvoiceID Type: string [0..1]	<i>direct</i>	Invoices/Invoice/ID Type: xs:int
.../Invoice/InvoiceHeader/Date Type: string [0..1]	<i>direct</i>	Invoices/Invoice/IssueDate Type: xs:int
.../Invoice/InvoiceHeader/CustomerID Type: string [0..1]	<i>direct</i>	.../Party/PartyIdentification/ID Type: xs:int
.../Invoice/InvoiceHeader/CustomerName Type: string [0..1]	<i>direct</i>	.../Party/PartyName/Name Type: xs:string

.../Invoices/Invoice/InvoiceLines Type: [0..∞]	<i>direct</i>	Invoices/Invoice/InvoiceLine Type: restriction of xs:anyType [1..∞]
.../Invoice/InvoiceLines/Field Type: string [0..1]		
.../Invoice/InvoiceLines/LineNo Type: string [0..1]	<i>direct</i>	.../Invoice/InvoiceLine/ID Type: xs:byte
.../Invoice/InvoiceLines/ItemID Type: string [0..1]	<i>direct</i>	.../Item/SellersItemIdentification/ID Type: xs:int
.../Invoice/InvoiceLines/ItemName Type: string [0..1]	<i>direct</i>	.../InvoiceLine/Item/Name Type: xs:string
.../Invoice/InvoiceLines/Quantity Type: string [0..1]	<i>direct</i>	.../Invoice/InvoiceLine/InvoicedQuantity Type: xs:byte
.../Invoice/InvoiceLines/Value Type: string [0..1]	<i>direct</i>	.../Invoice/InvoiceLine/LineExtensionAmount Type: xs:decimal

Output **test3_sh** (test3_sh.xsd)

Connections		Nodes
		File: test3_sh.xml Type: string
		Invoices Type: restriction of xs:anyType [0..1]
Root/Invoices/Invoice Type: [0..∞]	<i>direct</i>	Invoices/Invoice Type: restriction of xs:anyType [1..∞]
.../Invoice/InvoiceHeader/InvoiceID Type: string [0..1]	<i>direct</i>	Invoices/Invoice/ID Type: xs:int
.../Invoice/InvoiceHeader/Date Type: string [0..1]	<i>direct</i>	Invoices/Invoice/IssueDate Type: xs:int
core.constant("380")	<i>direct</i>	Invoices/Invoice/InvoiceTypeCode Type: xs:short
		Invoices/Invoice/AccountingCustomerParty Type: restriction of xs:anyType
		.../Invoice/AccountingCustomerParty/Party Type: restriction of xs:anyType
		.../AccountingCustomerParty/Party/PartyIdentification Type: restriction of xs:anyType
.../Invoice/InvoiceHeader/CustomerID Type: string [0..1]	<i>direct</i>	.../Party/PartyIdentification/ID Type: xs:int
		.../AccountingCustomerParty/Party/PartyName Type: restriction of xs:anyType
.../Invoice/InvoiceHeader/CustomerName Type: string [0..1]	<i>direct</i>	.../Party/PartyName/Name Type: xs:string
.../Invoices/Invoice/InvoiceLines Type: [0..∞]	<i>direct</i>	Invoices/Invoice/InvoiceLine Type: restriction of xs:anyType [1..∞]
.../Invoice/InvoiceLines/LineNo Type: string [0..1]	<i>direct</i>	.../Invoice/InvoiceLine/ID Type: xs:byte
.../Invoice/InvoiceLines/Quantity Type: string [0..1]	<i>direct</i>	.../Invoice/InvoiceLine/InvoicedQuantity Type: xs:byte
.../Invoice/InvoiceLines/Value Type: string [0..1]	<i>direct</i>	.../Invoice/InvoiceLine/LineExtensionAmount Type: xs:decimal
		.../Invoice/InvoiceLine/Item Type: restriction of xs:anyType
.../Invoice/InvoiceLines/ItemName Type: string [0..1]	<i>direct</i>	.../InvoiceLine/Item/Name Type: xs:string
		.../InvoiceLine/Item/SellersItemIdentification

		Type: restriction of xs:anyType
.../Invoice/InvoiceLines/ItemID Type: string [0..1]	<i>direct</i>	.../Item/SellersItemIdentification/ID Type: xs:int

Constants

core.constant (-> test3_sh)	
Value	380

Mapping documentation generated by [Mapforce](#) Graphical data mapping tool <http://www.altova.com/mapforce>

DOKUMENTĀRĀ LAPA

Maģistra darbs: STRUKTURĒTU DATU TRANSFORMĀCIJAS E-KOMERCIJĀ

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

Autors: _____
(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par **pieņemotu / nepieņemotu** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs: _____
(Vadītāja paraksts)

Darbs iesniegts **maģistrantūras sekretariātā** _____.
(Iesniegšanas datums)

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

Studiju metodiķe: _____
(Metodiķes paraksts)

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

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē
_____ prot. Nr. _____, vērtējums _____
(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____
(Sekretāra paraksts)