

TALLINNA ÜLIKOOL
Matemaatika-loodusteaduskond
Informaatika osakond

**METAANDMETE HALDUS
ARUANDLUSPROTSESSIS**

Bakalaureusetöö

Autor: Marek Mägi
(allkiri) (kuupäev)

Juhendaja: Jaagup Kippar
(allkiri) (kuupäev)

Osakonna juhataja: Katrin Niglas
(allkiri) (kuupäev)

Tallinn 2006

Sisukord

| | |
|---|--------------------|
| Sisukord..... | 2 |
| Sissejuhatus..... | 4 |
| 1. Ülevaade andmelaondusest..... | 6 |
| 1.1. Ajalugu..... | 6 |
| 1.2. Definitsioon..... | 7 |
| 1.3. Omadused..... | 8 |
| 1.4. Eesmärgid..... | 9 |
| 2. Metaandmed..... | 10 |
| 2.1. Definitsioon..... | 10 |
| 2.2. Tüübid..... | 11 |
| 3. Aruandluse tarneahel..... | 12 |
| 3.1. Üldine arhitektuur..... | 12 |
| 3.1.1. Algallikad..... | 12 |
| 3.1.2. ETL-vahend..... | 13 |
| 3.1.3. Andmehoidla..... | 13 |
| 3.1.4. Andmevakad..... | 14 |
| 3.1.5. Nõustussüsteemid..... | 14 |
| 3.1.6. Väljund..... | 14 |
| 4. Kasutajad ja funktsioonid..... | 15 |
| 4.1. Laadija..... | 15 |
| 4.2. Arhitekt..... | 16 |
| 4.3. Andmeanalüütik..... | 16 |
| 4.4. Nõustussüsteemi peakasutaja..... | 16 |
| 4.5. Ärianalüütik..... | 16 |
| 4.6. Lõppkasutaja..... | 17 |
| 5. Metainfo vajadused kasutajagruppide lõikes ettevõtte X näitel..... | 18 |
| 5.1. Laadija..... | 18 |
| 5.2. Andmeanalüütik..... | 19 |
| 5.2.1. Sõbranimbrite näide..... | 21 |
| 5.3. Nõustussüsteemi peakasutaja, ärianalüütik..... | 22 |
| 5.3.1. Päringu koostamise näide..... | 23 |
| 5.4. Lõppkasutaja..... | 24 |

| | |
|---|----|
| 6. Senine olukord ettevõttes X..... | 25 |
| 6.1. Olemasolevad vahendid ja kaetud vajadused..... | 25 |
| 6.2. Hetke kitsaskohad..... | 26 |
| 7. Seatud eesmärgid..... | 27 |
| 8. Võimalikud lahendusvariandid..... | 28 |
| 8.1. Keskne metahoidla | 28 |
| 8.1.1. Eelised..... | 29 |
| 8.1.2. Riskid ja puudused..... | 30 |
| 8.2. Tootjapõhine terviklahendus..... | 30 |
| 8.2.1. Eelised..... | 31 |
| 8.2.2. Riskid ja puudused..... | 31 |
| 8.3. Metaandmete abivahend andmehoidlale..... | 32 |
| 8.3.1. Eelised..... | 33 |
| 8.3.2. Riskid ja puudused..... | 34 |
| 8.4. Andmebaasi dokumenteerimistarkvara soetamine..... | 34 |
| 8.4.1. Eelised..... | 35 |
| 8.4.2. Riskid ja puudused..... | 35 |
| 9. Analüüsi tulemused..... | 36 |
| Kokkuvõte..... | 37 |
| Resümees (Summary)..... | 38 |
| Kasutatud kirjandus..... | 39 |
| Lisa 1..... | 41 |
| 1.1. Kasutatud lühendid ja spetsiifilised terminid..... | 41 |

Sissejuhatus

Kui veel pisut üle kümnekonna aasta tagasi ei olnud haruldane ettevõtte, kus puudus igal kontoritöökohal oma arvuti, siis tänaseks oleme jõudnud sootuks teistsugusesse aega. Vähe sellest, et peaaegu iga statsionaarse töökoha lahutamatuks osaks on arvuti, on viimase aja juhtmevabade võrkude tehnoloogiline areng ja kiire levik tinginud selle, et ka üha rohkem mobiilseid töökohti on varustatud süle- või pihuarvutiga. Igal sellisel töökohal, olgu see siis mobiilne või mitte, tekitatakse päevas lugematul hulgal andmeid erinevatesse infosüsteemidesse. Enamuses tänapäeva suurtes ettevõtetes püütakse oluline osa nendest andmetest muuta infoks, mis aitaks ettevõtte juhtimis-, müügi- ja muudel üksustel oma tööd efektiivsemaks muuta. Selle teevad võimalikuks andmehoidlad ja nende pealt töötavad erinevad aruandlus- ja nõustussüsteemid. Andmehoidla näol on tegemist suure andmebaasiga, kuhu laetakse üle ettevõtte erinevatest infosüsteemidest tekkinud andmed selliselt, et sealt oleks võimalik teatud abivahenditega leida vastuseid ettevõtet puudutavatele strateegilistele küsimustele. Sellised andmebaasid on tehniliselt ja ideoloogiliselt kohandatud väga suurte andmehulkadega hakkama saamiseks. Samuti on selleks kohandatud vastavad aruandlus- ja nõustussüsteemid.

Andmete teekond nende tekkimisest konkreetse aruandeni jõudmiseks on enamasti üsna pikk. Samuti puudutab see erinevaid inimesi. Protsessijuhid mõtlevad välja, kuidas üks või teine tegevus on mõistlik ettevõttes käima panna. Infosüsteemi arendajatest sõltub mis, kuidas ja millisel kujul sellest tegevusest tekib infosüsteemidesse. Andmehoidla arendajatest omakorda sõltub mis, kuidas ja millisel kujul jõuab andmehoidlasse ja sealt tavaliselt veel edasi vaadete vahekihti. Aruande koostajast sõltub mis ja millisel kujul jõuab aruannetesse ja lõppkasutajani. Kõik etapid peavad suutma andmeid tõlgendada just selliselt nagu nad antud etapis on. Sellest tulenevalt on üsna selge, kui kriitilise tähtsusega on dokumentatsiooni ja metainfo olemasolu kogu elutsükli ulatuses. Seda nii andmekvaliteedi seisukohalt kui ka töövoogu kiiremaks ja tulemuslikumaks muutmisel.

Tänaseks on juba üsna mitmeid andmehoidlaid puudutavaid töid kirjutatud erinevates ülikoolides, kajastades enamjaolt andmehoidla loomise vaatenurgast näiteks dimensionaalse andmeanalüüsi või andmehoidla modelleerimisega seonduvat. Samas võib väita, et tänapäeval juba enamikes suurorganisatsioonides kasutuses olev andmelaondus on oma algusaegadega võrreldes küllaltki palju muutunud. Päevakorral pole enam mitte niivõrd see, kuidas ehitada andmehoidlat, vaid pigem keskendutakse kitsamatele teemadele, mis tõstaksid juba töötava

andmelao efektiivsust. Nii räägitakse viimasel ajal üha sagedamini näiteks andmekaevest, andmekvaliteedi vahenditest, aruannete visualiseerimisvahenditest, metaandmete haldusest ja teistest sarnastest teemadest.

Käesolev töö püüab anda lühiülevaate andmehoidlatest ja selle toimimisest organisatsioonis ning keskendub metainfo vajadustele ja haldamisele aruandluse tarneahelas. Töö esimeses teoreetilises pooles (peatükk 1-4) kajastatakse andmelaondusega seotud üldisi teemasid nagu ajalugu, definitsioonid, arhitektuur, kasutajad, funktsioonid jm ning praktilisema osa eesmärgiks ongi analüüsida metainfo vajadusi ettevõtte X näitel ning leida võimalikke lahendusvariante kitsaskohtadele selles vallas. Selleks üritab autor leida ja kirja panna peamised infovajadused metaandmestiku osas läbi aruandluse tarneahela, lähtudes konkreetsetest kasutajatest ja rollidest (peatükk 5). Seejärel on võimalik leida ka erinevaid lahendusi metaandmete haldamiseks (peatükk 8-9).

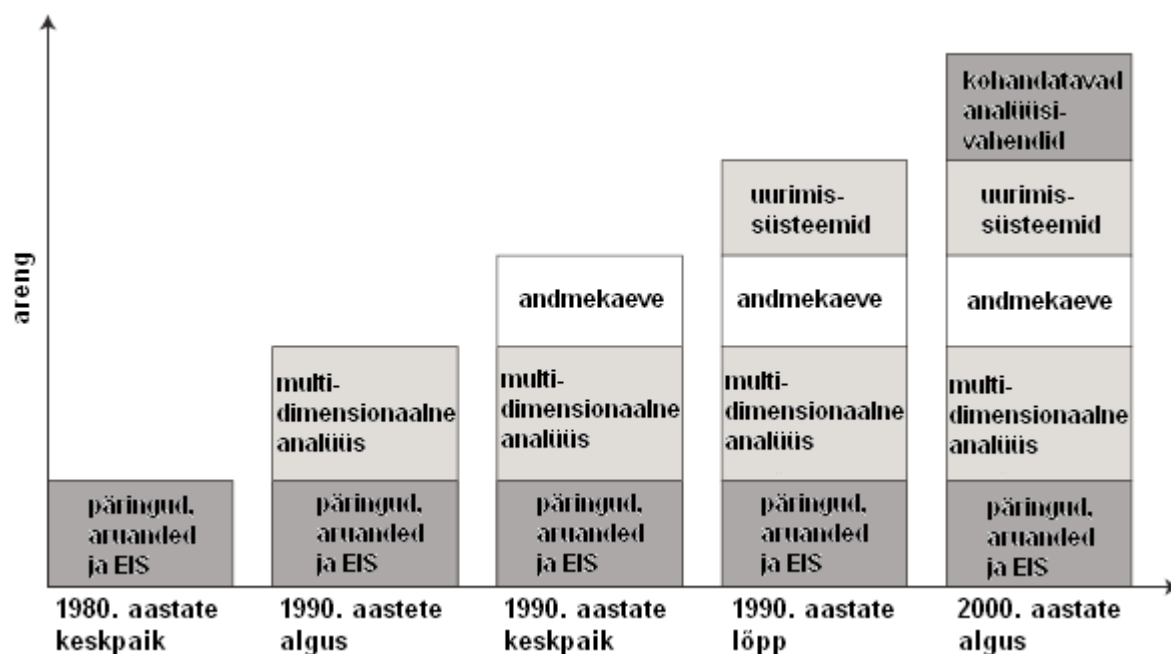
1.Ülevaade andmelaondusest

1.1.Ajalugu

Andmelaonduse ja otsuse toetamise süsteemide (*decision support systems, DSS*) juured ulatuvad kaugesse arvutite ja infosüsteemide algusaegadesse. Juba 1960. algusaastatel koostati individuaalseid rakendusi, mis töötasid juhtfaile (*master files*) kasutades. Rakendused kirjutati harilikult varastes programmeerimiskeeltes nagu Fortran või COBOL. [6]

Esimesed tavapärasest andmebaasist erinevad spetsiaalsed andmehoidlad pärinevad 1980. aastate lõpust ja 1990. aastate algusest. Neid arendati seoses kasvava vajadusega informatsiooni hallata ja analüüsida, millega ei saanud operatiivsüsteemid hakkama. [23] Umbes sellel ajal hakkasid ka laiemalt levima andmebaasihaldustarkvarad (*distributed database management system, DBMS*), mis hõlbustasid tööd andmebaasidega, muutes lihtsamaks nende loomise, haldamise ja kasutamise.

Kahtlemata oluline sündmus andmelaonduse vaatevinklist oli andmehoidla defineerimine 1990. aastal Bill Inmon'i poolt [23]. Oluline just seetõttu, et see eristas andmehoidlat tavalisest andmebaasist ja definitsioon ise on ka tänaseni igati aktsepteeritav.



Joonis 1. Ärianalüüsivahendite areng

Andmehulgad kasvasid hoogsalt. Juba 1990. aastate alguseks olid olemas ka esimesed töötavad süsteemid, mis opereerisid üle 1 terabaidiste andmehulkadega. Ühelt poolt see ja

teiselt poolt ka kasutajaskonna kiire kasvuga hakkasid kiiresti arenema ka ärianalüüsivahendid. Claudia Imhoff, Nicholas Gallema ja Jonathan G. Geiger on oma raamatus "*Mastering Data Warehouse Design*" ärianalüüsi (*business intelligence, BI*) tehnoloogilisi arenguid kajastanud graafilise pildi abil (joonis 1).

- Juhtimisinformatsioonisüsteem (*Executive Information System, EIS*)

See on vahend, mis on taganud ärijuhtidele juurdepääsu informatsioonile. Varased juhtimisinformatsioonisüsteemid nurjusid sageli tugeva toetava arhitektuuri puudumise tõttu.

- Multidimensionaalne analüüs (*Multi-Dimensional Analysis*)

Tegemist on süsteemiga, mis võimaldab mitmedimensionaalset, kontseptuaalset vaadet andmetest, kaasa arvatud toetust dimensioonide hierarhiatele

- Andmekaeve (*Data Mining*)

Tegemist on mahukate andmete analüüsiga automaatsete või poolautomaatsete vahendite abil leidmaks uusi mustreid ehk seaduspärasusi ja ootamatuid seoseid, mille sisust on võimalik välja lugeda uut ja kasulikku informatsiooni.

- Uurimissüsteem (*exploration*)

See uus tehnoloogia täiendas suuresti seniseid vahendeid võimalusega uurida andmeid erinevates vaadetes ning luua neist kiirelt ja efektiivselt prototüüpe.

- Kohandatavad analüüsivahendid (*Customizable analytical applications*)

See lisavõimalus lubas lihtsa vaevaga ja efektiivselt kohandada rakenduse funktsionaalsust vastavalt oma spetsiifilistele vajadustele. [5]

1.2. Definitsioon

Olgugi, et Eestis on andmehoidlad suhteliselt hiljaaegu kasutusele võetud ja seda ka ainult meie mõistes väga suurtes ettevõtetes, võib erinevates vastavasisulistest emakeelsetes allikates kohata lisaks terminile andmehoidla veel mitmeid teisi sünonüüme sellele. Näiteks mõeldakse üldjuhul terminite andmeait, andmevaramu ja andmelao all täpselt seda sama. Varasematel aegadel on esinenud veel mõiste informatsiooni hoidla (*information warehouse*), mida kasutasid IBM ja teised tootjad [5]. Tänapäevase vastava ingliskeelse termini *data warehouse* võttis 1990. aastal kasutusele Bill Inmon, kelle poolt välja mõeldud definitsioon sellele on samuti laialt levinud [9].

Lisaks eelnimetatud terminitele võib emakeelsetes kirjutistes kohata veel terminit andmevakk või erihoidla, mille ingliskeelne vaste oleks *data mart*. Selle all mõistetakse kohati ka pisemat iseseisvat andmehoidlat, kuid enamike allikate kohaselt see

siiski nii ei ole. Pigem on tegemist kindlale kasutajagrupile suunatud ja kindla teemaga piiritletud andmehoidla alamosaga, mille sisu pärineb tsentraalsest andmehoidlast.

Nagu ka juba eelpool mainitud, pärineb andmehoidla enim tuntud definitsioon Bill Inmon'ilt aastast 1990 ja ütleb järgmist:

Andmehoidla on andmebaas, millel on järgmised neli omadust:

- teemale orienteeritud (*subject oriented*)
- ajateisendlik (*time variant*)
- püsiv (*nonvolatile*)
- integreeritud (*integrated*)

Teemale orienteeritus tähendab, et andmed andmebaasis on koondatud ümber teemade. Ajateisendlikkus tähendab, et jälgitakse muutusi algsüsteemi andmetes ja salvestatakse need andmebaasi selliselt, et aruannetes on võimalik muutusi tagasiulatuvalt kajastada. Püsivuse all mõeldakse, et kord andmebaasi paigutatud andmed enam kunagi ei muutu. Neid ei kirjutata üle ega ka kustutata. Integreeritus tähendab, et andmebaas sisaldab ühildatud andmeid enamuse või kõigist organisatsiooni operatiivsüsteemidest. [23]

1.3.Omadused

Lisaks andmehoidla definitsioonis kajastatud ideoloogilistele omadustele omab tänapäevane andmehoidla ka rida tehnoloogilisi iseärasusi võrreldes tavalise andmebaasiga. Enamasti tulenevad need vajadusest andmebaase kohandada väga suurte andmehulkadega hakkama saamiseks.

Enamik andmehoidlatele tarkvaralahendusi pakkuvad tootjad on välja töötanud omi erinevaid lahendusi kiiruse ja kasutusmugavuse tõstmiseks. Nii võib leida iseärasusi *Teradata*, *Oracle'i*, *Sybase'i* jt vastavate tootjate vahendite andmebaasimootori tööpõhimõtetes, päringukeele funktsioonides ja mujalgi.

Andmehoidlas päritakse andmeid tavaliselt üle terve tabeli erinevalt operatiivsüsteemist, kus käib töö tavaliselt kirje tasemel. Seetõttu on üks olulisemaid küsimusi andmehoidla jaoks, kuidas tagada suurte andmehulkadega kiirem opereerimine. Peamiseks lahenduseks sellele on enamikel tootjatel paralleelsuse tagamine andmete töötlemisel ja puhverdamise võimalused, millega justkui nihutatakse raskuskeset ja optimaalse andmehulga piiri. Nii on näiteks *Teradata* välja töötanud oma pöördusmoodul-protssessorid (*access module processor, AMP*). Selle töötamise üheks eelduseks on kõigi tabelite indekseerimine ühe või mitme välja järgi nii, et indeksid jaguneksid võrdselt AMP'ide vahel ära. Iga AMP suhtleb otse kettaga, mis tagab paralleelse töötuse.

1.4.Eesmärgid

Andmelaonduse esmane eesmärk on aidata analüüsida organisatsiooni puudutavaid strateegilisi küsimusi. Üheks eraldi andmehoidlate väljatöötamise põhjusi oli ka operatiivsüsteemide andmebaaside säästmine suurtest andmepäringutest, mis häiriks tugevalt operatiivsüsteemide tööd. Samuti ei salvestata operatiivsüsteemide andmebaasidesse andmete muutumist ajas. See oleks operatiivsüsteemis kiiruse seisukohalt ebaotstarbekas. Seega üheks eesmärgiks oli kindlasti ka ajalootabelite vajadus, kuna tendentse saab ikkagi jälgida ainult pikemaajalisi perioode tagasiulatuvalt uurides, st ei piisa ainult hetkeseisu vaatlemisest. Suurte organisatsioonide puhul oli kahtlemata oluline ka vajadus andmete ühtlustamise järele. Eraldi andmehoidla näol oli selleks suurepärase võimalus andmed eri süsteemidest ühildada ja ühtse loogika alusel kokku paigutada selliselt, et andmed hakkaksid organisatsiooni jaoks kandma olulist informatsiooni.

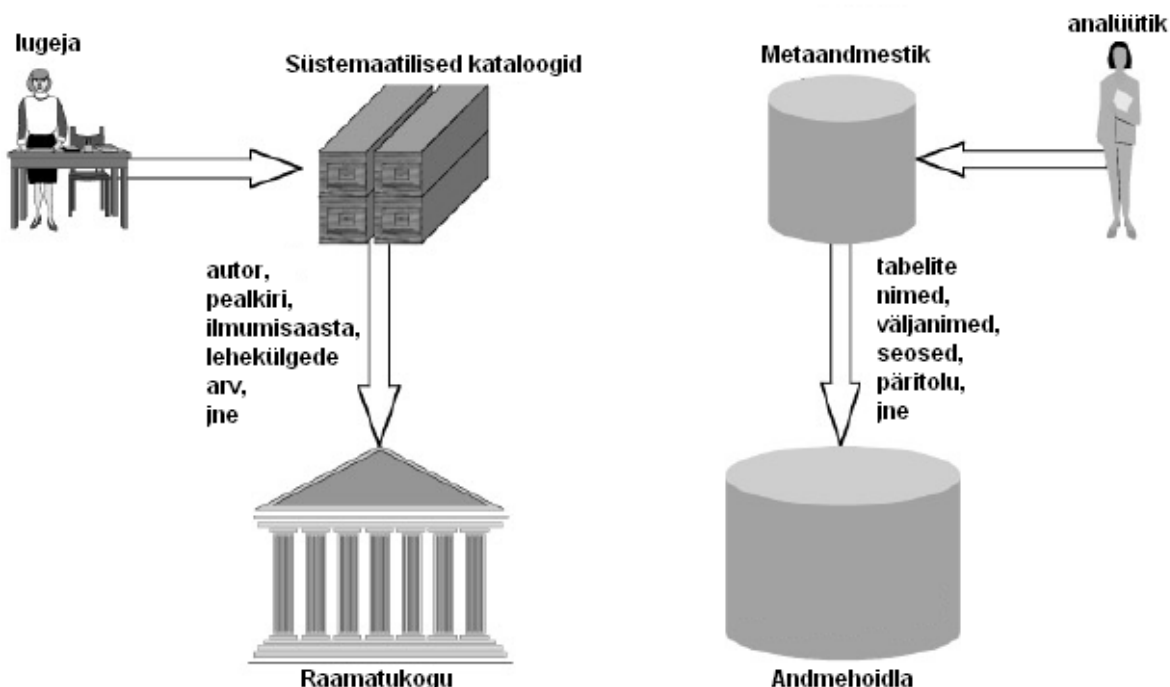
Tänapäevaks on olemas juba mitmeid eri tüüpi andmehoidlaid, mis vastavalt oma tüübile on ka oma eesmärgilt üksteisest veidi erinevad. Näitena võib välja tuua üha enam päevakorrale tõusva reaalaegse andmehoidla (*real-time data warehouse*). Selline andmehoidla peaks töötama transaktsiooni või sündmuse põhised. Seega on andmehoidla tihedalt seotud algsüsteemidega ning andmed uuenevad päästikprotsesside (*trigger*) abil kohe, kui algsüsteemis toimub vastavates andmetes muutusi. On selge, et selline andmehoidla võimaldab tema peale ehitada teistsuguseid lahendusi kui eilsete või veelgi varasemate andmetega andmehoidlale. Ometi on viimasel ajal ka üsna laialt levinud skeptilised arvamused selles osas. Vastavasisulistel seminaridel ja foorumitel leitakse, et reaalaegsed andmed pole eesmärk omaette, vaid eesmärgiks peaks seadma õigeaegsuse (*right-time*), mis enamikel juhtudel ei ole reaalaeg.

2. Metaandmed

2.1. Definitsioon

Kõige lihtsama ja levinuma definitsiooni võib sõnastada ühe lausega. Metaandmed on andmed andmete kohta. Marco D. on oma raamatus "*Building and Managing the Metadata Repository*" aga püüdnud sellele leida antud kontekstis veidi täpsemat definitsiooni. Tema definitsioon ütleb järgmist: metaandmeteks on kõik füüsilised andmed ja kirjeldused organisatsiooni seest või väljast, mis sisaldavad informatsiooni füüsiliste andmete, tehniliste ja äriprotsesside, andmestike reeglite ja piirangute ning andmestruktuuride kohta üle terve organisatsiooni.[12]

Mõistmaks paremini metaandmete olulisust andmelaonduses vaatleme hetkeks raamatukogude süstemaatilisi katalooge (joonis 2). Kataloogide kaudu on tuvastatav, millised raamatud on raamatukogus olemas ja kus need füüsiliselt paiknevad. Raamatuid on võimalik otsida teema, autori või pealkirja järgi. Samuti on kataloogikaartidel reeglina kirjas mitmeid muid andmeid raamatu kohta (lehekülgede arv, väljalaske aeg jne.) Süstemaatiliste kataloogide puudumisel oleks raamatute leidmine vaevaline ja aeganõudev.



Joonis 2. Kataloogisüsteemi ja metaandmestiku paralleel

Piltlikult võib öelda, et süstemaatiline kataloog raamatukogus on andmelaonduse mõistes metaandmestik. [12]

2.2. Tüübid

Et metaandmestikus paremini orienteeruda ja seda paremini hallata, liigitatakse metaandmeid tihti kasutuse järgi kaheks: ärilised metaandmed ja tehnilised metaandmed. Nii peaksid ärilised metaandmed aitama eelkõige ärikasutajaid e lõppkasutajaid ning näiteks kirjeldama täpsetes äriterminites, mida andmeelemendid tegelikult tähendavad. Tehnilised metaandmed peaksid hõlbustama eelkõige tehniliste kasutajate tööd, kelleks on andmehoidla arendajad ja haldajad. Sellised andmed peaksid andma muuhulgas informatsiooni algsüsteemide, andmete transformatsiooni, seoste, eraldamise, kasutuse, varundamise kohta.

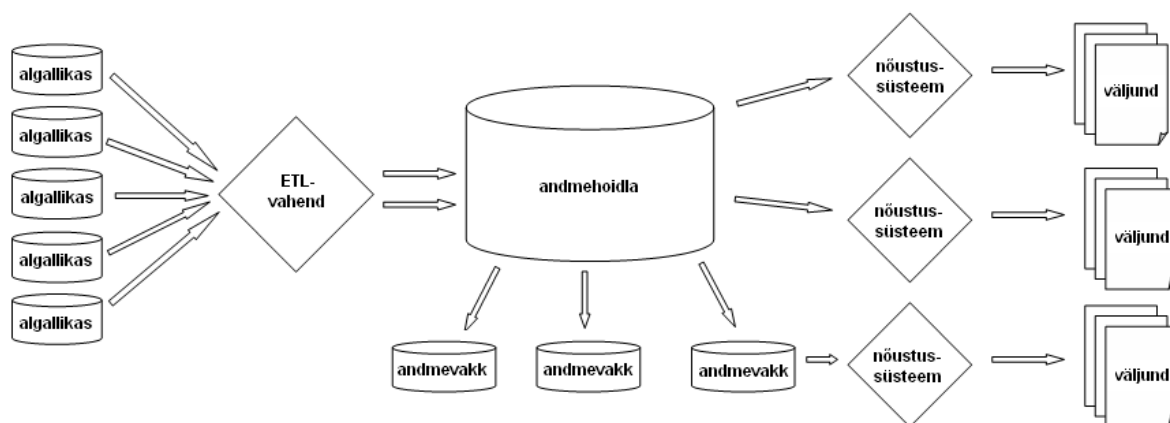
Metaandmete kasutusel siiski päris ranget joont vahele tõmmata ei saa, kuna tihti on ka tehnilisel kasutajal vaja olla kursis ärireeglite ja -terminoloogiaga ning vahel ka vastupidi.

Lisaks liigitusele kasutuse järgi võib kohata ka liigitust metaandmete tekke järgi. Nii jagunevad tehingusüsteemide, üldhoidla, erihoidlate ning hõiveprotseduuride kavandamise/projekteerimise ja modelleerimise metaandmed kavandi metaandmeteks, andmehoidla haldamise metaandmeteks ning andmehoidla kasutamise metaandmeteks [9].

3. Aruandluse tarneahel

3.1. Üldine arhitektuur

Kogu aruandlusprotsess kulgeb läbi teatud osade. Nii võib aruandeks vajalike andmete teekonda läbi aruandlusprotsessi osade nimetada aruandluse tarneahelaks. Nagu jooniselt 3 näha, on üldjuhul aruandluse tarneahelas aruandeni jõudmiseks vaja andmetel läbida 5 kuni 6 peamist etappi. Nendest tuleb põgusalt juttu järgnevatel alapunktides.



Joonis 3. Aruandluse tarneahel

3.1.1. Algallikad

Suures osas on andmehoidla algallikateks enamasti erinevad organisatsioonisisemed infosüsteemid, mida antud teema kontekstis nimetatakse ka lähte- või operatiivsüsteemideks. Operatiivsüsteemide nimetus tuleneb just sellest, et vastandada neid andmehoidlale oma funktsionaalsuselt. Teisisõnu operatiivsüsteemides tegeldakse operatiivse info vastupidiselt andmehoidlale, mis on eelkõige strateegilise informatsiooni allikas. Vahe seisneb lihtsalt selles, et operatiivinfo kajastab antud hetke, aga strateegilise info abil jälgitakse ajalugu ja trende. Nii otsime vastust küsimusele „Mis tooted on hetkel laos?“ operatiivsüsteemist, aga andmehoidlast tahame teada näiteks „Mis tooted on hooajalised?“. Selliste süsteemide puhul luuakse ühendus otse operatiivsüsteemi andmebaasi peale ja laetakse sealt vajalikud andmed vajalikul kujul ja vajaliku sagedusega andmehoidlasse. Harvem juhtub ka seda, et algallikaks on mõni organisatsiooniväline operatiivsüsteem.

Lisaks operatiivsüsteemidele võib andmete algallikaks olla ka näiteks lihtne andmefail olles siis teksti, tabeli või mõnel muul kujul. Enamik ETL-vahendeid saavad nende lugemise ja seal andmete muutumise korral automatiseeritud laadimisega suurepäraselt hakkama. Sellist

lähenemist kasutatakse näiteks juhtudel, kui kindla spetsiifilise aruande tarvis on vaja kindlat spetsiifilist andmete grupeerimist, mida ei ole otstarbekas juurutada operatiivsüsteemi pidevate aruande kasutaja vajadustest tingitud ebareeglipäraste muutuste tõttu.

3.1.2.ETL-vahend

ETL-vahendi näol on lihtsustatult tegemist vahendiga, mille abil liigutatakse vajalikud andmed andmehoidlasse. Tegelikult on see tegevus muidugi kõike muud kui lihtne. Nagu ka pikast ingliskeelsest nimest *Extract - Load - Transform* näha, tehakse enne andmete liigutamist veel rida muid tegevusi. Lugemise (*Extract*) käigus avastatakse muuhulgas kõik muutused algallikates alates eelmisest laadimisest. Teisendamise (*Transform*) faasis viiakse uued andmed vajalikule kujule (normaliseeritakse, agregeeritakse jne). Laadimise (*Load*) faasis indekseeritakse ja laetakse andmed andmehoidlasse. Lisaks on iga laadimine seotud veel kvaliteedikontrolli, konsolideerimise, filtreerimise, sorteerimise ja paljude muude tegevustega, millele siinkohal pikemat tähelepanu ei pööraks. [6]

Kui andmelaadimise algusaegadel tehti kogu see töö ära käsitsi suurte ja keeruliste SQL-lausetega, siis tänapäeval toodavad juba väga paljud andmebaaside või andmehoidlatega seotud tarkvaratootjad graafilise kasutajaliidese, automaatse koodi genereerimise, päästikprotsesside ja kontrollmehhanismidega ETL-vahendeid. Nende hulgas võib nimetada *Business Objects'i*, *Microsoft'i*, *Oracle't* jt.

3.1.3.Andmehoidla

Aruandluse tarneahela seisukohalt on andmehoidla näol tegemist tsentraalse andmebaasiga, kuhu on vastavalt kindlale üldisele mudelile ühildatult kokku kogutud organisatsiooni hulgalistest infosüsteemidest organisatsioonile strateegiliselt vajalik andmestik. Samuti säilitatakse seal ka andmete muutumine ajas. Sellest tingituna on tavaliselt tegemist väga suurte andmehulkadega, mille maht ulatub tihti terabaitideni. Andmeid andmehoidlast kasutavad erinevad ärianalüüsi vahendid otse tsentraalsest andmehoidlast või spetsiaalselt neile moodustatud füüsilistest teemale orienteeritud alamosadest e andmehoidlastest. Andmestikul andmehoidlas võib olla mitu kihti. Reeglina moodustatakse analüüsitööriistade tarbeks füüsiliste andmebaasitabelite peale veel vaadete vahekiht, mis luuakse vastavalt analüüsitööriistade vajadustele st saab veel andmeid peita, agregeerida või muul viisil teisendada.

3.1.4. Andmevakad

Andmevakk (*data mart*) on definitsiooni kohaselt andmete kogum, mis on integreeritud andmehoidla alamhulk. See on orienteeritud kindlale teemale ja reeglina suunatud ka kitsale kasutajagrupile. Üldjuhul paikneb andmevakk füüsiliselt iseseisval serveril, kuid teatud juhtudel võib see paikneda ka samas andmebaasis. Üsna tihti võib kohata seisukohti, et andmevakk võib olla iseseisev ehk sõltumatu tsentraalsest andmehoidlast, moodustades justkui ise odavama ja lihtsama alternatiivi andmehoidlale. Paraku läheb see aga selgelt vastuollu andmehoidla üldise ideoloogiaga, põhjustades varem või hiljem ka terve rea küsimusi andmekvaliteedi ja ressursi osas.

3.1.5. Nõustussüsteemid

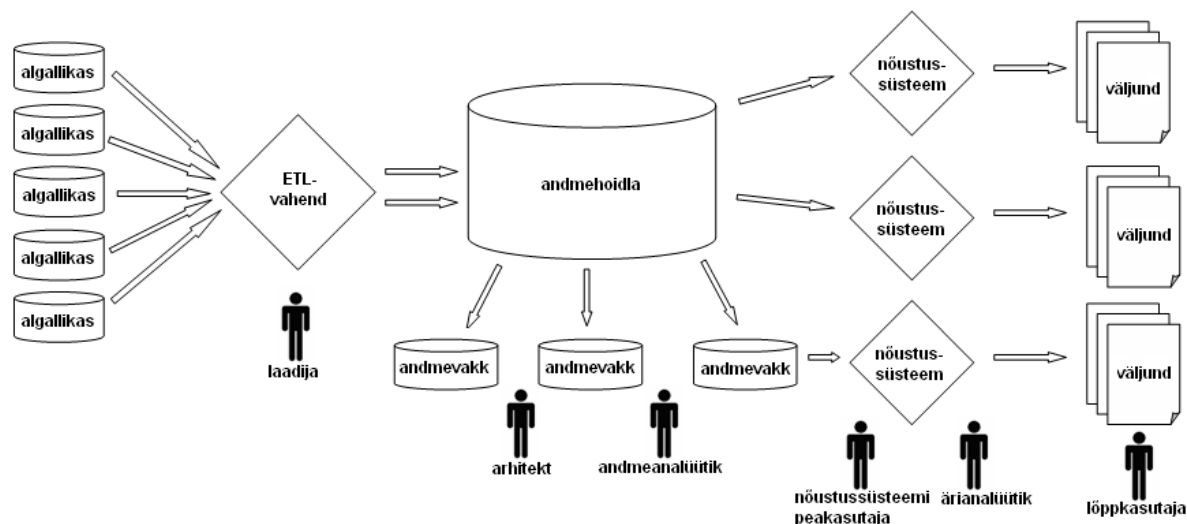
Selle termini all peab autor silmas kõikvõimalikke andmehoidla andmete pealt töötavaid analüüsi- ja aruandevahendeid. Selliste vahendite tootmisega tegeleb tänapäeval juba üsna suur hulk tarkvaratootjaid. Suuremate hulgas võib ära nimetada *Business Objects*'i, *SAS*'i, *Oracle*'i, *IBM*'i, *Sybase*'i jt. Iga vahend on mõeldud kindla spetsiifilise funktsiooni täitmiseks. Nii on näiteks müügiettevõtte turundusüksusele olemas vahend, mis aitab klientide seast hõlpsasti leida kindlate müügikampaaniate tarvis õigeid sihtgrupe ning ettevõtte tippjuhtkonnale suunatud vahend, mis võimaldab neile graafiliselt kuvada ettevõtte olulisi finantsnäitajaid, mis numbrite keeles poleks kaugeltki sama informatiivsed ja lihtsasti loetavad. Üldiselt koondatakse sarnaseid tooteid tihti ka ingliskeelsete mõistete *decision support system* (DDS) või viimase kümnekonna aastaga väga laialt levinud *business intelligence* (BI) alla. Ka antud valdkonna kontekstis tihti mainitud ja kasutatav rakenduste klass OLAP (*Online Analytical Processing*) ei ole midagi muud, kui üks kindlatele tingimustele vastav ärianalüüsivahend. [6]

3.1.6. Väljund

Aruandluse tarneahela viimane lüli on alati mingil kujul vastus otsitud küsimusele. Autor nimetaks seda lihtsalt väljundiks, kuna see võib olla nii tabelkujul keerukas erinevate mõõdikute ja dimensioonidega aruanne, lihtsalt mingi arv, graafiline diagramm või mingil muul kujul informatsioon lõppkasutajale. Väga erinev on ka periood, mille lõikes jälgitakse andmete muutusi väljundis. Suuresti kõigub see vahemikus ühest päevast kuni ühe kuuni. Enamasti jälgitakse lihtsalt seda, kuidas miski asi kulges ja oskuslikult küsimusi püstitades, on võimalik ka prognoose teha tuleviku tarbeks.

4. Kasutajad ja funktsioonid

Andmehulgad ja aruandlusvajadused erinevates organisatsioonides on väga erinevad. Osalt ilmselt tingituna organisatsioonide suuruse erinevustest ja teisalt kindlasti ka tegevusvaldkonnast. Samuti mängib oma rolli siin ka organisatsiooni arengutase. Seepärast ei ole ilmselt olemas ka ühte õiget lahendust kasutajate tööjaotuse organiseerimiseks, vaid edu tagamiseks selles osas on vaja eeskätt lähtuda organisatsiooni iseärasustest. Seega näitab autor siin üht võimalikku varianti, tuginedes antud peatükis osaliselt ka ettevõttes X hästi toimivale näitele kasutajate ja nende funktsioonide kirjeldamisel. Kuna käesoleva töö üheks eesmärgiks on analüüsida metainfo vajadusi läbi aruandluse tarneahela ja leida nendele võimalikke lahendusi, siis ei käsitle autor siin andmelaonduse riistvaralise poolega tegelevaid kasutajaid. Nemad üldjuhul ei puutu üldse reaalsete andmetega kokku. Andmetega tegelevaid kasutajaid ja nende paiknemist aruandluse tarneahelas illustreerib allolev joonis 4.



Joonis 4. Kasutajad aruandluse tarneahelas

4.1. Laadija

Nagu ka jooniselt 4 paiknemise järgi näha, on laadija tööriistaks ETL-vahend. Selle abil kirjutatakse valmis vajalike andmete laadimispaketid, mis kindlas koosseisus ja kindlate päästik- ja kontrollmehhanismide abil kindla regulaarsusega käivitatakse. Laadija vastutabki selle eest, et kõik vajalikud laadimised toimuksid õigeaegselt ja terviklikult. Tema ei süüvi andmete tähendusse. Sisend informatsiooniga, millised andmed, kui tihti ja kuhu peavad jõudma, tuleb andmeanalüütiku käest.

4.2. Arhitekt

Arhitektil on oluline roll täita andmehoidla loomise ja hiljem vähesemal määral ka arendamise juures. Arhitekt peaks töötama välja ja tagama kõik üldised reeglid, andmestruktuurid, mudelid ja standardid. Tema ei kuulu otseselt andmehoidla meeskonda, vaid organisatsiooni üldise IT arhitektuuri meeskonda. See tagab olukorra, et arhitekt näeb organisatsiooni terviklikult ja koostöös andmeanalüütikuga suudab välja töötada ja tagada ka andmehoidlas võimalikult analoogse pildi asjadest, mis kattuvad muu organisatsiooniga. Samuti välistab see ka kord juba kusagil organisatsioonis katsetatud ja ennast mitte õigustanud tegevused selles vallas.

4.3. Andmeanalüütik

Väga kokkuvõtlikult öeldes tegeleb andmeanalüütik andmehoidla arendamise ja haldamisega (teatud juhtudel võib haldamisega tegeleda ka eraldi kasutaja). Tema tagab selle, et kõigil lõppkasutajatel ja ärianalüütikutel oleks võimalik andmehoidlast saada kätte neile vajalikku infot. Selleks otsib üles andmestiku algallikatest või vajadusel tellib vastava arenduse algsüsteemile, et vajalikud andmed hakkaksid sinna tekkima. Seejärel leiab uutele andmetele õige koha ja kuju andmelao andmestiku füüsilises mudelis ja kirjutab laadijale ette skeemid, kuidas ja kuhu need andmed laadida. Kui andmed on jõudnud andmehoidlasse, loob andmeanalüütik vajalikud vahekihid nõustussüsteemidele, kus on võimalik lõppkasutajale aruanded koostada.

4.4. Nõustussüsteemi peakasutaja

Teatud nõustussüsteemid tekitavad veel ka endale andmetest vahekihi, mida nad kasutavad. Seda nii piisava kiiruse tagamiseks, aga ka andmete omavaheliseks seostamiseks. Neid vahekihte on vaja pidevalt täiendada ja hallata. Selleks ongi vajalikud teatud nõustussüsteemidel peakasutajad, kes hoolitsevad vahekihi eest ja aitavad lõppkasutajatel oma aruandeid nende pealt kätte saada. Näitena sellisest nõustussüsteemide vahekihtidest võib tuua välja *Business Objects*'i universumid (*universe*), kuhu ehitatakse kontekstipõhised seosed andmetest, mida on ühe aruande raames võimalik kasutada. Teise näitena võib välja tuua *SAS Marketing Automation* 'i, mis kasutab päringute kiiruse tagamiseks omaenda andmebaasi.

4.5. Ärianalüütik

Ärianalüütik on kasutaja, kes ei vaata ainult valmisaruandeid, vaid kasutades vastava suunitlusega nõustussüsteeme, analüüsib olemasolevaid andmeid. Need vahendid

võimaldavad reeglina graafilise kasutajaliidese abiga koostada ärianalüütikul üsna keerulisi päringuid oluliselt ise SQL-päringukeelest aru saamata. Näiteks võib jälle tuua *SAS Marketing Automation*'i, mis turundusanalüütikutel võimaldab läbi mängida erinevaid müügikampaaniaid.

4.6.Lõppkasutaja

Aruandluse tarneahela viimane kasutaja ongi lõppkasutaja. Tema on kasutaja, kes reeglina tahab näha päris valmisaruannet. Seda kuvatakse talle üldjuhul ka veebiliidese või mõne tabelarvutusprogrammi abil, st lõppkasutaja ei oma reeglina ka mingit spetsiaaltarkvara aruannete vaatamiseks. Lõppkasutaja rollis esinevad tihti organisatsiooni juhtivtöötajad, protsessihaldurid ja vahel ka ärianalüütikud.

5. Metainfo vajadused kasutajagruppide lõikes ettevõtte X näitel

Nagu juba ka sissejuhatuses mainitud, on andmehoidla efektiivsuse ja kvaliteedi seisukohalt äärmiselt oluline roll metainfo olemasolul üle terve aruandluse tarneahela. Järgnevates alapunktides toob autor välja infovajadused kasutajate lõikes, et saaks seejärel analüüsida hetkeolukorda, leida puudujäägid ja võimalusel leida üldistatud lahendus. Siinkohal ei käsitle autor eraldi arhitekti vajadusi, kuna tema ei osale aktiivselt igapäevases aruandluse tarneahelas ega puutu üldjuhul kokku ka nende andmetega.

5.1. Laadija

Kuigi laadija vastutab üldjuhul ainult laadimispakettide töötamise eest ja laadimisskeemid kirjeldab ära andmeanalüütik, on võimalike vigade ilmnemisel tal siiski sarnane vajadus võrreldes andmeanalüütikuga. Vastavad vajadused võib grupeerida allika järgi kolmeks. Need oleksid järgmised:

- algallika metaandmed
- ETL-vahendi metaandmed
- andmehoidla metaandmed

Algallika kohta peab laadija nägema järgmist informatsiooni:

- andmete tekkimise loogika

Laadijal on vaja teada, milliseid tabeleid tuleb jälgida uute andmete tekkimisel. Vahel kajastab laadimiseks vajalikku informatsiooni (aeg, tabel, väli) andmete tekkimisest hoopis eraldi tabel.

- andmete muutmise loogika

Sarnaselt andmete tekitamise informatsiooniga on laadijal vaja teada, kust saab jälgida mingite andmete muutmist.

- andmete kustutamise loogika

Väga oluline inkrementaalsete laadimiste koostamisel on arvestada ka andmete võimaliku kustutamisega.

- tabelite väljade tüübid

Väljatüüpe on laadijal vaja jälgida laadimispakettides andmeid vajalikule kujule viies või lihtsalt kontrolliks, et välistada andmete paigutamist kohta, kuhu nad ei sobi.

ETL-vahend on laadija peamine töörist ja seega näeb laadija sealt talle vajalikku informatsiooni igapäevaselt ning ei vaja mujalt lisainformatsiooni. Seega ei hakka siinkohal täpseid vajadusi ka lahkama.

Andmehoidla kohta peab laadija nägema järgmist informatsiooni:

- tabelite väljade tüübid

Samadel põhjustel nagu algallika väljatüüpe peab laadija nägema ka andmehoidla väljatüüpe.

- tabelite väljade vastavusskeem algallikaga (*mapping*)

Suvalise andmehoidla andmeosa kohta peab olema võimalik näha, millisest algallikast ta pärineb.

5.2. Andmeanalüütik

Nagu ka jooniselt 4 näha, asub andmeanalüütiku rollis kasutaja aruandluse tarneahela mõttes andmete teekonna keskel. Sellest tulenevalt puutub andmeanalüütik kokku ka tarneahela mõlemas otsas asuvate kasutajatega ja esineb tihti justkui vahendaja või tõlgina algsüsteemi arendaja ja aruande tellija vahel. Seetõttu on andmeanalüütiku vajadused metainfo osas ka kõige suuremad võrreldes teiste kasutajatega.

Samuti nagu laadija puhul võib andmeanalüütiku peamised metainfo vajadused grupeerida allika järgi kolmeks. Need oleks järgmised:

- algallika metaandmed
- ETL-vahendi metaandmed
- andmehoidla metaandmed

Algallika kohta peab andmeanalüütik nägema järgmist informatsiooni:

- andmete tekkimise loogika

Andmeanalüütikul on laadimisskeemide kirjutamisel oluline teada, mis tegevuse peale kuhu ja kui palju andmeid tekitatakse. Näiteks võib algsüsteemi kasutajaliides kiiruse ja lihtsuse huvides töötada selliselt, et kasutaja mingi nupuvajutuse peale kirjutatakse üle ka need väljad, mida tegelikult ei muudetud. Sellisel juhul tuleb hiljem andmeid analüüsides välistada need kirjed, mille uus ja vana väärtus on sama.

- andmete muutmise loogika

Muutuste puhul on sarnaselt tekitamisegagi vaja andmeanalüütikul hilisema analüüsi huvides mõista muutuste tagamaid.

- andmete kustutamise loogika

Andmete kustutamise reeglid on samuti olulised õige käsitlemise huvides. Samas on väga oluline inkrementaalsete laadimiste kirjeldamisel.

- tabelite väljade tüübid

Algsüsteemi väljade tüübid on samuti vajalikud laadimisskeemide kirjeldamisel. Kuna andmed andmehoidlasse tekivad erinevatest süsteemidest, siis tuleb defineerida andmehoidlasse optimaalne väljatüüp.

- tabelite ja andmete omavahelised seosed

Tihti ei ole võimalik andmebaasi luua täiesti relatsioonilise andmebaasina. Seega ei paista seosed kusagilt välja ning tuleb ära kirjeldada kõik juhud ja tingimused, kuidas andmed omavahel seotud on.

- tabelite väljade sisulised vasted ehk ärinimed ja definitsioonid

Väga oluline roll on ühtsete äriterminite defineerimisel üle kogu organisatsiooni. See tagab kasutajate üksteise mõistmine ja töövoogu kiirema funktsioneerimise läbi kogu aruandluse tarneahela. Vastasel juhul võib tekkida olukord, kus mõni kasutaja räägib oma väljakujunenud terminite keeles, mõni kasutajaliidesest tulenevast keeles, mõni andmehoidla väljanimedate keeles, mõni algsüsteemi väljanimedate ja tegelike vajaduste väljaselgitamine kujuneb omaette aeganõudvaks ülesandeks.

ETL-vahendi kohta peab andmeanalüütik nägema järgmist informatsiooni:

- muutused ehk transformatsioonid andmetes

Harva laetakse andmed andmehoidlasse täpselt selliselt nagu nad eksisteerivad algsüsteemides. Seetõttu on oluline kirjeldada ära kõik muutused, mis andmetega laadimise käigus tehakse.

- laadimispakettide käivitamise reeglid

On oluline teada, kui tihti või mille alusel mingeid andmeid andmehoidlasse laetakse. Enamasti on laadimised mingi konstantse regulaarsusega, kuid teatud andmeid laetakse muudatuse peale algsüsteemist.

- laadimispakettide käivitamise ajalugu

Laadimispakettide käivitamise ajalugu on oluline eelkõige andmekvaliteedi kontrolliks. Peab olema võimalus vaadata, kas plaanitud laadimised on ka korrektselt lõpuni viidud või on seal esinenud tõrkeid.

Andmehoidla kohta peab laadija nägema järgmist informatsiooni:

- tabelite väljade tüübid

See info on eelkõige vajalik uute andmete sobitamisel olemasolevasse füüsilisse andmemudelisse andmehoidlas veendumaks kas ja kuidas on neid võimalik kohandada.

- tabelite ja andmete omavahelised seosed

Sarnaselt algsüsteemidega tuleb ära kirjeldada kõik juhud ja tingimused, kuidas andmed omavahel seotud on.

- tabelite väljade sisulised vasted ehk ärinimed ja definitsioonid

Samuti nagu algsüsteemi puhul on ka andmehoidlas vajalik andmeosad vastavusse seada organisatsiooni üldiste äriterminitega.

- tabelite väljade vastavusskeem algallikaga (*mapping*)

Suvalise andmehoidla andmeosa kohta peab olema võimalik näha, millises algallikast ta pärineb.

- reeglistik uute objektide loomiseks

Siin peab autor silmas kõiki reegleid, mida on võimalik üldistada uute andmebaasiobjektide loomisel. Näiteks lühendite kavand, mudeldusreeglid jne.

- tabelite ja vaadete kasutamise ajalugu

Seda infot on vaja eelkõige andmebaasiobjektidesse muudatuste tegemisel. Samuti on selle info alusel võimalik eemaldada aegunud või kasutuks muutunud objekte.

- tabelite kettale liigendamise ehk partitsioneerimise info

Väga suurte andmetabelite puhul kasutatakse liigendamist, mis piltlikult jagab tabeli mingi indeksi alusel loogilistele ketastele laiali. Päringu kirjutamisel tuleb seda arvestada päringu kiiruse huvides.

- indekseerimisinfo

See info on samuti oluline eelkõige päringu kiiruse tagamise seisukohalt.

5.2.1.Sõbranumbrite näide

Et eelnev jutt ei jääks liiga teoreetiliseks vaatleme siinkohal ka üht näidet metaandmete vajalikkusest andmete tekkimise ja muutmisloogika osas. Tabelist 1 on näha, et mingil

telefonitootele on omistatud 3 sõbranumbrit (numbrid valitud juhuslikud). Lähemal uurimisel selgub, et mõnedel juhtudel on tabelis muutusena kajastatud kirjel jäänud tegelikult number samaks. On üsna selge, et selline kirje ei kanna andmehoidla jaoks mingit kasulikku informatsiooni. Pigem muudab selline olukord hilisema aruandluse sõbranumbrite lisamise, muutmise ja lõpetamise osas üsna keerukaks. Igati mõislik on juba andmete andmehoidlasse laadimise käigus tõsta kokku need kirjed, kus number on jäänud samaks ja teha seda selliselt, et alguskuupäev võetakse esimeselt ja lõpukuupäev viimaselt kirjelt.

Alati ei pruugi analoogne olukord andmetele peale vaadates kohe välja paista ning metaandmete kirjeldamine omaks sellises olukorras veelgi kriitilisemat tähtsust.

| TOODE_ID | TRIBUUT | NUMBER | ALGUS_KP | LOPP_KP |
|----------|---------|---------|-----------|-----------|
| 4565644 | SOBER1 | 7733921 | 4.01.2002 | 3.06.2002 |
| 4565644 | SOBER1 | 7733921 | 4.06.2002 | 8.02.2003 |
| 4565644 | SOBER1 | 4477505 | 9.02.2003 | |
| 4565644 | SOBER2 | 3847441 | 4.01.2002 | 3.06.2002 |
| 4565644 | SOBER2 | 3847441 | 4.06.2002 | 8.02.2003 |
| 4565644 | SOBER2 | 3847441 | 9.02.2003 | |
| 4565644 | SOBER3 | 4797146 | 4.06.2002 | 8.02.2003 |
| 4565644 | SOBER3 | 4797146 | 9.02.2003 | |

Tabel 1. Sõbranumbrid

5.3.Nõustussüsteemi peakasutaja, ärianalüütik

Sarnaste vajaduste tõttu andmeid kajastava informatsiooni osas võib siinkohal käsitleda nõustussüsteemi peakasutajat ja ärianalüütikut koos ühe sihtgrupina. Kui vaadelda seda sihtgruppina samadel alustel kui eelmisi, siis allika järgi grupeerituna võib metaandmete vajadused jagada samuti kolmeks:

- ETL-vahendi metaandmed
- andmehoidla metaandmed
- nõustussüsteemi metaandmed

ETL-vahendi kohta peavad nõustussüsteemide peakasutajad ja ärianalüütikud nägema järgmist informatsiooni:

- muutused ehk transformatsioonid andmetes

Samuti nagu andmeanalüütikul on vaja ka sellel sihtgrupil näha kõiki muutusi, mis andmetega laadimise käigus tehtud on.

- laadimispakettide käivitamise reeglid

Andmete uuendamise sagedusest sõltub ka aruannete uuendamise sagedus. Tihti ka vastupidi, st võimalusel lähtutaksegi laadimise regulaarsuse paika panemisel aruandluse vajadustest.

- laadimispakettide käivitamise ajalugu

Vigade otsimisel aruannetes on oluline esmalt veenduda, kas andmed on korrektselt andmehoidlasse kohale jõudnud.

Andmehoidla kohta peavad nõustussüsteemide peakasutajad ja ärianalüütikud nägema järgmist informatsiooni:

- tabelite väljade tüübid

Aruannetes on vahel vajalik andmeid näidata erinevalt algkujust. Seega on antud sihtgrupil vaja veenduda, kas ja kuidas on võimalik andmeid kohandada. Samuti on väljatüüp oluline piirangute ja lisatingimuste seadmisel nõustussüsteemis.

- tabelite ja andmete omavahelised seosed

Aruannete koostamisel on vaja teada kõiki juhte ja tingimusi, kuidas andmed omavahel seotud on.

- tabelite väljade sisulised vasted ehk ärinimed ja definitsioonid

Aruannetes on kindlasti vaja järgida ühtseid äritermineid andmete kuvamisel.

- tabelite kettale liigendamise ehk partitsioneerimise info

Tingimuste seadmisel nõustussüsteemides tuleb seda arvestada päringu kiiruse huvides.

- indekseerimisinfo

See info on samuti oluline kiiruse tagamise seisukohalt tingimuste seadmisel nõustussüsteemises.

Nõustussüsteemide kohta peavad nõustussüsteemide peakasutajad ja ärianalüütikud nägema järgmist informatsiooni:

- aruannete käivitamise ajalugu

Peab olema võimalus analüüsida, kui tihti ja kui suur hulk kasutajaid mingit aruannet käivitab.

5.3.1.Päringu koostamise näide

Et tõsta kirjeldatud metaandmete vajadustele veidi konkreetsust, vaatleme järgnevalt ka üht näidet. See illustreerib tabelite liigendamise kirjeldamise vajadust. Suurtes ettevõtetes on

arvete ja arveridadega seotud tabelid väga suured ja tihti liigendatud näiteks lisavälja periood järgi. Vaatlemegi nüüd näitena sellist olukorda ja päringut (päring 1), millega tahetakse näidata detsembri esimesel nädalal üle 10 000 kr väärtuses arveldanud kliente. Selline lihtne päring võib halvemal juhul käia tabeli suure mahu tõttu mitmeid kümneid minuteid. Kui aga lisame päringule (päring 2, rida 6) ainult üheainsa tingimuse, mis määrab ära ka perioodi, siis olenevalt ketta loogilise jaotuse suurusest saame tulemuse kordi või isegi kümneid kordi kiiremini. Andmebaasi päringu optimeerija oskab perioodi veeru järgi kergesti alles jätta vaid vastavad read, st kiiresti saavad välistatud mittesobivad read ning päringu edasine täitmine toimub juba oluliselt väiksema andmehulga pealt. Ligikaudu sama tulemuse võiks anda ka klasterdatud indeksi tegemine arvete kuupäevaltuba järgi.

```
Päring 1.  SELECT a.kliendi_id, k.kliendi_nimi
           FROM arved a, kliendid k
           WHERE a.kleindi_id = k.kliendi_id
              AND a.arve_kp BETWEEN '2006-12-01' AND '2006-12-07'
              AND a.summa > 10000
           GROUP BY 1,2
           ;
```

```
Päring 2.  SELECT a.kliendi_id, k.kliendi_nimi
           FROM arved a, kliendid k
           WHERE a.kleindi_id = k.kliendi_id
              AND a.arve_kp BETWEEN '2006-12-01' AND '2006-12-07'
              AND a.summa > 10000
              AND a.period = 200612
           GROUP BY 1,2
           ;
```

5.4.Lõppkasutaja

Lõppkasutajal pole otseselt vaja metaandmete osas muud, kui näha aruannetes kajastatavatele ärinimedele vastavaid definitsioone. Soovituslikult võiks olla lisaks veel aruandlusprotsessist paremini aru saavatel lõppkasutajatel võimalus jälgida laadimiste ajalugu.

6.Senine olukord ettevõttes X

6.1.Olemasolevad vahendid ja kaetud vajadused

Hetkel töötab ettevõttes X enamus andmehoidla algallikatena kasutuses olevaid algsüsteeme *Oracle*'i andmebaaside peal. Vähesel määral leidub ka teisi andmebaasiservereid nagu näiteks *MySQL*, *PostgreSQL*, *Microsoft SQL* jt. Samuti leidub mõni algallikas faili kujul, mis on reeglina käsitsi muudetav klassifikaatorite tabel kindla spetsiifilise aruande jaoks. ETL-vahendina kasutatakse Microsofti poolt toodetud vahendit *Data Transformation Services (DTS)*. Ettevõttes kasutusel olev andmehoidla töötab Teradata vastavas serveris. Nõustussüsteemidena võib ettevõttest leida erineva suunitlusega *Business Objects*'i aruandemoduleid, *Hummingbird*'i analüüsivahendeid, *SAS*'i analüüsivahendeid jm. Suur osa väljunditest kuvatakse *Business Objects*'i veebipõhise klientprogrammi kaudu filtreeritavate aruannetena või eksporditakse mõnda tabelarvutusfaili. Teatud osas kasutatakse ka *Business Objects*'i „paksu” klientprogrammi, mis on oma võimalustelt oluliselt mitmekülgsem. Samuti võib kohata andmehoidla väljundeid erineval kujul veel mujalgi. Näiteks kuvatakse ettevõtte siseportaali ettevõtte jaoks võtmetähtsusega mõõdikuid, kasutades selleks andmehoidlas spetsiaalselt kirjeldatud makrode vahekihti.

Senine olukord ja olemasolevad vahendid võimaldavad kasutajatel metaandmeid lugeda üsna vähesel määral. Osaliselt on algsüsteemidel olemas koht, kuhu on võimalik kirjeldada metaandmestikku. Seni on osaliselt kirjeldatud algsüsteemide järgmised metaandmed:

- tabelite ja andmete omavahelised seosed
- tabelite väljade sisulised vasted
- tabelite väljade tüübid

ETL-vahendi metaandmete osas on seni võimalus vaadata järgmist informatsiooni:

- pakettide käivitamise ajalugu

Andehoidlat puudutavad metaandmed on seni olemas järgmises osas:

- tabelite väljade tüübid
- tabelite väljade vastavusskeem algallikaga (*mapping*)
- tabelite ja vaadete kasutamise ajalugu

Kuna nõustussüsteemide metaandmete vajadused esinevad sisuliselt ainult peakasutajate huvides, siis võib öelda, et ka need vajadused on kaetud, kuna enamik selliseid vahendeid võimaldavad juba ise ajaluliselt vaadelda aruannete käivitamist kasutajate lõikes.

6.2.Hetke kitsaskohad

Seni üldse kirjeldamata metaandmete osa on üsna suur. Välja tuleks tuua järgmised kitsaskohad selles osas:

- algallika andmete tekkimise, muutmise ja kustutamise loogika
- laadimisel tehtavad muutused ehk transformatsioonid andmetes
- laadimispakettide käivitamise reeglid
- tabelite ja andmete omavahelised seosed
- tabelite väljade sisulised vasted ehk ärinimed ja definitsioonid
- reeglistik uute objektide loomiseks
- tabelite kettale liigendamise ehk partitsioneerimise info
- indekseerimisinfo

Lisaks puudulikele andmetele leidub puudujääke ka olemasolevate metaandmete kajastamises. Seosed, mis on kirjeldatud algsüsteemide kohta ei ole ühtsel kujul ja ühest kindlast kohast leitavad. Samuti kuna *Microsoft DTS* vahendil puudub automaatne dokumenteerimise funktsionaalsus ehk aruandlussüsteem, mis võimaldaks kiiresti lokaliseerida vajalike andmete asukoha ja seosed erinevates laadimispakettides ja võimaldaks seda infot loetavalt kasutada ka teistel andmehoidla kasutajatel, kirjeldatakse hetkel ka andmehoidla andmete vastavusskeemid käsitsi tabelitesse. See aga ei ole täna enam selgelt rahuldav lahendus.

Kõik need puudused tingivad olukorra, kus üsna suures osas andmehoidlal tehtavad toimingud on vaevalised ja aeganõudvad ning tihti on erinevateks analüüsitöödeks vajalik kaasata ka algsüsteemide arendajaid.

7.Seatud eesmärgid

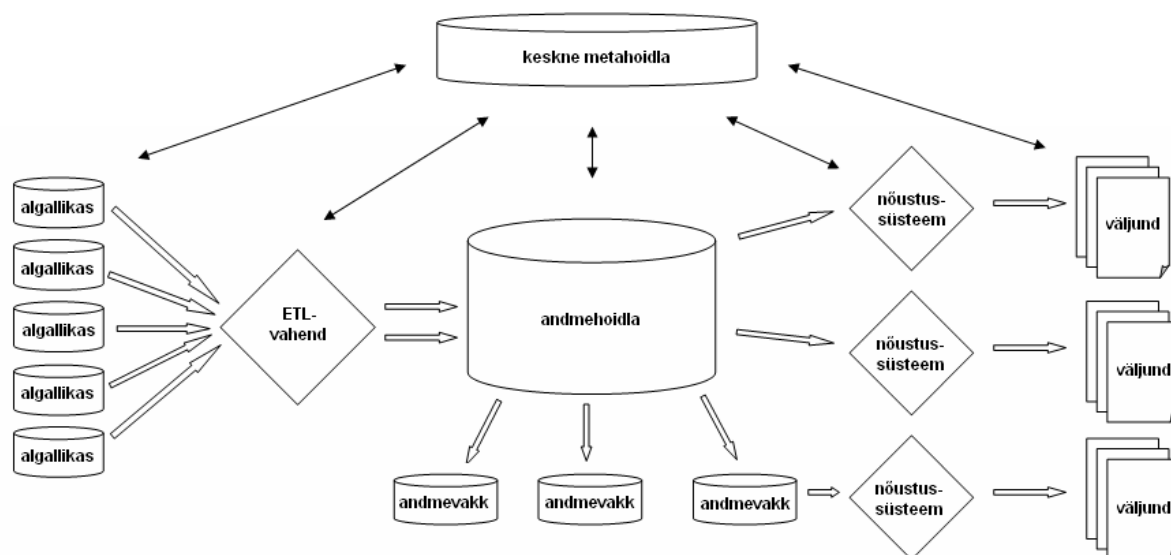
Peamiseks eesmärgiks on tutvuda erinevate lahendusvariantidega ning leida nende seast ettevõttele sobiv lahendus või suund, kuhu poole püüelda. Lahendustevariantide analüüsi käigus peaks selguma, kas probleemidele on võimalik üks terviklahendus, mis kataks ära kõik kitsaskohad või peaks probleemidele lähenema üksikutest vajadustest või kasutajagruppidest lähtuvalt. Kuna ettevõtte on suur, andmed jooksevad kokku erinevatest üksteisest otseselt sõltumatutest üksustest ning ärireeglid on pidevas muutuses, peavad aruandlusprotsessi kõik osad olema võimalikult läbipaistvad, hästi dokumenteeritud ja standardiseeritud. Võimaliku lahenduse leidmisel peaks metaandmestik võimaldama suuremal hulgal kasutajatest oma aruannete koostamisega ise toime tulla. Samuti peaks seeläbi vähenema aruandlusprotsessi teistes osades märgatavalt aja- ja tööjõukulu. Lihtsustama peaks testimine, haldamine ja arendustööde läbiviimine andmehoidlas. Hea lahenduse leidmisel peaks kindlasti arvestama ka seda töömahtu, mis on vajalik metaandmete tekitamisel. Teisisõnu peaks püüdlema võimalikult automatiseeritud lahenduse poole nii, et käsitsi andmete kirjeldamine oleks vajalik ainult seal, kus teisiti ei saa.

8.Võimalikud lahendusvariandid

Selles peatükis käsitleb autor mõningaid võimalikke lahendusvariante seatud eesmärkidele. Lahendused on oma mahult erinevad ja sellest tulenevalt katavad ära ka erineva hulga vajadustest. Lahenduste kajastamisel ei aeta taga suurt tehnilist detailsust, vaid pigem üritatakse anda põgus ülevaade selle olemusest. Iga lahenduse juures kirjeldatakse selle eeliseid ja puudusi.

8.1.Keskne metahoidla

Üheks võimalikuks lahendusvariandiks tagamaks kõikidele kasutajatele neile vajaliku metainfo vajadused, on ehitada keskne metahoidla (*metadata repository*). See oleks piltlikult nagu iseseisev metaandmetega andmehoidla, kust on võimalik lugeda ja tekitada vajalikku metainfot igal aruandluse tarneahela lülil (joonis 5). Nii näiteks tekitavad ja kasutavad algallika arendajad samu äritermineid nagu lõppkasutajad oma aruannetes. Seega ei teki olukorda, kus näiteks mõiste „toode” all mõeldakse organisatsiooni erinevates osades erinevaid asju või tehingukuupäeva nimetatakse igas aruandes isemoodi. Metahoidla oskusliku väljaehitamise ja arendamisega, peaks olema võimalik katta ära praktiliselt kõik eelpool kirjeldatud metaandmete vajadused.



Joonis 5. Aruandluse tarneahel ja metahoidla

Suures organisatsioonis on keskse metahoidla juurutamine kahtlemata äärmiselt ressursimahukas ettevõtmine. Seda nii ajaliselt, tööjõult kui ka riistvaraliselt. Samas on see ka olulisel määral kõigist teistest aruandluse tarneahelas kasutatavatest vahenditest ja neis

kasutatavatest andmestandarditest ning eksportimisvõimalustest. Küsimus seisneb suuresti ka selles, kui palju andmeid tuleb metahoidlasse kirjeldada käsitsi ja kui palju on võimalik automaatselt kätte saada.

Metahoidla juurutamisel tuleks suuremat tähelepanu pöörata mitmetele aspektidele. Need oleksid järgmised:

- vahenditega tutvumine ja hindamine

Metaandmete vahendid võib liigitada kahte gruppi. Integreerimisvahendite abil on võimalik erinevad metaandmed kesksesse metahoidlasse toimetada ning haldusvahendid võimaldavad uurida ja hallata andmeid metahoidlas.

- vajaduste selgitamine

Kasutajate vajaduste täpne uurimine ja selgitamine. Kuidas neid ühendada erinevate algallikate metaandmetes, on kriitilise tähtsusega eduka metahoidla loomisel.

- projektplaani koostamine

Samm sammu haaval tuleks ära kirjeldada projekti tähtsamad faasid ja sellest tekkinud lahendamist vajavad ülesanded.

- arhitektuuri väljatöötamine

Selle käigus tuleks paika panna metahoidla peamised arhitektuurilised elemendid ja kirjeldada, kuidas seotakse omavahel andmehoidla arhitektuur ja metahoidla arhitektuur.

- andmekvaliteedi tõstmine

Projekti käigus peaks tähelepanu pöörama ka asjaolule, et metaandmete abiga on võimalik kontrollida andmete kvaliteeti andmehoidlas. Äriksutajatel peaks vähenema võimalus andmeid valesti tõlgendada, tehnilistel kasutajatel peaks lisanduma rida võimalusi andmete terviklikkuse kontrolliks jne.

- metamudeli loomine

Kahtlemata on metamudeli loomine üks keerulisemaid ja aeganõudvamaid etappe. Selle kvaliteedist sõltub suuresti ka kogu projekti edu.[12]

8.1.1.Eelised

Luues oskuslikult metamudeli, on selle lahenduse puhul võimalik kirjeldada ja kajastada kogu metaandmete vajaduste osa ettevõttes. Võimalik on ehitada metahoidla suhteliselt universaalsete tehnoloogiate abil, mis tagab konkreetsest tootjast ja platvormist sõltumatus. Kuna kogu aruandluse tarneahela senised komponendid jäävad endiseks ja lisandub lihtsalt

eraldiseisev metahoidla, siis ei kaasne juurutamisega vajadust midagi olemasolevate laadimispakettide või aruannete osas ringi teha.

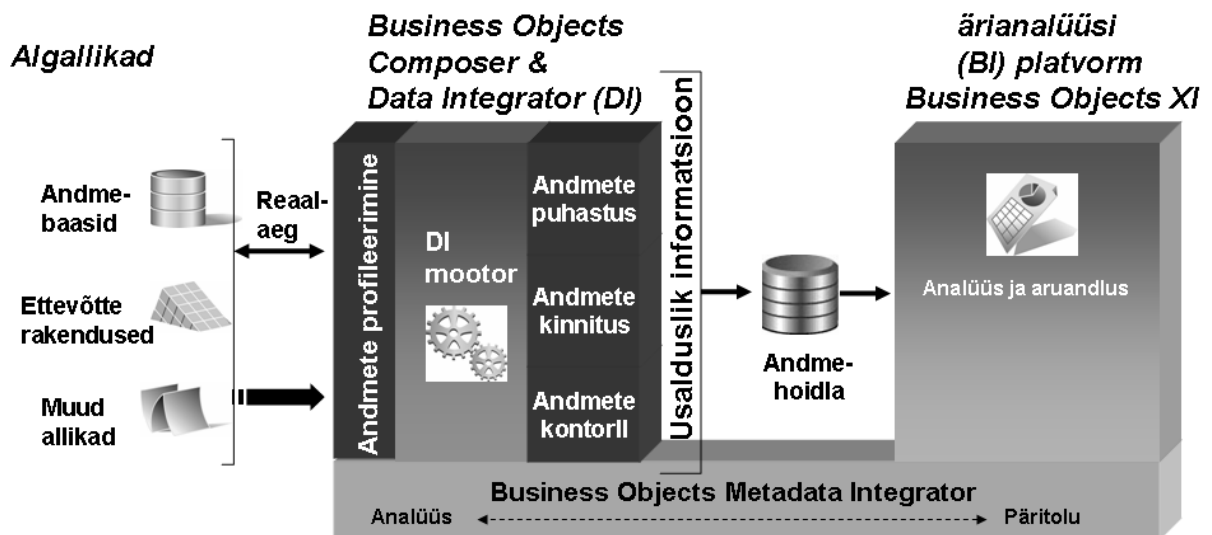
8.1.2.Riskid ja puudused

Sellise projekti käivitame ja lahenduse juurutamine on äärmiselt ajamahukas ja töörohke ettevõtmine ning kujuneb seeläbi ettevõttele ka päris kulukaks. Projekti venimise või hoopis ebaõnnestumise riski suurendab ajaolu, et projekti on raske kaasata välispartnerit, kes suudaks adekvaatselt hoomata ja abistada probleemide puhul erinevate tehnoloogiatega seotud aruandlusprotsessi osade juures algallikatest aruandlusvahenditeni. Samuti puudub ettevõttesiseselt kogemus sarnase projekti läbiviimisel. Automatiseeritud metaandmete tekitamise metahoidlasse võib osutada äärmiselt keerukaks just erinevate tehnoloogiatega olemasolu tõttu aruandluse tarneahela lõikes.

8.2.Tootjapõhine terviklahendus

Antud lahendus põhineb ühe tootja vahenditel kogu aruandlusprotsessi ulatuses. See tähendaks, et peale algallikate ja andmehoidla enda, oleks tarneahel ühe tootja vahenditega kaetud. Seeläbi on tagatud tihe seos kõikide tarneahela komponentide vahel. See võimaldab muuta andmete liikumise läbipaistvaks ja hästi dokumenteeritavaks.

Sellist lahendust pakub *Business Objects*, kelle tooted *Business Objects Composer*, *Business Objects Data Integrator* ja *Business Objects XI*. Esimesed kaks toodet on seotud andmete laadimisega ja moodustavad üheskoos piltlikult ETL-vahendi. *Composer* on abivahend laadimistöö korraldamiseks ja dokumenteerimiseks. Seal defineeritakse ärivajadused, kaardistatakse algandmed ja kirjeldatakse andmete vastavustabelid (*mapping*). Sealt minnakse edasi *Data Integrator*i'sse, mille abil viiakse läbi reaalsed andmete laadimised. *Business Objects XI* näol on tegemist aruandluskeskkonnaga. Andmetest tekkivaid ja kirjeldatavaid metaandmeid jagatakse kogu lahenduse ulatuses. Siin tuleb appi kogu lahenduse mõttes võtmetähtsust omav abivahend ehk *Business Objects Data Integrator*'i komponenttoode nimega *Metadata Integrator*, mis installeeritakse *Business Objects XI* juurde. See tagab sünkroonsuse metaandmetes ETL-vahendite ja ärianalüüsi platvormi vahel. Joonisel 6 ongi üritatud visualiseerida eelneva jutu sisu. [3]



Joonis 6. *Business Objects*'i lahendus

Sarnaselt metahoidla lahendusega kataks antud lahendus ära olulise osa kogu metaandmete vajadusest ettevõttes. Tarnija hinnangul oleks see ligikaudu 90%.

8.2.1. Eelised

Selle lahenduse suurimaks eeliseks oleks garanteeritud automaatne tehniliste metaandmete teke. Äriterminid ja definitsioonid tuleb suvalise lahenduse puhul ühekordselt käsitsi kirjeldada. Kuna projekt viidaks läbi tihedas koostöös tarkvaratarnijaga, siis ei kaasne sellega ka olulisi riske kompetentsi leidmise osas.

8.2.2. Riskid ja puudused

Suurim risk on vanalt süsteemilt uuele migreerumises. Eeldatavalt tuleb kõik senised laadimised otsast peale uuesti teha. Samuti tuleb suures osas aruandlus ringi teha. Vanade asendamisel uutega võib tulla esile igasugu loogilisi ja tehnilisi viperusi. Migreerimine ise kestab seetõttu ka suhteliselt pikalt ning vajab palju ettevõttesisest tööjõudu. Lisaks tuleb soetada terve rida tarkvaralitsentse misstõttu projekti hind kujuneb üsna kalliks. Uute spetsiifiliste vahendite kasutuselevõtu puhul tuleb kindlasti ka terve hulga töötajate koolitamisega arvestada. Kuna antud lahenduse puhul tagab metainfo olemasolu mitmete ühe tootja komponentide olemasolu aruandluse tarneahelas, jääks ettevõtte neist tugevalt sõltuma. Lahendus ei võimaldaks kajastada informatsiooni algallika andmete muutumise loogika.

8.3. Metaandmete abivahend andmehoidlale

Kuna hetkel on ettevõttes andmehoidlana kasutatud *Teradata* serverit, siis ühe lahendusena, on võimalus soetada lisaks *Teradata* poolt välja töötatud abivahend metaandmete haldamiseks andmehoidlates. Seda lahendust nimetab tootja ise *Teradata Meta Data Services'ks (MDS)* ja see on optimeeritud töötama just *Teradata* teiste andmehoidla vahenditega. Peamisteks eeldusteks ongi andmehoidlana *Teradata* serveri versioon V2R6.0 või V2R5.1 ning operatsioonisüsteemina on sertifitseeritud *UNIX MP-RAS 3.02*, *Windows XP*, *Windows 2000* ja *Windows server 2003*. Kasutajaliideste töötamiseks on vajalik *Internet Explorer 6.0*, *Netscape 7.1* või nendest uuemad versioonid [13]. Need eeldused on ettevõttes kenasti täidetud.

Antud abivahendi juurutamisele peaks tootja sõnul mõtlema ettevõtte, kelle majandustöötajatel on tekkinud küsimused nagu

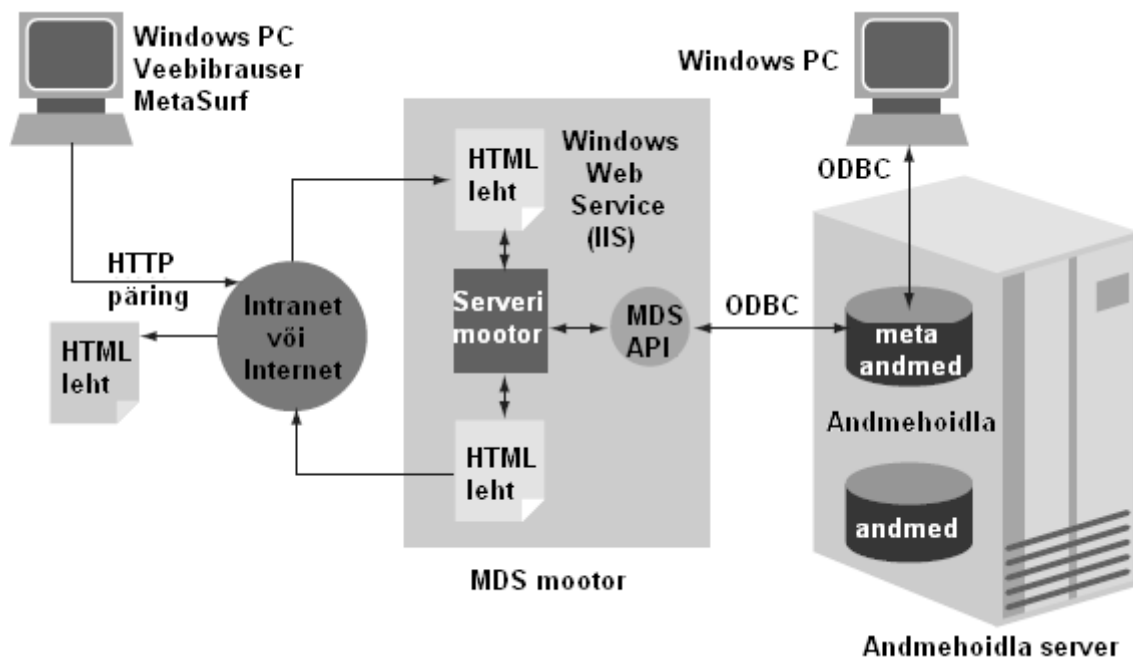
- Mis andmetele on mul juurdepääs?
- Mis on andmete kirjeldused äriterminites?
- Kust andmed pärinevad?
- Mis on andmete lubatud väärtused?
- Kus asuvad äriküsimustele vajalikud andmed?
- Millised on olnud andmemuundused andmete laadimisel andmehoidlasse?
- Kui uued on andmed andmehoidlas?

ning kus tehnilised töötajad otsivad vastuseid küsimustele nagu

- Millised andmeida piirkonnad kasvavad ja kui kiiresti need kasvavad?
- Kes, kui tihti ja mis andmeid kasutab?
- Kas leidub liigseid andmeid?
- Millal ja kuidas andmeid viimati uuendati?
- Kuidas mõjub andmete muutmine või kustutamine teistele tabelitele, vaadetele või rakendustele? [13]

Viimase versiooni uuendustena märgitakse veel eraldi ära makrode toetus, mis võimaldab kasutajatel vaadelda ja analüüsida nende seoseid. Samuti on täiendatud seoste kirjeldamist välja tasemel ning objektidele lisatakse nüüdsest alati ka nende looja. [14]

Kogu lahenduse üldine lihtsustatud arhitektuur näeb välja analoogne nagu on kujutatud joonisel 7.



Joonis 7. Teradata MDS'i arhitektuur

Kasutatud on *Internet Information Server*'it (IIS), mis suhtleb andmehoidla serveriga üle *ODBC (Open Database Connectivity)* draiveri. Vahepealne lüli *MDS API (MDS Application Programming Interface)* ongi vajalik metaandmete objektide loomiseks, muutmiseks ja kustutamiseks. Nagu ka jooniselt 6 on võimalik välja lugeda, on peamine kasutajaliides *MetaSurf* veebipõhine. See on mõeldud salvestatud andmete otsimiseks, vaatlemiseks ja analüüsimiseks. Lisaks kuulub lahenduse koosseisu veel kaks kasutajaliidest, milleks on *MetaBrowse* ja *MetaManager*. Esimene pakub *MMC*'ga ühilduva nõ puuvaadet, mis võimaldab metaandmeid detailsemalt vaadata. Teine võimaldab metaandmete halduril kasutajaid ja õigusi seadistada ning muid MDS-süsteemi parameetreid muuta. [14]

8.3.1. Eelised

Antud lahendus kataks samuti ära päris arvestatava osa metaandmete vajadusest. Kuna andmehoidla server on samuti *Teradata* oma, on metaandmete automaatne tekkimine samuti garanteeritud. Kuna puudub vajadus olemasolevate laadimispakettide ja aruannete osas midagi ringi kirjutada, lahendusega ei kaasne hulgaliselt litsentside ostmist ning abivahendi enda keerukus ei eelda ilmtingimata töötajate koolitust, siis võib lahenduse lõplik hind kujuneda üsna soodsaks. Ettevõtte on olemas tootjapoolne kontakt ja tehniline tugi, mis kindlasti kiirendab ja lihtsustab lahenduse juurutamist ettevõttesse.

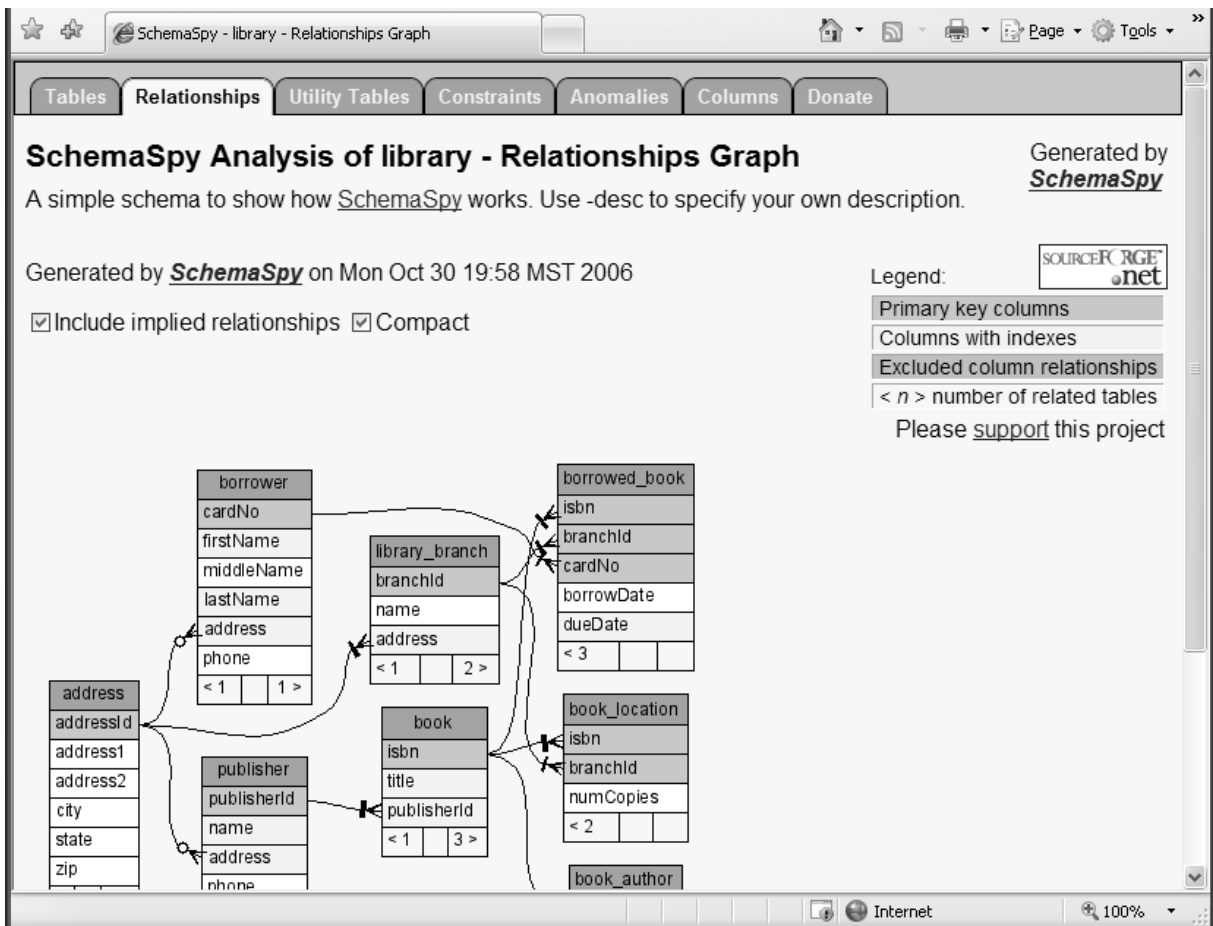
8.3.2.Riskid ja puudused

Kui peaks tekkima vajadus vahetada andmehoidla server teise tootja omaga, tekib probleeme metaandmete eksportimisega uude süsteemi. Lahendus ei võimaldaks kajastada informatsiooni algallika andmete muutumise loogika ja laadimispakettide käivitamise osas.

8.4.Andmebaasi dokumenteerimistarkvara soetamine

Juhul, kui ükski eelnev lahendusvariant ei peaks realiseeruma ressursimahu tõttu, siis on olemas ka lihtsamaid lahendusi. Üheks ilmselt kõige lihtsamaks lahenduseks parandamiseks veidigi senist olukorda metaandmete osas, on leida sobiv andmebaasi dokumenteerimistarkvara. On olemas hulgaliselt nii vabavaralisi kui ka tasulisi dokumenteerimiskeskondi andmebaasidele, mis enamjaolt on veebipõhise väljundiga. Üheks selliseks on näiteks avatud lähtekoodiga *SchemaSpy*, millel tänaseks väljas versioon 3.1. Kui täna kirjeldatakse enamuse andmehoidlat puudutavat informatsiooni päris käsitsi *Microsoft Excel*'i failidesse, siis *SchemaSpy* suudab vähemalt suure osa andmehoidla metaandmeid automaatselt üle JDBC (*Java Database Connectivity*) ühenduse andmebaasist välja lugeda. Lisaks otsib ja avastab programm mitmeid anomaaliaid nagu indeksita tabelid, ühe väljaga tabelid jne ning leiab üles võimalikule seosele viitava olukorra, kus välja nimi ja tüüp viitab mõne teise tabeli primaarvõtme väljale.

Kasutajaliides (joonis 8) võimaldab lihtsasti liikuda tabelite, väljade ja seoste vahel ning kuvada erinevaid vaateid metaandmetest. [20]



Joonis 8. *SchemaSpy* kasutajaliides

Kuna tegemist on avatud lähtekoodiga tarkvaraga, mis on kirjutatud laialt levinud programmeerimiskeeles *Java*, siis on seda ka lihtsa vaevaga võimalik kohendada oma vajadust mööda. Samuti on võimalik ära kasutada lisavälju kirjeldamiseks sinna näiteks andmete päritolu ehk algallika andmeid.

8.4.1. Eelised

Selle lahenduse kõige suurimaks plussiks on ilmselt hind. Kuna tegemist on lihtsa vabavaralise tootega, siis ei kaasne sellega olulisi kulutusi ning seda on võimalik mingis osas ka vajadustega kohendada.

8.4.2. Riskid ja puudused

Oluliseks miinuseks on vähene vastavus vajadustele. Päris suur osa metaandmetest jäaks kirjeldamata, mis puudutavad algsüsteeme, ETL-vahendit ja andmehoidlat ennast. Selle lahenduse puhul puuduks ettevõttel tehniline tugi ja partner väljastpoolt.

9. Analüüsi tulemused

Kõigepealt peaks ära mainima, et töös kajastatud lahendused ei ole kindlasti ainuvõimalikud, kuid siinsel juhul ja olukorras tundusid autorile need lahendused kaalumist väärivad ning seetõttu keskendutigi just neile. Mõnes muus olukorras võiksid kaalumist vajada hoopis teistsugused lahendused. Näiteks teiste tootjate terviklahendused, ise lihtsa veebipõhise dokumenteerimisvahendi loomine vms.

Nagu ka eelpool selgeks sai, olid oma head ja vead olemas kõigil kajastatud lahendustel. Valiku tegemisel peab autor kõige määravamaks tasakaalu kahes aspektis. Ühelt poolt kaasnevad kulutused lahenduse juurutamisel ja teiselt poolt saadav tulu vajaduste osas pärast seda. Sellest lähtuvalt peab autor keskse metahoidla lahendust ja tootjapõhist terviklahendust antud hetkel liiga töömahukas. Viimase ehk andmebaasi dokumenteerimistarkvarast ei näe autor aga piisavalt palju tulu. Seega pidas autor antud olukorras kõige otstarbekamaks lahenduseks andmehoidla metaandmete abivahendi *Teradata Metadata Services (MDS)* soetamist. Otsusele aitas kaasa ka asjaolu, et erinevalt andmebaasi dokumenteerimistarkvarast, on antud tarkvara mõeldud spetsiaalselt andmehoidlale ning selle abiga on võimalik optimaalsete kulutustega katta ära enamik kriitilise tähtsusega metaandmete vajadustest. Lahenduse kasuks räägib ka asjaolu, et *Teradata* andmehoidla serverina on ennast hästi õigustanud ja on tagatud ka piisav tootjapoolne tehniline tugi. Samuti ei jää sellise lahenduse puhul ETL-vahend ja aruandlussüsteemid tootjast sõltuvaks ning nende osas ei kaasne seoses muudatustega andmehoidlas mingit lisatööd.

Kokkuvõte

Vastavalt seatud eesmärgile sai töö teoreetilises osas kajastatud mitmeid peamisi andmelaondusega seotud teemasid nagu ajalugu, definitsioonid, arhitektuur, kasutajad, funktsioonid jm. Seega julgeb autor loota, et seni rohkem andmehoidla arendusele keskendunud tööde kõrval andis käesolev töö oma esimeses pooles piisavalt ülevaatliku pildi ka andmehoidla toimimisest ja kogu aruandlusprotsessist üldse. Kogu teoreetilise jutu kokkuvõtteks võib lühidalt öelda, et andmehoidla on juba mõnikümmend aastat tänapäeval kujul eksisteerinud eelkõige suurte organisatsioonide abivahend, kuhu koondatakse kokku organisatsiooni jaoks strateegilise tähtsusega andmed. See erineb tavapärastest andmebaasist nii ideoloogiliselt kui ka tehniliselt lahenduselt. Kui me räägime andmelaonduse toimimisest, peame vaatlema aruandlusprotsessi terviklikult kogu aruandluse tarneahela ulatuses. Nii kajastab ka käesolev töö protsessi erinevaid osi, nendega seotud kasutajaid ning funktsioone. Töö teoreetilise osa püstitatud eesmärgiks oli leida esmalt erinevate andmehoidla kasutajagruppide vajadused metaandmestiku osas. Seejärel leida puudujäägid nende osas ettevõtte X näitel ning lõpuks püüda leida hea lahendus probleemsetele kohtadele. Viimase eesmärgi täitmiseks tutvuti töös mitmete erineva kaaluga lahendustega ja püüti leida nende hulgas sobivaim. Autori arvates antud ettevõttele hetkel kõige sobivamaks lahenduseks kujunes andmehoidla metaandmete abivahend *Teradata Metadata Services (MDS)*, mis kujutab endast veebiväljundiga spetsiaalset andmehoidla dokumenteerimistarkvara.

Resümee (*Summary*)

Data warehousing that are used in most big organizations nowadays has made quite a big change compared to beginning years. In modern thinking the main worry isn't a fact how to build data warehouse, instead how to make already working data warehouse systems more flexible, faster and accurate. And that leads us to expressions like data mining, data quality equipment, reporting visualization, metadata managing and others, which are used more and more frequently.

Headline of the present work should be: „Managing Metadata through Reporting Process“. Following material tries to give a short overview of data warehousing and the way it is inculcated in organizations while takes a closer look at a narrow subject like metadata managing is. The first part of the work is theoretical. In chapter 1 history of data warehouse, its definitions, qualities and purposes will be reviewed. In second chapter you'll find definition and qualify of metadata. Chapters 3 and 4 are focused on reporting process and give a review of its architecture, users and functions.

In practical part of the work analyses the needs of metadata of company X and several different solutions, which are very different considering capacity of resource and arise from that should cover the needs of the company as good as possible. In 5th chapter a needs of metadata for company X will be discovered and in 6th chapter the situation of the metadata will be analyzed. In chapter 7 the purposes will be brought up the possible solution should solve and eventually in chapter 8 four possible solutions will be briefly reviewed.

Conclusion of the work (chapter 9) should lead us to optimum solution to manage company's metadata and a direction the company should move towards.

Kasutatud kirjandus

1. Almeida M.S. & Ishikawa M. & Reinschmidt J. & Roeber T. (1999) "Getting Started with Data Warehouse and Business Intelligence", California, IBM Corporation, International Technical Support Organization
2. Ashok K. Nag, (2006) „Managing with metadata”, Teradata Magazine, 09.2006, vol.6,no 3, lk. 10
3. Business Objects (2006) "Data Integrator Advanced Development and Migration Guide", Business Objects
4. Firestone, J.M. (1997) "Data Warehouses and Data Marts: A Dynamic View", kättesaadav: <http://www.dkms.com/papers/dwdmdv.pdf>, 07.06.2006.
5. Imhoff C. & Galemno N. & Geiger J.G. (2003) "Mastering Data Warehouse Design Relational and Dimensional Techniques", Indianapolis, Indiana, Wiley Publishing, Inc.
6. Inmon W.H. (2005) "Building the Data Warehouse, Fourth Edition", Indianapolis, Indiana, Wiley Publishing, Inc.
7. KeeleWeb. (2005) „IT terministandardi sõnastik”, kättesaadav: <http://keeleeveeb.edu.ee>, 2.10.2006.
8. Kimball, R. & Ross, M. (2002) "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd Edition", New York, John Wiley & Sons, Inc.
9. Leis P. (1999) "Andmehoidlad", kättesaadav: <http://deephought.ttu.ee/aa/modules.php?name=News&file=article&sid=2>, 13.11.2006
10. Longman, C. (2004) "Data Warehouse Lifecycle Management: Concepts and Principles". kättesaadav: <http://www.dmreview.com/whitepaper/WID1002691.pdf>, 18.06.2006.
11. Marco D. & Jennings M. (2004) "Universal Meta Data Models", Indianapolis, Indiana, Wiley Publishing, Inc.
12. Marco D. (2000) "Building and Managing the Metadata Repository: A Full Lifecycle Guide", New York, John Wiley & Sons, Inc.
13. NCR Corporation (2005) "Teradata Meta Data Services - Release Definition", NCR Corporation
14. NCR Corporation (2004) "Teradata Meta Data Services - The Encyclopedia for Your Data", NCR Corporation
15. Orr, K. (2000) „Data Warehousing Technology”, kättesaadav: www.kenorrist.com/dwpaper.html, 2.10.2006.
16. Paim, F.R.S. & Castro, J.F.B. "DWARF – An Approach for Requirements Definition and Management of Data Warehouse Systems", kättesaadav: http://www.di.ufpe.br/~ler/publicacoes/pub_2003/RE03_DWARF.pdf, 07.06.2006.

17. Perkins, A. (1998) "Data Warehouse Architecture: A Blueprint for Success",
kättesaadav: <http://www.dmreview.com/whitepaper/dwb.pdf>, 18.06.2006.
18. Politano, A.L. "Salvaging Information Engineerin Techniques in a Data Warehouse
Environment". kättesaadav: <http://www.dmreview.com/whitepaper/wid321.pdf>,
07.06.2006.
19. Poole J. & Chang D. & Tolbert D. & Mellor D. (2002) "Common Warehouse
Metamodel", New York, John Wiley & Sons, Inc.
20. SchemaSpy, "Graphical Database Schema Metadata Browser", kättesaadav:
<http://schemaspy.sourceforge.net>, 14.12.2006
21. Tozer G. (1999) "Metadata Management for Information Control and Business
Success", Boston, London, Artech House, Inc.
22. Vallaste, H. (2000) „E-teatmik”, kättesaadav: <http://www.vallaste.ee>, 2.10.2006.
23. Wikipedia, the free encyclopedia „Data warehouse”, kättesaadav:
http://en.wikipedia.org/wiki/Data_warehouse, 12.11.2006

Lisa 1

1.1. Kasutatud lühendid ja spetsiifilised terminid

| | |
|---|---|
| operatiivsüsteem (<i>operative system</i>) | informatsiooni kogumiseks, töötlemiseks, salvestamiseks, lugemiseks ja käitlemiseks |
| andmebaasimootor (<i>database engine</i>) | mõeldud rakendus andmebaasihalduri keskne funktsionaalne osa, mis salvestab ja otsib andmeid |
| inkrementaalsed laadimised (<i>incremental loading</i>) | laadimised, kus lisatakse ainult lisandunud ja uuenenud andmed |
| JDBC (<i>Java Database Connectivity</i>) | Javapõhine andmebaasipöördus |
| MMC (<i>Microsoft Management Console</i>) | Microsoft'i halduskonsool, millel on graafiline kasutajaliides (GUI) ja programmeerimisraamistik. Konsoole kasutatakse Windows'i põhise riistvara, tarkvara ja võrgutarkvara komponentide haldamiseks |
| ODBC (<i>Open Database Connectivity</i>) | avatud andmebaasipöördus - standardiseeritud meetod, mis võimaldab pöörduda mistahes tüüpi andmebaasi poole mistahes rakendusprogrammist ja seda sõltumatult andmebaasihalduri (DBMS) tüübist |
| päringukeel (<i>query language</i>) | spetsiaalne keel päringute tegemiseks andmebaasile |
| päästikprotsessid (<i>trigger</i>) | protsess, mis põhjustab protseduuri automaatse käivitumise |
| transaktsioon (<i>transaction</i>) | infovahetuse ja sellega seotud tegevuste jada |