

### References

1. Ozols, Y. and Borisov, A. (1997), "Pattern Classification and Feature Extraction on the Basis of Composition of Fuzzy Relations." *Proceedings of the Seventh International Fuzzy Systems Association World Congress, IFSA'97*, Vol. III, Prague, June 25–29, pp. 129–134.
2. Zadeh, L. A. (1965), "Fuzzy sets and systems." *Proceedings of the Symposium on Systems Theory*, Polytechnic Institute of Brooklyn, pp. 29–37.
3. Borisov, A., Ozols, Y., and Grekova, R. (2001), "Construction of Features and Decision Rules in Fuzzy Pattern Recognition Tasks." *Int. J. General Systems*, Vol. 30(1), pp. 23–43.

## DATU NOLIKTAVAS REALIZĒŠANAS UN IZSTRĀDES RĪKI, TO NOVĒRTĒŠANAS KRITĒRIJI

## DATA WAREHOUSE IMPLEMENTATION AND DEVELOPMENT TOOLS AND CRITERIA TO EVALUATE THEM

**Aleksejs Jevsejenko, LU Datorzinātņu maģistrantūras maģistrants,**

A/S "Dati", LR VID datu noliktavas projekta grupas vadītājs, aleksejs.jevsejenko@dati.lv

**Pēteris Gulāns, RTU ASTF Datorzinātņu maģistrantūras maģistrants,**

A/S "Dati", LR VID datu noliktavas projekta sistēmanalītiķis, e-pasts: peteris.gulans@dati.lv

***Abstract.** This paper provides an overview of data warehouse systems development and production tools. It also contains tools evaluation methodology. There are four types of tools used in data warehouse environment. The group of unique criteria is used for every tool evaluation. Each criterion has its own severity (critical, important, additional). Project specific features, chosen criteria set and its severity is a base for tools evaluation matrix. In addition, the unique criteria set for each type of tools is developed and described in this paper.*

*The results of research are used in Latvia State Revenue Service data warehouse system development.*

### 1. Ievads

Datu noliktavas (Data Warehouse) koncepcija radās astoņdesmito gadu sākumā. Pirms tam lielākais vairums informācijas apstrādes sistēmu bija transakciju sistēmas, kuras saturēja tikai pašreizējus datus, kuri nepārtraukti mainījās, kas ļāva risināt operatīvus, pašreizējus jautājumus un bija šauri orientētas viena biznesa uzdevuma risināšanai. Transakcijas sistēmu īpašības rada vairākas problēmas.

- ✓ Vēsturisko datu trūkums sistēmā.
- ✓ Liels informācijas apstrādes sistēmu skaits vienas organizācijas ietvaros.
- ✓ Datu integritātes problēma. Dažādās sistēmās dati tika glabāti un apstrādāti, izmantojot dažādus algoritmus, tāpēc gadījumos, kad tika veikti mēģinājumi apvienot sistēmas, tas bija neiespējams atšķirīgo datu struktūru dēļ.
- ✓ Problēmas ar lielu datu apjoma apstrādi, jo galvenais bija nodrošināt ātru un efektīvu ievades, glabāšanas un apstrādes iespēju, nevis nodrošināt nestandarta analīzes iespējas.
- ✓ Izmaiņu veikšana sistēmās bija dārgs un darbietilpīgs process.

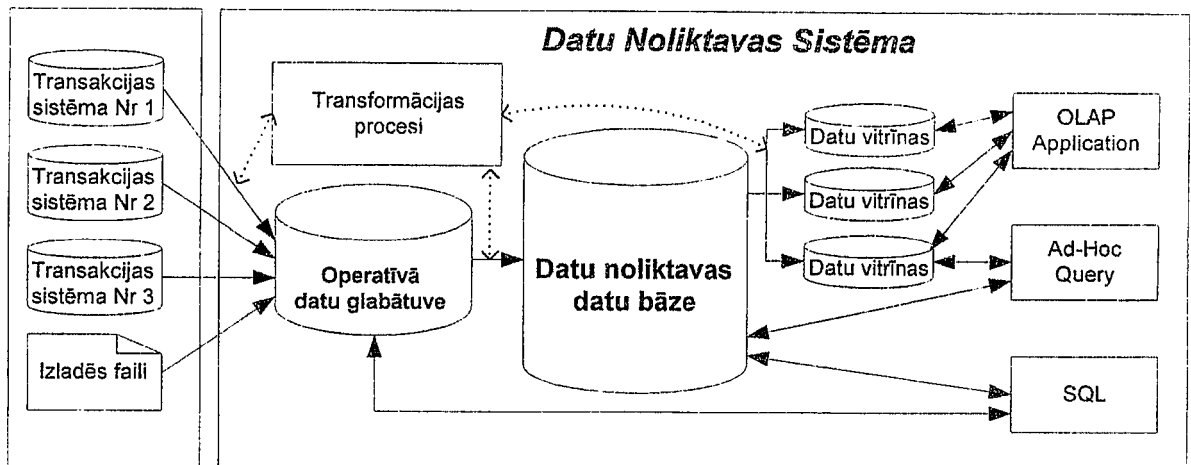
Tieši datu noliktavas (DN) sistēmas koncepcija tika izstrādāta, lai papildinātu esošo transakciju sistēmas un glabātu un analizētu detalizētu, vēsturisku, integrētu informāciju.

## 2. Datu noliktavas sistēmu attīstība

Pirmā datu noliktavas sistēma bija vienkārši liela datu bāze, kura glabāja integrētus datus no vairākām transakciju sistēmām. Divdesmit gadu laikā datu noliktavas sistēmas veica garu evolūcijas ceļu, kā rezultātā radās jauna kompleksa datu noliktavas sistēmas arhitektūra, jauni sistēmas elementi, programmatūras produkti, metodoloģijas un tehnoloģijas [1, 2].

- ✓ Datu operatīvā analītiskā apstrāde – On-Line Analytical Processing (OLAP). OLAP sistēmas ir tiešsaistes sistēmas, kuras ļauj analizēt datus, izmantojot dažādas zināšanu ieguves tehnikas, iegūstot informāciju.
- ✓ Datu vitrīnas (data mart). Datu vitrīnas ir specializētas datu noliktavas, kas nodrošina lēmumu pieņemšanu organizācijas darbības virzieniem. Datu vitrīnu aizpildei tiek izmantoti dati no korporatīvās datu noliktavas un parasti tie ir agregēti, kas ļauj paātrināt darbu ar datu vitrīnām.
- ✓ Operatīvā datu glabātuve un transformācijas rīki. Operatīvā datu glabātuve ir daļa no datu noliktavas sistēmas un ir paredzēta transakcijas sistēmu datu integrēšanai – transformācijas rīki ielādē datus šajā glabātavē un pēc tam izmanto tālākai datu noliktavas datu bāzes aizpildīšanai. Šo sistēmu var arī izmantot atgriezeniskās saites veidošanai uz datu avotiem.
- ✓ Daudzdimensionālā modelēšana. Speciālā modelēšanas tehnika datu noliktavas struktūras projektēšanai un veidošanai. Šī tehnika ir alternatīva ER datu struktūras modelēšanas teknikai. Ar daudzdimensionālo modelēšanu tiek attēlota tā pati informācija, kura ar ER tehniku, taču veidojamā struktūra ir simetriska un adaptēta vaicājumu ātrdarbībai, elastībai pret izmaiņām un lietotāju vieglākai izpratnei.
- ✓ Centralizētā informācijas ražotne – Corporate Information Factory (CIF). Tā ir integrēta zināšanu vide, kas nodrošina efektīvāku informācijas izplatīšanu organizācijā, kas ļauj veikt ātrāku un efektīvāku lēmumu pieņemšanu.

Datu noliktavas vispārējā arhitektūra ir attēlota 1.zīm.



1.zīm. DN sistēmas vispārējā arhitektūra

Tagad datu noliktava ir kompleksa sistēma, kas iekļauj arī iepriekš uzskaitītās jaunās tehnoloģijas un rīkus.

### 3. DN sistēmas uzbūve, rīki un to novērtēšana

Mūsdienu datu noliktavas sistēmai ir kompleksa arhitektūra, un, lai nodrošinātu tās uzbūvi un funkcionēšanu, tiek izmantots specializēts rīku kopums (skat. 1.tab.), kas sastāv no datorizētiem sistēmu projektēšanas jeb CASE (Computer–Aided Software/System Engineering), transformācijas jeb ETL (extraction, transformation, loading), DBVS un OLAP rīkiem.

Datu noliktavā iekļautie rīki

1.tab.

| Rīks  | Primārais uzdevums  | Piemērs   |
|---|---|---|
| <b>CASE</b><br>(Computer–Aided Software/System Engineering) | Primārais uzdevums ir datu noliktavas datu bāzes struktūras projektēšana, veidošana un uzturēšana   | ErWin, Power Designer                                       |
| <b>ETL</b> (extraction, transformation, loading)            | Primārie uzdevumi ir trīs: datu savākšana, datu transformācija (kas iekļauj sevī transformāciju, integrāciju, attīrīšanu, verifikāciju), datu ielādēšana. | Ardent Data Stage, Data Junction                            |
| <b>DBVS</b>   | Primārais uzdevums ir datu glabāšana, lietotāju vaicājumu apstrāde  | Sybase ASIQ, MS SQL, Informix, Oracle                       |
| <b>OLAP</b> (On–Line Analytical Processing)                 | Primārais uzdevums nodrošināt informācijas analīzes iespējas  | Business Objects, Cognos, Microstrategy, Hyperion Solutions |

Visiem šiem rīkiem ir specifiski uzdevumi, un tie izvirza dažādas prasības un attiecīgi arī dažādus kritērijus rīku novērtēšanai. Tālāk tiek aprakstīts kritēriju kopums šo rīku grupu novērtēšanai.

### 3.1. CASE rīku novērtēšana

CASE rīku novērtēšanai var izvirzīt šādus kritērijus:

- ✓ uzturēto DBVS saraksts;
- ✓ savietojamība ar datu bāzu vadības sistēmām (vai var veidot PK, FK, indeksus, skatus, skriptus), ja CASE rīkam atšķiras šīs iespējas attiecībā pret dažādām DBVS, tad šīs iespējas ir jānovērtē attiecībā pret katru DBVS;
- ✓ savietojamība ar dažādām modelēšanas tehnikām (daudzdimensionālā modelēšana, ER, UML, DFD utt.);
- ✓ savietojamība ar operētājsistēmām;
- ✓ vai rīkam ir *reengineering* iespējas;
- ✓ drošības iespējas;
- ✓ dokumentēšanas iespējas;
- ✓ ja rīks spēj veidot skriptu, vai šo skriptu kodu ir iespējams apskatīt un labot;
- ✓ vai no gatavas datu struktūras modeļa var izdalīt apakšmodeli;
- ✓ vai rīks spēj ieteikt izmantot indeksus un optimizēt jau esošu modeli;
- ✓ cik bieži parādās jaunas produkta versijas un cik problemātiski ir uzstādīt jaunu produkta versiju;
- ✓ vai pirkšanas izmaksas iekļauj tikai pašu produktu, vai tās ir atkarīgas no izstrādātāju skaita, datu avotu tipiem, servera procesa skaita utt.;
- ✓ vai par jaunajām versijām jāmaksā atsevišķi;
- ✓ vai ir tehniskais atbalsts.

### 3.2. Datu bāzu vadības sistēmu novērtēšana

Vērtējot DBVS, vajadzētu ņemt vērā šādus kritērijus:

- ✓ ātrdarbība ar dažādiem datu apjomiem un dažādu vienlaicīgi strādājošu lietotāju skaitu, ātrdarbība, strādājot ar lieliem datu apjomiem, darbojoties ar vienu, divām, trim tabulām;
- ✓ savietojamība ar operētājsistēmām;
- ✓ programmēšanas valodu uzturēšana;
- ✓ reģionālo valodu uzturēšana;
- ✓ lietotāju pieejas drošība (vai ir sava drošības sistēma, vai ir operētājsistēmas drošības sistēma, vai abas sistēmas tiek kombinētas);
- ✓ datu glabāšanas drošība;
- ✓ īpašības: indeksēšana, skati, saliktie indeksi;
- ✓ cik bieži parādās jaunas produkta versijas un cik problemātiski ir uzstādīt jaunu produkta versiju;
- ✓ vai pirkšanas izmaksas iekļauj tikai pašu produktu, vai tās ir atkarīgas no izstrādātāju skaita, datu avotu tipiem, servera procesa skaita utt.;
- ✓ vai par jaunajām versijām jāmaksā atsevišķi;
- ✓ vai ir tehniskais atbalsts.

### 3.3. Transformācijas (ETL) rīku novērtēšana

Tagad pasaulē ir pazīstami vairāk nekā 70 rīki, kuriem ir transformācijas funkcionalitāte. Tos ir iespējams saklasificēt un novērtēt pēc funkcionalitātes, uzbūves tipa, transformācijas mehānisma [3].

Nosacīti pēc funkcionalitātes šos rīkus var sadalīt trijās grupās. Pirmās grupas rīkus var nosaukt par datu migrācijas rīkiem, jo transformācijas elementi šādos rīkos ir vāji attīstīti un pārsvarā tiek nodrošināta datu migrācija. Otrās grupas rīkiem ir labi attīstīta datu transformācijas funkcionalitāte, bet ir problēmas ar pieeju izejas vai ieejas datu bāzei (netiek atbalstītas visas DBVS vai tiek izmantots tikai noteikts datu izlādes/ielādes avota tips (piemēram, flat faili)). Trešās grupas rīki nodrošina pilnu ETL funkcionalitāti.

Pēc uzbūves tipa (engine type) šos rīkus var sadalīt divās grupās. Pirmās grupas rīkiem transformācijas procesi tiek izpildīti uz servera, kas ļauj vienlaicīgi veikt vairākus transformācijas procesus, izmantot priekšrocības, kuras sniedz daudzprocesoru serveri, kā arī nodrošināt vienlaicīgu darbu vairākiem izstrādātājiem. Otrās grupas rīkiem transformācijas process notiek uz izstrādātāja darbstacijas, kas samazina ātrdarbību, liedzot veikt vairākus transformācijas procesus vienlaicīgi.

Pēc transformācijas mehānisma šos rīkus var sadalīt divās grupās. Lai veiktu transformāciju uz kodu bāzētajos rīkos, ir nepieciešams izstrādāt transformācijas procesā kodu specifiskā valodā (PERL, PL/SQL u.tml.). Uz grafisko lietotāja saskarsmi bāzētajos rīkos visas nepieciešamās darbības transformācijas procesa izveidei var veikt ar grafiskās saskarnes palīdzību, kas atvieglo un paātrina transformācijas procesu izstrādi, kā arī daži no šiem rīkiem ļauj automātiski dokumentēt transformācijas procesu.

Izvēloties transformācijas rīku, jāņem vērā šādi kritēriji:

- ✓ savietojamība ar dažādiem datu bāzes tiem;
- ✓ vai tiek atbalstīti citi datu tipi (piemēram, flat faili);
- ✓ vai ir nepieciešams datu avotu noformatēt noteiktā veidā, vai ir iespējams norādīt, kā datu avots ir formatēts;
- ✓ vai ir iespējams apskatīt, pārbaudīt, labot SQL vaicājumus, ar kuriem tiek iegūti dati no transakciju sistēmām;
- ✓ vai rīkam ir atklūdošanas un testēšanas iespējas;
- ✓ vienlaicīgi strādājošo izstrādātāju skaits;
- ✓ drošības sistēma – vai ir sava, vai izmanto operacionālās sistēmas drošības sistēmu, vai kombinē abas divas);
- ✓ vai spēj uzturēt un glabāt dažādas procesa versijas;
- ✓ maksimālais vienlaicīgi strādājošo procesu skaits un tam patērētie resursi;
- ✓ vai rīkam ir kļūdu arhīva žurnāls;
- ✓ maksimālais datu apjoms, ko var izlaist cauri vienam transformācijas procesam;
- ✓ cik viegli ir veikt transformācijas procesa izmaiņu;
- ✓ cik bieži parādās jaunas produkta versijas un cik problemātiski ir uzstādīt jaunu produkta versiju;
- ✓ vai pirkšanas izmaksas iekļauj tikai pašu produktu, vai tās ir atkarīgas no izstrādātāju skaita, datu avotu tiem, servera procesa skaita utt.;
- ✓ vai par jaunajām versijām jāmaksā atsevišķi;
- ✓ vai ir tehniskais atbalsts;
- ✓ savietojamība ar operētājsistēmām;

- ✓ savietojamība ar tehniku;
- ✓ vai nav konfliktu ar citiem rīkiem, kuri parasti darbojas uz servera;
- ✓ kādus datu pārraides protokolus spēj izmantot;
- ✓ vai rīks pats veido transformācijas procesa dokumentāciju;
- ✓ vai rīkam ir iebūvēti darba automatizācijas līdzekļi (piemēram, transformācijas procesu startēšanas plānotājs).

### 3.4. OLAP rīku novērtēšana

OLAP rīkus var iedalīt pēc to uzbūves un funkcionalitātes.

Pēc uzbūves OLAP rīkus var iedalīt trijās grupās: ROLAP, MOLAP un HOLAP. ROLAP rīki datu kuba glabāšanai izmanto relāciju datu bāzes struktūras, un, kad tiek veikts vaicājums, dati tiek iegūti no pamata datu bāzes. MOLAP rīki nedarbojas ar pamatdatu bāzi, jo dati tiek glabāti specializētā daudzdimensiju struktūrā, kurā visi dati tiek iekopēti no pamata datu bāzes. Tāpēc, kad tiek veikts vaicājums, pamata datu bāze nav nepieciešama, jo datu kubs tiek veidots no šīs otrās daudzdimensiju datu bāzes datiem. Datu apkopojumi (agregations) arī tiek glabāti daudzdimensiju struktūrā. HOLAP rīki darbojas daļēji kā ROLAP rīki – dati tiek atstāti pamata datu bāzē (kura ir relāciju datu bāze), no kurienes dati tiek iegūti ar SQL vaicājumu, bet datu apkopojumi tiek glabāti specializētā daudzdimensiju struktūrā.

Ja rīkus iedalām pēc funkcionalitātes, tad ir iespējams izdalīt 2 grupas. Pirmās grupas rīki ar SQL vaicājumu iegūst datu kubu, ar kuru tiek veiktas visas nepieciešamās darbības (datu analīze), un SQL vaicājums tiek startēts tikai, tad ja ir jāiegūst jauni dati. Otrās grupas rīki pēc katras jaunas darbības startē SQL vaicājumu, un izvada iegūtos datus.

Izvēloties OLAP rīkus, būtu jāņem vērā sekojoši kritēriji:

- ✓ analīzes iespējas (urbšanās, hierarhiju veidošana, daudzceļu hierarhijas veidošana);
- ✓ vai ir integrēts pārskatu veidošanas rīks, vai jāizmanto cits produkts;
- ✓ pārskatu veidošanas iespējas (sarežģītības pakāpe, formatēšanas iespējas, vai ir iespējams izmantot dažādus datu avotus);
- ✓ drošība (vai ir sava drošības sistēma, vai ir operētājsistēmas drošības sistēma, vai ir DBVS drošība, vai sistēmas tiek kombinētas, vai kāda no tām jāizvēlas);
- ✓ savietojamība ar operētājsistēmām;
- ✓ savietojamība ar DBVS (vai ir ierobežojumi DB izmantošanai, vai ir ierobežojumi DB metadatu glabāšanai);
- ✓ savietojamība ar tehniku;
- ✓ maksimālais vienlaicīgi strādājošo lietotāju skaits;
- ✓ iespēja darbam caur internetu, vai WEB versijai ir ierobežojumi, salīdzinot ar pilno klienta versiju;
- ✓ cik viegli ir apgūt šo rīku izstrādātājiem;
- ✓ cik viegli ir apgūt šo rīku lietotājiem;
- ✓ cik bieži parādās jaunas produkta versijas un cik problemātiski ir uzstādīt jaunu produkta versiju;
- ✓ vai pirkšanas izmaksas iekļauj tikai pašu produktu, vai tās ir atkarīgas no izstrādātāju skaita, datu avotu tipiem, servera procesa skaita utt.;
- ✓ vai par jaunajām versijām jāmaksā atsevišķi;
- ✓ vai ir tehniskais atbalsts;
- ✓ administrēšanas iespējas;
- ✓ vai ir optimizācijas iespējas;

✓ vai ir lietotāju darba žurnāls.

### 3.5. Kritēriju nozīmīgums

Visus iepriekšminētos kritērijus pēc nozīmīguma var nosacīti iedalīt trijās grupās:

- ✓ kritiskie kritēriji – kritēriji, kuru trūkums liedz izmantot novērtējamo programmatūru;
  - ✓ svarīgi kritēriji – kritēriji, kuru trūkums var atstāt iespaidu uz izstrādes procesu;
  - ✓ papildkritēriji – kritēriji, kuru trūkums nevar atstāt nopietnu iespaidu uz izstrādes procesu.
- Kritērija piederība noteiktai grupai nav viennozīmīga, un ir atkarīga no projekta specifikas. Turklāt no projekta specifikas ir arī atkarīgs, vai kritērijs vispār tiks izmantots produktu novērtēšanai.

### 3.6. Sākuminformācijas matrica rīku novērtēšanai

Datu noliktavas sistēmas projektēšanas brīdi ir nepieciešams novērtēt pieejamos izstrādes un realizēšanas rīkus, lai izvēlēties piemērotākos.

Rīku novērtēšanai var izmantot rīku novērtēšanas sākuminformācijas matricu (skat. 2.tab.). Lai izveidot rīku novērtēšanas matricu, jāveic trīs soļi:

- 1) jāizvēlas novērtēšanas kritēriji (skat. nodaļas 3.1. – 3.4.);
- 2) jānovērtē katra kritērija nozīmīgums (skat. nodaļu 3.5.);
- 3) pamatojoties uz pirmajos divos soļos iegūto rezultātu, jāizveido rīku novērtēšanas sākuminformācijas matrica.

Projekta DN\_BANKA rīku novērtēšanas matrica (*fragments*)

2.tab.

| <b>3. Transformācijas programmatūras novērtēšana</b>   |                  |                |
|--|------------------|----------------|
| <i>Produkta nosaukums:</i><br><i>Versija:</i><br><i>Firma–izstrādātājs:</i><br><i>Firma–piegādātājs:</i>   |                  |                |
| <b>3.1. Kritiskie kritēriji</b>  | <b>Vērtējums</b> | <b>Piezīme</b> |
| 3.1.1. Savietojamība ar DBVS (ir/nav):<br>Informix Dynamic Server 7.3.0<br>Sybase ASIQ 12<br>Oracle 9<br>3.1.2. Atklādošanas un testēšanas iespējas (ir/nav)<br>3.1.3. Savietojamība ar operacionālajām sistēmām (ir/nav):<br>Windows 95/98<br>Windows NT<br>Windows 2000<br>Unix AIX<br>... |                  |                |
| <b>3.2. Svarīgi kritēriji</b>  | <b>Vērtējums</b> | <b>Piezīme</b> |
| 3.2.1. Vienlaicīgi strādājošo izstrādātāju skaits (cilv.)<br>3.2.2. Spēja uzturēt un glabāt procesa versijas (ir/nav)<br>3.2.3. Vai ir iebūvēti darba automatizācijas līdzekļi (ir/nav)<br>...   |                  |                |
| <b>3.3. Papildkritēriji</b>  | <b>Vērtējums</b> | <b>Piezīme</b> |
| ...  |                  |                |
| <b>3.4. Papildinformācija (brīvā tekstā)</b>   |                  |                |
| ...  |                  |                |

Pareizi izvēlēti rīku novērtēšanas kritēriji, un uz to pamata izstrādātā rīku novērtēšanas sākuminformācijas matrica ir viens no svarīgākajiem posmiem veiksmīgā datu noliktavas sistēmas izstrādes procesā.

#### 4. Nobeigums

Datu noliktavu sistēmu projektēšana un izstrāde ir ilgstošs, dārgs un sarežģīts process. To apliecina 40% ieviesto sistēmu, kuras ir bijušas pilnībā nerentablas. Savukārt 80% sistēmu nav devušas vēlamu rezultātu. Lai veiksmīgi izstrādātu datu noliktavas sistēmu, ir ārkārtīgi svarīgi izmantot ne tikai pareizu projektēšanas metodoloģiju, bet arī pareizi novērtēt un izvēlēties datu noliktavas sistēmas izstrādes un realizēšanas rīkus.

Rīku novērtēšanas metodoloģiskie principi tika izvirzīti un noformēti LR VID datu noliktavas projekta izstrādes gaitā. Izvirzītie principi tika izmantoti minētajā projektā rīku vērtēšanai un izvēlei, kā arī paredzēti jaunu datu noliktavu projektu izveides atvieglšanai.

#### Literatūra

1. Kimball R., Reeves L., Ross M., Thornthwaite W. The Data Warehouse Lifecycle Toolkit, – NY.: John Wiley & Sons, Inc., 1998, 772 p.
2. Adelman S., Moss L. T. Data Warehouse Project Management, – Boston: Addison-Wesley, 2000, 405 p.
3. Meyer S. Which ETL Tool is Right for You?, DM Review, March 2001, [http://www.dmreview.com/portal\\_ros.cfm?NavID=91&EdID=3084&PortalID=7](http://www.dmreview.com/portal_ros.cfm?NavID=91&EdID=3084&PortalID=7)

## SILTUMA PĀRNESES PROCESU AR IZSTAROŠANU SKAITLISKĀ MODELĒŠANA HORIZONTĀLĀ PLĀKSNĒ SIMULATION OF THE HEAT TRANSPORT PROBLEMS WITH RADIATION IN PLATE

H. Kalis, Latvijas Universitāte, I. Kangro, Rēzeknes Augstskola

*Abstract.* In the literature [1–5] simple and effective algorithms for mathematical modelling processes of distribution of heat in multilayered spaces are created. In the given work the way of improvement of accuracy of algorithms is considered at approximation of integrals derivatives more the supreme orders are used.

### 1. Matemātiskais modelis

Plāksne ar biezumu  $\tilde{l}$  tiek karsēta krāsnī. Pieņemsim, ka plāksne aizņem apgabalu  $\Omega = \{0 \leq \tilde{x} \leq \tilde{l}, -\infty \leq \tilde{y} \leq +\infty, -\infty \leq \tilde{z} \leq +\infty\}$  krāsnī. Siltums tiek pievadīts no augšas ( $\tilde{x} > \tilde{l}$ ) un no apakšas ( $\tilde{x} < 0$ ). Apskatām robežproblēmu temperatūras  $\tilde{T}(\tilde{t}, \tilde{x})$  sadalījumam plāksnē [1–5]:

$$\rho c_p \left( \frac{\partial \tilde{T}}{\partial \tilde{x}} \right) = \partial \left( k \frac{\partial \tilde{T}}{\partial \tilde{x}} \right), \quad \tilde{x} \in (0, \tilde{l}), \tilde{t} > 0, \quad (1)$$

$$\frac{\partial \tilde{T}(\tilde{t}, 0)}{\partial \tilde{x}} = f_1(\tilde{T}(\tilde{t}, 0)), \quad (2)$$