Quality of integrated data

Lt.col. Otilia PÎRLOG
Ministerul Apărării Naţionale

The multitude of data sources available leads to the necessity of determining the optimal solution for configuring a data warehouse, through which accurate and timely replies can be given to internal and external beneficiary. The Extraction, Transforming and Loading technique (ETL) allow for designing data warehouse configuration projects, by defining the underlying conceptual and logical models. On these grounds, the transformation flow is designed and analyzed, the main final target being the activity areas. The present article presents a modality for quantifying the efficiency, productivity and tolerance of the analyzed solutions. Also, by using the matrixes, it is possible to determine the usefulness hierarchies or the vulnerability of the network nodes.
Keywords: *data quality, data warehouses, source integration, ETL technique.*

Configurarea depozitelor

Procesele operaţionale de construire a depozitelor de date constau din extragerea, transformarea, integrarea, curăţarea şi transportul datelor. În afară de curăţarea datelor, celelalte procese se subscriu unei abordări comune prin *tehnica extragere, transformare, încărcare (ETL)*, care a fost elaborată în scopul facilitării conducerii şi optimizării acestor procese. Mecanismul general de utilizare al acestei tehnici constă din elaborarea unui scenariu general pe baza unor formate flexibile, care sunt apoi particularizate la cerinţele specifice ale utilizatorului.

În figura 1 este prezentat mediul de dezvoltare al tehnicii ETL. Corespunzător fluxului datelor şi al metadatelor, pot fi evidenţiate două planuri de manifestare, fizic şi logic. Pe parcursul stadiilor ciclului de viaţă al depozitului, conexiunile dintre cele două planuri au sensuri de acţiune diferite, care pot fi uni sau bidirecţionale. Sunt sugerate astfel modalităţi de abordare a depozitelor, categoriile de activităţi prin care să se configureze scenariul propus şi funcţiile aferente acestora.

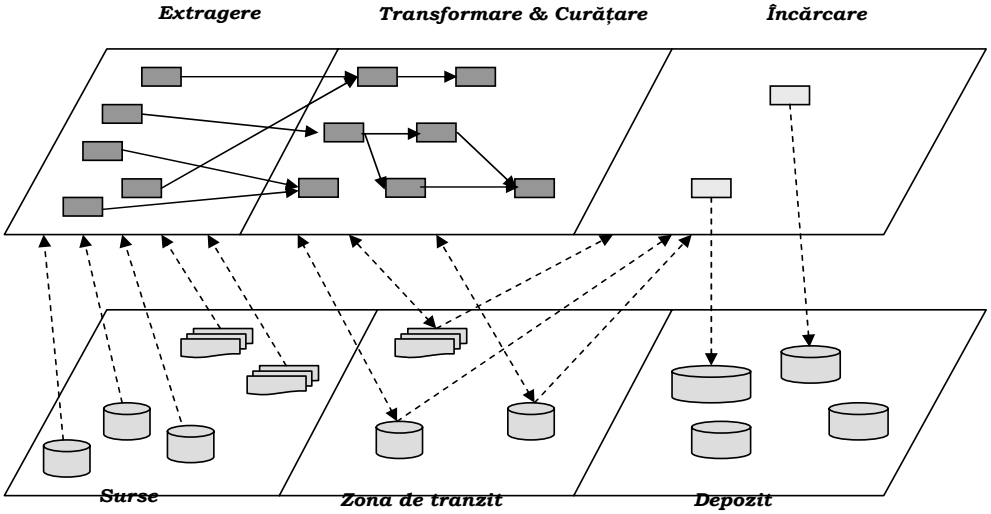

Fig. 1. Mediul de dezvoltare al procesului ETL

Planul fizic al procesului ETL cuprinde şi o zonă de tranzit. Aceasta constituie locul în care au loc toate transformările asupra datelor extrase din sursele iniţiale, astfel încât să corespundă cerinţelor formulate pentru depozit. Se obţine astfel o imagine liniară de procesare, în care sunt reliefate elementele şi activităţile majore de constituire a depozitu-

lui.

Ciclul de viață al unui depozit de date cuprinde următoarele faze:

– ingineria inversă a surselor şi colectarea cerințelor;
– proiectarea logică;
– descrierea completă a activității;
– elaborarea softului.

În prima fază sunt analizate sursele de date în scopul definirii structurii şi a conținutul acestora. Cerințele utilizatorilor stabilesc condițiile pe care trebuie să le satisfacă depozitul de date care va fi constituit. Drept rezultat, se elaborează un model conceptual pentru depozit, datele memorate şi activitățile implicate. În faza următoare, elementele analizate anterior sunt structurate într-un proiect logic. Se detaliază apoi fiecare element al modelului logic, astfel încât să permită rafinarea elementelor specifice ale structurii fizice a depozitului de date şi a parametrilor de execuție aferenți mediului de dezvoltare propriu. Se obține deci o descriere completă a activităților, cu stabilirea proceselor operaționale şi a metricilor de evaluare. Toate eforturile de proiectare depuse în urma parcurgerii primelor trei etape permit continuarea cu elaborarea, testarea şi evaluarea programului soft. Tot acum se elaborează o versiune inițială a depozitului şi se face o evaluare calitativă a proiectării realizate. Se impune de asemenea trecerea la administrarea depozitului, care necesită metrici specifice de întreținere şi monitorizare.

Apoi ciclul reîncepe, cerințele utilizatorului şi condițiile de răspuns formulate de proiectant, prin intermediul depozitului de date, fiind ameliorate continuu.

Elaborarea modelului conceptual

În stadiul inițial, de configurare a depozitului, modelul conceptual surprinde relațiile dintre atributele datelor şi activitățile al căror efect se răsfrânge asupra lor. Elementul central al modelului conceptual îl constituie modul de documentare / formalizare al particularităților datelor. Se adresează astfel două aspecte majore ale procesului ETL:

– relațiile dintre atribute şi concepte;
– transformările realizate pe timpul încărcării depozitului.

În perioada de stabilire şi analizare a cerințelor, este necesară rezolvarea de către proiectant a problematicii aferente următoarelor etape majore:

- identificarea surselor care corespund cererii formulate;
- stabilirea candidaților pentru datele referite;
- realizarea diagramei atributelor dintre furnizori şi consumatori;
- adnotarea diagramei cu restricțiile în timp real.

Realizarea diagramei atributelor reprezintă etapa cea mai dificilă. În urma discuțiilor cu administratorii surselor de date, se clarifică problemele vizând codificarea, regulile şi valorile implicite. De asemenea, în scopul identificării incidentelor posibile, sunt elaborate şi testate scenarii diverse, prin care se simulează diverse cereri care pot fi adresate în exploatarea reală. Efortul se concentrează pe urmărirea transformărilor şi operațiilor de curățare care se execută pentru elaborarea diagramei de corelație a atributelor. Separat de acest flux, este necesar să fie definiți o serie de alți parametri, prin adnotarea diagramei cu restricțiile în timp real. Acestea se referă la: planul de derulare în timp al evenimentelor, elemente de monitorizare a procesului de către administratorul depozitului, tratarea modalităților de violare a regulilor, definirea rutinelor de refacere a deteriorărilor posibile etc.

Elaborarea modelului logic

Se realizează în scopul asistării proiectării depozitului şi constă dintr-un graf prin care se structurează arhitectura acestuia, se descrie fluxul datelor de la surse la depozit, modul de compunere a activităților şi semantica derivată. Cu ajutorul lui se poate determina partea din scenariu care este afectată la ştergerea unui atribut şi aprecierea calității scenariului elaborat.

Elementele componente ale grafului sunt noduri şi arce. Un scenariu complet este constituit din activități, seturi de înregistrări şi funcții, care sunt conectate prin intermediul relațiilor corespunzătoare. La acestea se adaugă elemente de definire, cum sunt tipul datelor / funcțiilor, constante şi atribute. Princi-

palele noduri ale grafului sunt reprezentate din activităţi. Acestea sunt abstracţii logice care conţin o secvenţă de module de cod sau reprezentarea completă a unui proces.
O activitate elementară este descrisă formal prin următoarele elemente: nume, schema /schemele de intrare, schema de ieşire, schema de respingere şi lista parametrilor.
Prin conexiunile grafului sunt evidenţiate relaţiile dintre entităţile reţelei grafice.
Principalele tipuri de relaţii pentru definirea schemei depozitului sunt: de apartenenţă, de particularizare, de furnizare, de reglare şi de furnizare derivate.
Relaţia de apartenenţă leagă atributele şi parametrii de respectivele activităţi, set de înregistrări sau funcţii, la care ele aparţin. Este modalitatea prin care se reliefează setul de atribute din compunerea unui concept.
Relaţia de particularizare conectează tipul datei / funcţiei la modul particular de manifestare al acesteia.
Relaţia de furnizare poate să vizeze un număr variat de surse, situaţia limită fiind cea de constituire a depozitului pe baza unui singur furnizor. Un set de atribute de intrare este într-o relaţie de furnizare cu un set de atribute de ieşire prin intermediul unei transformări relevante. Există două modalităţi de abordare a acestei relaţii, funcţie de numărul atributelor sursă şi destinaţie corelate:
- relaţia simplă (1:1): un atribut sursă populează atributul din depozit printr-o transformare corespunzătoare;
- relaţia multiplă (N:M): cuprinde un set finit de atribute de intrare şi ieşire, puse în corelaţie unele cu celelalte printr-o relaţie de alocare;
Relaţia de reglare stabileşte parametrii activităţilor şi termenii care le populează.
Relaţia de furnizare derivată este un caz special al relaţiei de furnizare. Diferenţa dintre ele constă în elementele pe baza cărora se calculează atributele de ieşire. Pentru relaţia de furnizare acestea constau din atribute de intrare, respectiv parametri pentru cealaltă situaţie.
Datorită complexităţii grafului rezultat, deseori este necesară o analiză parţială a acestuia, prin delimitarea unei anumite ramificaţii sau prin evidenţierea unor tipuri de transformări de bază. Analiza numai a unei anumite transformări reprezintă un caz particular al analizei, deosebit de util. Drept rezultat, există două tipuri de transformări de bază:
- relaţii de populare: sunt reţinute numai relaţiile de furnizare şi cele de reglare, rezultând o machetă completă a descendenţei şi precedenţei atributive;
- fluxul principal: este o tehnică de analiză inversă, astfel încât se reţin numai elementele grafului care au implicaţii asupra configurării finale a depozitului.
Analizele care se fac de către proiectantul şi administratorul depozitului impun cu necesitate executarea unor scenarii diverse, compararea rezultatelor obţinute şi stabilirea variantei care satisface criteriul de optim vizat.

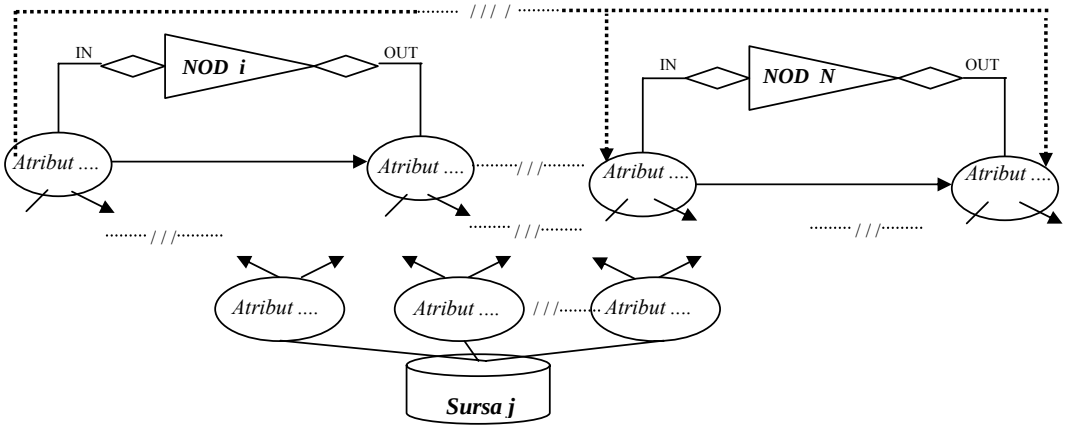

Fig. 2. Graful ipotetic de proiectare a unui depozit de date

Analiza fluxului datelor

Prin modelarea activităţilor ETL sub forma unui graf, este posibilă tratarea scenariului rezultat ca un schelet de bază al întregului mediu de proiectare. Pot fi folosiţi algoritmi prin care să obţinem graful de bază sau cel

critic. În schimb, acestea nu dau un rezultat cuantificabil privind eficiența, randamentul şi toleranța soluției analizate. De asemenea, nu se pot determina ierarhii de importanță sau vulnerabilitățile nodurilor rețelei. La aceste deziderate răspund metricile.

Figura 2 reprezintă un graf ipotetic de constituire a unui depozit, sugerând conexiunea şi codificarea nodurilor acestuia şi a surselor emitente ale datelor. În tabelul 1 este prezentat modul de definire şi notare a elementelor grafului.

Tabelul 1. Elementele grafului de proiectare a unui depozit de date

Simbol	Semnificație vector / matrice	Element	Dimensiune variabile	Domeniu
Nod(N)	Nodurile grafului	*Nod(i)*	N = numărul de noduri	$i = 1 \div N$
Sursa(S)	Sursele de date	*Sursa(s)*	S = numărul de surse de date	$s = 1 \div S$
Nr_atrib(S)	Numărul de atribute emise pe flux de către fiecare sursă de date	*Nr_atrib(s)*	S = numărul de surse de date	$s = 1 \div S$
Înloc(N)	Chei de înlocuire	*Înloc(i)*	N = numărul de noduri	$i = 1 \div N$
Nr_înloc(N)	Numărul de chei de înlocuire generate în fiecare nod al grafului	*Nr_înloc(i)*	N = numărul de noduri	$i = 1 \div N$
Tip	Tipul atributului	*Tip*	Tip $\in$ { 0, 1, 2, 3, 4 }	
Atr(M)	Atribute	*Atr(j)*	M = numărul total de surse	$j = 1 \div M$

Nod(N) este vectorul nodurilor grafului, unde *N* este numărul de noduri. Alături de sursele emitente de atribute ale datelor, respectiv *Sursa(S),* nodurile grafului constituie un tip special de surse, prin generarea de chei de înlocuire. Identificarea acestora este funcție de nodurile rețelei grafice, de aici şi notarea lor, *Inloc(N)*. Teoretic, pe lângă cele *M* surse de bază, mai pot exista alte *N* surse adiționale, numărul total de surse emitente fiind *S+N.* Un element din prima categorie, *Sursa(j)* , conține numele sursei respective, în timp ce furnizările de înlocuire se înscriu pe aceeaşi traiectorie de dezvoltare cu cea a atributelor datelor.

O tratare specială este cea referitoare la atribute. Acestea pot fi surprinse în una din următoarele ipostaze, fixate prin tipul atributului, *tip*: *tip = 0* - atribut de intrare în nodul grafului sau în schema finală a depozitului; *tip = 1* - atribut sursă; se află la prima apariție pe flux, prin emiterea de către o sursă; *tip = 2* - valoare de înlocuire generată în nodul emitent; *tip = 3* - atribut de ieşire dintr-un nod al grafului; *tip = 4* - atribut final; are destinația ultimul nod al grafului, *N,* deci schema depozitului.

Numărul total de atribute care traversează fluxul de prelucrare, *M*, se determină prin însumarea numărului de atribute emise de sursele de date, *N1*, cu cel al numărului de valori de înlocuire generate în diversele noduri de prelucrare, *N2*.

$M = N1 + N2$

Numărul de atribute emise cumulează atributele aferente fiecărei surse în parte. Fie vectorul *Nr_atrib(S)* al cărui element generic *Nr_atrib(s)* reprezintă numărul de atribute emise pe flux de către sursa *s*.

Atunci $N1 = \sum_{s=1}^{S} Nr\_atrib(s)$.

Numărul de valori de înlocuire generate se calculează asemănător, având drept bază de calcul vectorul *Nr_înloc(N)*, respectiv elementele acestuia, *Nr_înloc(i)*.

Deci $N2 = \sum_{i=1}^{N} Nr\_inloc(i)$.

Formalizarea fluxului de prelucrare

În continuare prezentăm o modalitate de algoritmizare a fluxului de prelucrare pentru definirea schemei depozitului de date.

Fie *N* nodurile rețelei de prelucrare proiectate şi *M* atributele care traversează aceste noduri. În tabelul cu dublă intrare aferent, de dimensiune *(N, M)*, înscriem elementele pereche de forma *(K1(i,j), K2(i,j))*, cu următoarea semnificație:

- *K1(i,j)* nodul emitent al atributului *j* către nodul *i*; în cazul în care atributul este generat în nodul *i, K1(i,j)=0*;
- *K2(i,j)* nodul destinație al atributului *j* emis de nodul *i*; în cazul în care atributul converge către schema finală a depozitului,

K2(i,j)=N.
Practic, prin înscrierea în matrice a acestor perechi de valori, structura acesteia definită inițial primeşte o nouă dimensiune de referință (tabel 2).

Tabelul 2. Formalizarea fluxului de prelucrare pentru stabilirea schemei depozitului

Atributele datelor	*Nodurile fluxului de prelucrare*							
	Nod(1)	*Nod(2)*	. . .	. . .	*Nod(i)*	. . .	*Nod(N-1)*	*Nod(N)*
Atr(1)								
Atr(j)					***(K1(i,j),K2(i,j))***			
...	...							
Atr(M)								

Această modalitate de formalizare permite analiza fluxurilor atât la nivel de atribute, deci pe liniile matricei, cât şi pe noduri, corespunzător coloanelor acesteia.
Avem acum la dispoziție toate elementele prin care se pot defini metricile vizate.
Dependența locală (Dep_loc) reprezintă numărul de noduri cu conexiuni de tip furnizare aferente nodului evaluat, pe parcursul schemei globale de transformare. Se reliefează astfel numărul atributelor primare sau intermediare care concură la definirea nodului respectiv. Este evidentă tratarea dinspre sursele de date către nodul analizat.
Dep_loc (i) = Card {[K1(i,j),K2(i,j)]/ K1(i,j) ≠ 0, j=1,M }
Responsabilitatea locală (Resp_loc) stabileşte numărul de noduri dependente de nodul evaluat. În acest caz, calculul se face pe baza elementelor care converg de la nodul grafului către depozit.
Resp_loc (i) = Card {[K1(i,j),K2(i,j)]/ K1(i,j) ≠ 0, j=1,M }
În timp ce dependența locală exprimă numărul de noduri care trebuiesc activate în scopul populării unui anumit nod, responsabilitatea locală caracterizează numărul de noduri care depind de activarea nodului considerat, în scopul primirii datelor.
Nivelul de determinare locală (Niv_dt_loc) stabileşte intensitatea fluxului aferent nodului. Se calculează prin însumarea valorilor obținute pentru metricile definite anterior.
Nr_dt_loc (i) = Dep_loc (i) + Resp_loc (i)
Nivelul total local (Niv_tot_loc) este o expresie a gradului de aglomerare al nodurilor implicate în scenariu.

$$Niv\_tot\_loc = \sum_{i=1}^{N} Niv\_dt\_loc(i)$$

Dependența tranzitivă (Dep_tranz) reprezintă numărul de noduri anterioare traversate de atributul respectiv, pe traiectoria fluxului de prelucrare, până la intrarea în nodul evaluat.
Modalitatea de calcul a dependenței tranzitive presupune refacerea traiectoriei parcurse de atributul *j* până la intrarea sa în nodul *i*. De aici necesitatea transpunerii algoritmice a fluxului printr-o structură repetitivă condiționată de atingerea unei valori *K1 = 0*, deci a sursei emitente / generatoare a atributului.

```
Depend = 0
ii := i
repeat until K1 (ii) = 0
        Depend := Depend + 1
        ii := K1 (ii)
continue
Dep_tranz (i) = Depend
```

Responsabilitatea tranzitivă (Resp_tranz) stabileşte numărul de noduri ulterioare traversate de atribut până la schema finală a depozitului.

```
Resp = 0
ii := i
repeat until K2 (ii) = 0
        Resp := Resp + 1
        ii := K2 (ii)
continue
Resp_tranz (i) = Resp
```

Nivelul de tranzitivitate (Niv_tranz) este cumulul metricilor determinate anterior.
Niv_tranz (i)=Dep_tranz (i) + Resp_tranz (i)
Nivelul de dependență (Niv_dep) se obține prin însumarea dependențelor locale şi tranzitive.
Niv_dep (i) = Dep_loc (i) + Dep_tranz (i)
Nivelul de responsabilitate (Niv_resp) este

suma responsabilităţilor locale şi tranzitive.
Niv_resp (i) = Resp_loc (i) + Resp_tranz (i)
Nivelul global de determinare ((Niv_glob_dt) rezultă prin însumarea nivelurilor de dependenţă şi responsabilitate.

$$Niv\_glob\_dt = \sum_{i=1}^{N} Niv\_dep(i) + \sum_{i=1}^{N} Niv\_resp(i)$$

Bibliografie

[Bech00], Bechtel Hanford Inc. Procedure: *Data Quality Objectives*, BHI-EE-01, Environmental Investigations Procedure: Procedure Number 1.2; Revision 3; September 30, 1999.

[Imho03], Imhoff, C., Galemmo, N., Geiger, J.: *Mastering Data Warehouse Design : Relational and Dimensional Techniques,* Publisher: Wiley; July 2003.

[Kimb02], Kimball, R., Ross, M.: *The Dta Warehouse Toolkit: The Complete Guide to Dimensional Modeling,* Publisher: Wiley; 2 edition, April 2002.

[Kimb04], Kimball, R., Caserta, J.: *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data,* Publisher: John Wiley & Sons, September 2004.

[LeeJ03], Lee, J., Wyner, G.M.: *Defining Specialization for Dataflow Diagrams*, Information Systems, Volume 28, Number 6, September 2003, 651-671.

[Redm03], Redman, T.C.: *Ending 'Garbage In, Garbage Out': IT's Role in Improving Data Quality,* Guest Editor, Cutter IT Journal, January 2003, Vol. 16, No. 1.

[Wang01], Wang, R.Y., Ziad, M., Lee, Y.W.: *Data Quality*, Kluwer, 2001.