Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Matija Pipan

Avtomatizacija in nadzor izvajanja procesov ETL v sistemu SAS

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: izr. prof. dr. Zoran Bosnić

Ljubljana, 2014

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Integracija podatkov je močno vezana na postopke ETL (extract, transform, load - zajemanje, obdelava in polnjenje podatkov), ki jih je za kakovostno in učinkovito izvajanje programskih rešitev potrebno avtomatizirati in nadzorovati njihovo izvajanje. Programska oprema SAS ponuja orodja za implementacijo procesov ETL, ki omogočajo njihovo sistematično implementacijo in so konkurenčna drugim sorodnim orodjem.

Kandidat naj v diplomskem delu predstavi trenutno stanje področja procesov ETL in predstavi programsko opremo, ki se za ta namen uporablja znotraj sistema SAS. V zaključku diplome naj opisano rešitev primerja s sorodnimi rešitvami.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Matija Pipan, z vpisno številko **63000406**, sem avtor diplomskega dela z naslovom:

Avtomatizacija in nadzor izvajanja procesov ETL v sistemu SAS

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Zorana Bosnića,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 17. avgust 2014 Podpis avtorja:

Zahvala

Največja zahvala velja mentorju izr. prof. dr. Zoranu Bosniću, ki mi je s strokovno pomočjo, z jeklenimi živci, z neizmerno mero potrpežljivosti in z neomajno spodbudo pomagal pri izdelavi diplomskega dela.

Naslednja zavala je namenjena gospe Metki Runovc v referatu za vztrajno opominjanje in pomoč pri izpolnjevanju pogojev za oddajo diplomskega dela.

Nazadnje bi se zahvalil svojmi staršem, ki me zdaj ne bodo več spraševali: *„Kdaj bo diploma?"*

Svojemu očetu Stankotu.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
aCRM	Analytical Customer Relationship Management	analitični sistem za upravljanje odnosov s strankami
ETL	Extract, Transform, Load	zajem, obdelava, polnjenje
RDBMS	Relational Database Management System	sistem za upravljanje relacijskih podatkovnih baz
SCD2	Slowly Changing Dimensions Type 2	sledenje podatkovnim posodobitvam tipa 2
CDC	Change Data Capture	zajem podatkov s pomočjo zaznavanja sprememb
OLAP	Online Analytical Processing	sprotna analitična obdelava podatkov
SAS	Statistical Analytical Software	programska oprema za analizo in obdelavo podatkov
DIS	Data Integration Studio	orodje za podatkovno integracijo
EG	Enterprise Guide	orodje za poročanje

Povzetek

Postopek podatkovne integracije se danes uporablja tako na komercialnih kot tudi na znanstvenih področjih. Implementacije zajemajo vrsto različnih procesov ETL za obdelavo in pripravo podatkov, kjer sta avtomatizacija ter nadzor izvajanja procesov ključnega pomena vsake rešitve.

Pri svojem delu redno sodelujem pri implemetacijah rešitev za podatkovno integracijo. Skozi projekte sem razvil ogrodje ETL, ki standardizira način izdelave procesov ter omogoča njihovo avtomatizacijo in nadzor pri izvajanju.

Diplomska naloga opisuje način uporabe ogrodja pri implementaciji in funkcionalnost, ki je z njim podprta. Na začetku je predstavljen podatkovni model, ki se tipično uporablja v poslovnem svetu. Sledi predstavitev programske opreme, uporabljene pri razvoju ogrodja in opredelitev sklopov ETL procesov. Tu so tudi opisane različne metode izvajanja vsakega sklopa. Nato bo podrobno predstavljeno ogrodje ETL, ki se v nadaljevanju primerja z drugimi rešitvami. Na koncu je opisan način uporabe ogrodja s strani uporabnika in sklepne ugotovitve.

Ključne besede: procesi ETL, ogrodje ETL, integracija podatkov, avtomatizacija, nadzor.

Abstract

The process of data integration is used today in both commercial as well as in scientific fields. Implementations are comprised of many different ETL processes to achieve proper transformation of data. Automation and control of entire process is a crucial aspect for each solution.

Throughout my previous work engagements I have developed ETL framework, which standardizes approach of ETL processes development and provides automation and control over entire implementation.

Thesis describes how ETL framework is used with implementation and which functionality are supported. We start by describing the data model, that is typically used in the business world. Followed by the Presentation of software used for framework development and defining different sets of ETL process flows. Here are also described different methods of processing used by each set of processes. Next, we provide detailed presentation of ETL framework, which is afterwards compared with other solutions. We finalize the thesis by describing the ETL framework interaction from the user perspective, and by drawing conclusions.

Keywords: ETL processes, ETL framework, data integration, automation, control.

Poglavje 1

Uvod

Vedno več podjetij se srečuje z vprašanjem, kako izboljšati poslovni učinek, povišati prihodke, ohraniti obstoječi portfelj strank oziroma ga razširiti s pridobivanjem novih. Danes se podjetja poslužujejo različnih metod preiskovanja podatkov svojih strank in njihovega poslovnega življenjskega cikla s ciljem zasnovati novo poslovno idejo za povečanje poslovnega učinka. To je še posebej pomembno za podjetja, ki imajo na voljo veliko podatkov o svojih strankah in njihovi zgodovini poslovanja, ki jo lahko uporabljajo za analitične namene. Sem spadajo predvsem panoge, kot so telekumunikacije, bančništvo, zavarovanje, prodaja, ...

V poslovnem svetu so podjetja začela uporabljati informacijske sisteme, ki so danes že zelo razširjeni in poslovanje brez njih ni več konkurenčno. Primer takšnega sistema je sistem za upravljanje odnosov s strankami (v nadaljevanju aCRM – angl. *Analitical Customer Relationship Management System*), ki je danes nujno poslovno orodje za vsa večja podjetja.

1.1 Opis predmeta raziskovanja

Podatki, uporabljeni pri analizah, so shranjeni v podatkovnem modelu aCRM, ki ga lahko opredelimo kot relacijski podatkovni model za analitiko. Model predstavlja zbirka tabel, ki organizira podatke po predmetih določene panoge. Za pripravo podatkov se uporabljajo tako imenovani procesi ETL (angl. *Extract Transform Load*) – te procesi podatke zajemajo iz izvornih sistemov, jih pretvorijo v

določeno obliko ter napolnijo v ciljne tabele podatkovnega modela. Takšen postopek obdelave podatkov opišemo z izrazom *podatkovna integracija* [1, 2] (angl. *Data Integration Process*).

Dobro zasnovana rešitev podatkovne integracije avtomatizira celoten proces. V ta namen je bilo razvito ogrodje ETL (angl. *ETL framework*), ki omogoča centraliziran nadzor nad celotnim potekom izvajanja procesov ter njihovo avtomatizacijo in je predmet tega diplomskega dela.

1.2 Cilj diplomskega dela

Za razvoj procesov ETL je na voljo vrsta različnih orodij.

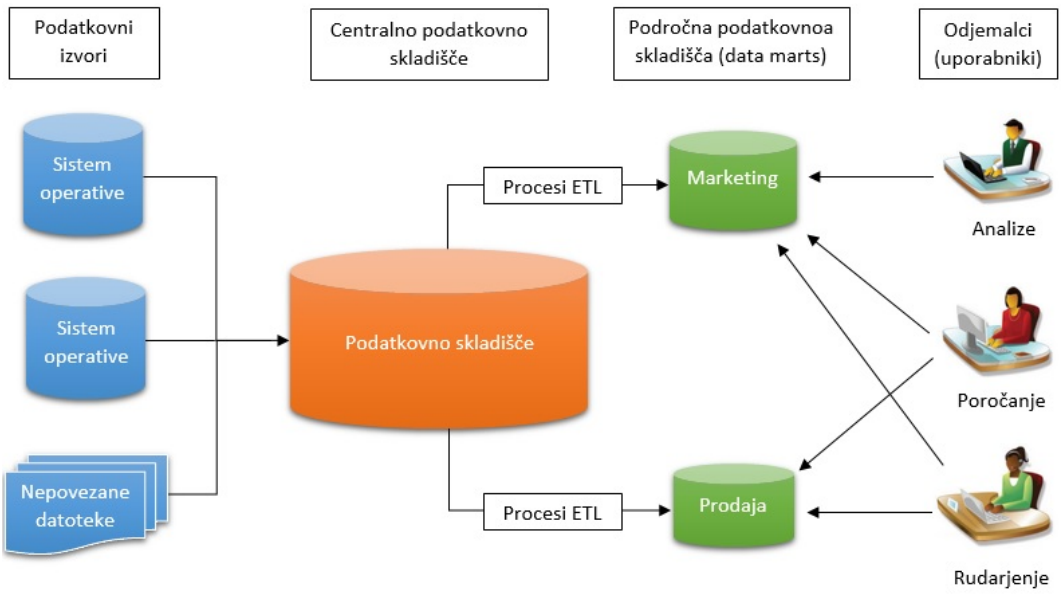
Cilj diplomskega dela je predstavitev ogrodja ETL, ki je bilo razvito z uporabo programske opreme SAS. Programska oprema nudi možnost upravljanja, spreminjanja, rudarjenja in pridobivanja podatkov iz vrste različnih podatkovnih virov in tudi njihovo analizo. Predstavili bomo katere funkcionalnosti ogrodje ETL podpira ter način uporabe pri implementaciji celovite rešitve za polnjenje aCRM podatkovnega modela.

Poglavje 2

Organizacija podatkovnega modela

Obseg podatkov, zajetih v podatkovnem modelu, je podmnožica podatkovnega skladišča (ali več različnih podatkovnih izvorov) nekega podjetja, ki je ponavadi prirejena za določen/specifičen oddelek in se v poslovnem svetu imenuje „področno podatkovno skladišče" (angl. *Data Mart*). Tako lahko podjetje organizira procese polnjenja več različnih področnih podatkovnih skladišč – vsako od njih na uporabniško prijazen način predstavlja določen podatkovni segment podjetja. Poslovni uporabniki tako ne dostopajo do podatkov, shranjenih v podatkovnem skladišču, ampak do podatkov, ki se nahajajo v določenem področno podatkovnem skladišču. Na ta način lahko vsak oddelek neodvisno uporablja, pretvarja in razvija svoje podatke, kot želi, ne da bi s tem vplival na izvorne podatke podatkovnega skladišča [2, 7] (slika 2.1).

Podatkovna struktura aCRM je predstavljena z relacijskim dimenzijskim modelom, ki opredeli podatke poslovnega procesa v dve kategoriji in je lažje razumljiv poslovnim uporabnikom. Model tako uporablja dva tipa tabel – tabela „dejstev" (angl. *Facts*), sestavljena iz metrik oziroma količin, in tabela „opisov" ali „šifrantov" (angl. *Dimensions*), s katerimi so „dejstva" opisana. Model z relacijami povezuje tabele v topologijo zvezde ali snežinke in je normaliziran v 3. normalno formo [13, 8].



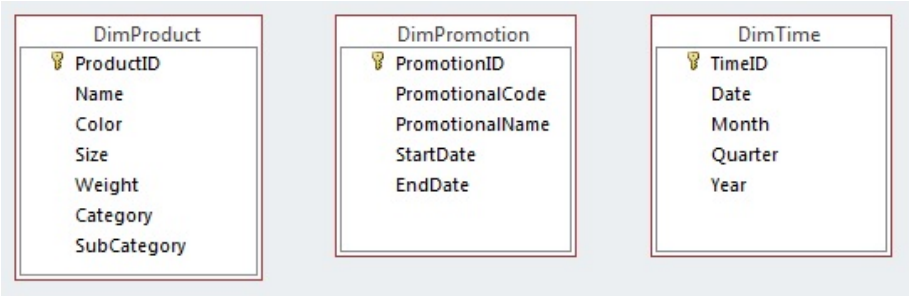
Slika 2.1: Organizacija podatkovnih sistemov

Sestavni deli opisnih tabel so: stolpci šifrantov (predstavljajo primarni ključ tabele) in stolpci z opisnimi (deskriptivnimi) atributi. Dobro zasnovana tabela se drži pravil, kjer so njene vrednosti:

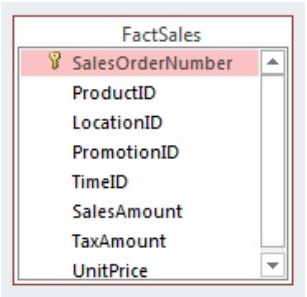
- deskriptivne (jasni opisi),
- celostne (brez manjkajočih vrednosti),
- diskretne (unikatne vrednosti),
- kvalitetne (brez slovničnih napak).

Takšne tabele (slika 2.2) se uporabljajo za hranjenje podatkov o proizvodih, promocijah, spolu, času ipd.

Tabele dejstev se običajno sestojijo iz metrik ter primarnih in tujih ključev. Sestavni deli tabele so torej: stolpci, ki določijo primarni ključ, stolpci tujih ključev opisnih tabel in stolpci metrik oziroma količin. Te tabele (slika 2.3) hranijo podatke, ki opisujejo dogodke, kot so: nakupi, naročila, transakcije ipd. V praksi se lahko pojavljajo tudi tabele dejstev, ki nimajo nobenih merljivih količin ali metrik (angl. *Factless Fact Table ali Cross-reference Table*). Takšne tabele se primarno uporabljajo za reševanje relacije „več-na-več”.



Slika 2.2: Primeri opisnih tabel



Slika 2.3: Primer tabele dejstev

Stopnjo podrobnosti zapisov tabel dejstev določa granularnost. Zapisi tabel z nizko stopnjo granularnosti nudijo podrobnejše informacije dogodkov kot zapisi tabel z visoko stopnjo. Določitev stopnje granularnosti igra pomembno vlogo pri zasnovi tabel dejstev. Prenizka stopnja lahko povzroči prekomerno količino podatkov v tabeli, s katero je zamudno operirati, medtem ko previsoka stopnja morda ne ohrani dovolj podrobnih podatkov za analiziranje [13, 9].

Poleg različne granularnosti zapisov delimo tabele dejstev tudi glede na način, kako se hrani zgodovina sprememb zapisov skozi čas.
Poznamo štiri osnovne tipe:

- Transakcijski posnetek (angl. *Transaction*): dodajanje samo novih zapisov iz izvorne tabele, ki še niso bili napolnjeni v ciljno tabelo. Ohrani se celotna zgodovina dogodkov.

- Periodični posnetek (angl. *Periodic Snapshots*): ciklično dodajanje celotne „slike" trenutnega stanja zapisov iz izvorne tabele. Vsi zapisi za določen datum odražajo veljavno stanje zapisov izvorne tabele v tistem trenutku. Ta koncept ohrani celotno zgodovino sprememb zapisov, je zelo preprost za uporabo, vendar prostorsko potraten.

- Akumulirani posnetek (angl. *Accumulating Snapshots*): spremembe zapisov tabele se skozi čas posodabljajo glede na spremembe v izvorni tabeli. Tak tip tabele ima veliko datumskih stolpcev in potrebuje zahtevnejšo logiko posodabljanja zapisov. Poleg tega moramo vedeti, koliko sprememb lahko pričakujemo v celotnem življenjskem ciklu zapisa. Prav tako kot pri prejšnjem tipu se ohrani celotna zgodovina sprememb zapisov ter prihranimo na prostoru.

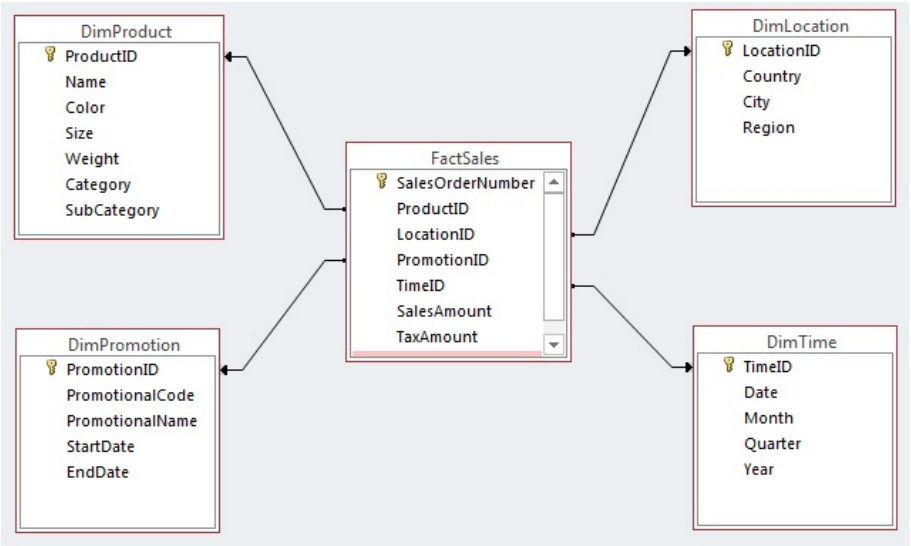
- Časovni posnetek (angl. *Temporal Snapshots*): spremembe zapisov tabele se skozi čas posodabljajo glede na spremembe v izvorni tabeli. Pri tem tipu uporabljamo samo dva dodatna datumska stolpca, ki predstavljata začetek in konec časovnega intervala, v katerem je bil zapis „aktualen". Vsaka sprememba vrednosti zapisa na izvoru povzroči dve spremembi v ciljni tabeli, in sicer:

- „zaprtje" trenutnega zapisa – posodobitev konca intervala trenutno veljavnega zapisa v ciljni tabeli,
- „odprtje" novega zapisa – dodan je nov veljaven zapis v ciljno tabelo z začetkom intervala na trenutni datum spremembe in „odprtim" končnim intervalom.

Na ta način ohranjamo celotno zgodovino sprememb zapisov, števila samih sprememb zapisov pa nam ni treba predvideti vnaprej. Ta način posodabljanja zapisov lahko uporabljamo tudi pri opisnih tabelah – s kratico mu pravimo SCD2 (angl. *Slowly Changing Dimensions Type 2*) in je najučinkovitejši način hranjenja zgodovine sprememb z najnižjo porabo prostora.

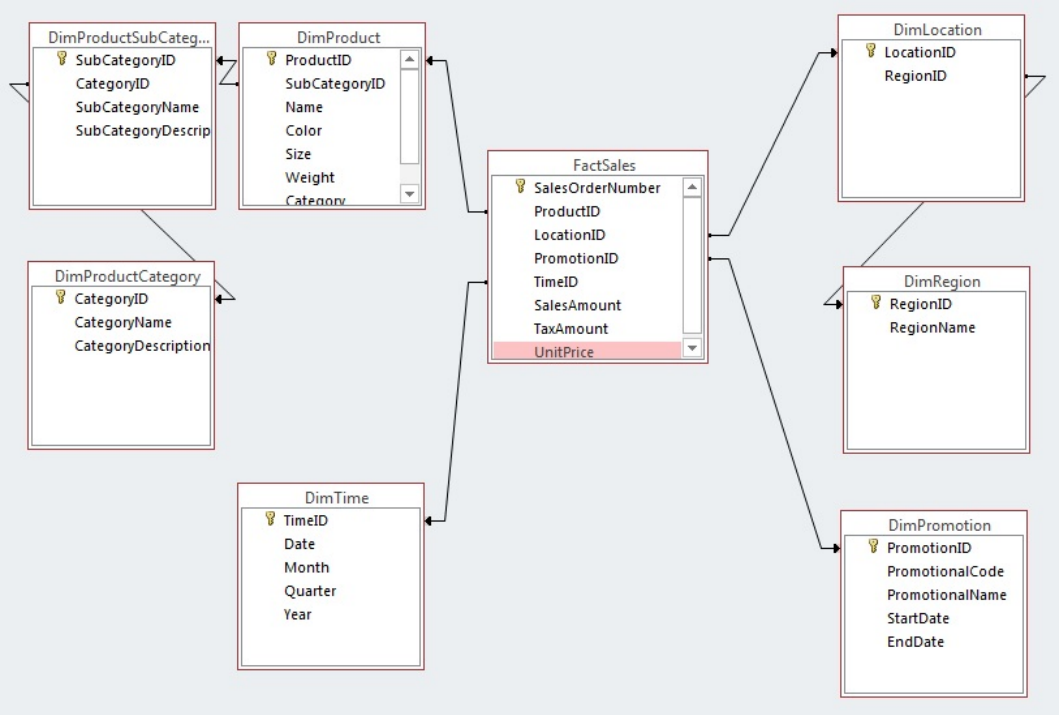
Kot omenjeno, so lahko tabele dimenzijskega modela organizirane v topologijo zvezde ali snežinke. Pri zvezdni topologiji so vse opisne tabele povezane neposredno s tabelami dejstev za razliko od snežinkaste topologije, kjer so opisne tabele lahko povezane med seboj [13, 12, 11].

V spodnjem primeru (slika 2.4) vsebuje opisna tabela DimProduct vse informacije o produktih na enem mestu [12].



Slika 2.4: Dimenzijski model - topologija zvezde

Če naknadno normaliziramo tabelo DimProduct, dobimo dve dodatni opisni tabeli DimProductSubCategory in DimProductCategory, ki sta zaporedno vezani na tabelo DimProduct. Tako smo dimenzijski model iz prejšnjega primera preoblikovali iz zvezdne v snežinkasto topologijo [12] (slika 2.5).



Slika 2.5: Dimenzijski model - topologija snežinke

Poglavje 3

Pregled povezanih tehnologij

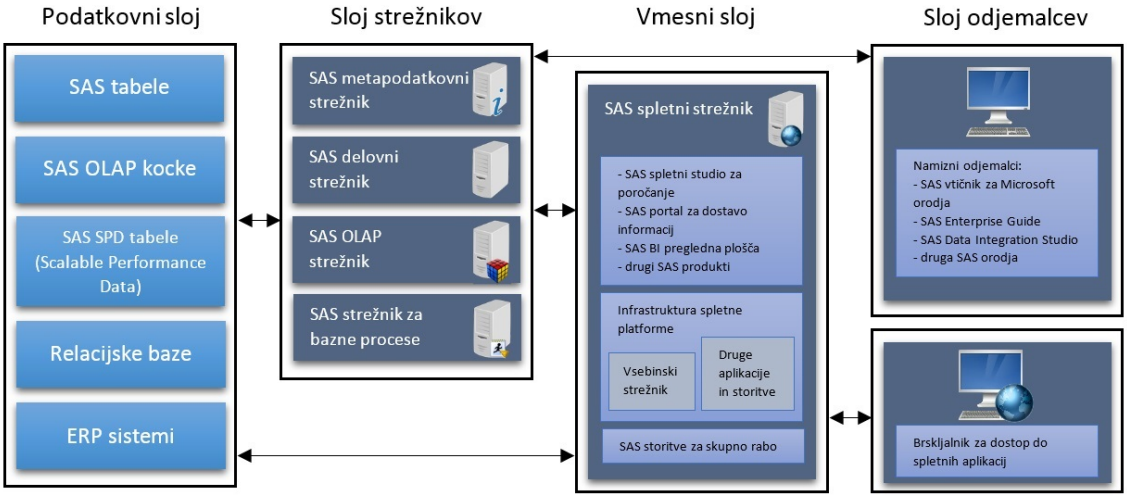
3.1 Predstavitev sistema SAS

Statistični analitični sistem (v nadaljevanju SAS – angl. *Statistical Analitical System*) je programska oprema, razvita v podjetju SAS Institute in namenjena poslovni inteligenci. Pod ta opis uvrščamo napredno analitiko, podatkovno upravljanje, podatkovno rudarjenje, analize predvidevanja ipd.

Programska oprema SAS nudi možnost upravljanja, spreminjanja, rudarjenja in pridobivanja podatkov iz vrste različnih podatkovnih virov in tudi njihovo analizo. Ponuja grafični uporabniški vmesnik za poslovne oziroma netehnične uporabnike in tudi okolje za programiranje v jeziku „SAS Base", s katerim je moč narediti praktično kakršne koli podatkovne transformacije in analize [10].

Arhitektura sistema oziroma pogramske opreme SAS sledi standardni organizaciji odjemalec – strežnik, katere sestava vsebuje vrsto različnih strežnikov, odjemalcev, povezovalnih storitev in metapodatkov.

Sistem opredeljuje večplastna topologija z naslednjo razvrstitvijo, kot prikazuje slika 3.1:



Slika 3.1: Plasti sistema SAS

- **podatkovna plast**: plast zajema zbirke podatkovnih virov, kot so podatkovne baze, tabele SAS, zunanje datoteke ipd,
- **plast strežnikov**: v tej plasti se nahajajo strežniki sistema, ki so primerljivi storitvam operacijskega sistema Windows. Njihova naloga je servisiranje zahtev aplikacij odjemalcev,
- **vmesna plast**: (včasih imenovan tudi „spletna plast") vsebuje vse potrebno za izvajanje spletnih aplikacij, kot so SAS Web Report Studio, SAS Information Delivery Portal, OLAP viewer, Stored Process Web Application ...
- **plast odjemalcev**: sem sodijo vsa orodja, ki jih odjemalci uporabljajo. Vsako je namenjeno svoji vrsti uporabe in za namene obdelave podatkov ter poročanja se najpogosteje uporabljata:
  - Enterprise Guide: prvotni namen orodja je izdelava poročil. Prav tako pa je mogoča obdelava podatkov za poročanje po meri in zahtevnejše obdelave z uporabo programske kode SAS Base.

– Data Integration Studio: prvotni namen orodja je izdelava procesov za podatkovno integracijo. Orodje je sorodno prejšnjemu s to razliko, da vsebuje vnaprej pripravljene procese (koncept črnih škatlic) za obdelavo podatkov. Tudi pri tem lahko z uporabo programske kode SAS Base dodamo poljubno funkcionalnost oziroma razvijemo svoje črne škatlice s točno določeno logiko obdelave.

Uporabniška orodja so prvotni vmesnik med končnimi uporabniki in sistemom SAS. Uporabnikom omogočajo prijavo na isti strežnik, tako da ne glede na vrsto orodja vsi dostopajo do istih podatkovnih struktur, podatkov in metapodatkov [14, 15].

Metapodatkovni strežnik: metapodatki služijo kot povezovalno sredstvo plasti. Metapodatki, ki opisujejo celotno okolje SAS, so shranjeni v repozitorijih, s katerimi upravlja metapodatkovni strežnik. Ta dostopa do repozitorijev, ko prejme zahtevo od aplikacije odjemalca ali drugih strežnikov SAS. Metapodatke same po sebi lahko opredelimo kot opise podatkov. Na primer, struktura neke tabele je predstavljena z metapodatki, ki definirajo ime tabele, imena stolpcev, tipe in formate stolpcev, primarne in tuje ključe, indekse ipd. Metapodatki pa niso v uporabi zgolj za opise struktur tabel, ampak hranijo tudi informacije o nastavitvah celotnega sistema SAS – konfiguracijo ostalih strežnikov, uporabniške račune, pravice dostopa, definicije knjižnic itd [15]. Primeri servisiranja zahtev metapodatkov–nega strežnika:

- ko uporabnik požene neko orodje in se prijavi v sistem s svojimi uporabniškimi podatki, metapodatkovni strežnik preveri podane podatke v repozitoriju za danega uporabnika in določi, ali mu je dovoljena raba orodja ali ne;

- ko uporabnik požene bazni proces za poročanje ali obdelavo podatkov, metapodatkovni strežnik dobi zahtevo iz uporabniške aplikacije za izvajanje baznega procesa; metapodatkovni strežnik preveri, kateri od strežnikov za izvajanje baznih procesov je na voljo ter mu pošlje proces v izvršbo.

3.2 Programsko okolje SAS Base

SAS Base je integrirano programsko okolje, zasnovano za dostop do podatkovnih virov, pretvorbo, manipulacijo in analizo podatkov ter poročanje. Komponente oziroma tehnike programiranja lahko glede na funkcionalnost razdelimo v naslednje kategorije [16]:

- programski jezik za pripravo/obdelavo in analizo podatkov (angl. *DATA Step Language*),
- makro programski jezik (angl. *Macro Facility Language*),
- sistem za poročanje rezultatov (angl. *Output Delivery System*).

Programi, napisani v programskem jeziku SAS Base, lahko uporabljajo eno in/ali vse od naštetih komponent. Kodo se izvaja sekvenčno po korakih, ki so zaključen deli procesne logike obdelave podatkov. Vsak korak pa je lahko eden od naslednjih dveh tipov:

- korak za obdelavo podatkov – podatkovni korak (angl. *DATA Step*),
- korak za analizo podatkov – procesni korak (angl. *PROC Step*).

Podatkovni korak je sestavljen iz niza ukazov. Ukazi so lahko deklarativni ali manipulacijski. S pomočjo deklarativnih ukazov lahko na primer določimo, katero tabelo bomo obdelovali in kakšna naj bo njena izhodna struktura – torej vplivamo na videz oziroma predstavitev podatkov. Rezultati manipulacijskih ukazov pa so vsebinske spremembe podatkov samih.

Procesni koraki se sestojijo iz procesnih ukazov oziroma klicev procedur in funkcij. Ti izvajajo analize podatkov (računanje statistik, generiranje grafov in poročil, razvrščanje tabel ipd.), njihov rezultat pa se zapiše v novo tabelo za nadaljnjo obdelavo ali v obliki poročila. Obstaja več kot 300 funkcij in procedur. Vsaka od njih vsebuje številne parametre in obsežen delež programske logike, ki jo lahko poljubno prilagajamo analitičnim potrebam [17, 18, 26].

Vredno je tudi omeniti možnost uporabe jezika SQL. To omogoča ena od procedur, s pomočjo katere lahko uporabljamo standardno sintakso SQL v kombinaciji z ukazi SAS in makro programskim jezikom.

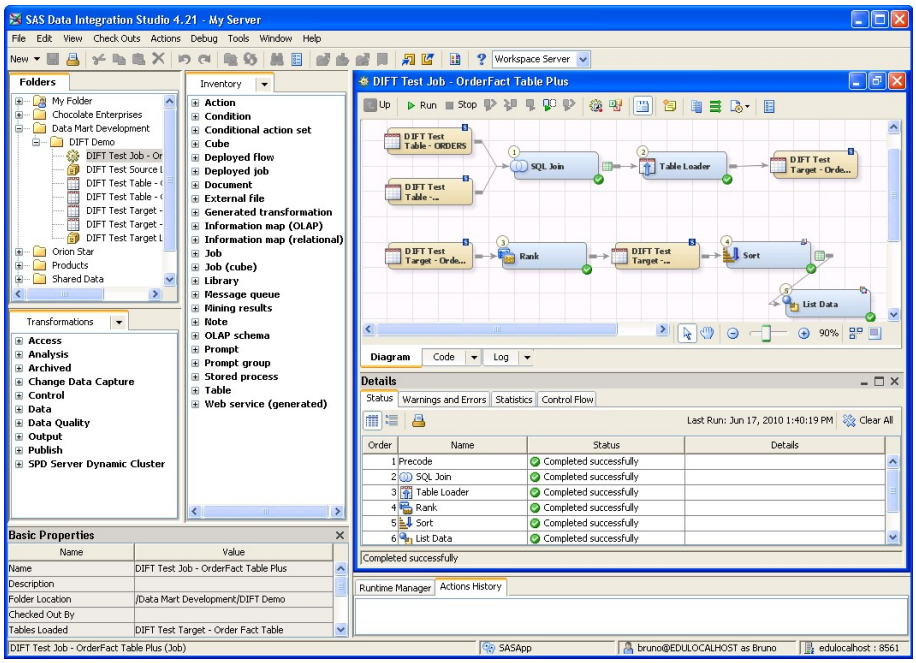
Programsko kodo SAS je moč dodatno razširiti z uporabo makro programskega jezika. Jezik omogoča razširitev funkcionalnosti, parametrizacijo in prilagajanje programov SAS – odvisno od načina uporabe. S pametno uporabo pa lahko tudi zmanjšamo število vrstic kode, potrebnih za izvajanje istih ukazov oziroma korakov. Tako lahko uporabnik imensko definira makro spremenljivke, katerih vrednosti so poljubni nizi znakov. To je lahko določen ukaz ali kos programske kode, ime tabele, vrednost nekega parametra ipd. Ko je makro spremenljivka definirana, se v kodi programa sklicujemo na njeno ime, le-ta pa se razreši oziroma pretvori v definirane nize znakov, ko program pošljemo v izvajanje. Na isti način je moč definirati makro procedure, s katerimi lahko sestavimo večje segmente kode SAS Base [27, 19].

3.3 Orodje za izdelavo procesov ETL - Data Integration Studio

Integracija podatkov je proces pretvorbe, konsolidacije in poenotenja podatkov iz vrste različnih izvorov. Za izvajanje teh aktivnosti se v sistemu SAS uporablja orodje Data Integration Studio (v nadaljevanju: DIS), kot da prikazuje slika (slika 3.2). To je vizualno orodje za razvoj in implementacijo procesov za integracijo podatkov ne glede na tipe podatkovnih izvorov in platforme, kjer je orodje nameščeno. Večuporabniško okolje nudi kolaboracijski pristop razvoja procesov med razvijalci, z uporabo sistema za vodenje sprememb (angl. *Change Management Facility*).

Uporabniki s privzetimi gradniki orodja ali z razvojem lastnih gradijo procesne diagrame za pretvorbo podatkov (diagrame poteka pretvorb), ki jim pravimo „opravila” (angl. *jobs*). Implementirana opravila so shranjenja v obliki metapodatkov na strežniku za metapodatke – do njih lahko dostopajo vsi uporabniki, ki imajo za to pravice [20, 21].

DIS ob sprožitvi nekega opravila v ozadju sestavi kodo SAS iz vseh uporabljenih gradnikov procesnega diagrama in jo pošlje v izvajanje na strežnik. Tako lahko opredelimo orodje DIS kot generator kode SAS, kjer je proces generiranja in izvajanja kode transparenten za uporabnika.



Slika 3.2: Orodje Data Integration Studio [22]

Vsi gradniki v orodju DIS so metapodatkovni objekti, ki so registrirani na trenutnem strežniku za metapodatke. Vsak gradnik se glede na svoj tip nahaja v določeni skupini.

Pomembnejši gradniki za izdelavo opravil so:

Knjižnice: Knjižnica v sistemu SAS opredeljuje povezavo sistema s podatkovnim izvorom. Ta je lahko zbirka fizičnih datotek SAS oziroma tabel in katalogov, shranjenih na trdem disku, podatkovno skladišče v ločenem sistemu za upravljanje s podatkovnimi bazami (na primer Oracle, MySQL, DB2 ...), datoteke Excel ipd.

Sistem SAS nudi možnost povezave s praktično vsemi tipi podatkovnih izvorov. Ima že vnaprej opredeljene tipe povezav vseh sistemov za upravljanje s podatkovnimi bazami. V primeru povezovanja s sistemom Oracle se uporabi tip povezave za ta sistem, ki knjižnici določa vrsto in število parametrov, potrebnih za vzpostavitev povezave – te vsebujejo: ime knjižnice, tip povezave za Oracle strežnik, naslov strežnika, ime sheme, uporabniško ime ter geslo. V primeru, da dostopamo do

zbirke fizičnih datotek SAS, uporabljamo parametre: ime knjižnice, tip povezave za datoteke SAS ter absolutno pot do imenika, kjer se datoteke nahajajo [21, 23].

Vsi parametri knjižnic so shranjeni na strežniku za metapodatke, do katerih dostopajo odjemalci. Knjižnico lahko dodelimo odjemalcu za aktivno sejo po potrebi oziroma ko uporabnik to zahteva, lahko pa je samodejno dodeljena pri vsaki seji, ki jo odjemalec vzpostavi z strežnikom.

Knjižnice delimo tudi po načinu dodelitve, in sicer na trajne in začasne. Trajne knjižnice so vse, ki so shranjene na strežniku za metapodatke in se dodelijo vsaki odjemalčevi seji samodejno ali na željo uporabnika. Začasne knjižnice si lahko uporabnik dodeli sam v trenutno aktivni seji z ukazom libname in se po zaključku seje izbrišejo.

Tabele: tabele v orodju DIS so opisane z metapodatki in so kot vsi drugi gradniki shranjene na strežniku za metapodatke. Vsako tabelo, do katere podatkov želimo dostopati, moramo najprej registrirati v sistem SAS, da je na voljo odjemalcem.

Seveda je mogoče operirati tudi neposredno s fizičnimi tabelami, ne le z njihovimi metapodatkovnimi reprezentacijami. Vsa orodja SAS, ki podpirajo programiranje v kodi SAS Base, lahko dostopajo do fizičnih tabel z deklaracijo začasne knjižnice. V teh primerih lahko uporabljamo tabele le v programskih gradnikih in niso vidne uporabnikom kot metapodatkovni objekt v orodju samem, če jih predhodno ne registriramo.

Proces registracije tabele je izgradnja metapodatkovne predstavitve njene strukture. Preden lahko sprožimo proces registracije, moramo definirati trajno knjižnico, za katero želimo tabele registrirati. Ob sprožitvi procesa sistem SAS uporabi parametre knjižnice za vzpostavitev povezave za dani podatkovni izvor. Nato zgradi zbirko vseh fizičnih tabel, ki so na voljo in uporabniku ponudi izbiro, katere od njih želi registrirati. Po potrditvi selekcije sistem SAS vsaki izbrani tabeli skenira sestavo in naredi njen ekvivalent v obliki metapodatkovnega objekta z naslednjimi informacijami [21, 23]:

- ime tabele,

- imena, labele, formate, dolžine, tipe stolpcev tabele,
- definicije indeksov, prvotnih ključev, tujih ključev in stolpičnih omejitev,
- način kodiranja,
- pravice dostopa itd.

Vse registrirane tabele so nato uporabnikom glede na njihove pravice dostopne v vsakem orodju SAS.

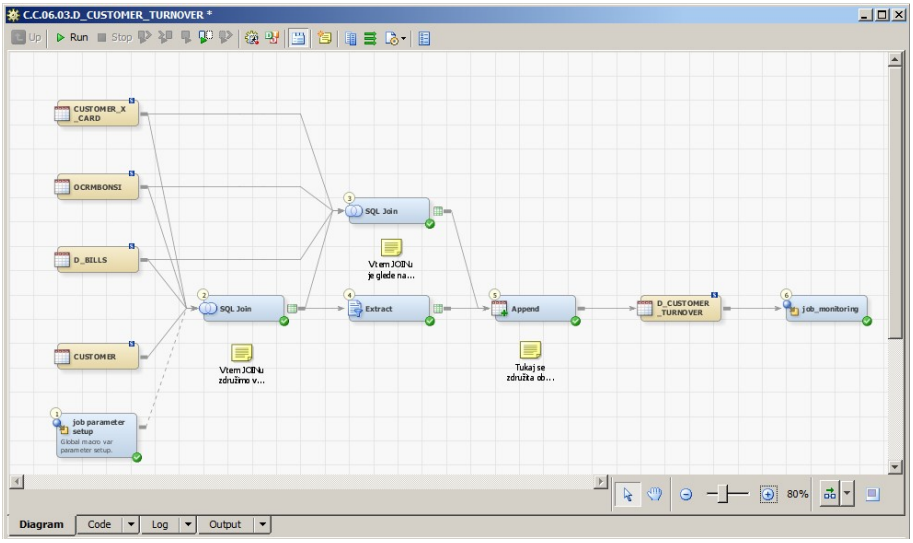
Transformacije: predstavljajo zaključene procesne enote, ki izvršijo neko nalogo. Lahko jih opredelimo kot črne škatlice, ki imajo vnaprej definirano procesno logiko. Vsaka transformacija tipično uporablja vhodne podatke, nastavitvene parametre in izhodni rezultat. V skupine jih delimo glede na njihovo funkcionalnost in namen uporabe:

- dostop do podatkov,
- analiza podatkov,
- manipulacija podatkov,
- kontrola procesov,
- upravljanje s kakovostjo podatkov,
- upravljanje z zajemom sprememb podatkov itd.

Vsaki transformaciji lahko uporabnik nastavlja ponujene parametre, s katerimi vpliva na njeno obnašanje. Poleg nastavljanja parametrov je mogoče dodati poljubno programsko kodo, ki se izvrši pred in/ali po končanem izvajanju integrirane procesne logike, in tudi spreminjati kodo integrirane procesne logike.

V primeru, da obstoječe transformacije ne zadovoljujejo vseh uporabnikovih zahtev, lahko uporabnik sam definira lastne transformacije z ustrezno procesno logiko, ustreznim številom vhodov in izhodov ter ustrezno parametrizacijo. To je še posebej uporabno v primerih, ko imamo podoben proces obdelave podatkov v več različnih opravilih. Prav tako je na voljo transformacija „uporabniška koda" (angl. *User Written Code*), v kateri lahko uporabnik z uporabo programske kode SAS razvije poljubno procesno logiko [21].

Opravila: opravilo (angl. *Job*) je zbirka gradnikov, ki v določenem zaporedju izvajajo naloge. V orodju DIS je opravilo predstavljeno kot diagram procesnega toka. Izdelava diagrama poteka v urejevalniku opravil, kjer z dodajanjem gradnikov (tabel, transformacij, drugih opravil ...) in medsebojnim povezovanjem le-teh gradimo vizualno predstavitev procesnega toka obdelave podatkov (slika 3.3).



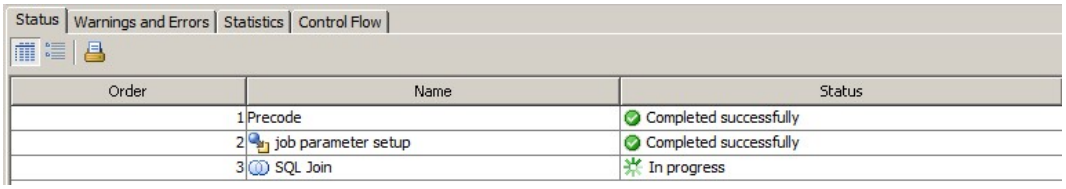
Slika 3.3: Primer opravila

Kadar razvijamo opravila za integracijo podatkov, se prvotno poslužujemo gradnikov transformacij in tabel. Vsaki transformaciji v procesnem toku je dodeljena številka, s pomočjo katere nadzorujemo zaporedje izvajanja in posredno vplivamo na proces generiranja kode.

Kot omenjeno, ima večina transformacij definiranega enega ali več vhodov, na katere povežemo tabele (pri nekaterih transformacijah tudi druge gradnike), in nič ali več izhodov, kamor transformacija zapiše rezultat – ponavadi je to nova tabela ali pogled na tabelo.

Ko sprožimo izvajanje opravila, orodje najprej generira programsko kodo vsake uporabljene transformacije v diagramu ter jo pošlje na strežnik v izvajanje. Pri tem strežnik pripravi novo sejo, v kateri se opravilo izvaja, ter dodeli začasno delovno knjižnico (angl. *Work*), ki služi kot delovni prostor opravila, kamor se shranjujejo

vse izhodne tabele transformacij [21, 24]. Ob prekinitvi seje se knjižnica samodejno pobriše. Med izvajanjem so vidni statusi vsake izvedene transformacije in katera je trenutno v izvajanju, kot je prikazano na sliki 3.4.



Slika 3.4: Statusi transformacij pri izvajanju opravila

Poglavje 4

Integracija podatkov in procesi ETL

Za podjetja z veliko različnimi izvornimi sistemi je še posebej pomembno, da imajo enoten pregled nad svojim podatkovnim premoženjem. Proces integracije podatkov je proces združevanja podatkov iz več različnih izvornih sistemov z namenom, da bi zagotovili enoten pogled nad njimi. Ti sistemi v veliki večini primerov uporabljajo različne tehnologije, kar predstavlja dodaten izziv za poenotenje in preureditev velikih količin podatkov, ki jih implementirani procesi izvajajo [3].

Najprepoznavneša oblika uporabe podatkovne integracije je izgradnja podatkovnega skladišča, kamor se stekajo vsi podatki podjetja, ki jih generirajo izvorni sistemi. Pod drugo uporabo lahko uvrstimo izgradnjo podatkovnega modela, ki je manjša podmnožica podatkovnega skladišča, prirejena izključno določenemu oddelku za potrebe analiz in/ali poročanja.

Kadar imamo opravka z integracijo podatkov, se srečujemo s tako imenovanimi procesi ETL. Angleška kratica ETL, sestavljena iz besed Extract, Transform in Load, robustno opiše tri sklope procesov, ki se jih poslužujemo pri implementacijah [3, 4].

Torej, procese glede na njihovo funkcionalnost delimo v naslednje tri sklope:

- procesi za zajem podatkov iz podatkovnih virov (črpanje),

- procesi za obdelavo ali preoblikovanje podatkov (transformiranje),
- procesi za nalaganje podatkov (polnjenje).

Pri navedenih sklopih na prvi pogled manjka proces zagotavljanja kakovosti podatkov. Ta je še posebej pomemben pri celotnem procesu integracije podatkov pri izgradnji podatkovnih skladišč in tudi manjših podatkovnih modelov. V praksi se ponavadi izkaže, da podjetja, ki naj bi uporabljala čisto centralizirano podatkovno skladišče, iz katerega črpajo podatke za analitične potrebe, še vedno posedujejo podatke relativno vprašljive kakovosti.

Procesi za zagotavljanje kakovosti podatkov so ponavadi implementirani kot samostojni procesi, lahko pa njihovo logiko pravil dodamo procesom za transformacijo podatkov. Od omenjenih dveh pristopov je elegantnejši prvi. Tako lahko razširimo naštete tri sklope z dodatnim sklopom procesov [4]:

- procesi za zajem podatkov iz podatkovnih virov (črpanje),
- procesi za zagotavljanje kakovosti podatkov (čiščenje),
- procesi za obdelavo ali preoblikovanje podatkov (transformiranje),
- procesi za nalaganje podatkov (polnjenje).

Zajem podatkov – črpanje

Naloga procesov prvega sklopa je prenos vseh potrebnih podatkov iz izvornih sistemov v pripravni prostor (angl. *Staging Area*), od koder so dostopni nadaljnjim procesom. Bistveno je, da so procesi zmožni opraviti svojo nalogo s čim manjšo bremenitvijo razpoložljivih sistemskih virov in da njihovo izvajanje nima negativnega vpliva na izvorni sistem s stališča zmogljivosti, odzivnega časa in ekskluzivne rabe (zaklepanja) objektov.

Poznamo naslednje metode za zajem podatkov, katerih izbor je odvisen od karakteristik izvornega sistema [2, 6, 5]:

- Zajem z obvestilom spremembe: nekateri izvorni sistemi so sposobni obveščanja o podatkovnih spremembah in podajanja opisov sprememb. To je najlažji

način zajema podatkov, kjer izvorni sistem sam posreduje informacije, s pomočjo katerih procesi zajamejo posodobljene podatke – v najboljšem primeru izvorni sistem pripravi samo posodobljene podatke, katerih proces zajame (ostali nespremenjeni podatki niso dostopni procesu zajema).

- Inkrementalni zajem oziroma zajem s pomočjo zaznavanja sprememb (CDC – angl. *Change Data Capture*) – zajem posodobljenih podatkov: pri tej metodi beležimo zgodovino sprememb podatkov ali spremembe po času in jo lahko implementiramo na več načinov:

  - uporaba časovnega žiga: če izvorni sistem beleži datum in čas vsake spremembe podatkov, lahko z njegovo pomočjo prepoznamo tiste, ki so bili dodani ali spremenjeni od trenutka zadnjega zajema;
  - uporaba časovnih particij: nekateri sistemi particionirajo podatke po datumu veljavnosti; podatki so razdeljeni v particije za določeno časovno obdobje; s pomočjo particij enostavno zajamemo podatke želenega obdobja;
  - uporaba datuma nastanka: podobno kot pri uporabi časovnega žiga izvorni sistem beleži datum in čas nastanka podatkov; razlika je v tem, da pri tem načinu izvorni sistem ob spremembi ne posodablja obstoječih podatkov, ampak jih doda kot nov zapis;
  - primerjanje vsebin: najbolj časovno potraten način, ki se ga poslužujemo samo v primerih, ko nimamo druge izbire; procesi vedno zajamejo vse razpoložljive podatke iz izvora ter s primerjavo podatkov predhodnega zajema opredelijo, kateri so bili posodobljeni, spremenjeni, izbrisani.

  Vsi omenjeni načini (z izjemo zadnjega) imajo pomanjkljivost pri ravnanju z izbrisanimi zapisi podatkov, ki se jih, če sistem sam ne beleži izbrisa, ne vidi.

- Celoten zajem: zajem vseh razpoložljivih podatkov iz izvornega sistema. Pri tej metodi ni treba slediti zgodovini sprememb podatkov, ker z vsakim zajemom dobimo trenutno sliko podatkov, kot so predstavljeni v izvornem sistemu.

Zagotavljanje kakovosti podatkov – čiščenje

Procesi za zagotavljanje kakovosti podatkov so sestavljeni iz pravil, ki služijo kot kontrolni mehanizmi za odkrivanje in posledično odpravljanje nedoslednosti podatkov. Proces s pravili preveri določene značilnosti podatka in glede na rezultat preverjanja izvede primeren ukrep.

Celoten potek izgradnje procesov za zagotavljanje kakovosti podatkov zajema naslednje naloge [25]:

- Profiliranje podatkov: proces pregledovanja podatkov in zbiranja informacij o napakah, nedoslednostih ter redundancah, podatkovnih nepopolnosti in drugih anomalij. Na primer: razponi vrednosti, vzorčenje tekstovnih vrednosti, porazdelitev vrednosti, odstopanja od povprečja, odstotek mankajočih vrednosti ipd.
- Definicija pravil in ukrepov: rezultat profiliranja podatkov služi kot predloga za definicijo pravil o najdenih anomalijah in definicijo ukrepov, potrebnih za njihovo reševanje.
- Razvoj procesov: integracija definiranih pravil z logiko reševanja/odpravljanja napak.

Pravila zagotavljajo podatkovno skladnost z definiranimi omejitvami podatkovnih struktur, kamor podatke v končni fazi polnimo, kot tudi njihovo doslednost, popolnost in enotnost – podatke standardizirajo.

Pravila, na primer izvajajo preverjanja:

- referenčne integritete,
- edinstvenosti primarnih ključev,
- integritetne omejitve.
- podvojenih vrednosti,
- poenotenja ali standardizacijie vrednosti,
- manjkajočih vrednosti,

- razpona vrednosti,
- drugih poslovnih pravil glede na panogo.

Obdelava in preoblikovanje podatkov – transformiranje

Procesi v tem sklopu podatke preoblikujejo v vnaprej določeno obliko, kot jo narekuje končna podatkovna struktura. V nekaterih primerih podatki ne potrebujejo preoblikovanja, v drugih primerih pa je treba uporabiti eno ali več transformacij za pretvorbe merljivih količin v standardne enote, agregacije, transponiranje, računanje novih metrik, združevanje podatkov iz različnih tabel, generiranje primarnih in/ali nadomestnih ključev, sortiranje podatkov itd [2].

Nalaganje podatkov – polnjenje

Zadnji sklop procesov napolni preoblikovane podatke v končni podatkovni model, ki je ponavadi podatkovno skladišče ali njegova manjša podmnožica. Tako kot pri prvem sklopu procesov poskušamo zagotoviti izvajanje s čim nižjo bremenitvijo sistemskih virov [2].

V praksi imamo na razpolago več različnih tehnik polnjenja podatkov. Izbor je odvisen od dejavnikov, kot so količina podatkov in hranjenje zgodovine, tipi tabele, tehnične omejitve ciljnega sistema in/ali strojne opreme, posebnosti poslovnih zahtev, predviden način uporabe končnega podatkovnega modela ipd.

Najbolj pogosto uporabljene tehnike polnjenja so:

- dodajanje (angl. *Append*): dodajanje ali priključevanje zapisov izvorne tabele ciljni tabeli – ohranjamo zgodovino podatkov;
- prepisovanje (angl. *Replace*): prepisovanje ciljne tabele z izvorno tabelo – ne ohranjamo zgodovine podatkov, ostane samo zadnje stanje;
- posodabljanje in dodajanje (angl. *Update/Insert*): posodabljanje obstoječih zapisov ter dodajanje novih;
- posodobitve časovnih posnetkov (angl. *Slowly Changing Dimensions Type 2*): tudi pri tej tehniki se izvaja posodabljanje obstoječih zapisov ter dodaja-

nje novih z uporabo dodatnih datumskih stolpcev (podrobnejši opis uporabe te tehnike v 2. poglavju).

Poglavje 5

Logična struktura ogrodja ETL

Poslovanje današnjih podjetij je močno odvisno od njihovega podatkovnega premoženja. Ažurnost in razpoložljivost podatkov za potrebe nemotenega izvajanja poslovnih procesov sta še kako pomembni.

Z avtomatizacijo izvajanja procesa podatkovne integracije zagotavljamo pravočasno razpoložljivost podatkov. Vendar še tako dobro zasnovan in avtomatiziran proces ni zmožen premostiti vseh problemov, s katerimi se tipično srečujemo – izpadi strežnikov, zamenjava podatkovnih izvorov, spremembe struktur izvornih tabel ipd. Majhna napaka ali sprememba imata posledično lahko velik vpliv na operativo poslovnega procesa. Zato je dober nadzor procesov podatkovne integracije kritičen za hitro odkrivanje in odpravljanje napak.

Logično ogrodje ETL je bilo zasnovano kot aplikativna rešitev za nadzor in upravljanje opravil. Sestavljajo ga uporabniško razvite makro procedure, nadzorne tabele, transformacije ter nadzorna oziroma glavna opravila, ki omogočajo centraliziran pregled in nadzor nad izvajanjem celotnega procesa ETL z upoštevanjem predpisanih medsebojnih odvisnosti opravil.

V tem poglavju so predstavljeni razviti gradniki ogrodja, njihov namen, način uporabe, potek izvajanja opravil ter avtomatizacija izvajanja. Vsa opravila z uporabo omenjenih gradnikov med izvajanjem posredujejo informacije, ki se hranijo v nadzornih tabelah. Z njihovo pomočjo takoimenovana glavna opravila nadzirajo potek in beležijo uspešnost izvajanja ter obveščajo o morebitnih zapletih.

5.1 Hierarhija povezovanja opravil v logične enote

Opravila so porazdeljena v logične zbirke skupin, imenovane *valovi*, ti pa so še naprej porazdeljeni v logične zbirke skupin, imenovane *sklopi*.

Opravilo: kot opisano v 3. poglavju, predstavlja opravilo samostojen proces obdelave podatkov od izvora do ponora. Med izvajanjem se vhodni podatki pretvorijo, kot je to definirano v procesnem diagramu opravila, ter odložijo na izhod. Med nekaterimi opravili veljajo medsebojne odvisnosti, ki vplivajo na njihov vrstni red izvajanja. Recimo, opravilo za preverjanje kakovosti podatkov ne more začeti z izvajanjem, dokler opravilo za zajem podatkov ni uspešno zaključeno.

Val: združitev posameznih opravil v logične skupine valov. Vsak val je tako sestavljen iz zbirke enega ali več opravil, katerih skupna lastnost je medsebojna neodvisnost. Opravila, zajeta znotraj vsakega vala, se lahko izvajajo sočasno.

Sklop: združitev posameznih valov v logične skupine sklopov. Vsak sklop je tako sestavljen iz zbirke valov in opredeljuje vrstni red zaporednega izvajanja. S tem ohranjamo medsebojne odvisnosti, kot so definirane na nivoju opravil.

Ogrodje ETL uporablja tako imenovana glavna opravila (angl. *Master Jobs*) za proženje vseh opravil. Ta z uporabo razvitih sestavnih delov ogrodja v pravilnem vrstnem redu proži sklope (posledično valove, ti pa posledično opravila) enega za drugim. Za pravilno opravljeno obdelavo podatkov skozi celoten potek zagotavlja strogo določen vrstni red proženja sklopov (kot tudi valov vsakega posameznega sklopa). Recimo, sklop opravil za polnjenje podatkov v končni podatkovni model ne more začeti z izvajanjem, dokler sklop opravil za pretvorbo podatkov ni uspešno zaključen.

Opredelitve vseh logičnih skupin (sklopov, valov) opravil, njihove medsebojne odvisnosti in zaporedje izvajanja so zapisane v eni od nadzornih tabel ogrodja, ki glavnim opravilom služi kot vodilo ali načrt za potek izvajanja.

5.2 Razvite komponente ogrodja in pravila uporabe

Nadzorne tabele:
Nadzorne tabele uporabljajo predvsem glavna opravila. Z njimi upravljajo s pomočjo razvitih makro procedur in/ali transformacij ogrodja. Vsaka tabela ima določen namen, ki podpira specifično funkcionalnost ogrodja.

Osnovna funkcionalnost ogrodja je podprta s privzetimi tabelami, ki služijo za beleženje povratnih informacij med izvajanjem opravil, upravljanje in nadzor izvajanja, definicijo vrednosti parametrov ipd.

V praksi se glede na uporabniške potrebe zbirko tabel razširi z dodatnimi. Ponavadi se te uporabljajo za dodatna preverjanja prenosov podatkov ali poročanja statistik.

Privzete tabele ogrodja so:

- Tabela zgodovine izvajanja opravil (angl. *Job Monitoring*): hrani zgodovino, rezultatov in osnovnih meritev, izvršenih opravil. Vsako opravilo (z izjemo glavnih) ob začetku izvajanja doda v tabelo nov zapis, ki ga ob zaključku izvajanja posodobi z naslednjimi informacijami:
  - ime uporabnika, ki je opravilo sprožil,
  - številka procesa, ki je bila opravilu dodeljena s strani operacijskega sistema,
  - datumsko obdobje, ki mu pripadajo podatki, uporabljeni pri izvajanju opravila,
  - ime opravila, ime ciljne knjižnice in ciljne tabele, ki jo opravilo polni,
  - število vrstic v ciljni tabeli pred in po izvajanju,
  - uspešnost izvajanja, čas začetka in konca izvajanja,
  - absolutna pot do dnevnika izvajanja.

- Tabela zgodovine izvajanja glavnih opravil (angl. *Master Job Executions*): hrani zgodovino rezultatov izvršenih glavnih opravil. Kot pri prvi tabeli se tudi tu hranijo informacije:
  - datumsko obdobje, ki mu pripadajo podatki, uporabljeni pri izvajanju opravil,
  - čas začetka in konca izvajanja glavnega opravila,
  - število pripravljenih opravil za izvajanje,
  - število uspešno zaključenih opravil,
  - število zaključenih opravil z opozorili,
  - število zaključenih opravil z napakami,
  - število neizvedenih opravil.
- Tabela zbirke opravil (angl. *Job List*): hrani zbirko vseh razvitih opravil, uporabljenih pri celotnem procesu obdelave podatkov. Vsak zapis določi logično podskupino vala ter sklopa, kateremu opravilu pripada in njihove medsebojne odvisnosti ter zaporedje izvajanja. Tabela je vodilo glavnim opravilom za proženje drugih opravil in zajema naslednje informacije:
  - edinstvena identifikacijska številka opravila,
  - ime opravila,
  - številka vala, ki mu opravilo pripada,
  - številka sklopa, ki mu val pripada,
  - odvisnost od drugih opravil,
  - zastavica za vključitev opravila v proces izvajanja,
  - zastavica za izključitev opravila iz procesa izvajanja.
- Tabela dodatnih nastavitev po meri (angl. *Custom Job Parameters*): zapisi v tabeli definirajo povezavo med opravili in uporabniško razvitimi makro procedurami, ki se izvajajo na začetku oziroma pred začetkom procesnega diagrama opravila. En zapis lahko definira uporabo makro procedure pri

določenem opravilu, zbirki opravil določenega vala ali celotnega sklopa opravil. Tega se poslužujemo v posebnih primerih (izjemah), ko želimo dodatno vplivati na izvajanje ali nadgraditi procesno logiko opravila (zakasnitve proženja opravil, zaznavanje posodobitev izvornih podatkov, predpripravo podatkov, nastavitev specifičnih parametrov ipd.).

Makro procedure:
Funkcionalnost ogrodja je podprta s številnimi uporabniško razvitimi makro procedurami. Klice procedur izvajajo uporabniško razvite transformacije, ki jih vključujemo v procesne diagrame opravil.

Procedure izvajajo naloge, kot so: nastavitev globalnih (skupnih) parametrov opravil, polnjenje nadzornih tabel, beleženje dnevnikov opravil, zaklepanje tabel, generiranje zbirk opravil za izvajanje ipd.

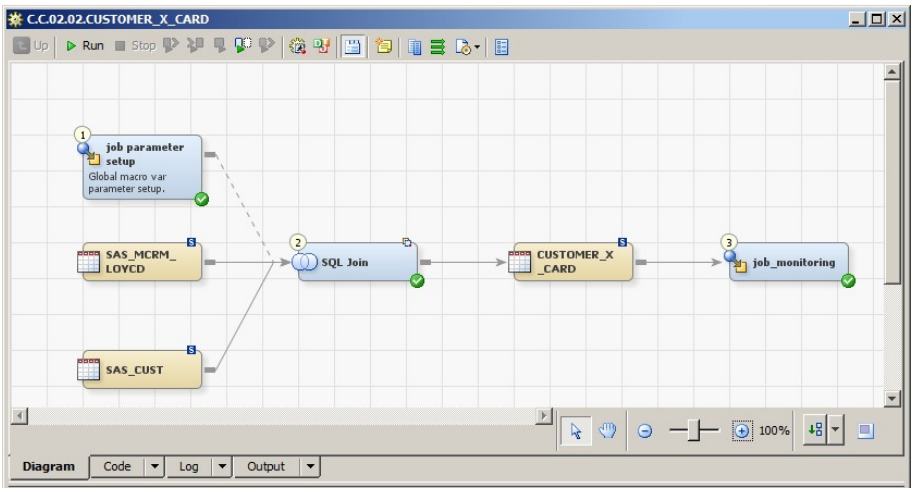
Vse makro procedure se nahajajo v posebnem imeniku sistema SAS, ki se samodejno dodeli vsaki aktivni seji in s tem omogoči opravilom njihovo rabo.

Transformacije:
Kot omenjeno, razvite transformacije ogrodja so vključene v procesne diagrame opravil. Njihova naloga je izvajanje makro procedur, ki podpirajo celotno funkcionalnost ogrodja. Njihov položaj v procesnih diagramih je strogo določen. Tako vsako opravilo za obdelavo podatkov sledi predpisanemu načinu uporabe transformacij, kar obenem zagotovi standardizacijo procesnih diagramov.

Standardizacija opravil in predpisi:
Za uspešno integracijo opravil z ogrodjem je predpisan način poimenovanja opravil in pravila izgradnje procesnih diagramov.

Imena opravil sledijo predpisani nomenklaturi. Sestavljajo ga črkovna koda (referencira sklop opravila), prvo število (referencira val opravila), drugo število (referencira zaporedno številko opravila v valu) in ime ciljne tabele, ki jo opravilo polni. Ime jasno določa, kateremu sklopu in valu opravilo pripada. Razlog takega poimenovanja ni le preglednost in organizacija, temveč je s tem podprta določena funkcionalnost ogrodja (nadgradnja procesne logike opravil z uporabniškimi makro procedurami).

Opravila pa morajo vsebovati določene privzete trasformacije ogrodja, in sicer kot je prikazano na sliki 5.1, se mora na prvem mestu procesnega diagrama nahajati transformacija *job parameter setup*, ter na zadnjem mestu transformacija *job monitoring*.



Slika 5.1: Pravilna raba transformacij ogrodja pri opravilu

5.3 Glavna opravila

Celotna funkcionalnost ogrodja pride do izraza pri glavnem opravilu. Njegova zasnova omogoča samostojno upravljanje in nadzorovanje poteka izvajanja opravil ter sprožitev celotnega procesa z enega mesta.

Ogrodje vsebuje eno privzeto glavno opravilo, ki se ga glede na zahteve uporabnikov uporabi za razvoj dodatnih. V praksi se v največji meri uporabljata dve glavni opravili. Prvo je namenjeno uporabnikom za ročno proženje preko orodja DIS, drugo pa samodejnemu proženju z uporabo sistemskih časovnih razporejevalnikov (angl. *System Task Scheduler*).

Obe opravili imata enako zasnovan procesni diagram. Razlika je pri opravilu za samodejno proženje, ki se ga nadgradi s procesno logiko za zaznavanje posodobitev izvornih podatkov. Ta je prilagojena glede na zmožnosti in način delovanja izvornih sistemov.

Glavno opravilo za ročno proženje vsebuje transformacijo (angl. *Master Execution Setup*), kjer uporabnik nastavi parametre izvajanja:

- datumsko obdobje, za katerega bodo podatki obdelani,
- izbira opravil oziroma sklopov opravil, vključenih v izvajanje,
- izbira opravil oziroma sklopov opravil, izključenih iz izvajanja.

Opravilo za samodejno proženje ima fiksno določene parametre – zbirke opravil oziroma sklopov opravil, ki so vključena oziroma izključena iz izvajanja. Parameter datumskega obdobja pa se nastavi samodejno z makro proceduro za posodobitve podatkov v izvornih sistemih.

Naloga glavnega opravila je torej proženje opravil ter upravljanje celotnega poteka izvajanja. Glavno opravilo uporablja dve tako imenovani *podporni opravili*, s katerima se upošteva hierarhija logičnih enot (sklop, val, opravilo). Obe podporni opravili, skupaj z glavnim, predstavljata vse tri nivoje hierarhije, in sicer:

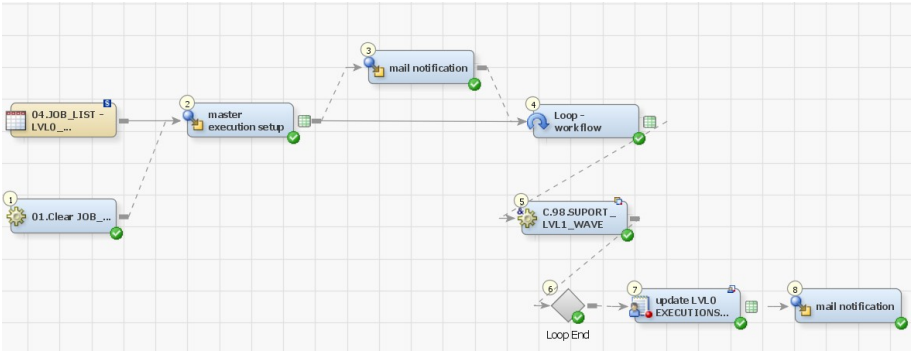
- glavno opravilo – nivo sklopov,
- prvo podporno opravilo – nivo valov določenega sklopa,
- drugo podporno opravilo – nivo opravil določenega vala.

Izvajanje se odvija skozi nivoje, kjer do sprožitve opravil za obdelavo podatkov pride pri zadnjem nivoju – to je omogočeno z medsebojnim gnezdenjem opravil. Tako ima glavno opravilo (operira na nivoju sklopa), vgnezdeno podporno opravilo (operira na nivoju vala), ta pa ima vgnezdeno naslednje podporno opravilo (operira na nivoju opravil), ki pošilja opravila v izvajanje. Opravila vsakega nivoja vsebujejo zanko, v kateri je vgnezdeno naslednje podporno opravilo. Zanka glavnega opravila v vsakem ciklu posreduje vrednosti vgnezdenemu opravilu, s katerimi se določijo parametri (obseg vrednosti) izvajanja njegove zanke. Na isti način se vrednosti za določanje parametrov zank prenašajo vse do najnižjega nivoja oziroma zadnjega podpornega opravila, kjer se zgradi zbirka opravil za izvajanje.

Potek izvajanja

Ko uporabnik sproži glavno opravilo, katerega procesni diagram prikazuje slika 5.2, se izvede naslednje zaporedje dogodkov:

- Glavno opravilo začne z izvajanjem:



Slika 5.2: Procesni diagram glavnega opravila

  - preverjanje vrednosti parametra datumskega obdobja: v primeru neveljavne vrednosti se v dnevnik opravila izpiše obvestilo o napaki in izvajanje se prekine;
  - označevanje opravil za vključitev oziroma izključitev iz izvajanja: prva transformacija glavnega opravila glede na nastavljene parametre za določitev zbirke opravil označi, katera opravila se bodo prožila in katera ne, s spreminjanjem vrednosti *zastavic* nadzorne tabele (Job List); proces iterativno označi vsa opravila, ki morajo biti vključena v izvajanje glede njihove predpisane odvisnosti od drugih opravil ne glede na to, ali so pogojena opravila zajeta v zbirki ali ne;
  - izgradnja seznama sklopov za izvajanje: seznam sklopov določijo opravila, ki so označena za izvajanje; proces zgradi edinstven seznam cifer sklopov (označenih) opravil iz nadzorne tabele (Job List) ter pripravi tabelo vhodnih parametrov zanke, kot jo prikazuje slika 5.3;

#	WORKFLOW	JOB_TYPE	LOAD_PERIOD	MASTER_START_EXECUTION_DTTM
1	1	EXTRACT ...	23MAR2014	12JUL2014:18:13:16
2	3	TRANSFORM ...	23MAR2014	12JUL2014:18:13:16
3	5	LOAD_ACRM ...	23MAR2014	12JUL2014:18:13:16

Slika 5.3: Tabela vhodnih parametrov zanke glavnega opravila

– obvestilo sprožitve procesa: v naslednjem koraku se pošlje e-pošta (slika 5.4) na naslove izbranim prejemnikom (shranjenih v eni od nadzornih tabel) z obvestilom o sprožitvi glavnega opravila; v e-pošti so našteti parametri, ki so dodeljeni opravilu s strani SAS strežnika, definirani parametri uporabnika ter seznam izbranih opravil za izvajanje;

SAS PROCESSING REPORT

```
Hostname:       arl047 (arl047)
Site:           70068130
Software:       9.04.01M0P061913
Environment:    X64_7PRO
Time:           12JUL2014 18:13
Batch ID:       7188
Process ID:     41D9A45C8E9CED914018000000000000
Process Name:   Object Server
User ID:        sasdemo

************************************************************
*** LVL0_MASTER job started PROCESSING REPORT
*** FLOW START TIME:  12JUL2014:20:56:23
*** LOAD PERIOD:      23MAR2014
************************************************************
* NUMBER OF JOBS SCHEDULED FOR EXECUTION:           7
************************************************************
```

JOB_ID	JOB_NAME	WORKFLOW	JOB_TYPE
1	C.B.01.01.OART_TXT	1	EXTRACT
2	C.B.01.02.SAS_CRMSI_MIARTICLE	1	EXTRACT
6	C.B.01.06.OECR_GR	1	EXTRACT
47	C.C.01.09.ECR_CATEGORY	3	TRANSFORM
50	C.C.01.12.D_PRODUCTS	3	TRANSFORM
73	C.E.01.09.ECR_CATEGORY	5	LOAD_ACRM
76	C.E.01.12.PRODUCTS	5	LOAD_ACRM

Slika 5.4: Elektronska pošta ob sprožitvi

– izvajanje zanke (nivo sklopov): telo zanke vsebuje vgnezdeno podporno opravilo, ki operira na nivoju valov; število ciklov zanke določa število zapisov tabele vhodnih parametrov; v vsakem ciklu zanka posreduje vrednosti parametrov (številko sklopa, datumsko obdobje) prvemu podpornemu opravilu ter ga sproži;

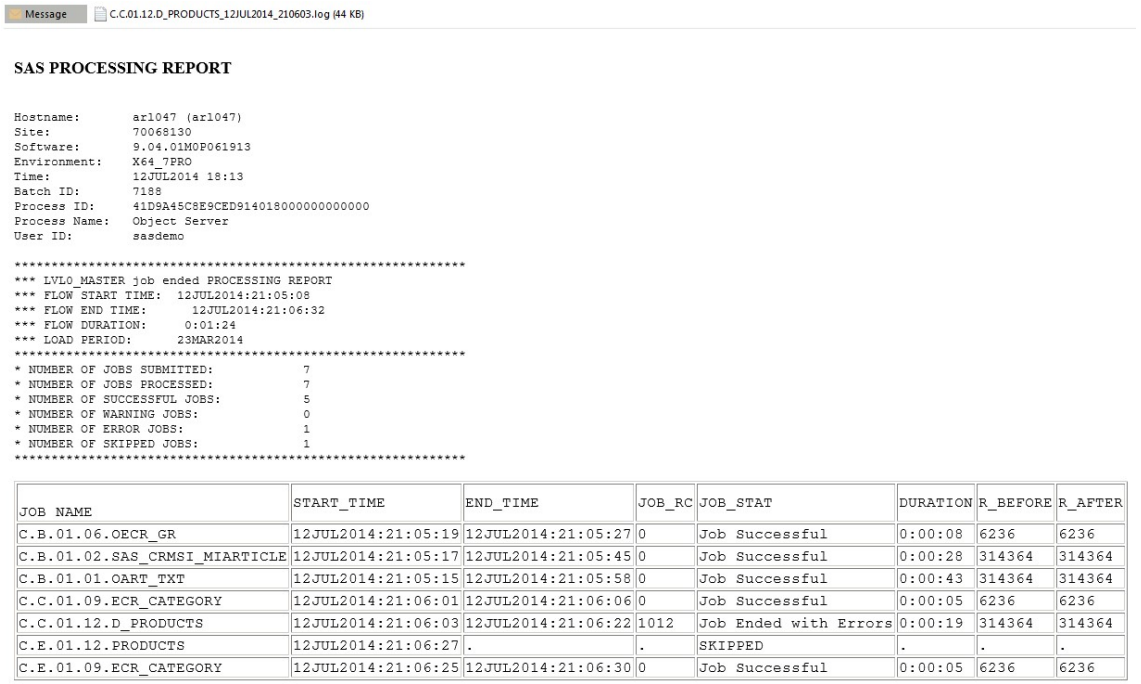
– analiza izvajanja: vsako opravilo za obdelavo podatkov ob zaključku zapiše rezultat svojega izvajanja v obliki statističnih informacij v nadzorno tabelo (angl. Job Monitoring); po zaključku zanke glavnega opravila se naredi povzetek rezultatov, ki se zapiše v nadzorno tabelo (angl. *Master Job Executions*) (slika 5.5);

#	LOAD_PERIOD	START_EXECUTION_DTTM	END_EXECUTION_DTTM
1	23MAR2014	12JUL2014:21:05:08	12JUL2014:21:06:31
2	23MAR2014	11JUL2014:16:00:15	11JUL2014:16:01:15
3	23MAR2014	11JUL2014:15:56:05	11JUL2014:15:57:24
4	23MAR2014	15APR2014:11:26:27	15APR2014:11:26:42

JOBS_FOR_EXECUTION	SUCCESSFUL	WARNINGS	ERRORS	SKIPPED	UNKNOWN_RUNNING
7	5	0	1	1	0
4	2	0	1	1	0
4	4	0	0	0	0
2	2	0	0	0	0

Slika 5.5: Povzetki izvajanja glavnih opravil

– obvestilo o zaključku procesa: v zadnjem koraku se pošlje e-pošta (slika 5.6) na naslove izbranih prejemnikov (shranjenih v eni od nadzornih tabel) z obvestilom o zaključku glavnega opravila; v e-pošti so našteti parametri, ki so dodeljeni opravilu s strani SAS strežnika, parametri izvajanja, ki jih je nastavil uporabnik, povzetek rezultatov izvedenih opravil, statistične informacije vsakega izvedenega opravila ter priponka z dnevniki neuspešno izvedenih opravil.



Message C.C.01.12.D_PRODUCTS_12JUL2014_210603.log (44 KB)

SAS PROCESSING REPORT

```
Hostname:       arl047 (arl047)
Site:           70068130
Software:       9.04.01M0P061913
Environment:    X64_7PRO
Time:           12JUL2014 18:13
Batch ID:       7188
Process ID:     41D9A45C8E9CED914018000000000000
Process Name:   Object Server
User ID:        sasdemo

*************************************************************
*** LVL0_MASTER job ended PROCESSING REPORT
*** FLOW START TIME:  12JUL2014:21:05:08
*** FLOW END TIME:      12JUL2014:21:06:32
*** FLOW DURATION:     0:01:24
*** LOAD PERIOD:      23MAR2014
*************************************************************
* NUMBER OF JOBS SUBMITTED:            7
* NUMBER OF JOBS PROCESSED:            7
* NUMBER OF SUCCESSFUL JOBS:           5
* NUMBER OF WARNING JOBS:              0
* NUMBER OF ERROR JOBS:                1
* NUMBER OF SKIPPED JOBS:              1
*************************************************************
```

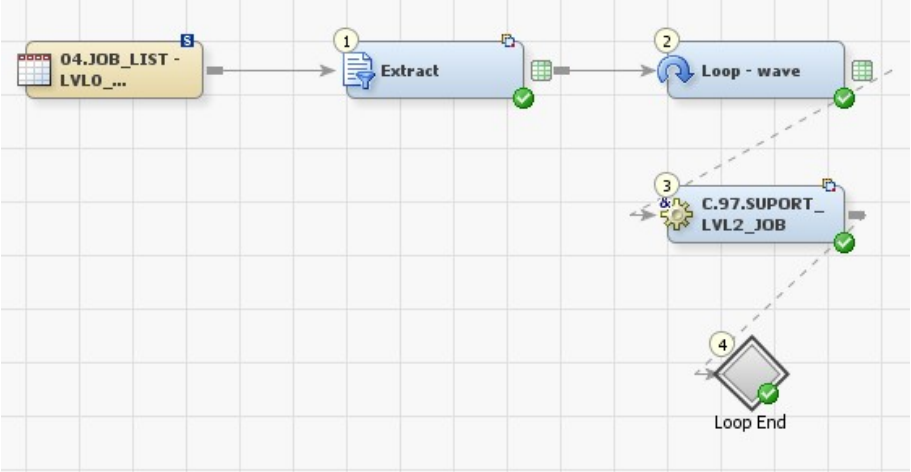
JOB_NAME	START_TIME	END_TIME	JOB_RC	JOB_STAT	DURATION	R_BEFORE	R_AFTER
C.B.01.06.OECR_GR	12JUL2014:21:05:19	12JUL2014:21:05:27	0	Job Successful	0:00:08	6236	6236
C.B.01.02.SAS_CRMSI_MIARTICLE	12JUL2014:21:05:17	12JUL2014:21:05:45	0	Job Successful	0:00:28	314364	314364
C.B.01.01.OART_TXT	12JUL2014:21:05:15	12JUL2014:21:05:58	0	Job Successful	0:00:43	314364	314364
C.C.01.09.ECR_CATEGORY	12JUL2014:21:06:01	12JUL2014:21:06:06	0	Job Successful	0:00:05	6236	6236
C.C.01.12.D_PRODUCTS	12JUL2014:21:06:03	12JUL2014:21:06:22	1012	Job Ended with Errors	0:00:19	314364	314364
C.E.01.12.PRODUCTS	12JUL2014:21:06:27	.	.	SKIPPED	.	.	.
C.E.01.09.ECR_CATEGORY	12JUL2014:21:06:25	12JUL2014:21:06:30	0	Job Successful	0:00:05	6236	6236

Slika 5.6: Elektronska pošta ob zaključku izvajanja

- Prvo podporno opravilo, katerege procesni diagram prikazuje slika 5.7
  Opravilo se sproži v zanki glavnega opravila. Pri tem so posredovane vrednosti parametrov za izvajanje, ki ga opisuje naslednje zaporedje dogodkov:

  – izgradnja seznama valov za izvajanje: posredovan parameter iz zanke glavnega opravila (številka sklopa) določi podskupino opravil v nadzorni tabeli (angl. Job List); iz nje se zgradi edinstven seznam cifer valov pripadajočemu sklopu opravil, ki so označena za izvajanje; iz seznama valov ter posredovanih parametrov se pripravi tabela vhodnih parametrov zanke (slika 5.8);

  – izvajanje zanke (nivo valov): telo zanke vsebuje vgnezdeno podporno opravilo, ki operira na nivoju opravil; število ciklov zanke določa število zapisov tabele vhodnih parametrov; v vsakem ciklu zanka posreduje vrednosti parametrov (številko sklopa, številko vala, datumsko obdobje) drugemu podpornemu opravilu ter ga sproži.



Slika 5.7: Procesni diagram prvega podpornega opravila

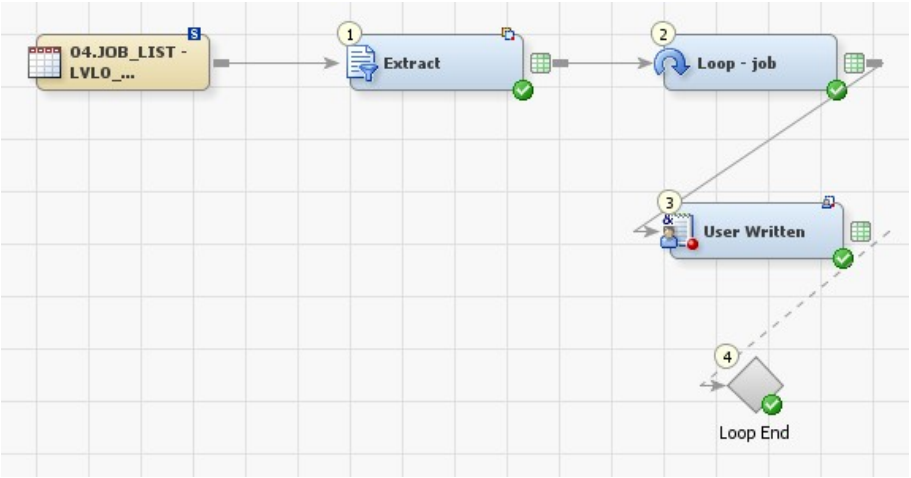
#	WORKFLOW_NUMBER	WAVE_NUMBER	LOAD_PERIOD	MASTER_START_EXECUTION_DTTM
1	1	1	23MAR2014	12JUL2014:20:57:02

Slika 5.8: Tabela vhodnih parametrov zanke prvega podpornega opravila

- Drugo podporno opravilo, katerege procesni diagram prikazuje slika 5.9 Opravilo se sproži v zanki prvega podpornega opravila. Pri tem so posredovane vrednosti parametrov za izvajanje, ki ga opisuje naslednje zaporedje dogodkov:

  – izgradnja seznama opravil za izvajanje: posredovana parametra iz zanke prvega podpornega opravila (številka sklopa, številka vala) določita podskupino opravil v nadzorni tabeli (angl. Job List); iz nje se zgradi seznam opravil, ki so označena za izvajanje; seznam izbranih opravil skupaj z vsemi posredovanimi parametri se zapiše v tabelo vhodnih parametrov zanke, kot prikazuje slika 5.10;

  – izvajanje zanke (nivo opravila): telo zanke vsebuje vgnezdeno transformacijo z uporabniško kodo za proženje opravil; število ciklov zanke določa število zapisov tabele vhodnih parametrov; v vsakem ciklu zanka posreduje vrednosti parametrov (številko sklopa, številko vala, ime opravila, datumsko obdobje) transformaciji ter jo izvede. Posebnost te zanke je, da transformaciji dodeli lastno sejo, v kateri se izvaja –



Slika 5.9: Procesni diagram drugega podpornega opravila

#	WORKFLOW	WAVE	JOB_NAME	LOAD_PERIOD	MASTER_START_EXECUTION_DTTM
1	1	1	C.B.01.01.OAR...	23MAR2014	13JUL2014:01:11:47
2	1	1	C.B.01.02.SAS_...	23MAR2014	13JUL2014:01:11:47
3	1	1	C.B.01.06.OEC...	23MAR2014	13JUL2014:01:11:47

Slika 5.10: Tabela vhodnih parametrov zanke drugega podpornega opravila

s tem je omogočeno sočasno izvajanje več transformacij (posledično opravil) hkrati.

Zanka se med ciklanjem ustavlja – na začetku preteče določeno število ciklov (sproži določeno število sej izvajanja) ter počaka, da se vse seje zaključijo. Nato steče dalje in postopek se ponavlja do konca vseh ciklov. Število sočasnih sej izvajanja določa eden od privzetih parametrov zanke, ki je prilagojen glede na zmožnosti strojne opreme.

– proženje opravil: opravila za obdelavo podatkov se izvajajo v sklopu vgnezdene transformacije zanke; posredovani parametri ji določijo ime opravila za izvajanje ter datumsko obdobje podatkov za obdelavo.

  Določene odvisnosti opravila (ko obstajajo) predstavljajo pogoj njegove sprožitve. Transformacija zbere rezultate izvajanja pogojenih opravil ter na njihovi podlagi izvede eno od naslednjih dveh akcij:

  * če so vsa pogojena opravila uspešno izvršena, transformacija opra-

vilo sproži,

* če je katero od pogojenih opravil neuspešno zaključilo izvajanje, transformacija opravila ne sproži ter v nadzorno tabelo (angl. Job Monitoring) zapiše, da je bilo izključeno iz izvajanja.

Poglavje 6

Primerjava z drugimi rešitvami

Na trgu obstaja veliko orodij in različnih proizvajalcev za razvoj procesov ETL ter samostojnih orodij oziroma modulov za njihov nadzor in avtomatizacijo. Slednja so ponavadi na voljo v paketih programske opreme, uporabljene za razvoj procesov.

Programska oprema sistema SAS pri tem ni izjema. Vendar v času nastanka ogrodja ETL standardni način avtomatizacije procesov ETL ni zagotavljal zadostnih povratnih informacij izvajanja. Pri tem so bila uporabljena samostojna orodja drugih proizvajalcev, kar je dodatno decentraliziralo nadzor in posledično podaljšalo čas pri odkrivanju in odpravljanju motenj.

Ogrodje ETL pa se ne smatra kot samostojno orodje oziroma modul za nadzor, kot to ponujajo današnje rešitve, ampak je integrirano v procese ETL med razvojem – tako se ne more primerjati s samostojnimi moduli. Kljub temu omogoča vse glavne funkcionalnosti avtomatizacije in nadzora izvajanja procesov, ki so primerljive z drugimi rešitvami [28, 29].

V tabeli 6.1 vidimo primerjavo glavnih karakteristik ogrodja ETL z drugimi rešitvami. Sistemi se po analiziranih kriterijih ne razlikujejo, razen pri možnosti razširitev procesov za izjeme. Dodatne prednosti ogrodja se kažejo pri ceni (za uporabo ogrodja ne potrebujemo dodatnih licenc) ter možnosti razširitve funkcionalnosti ogrodja ob predpostavki, da razvijalci/uporabniki razumejo njegovo delovanje ter posedujejo primerno znanje programiranja v programskem jeziku SAS Base. Pomanjkljivost ogrodja se kaže z načinom integracije/uporabe, kot ga

določajo pravila za implementacijo.

	Ogrodje ETL Data integration Studio	Informatica PowerCenter	Ascential Data Stage	Business Objects Data integrator
Definiranje procesnih tokov	Podprto	Podprto	Podprto	Podprto
Sočasno izvajanje procesov	Podprto	Podprto	Podprto	Podprto
Definicija odvisnosti med procesi	Podprto	Podprto	Podprto	Podprto
Nadaljevanje izvajanja od točke prekinitve	Podprto	Podprto	Podprto	Podprto
Samodejno obveščanje	Podprto	Podprto	Podprto	Podprto
Uporaba sistemskih časovnih razporejevalnikov	Omogočeno	Omogočeno	Omogočeno	Omogočeno
Razširitev procesov za izjeme	Omogočeno z dodatnim razvojem	Neznano	Neznano	Neznano

Tabela 6.1: Primerjava karakteristik ogrodja ETL z drugimi rešitvami

Poglavje 7

Uporaba s strani razvijalca

Končna implementacija ogrodja popolnoma avtomatizira proces podatkovne integracije, kar tudi zmanjša potrebo po uporabniški interakciji. Uporabnik (predvidoma) izvaja le splošen pregled rezultatov izvršenih glavnih opravil, ki jih prejme v obliki e-pošte, ter morebitna dodatna preverjanja statistik podatkovnih prenosov med sistemi, zabeleženih v nadzornih tabelah.

Glavne značilnosti ogrodja uporabniku omogočajo:

- standardiziran način razvoja opravil (ETL procesov),
- definicijo poteka celotnega procesa z enega mesta,
- definicijo medsebojnih odvisnosti opravil,
- možnost nadgradnje procesne logike opravil za izjeme,
- avtomatiziran (centraliziran) nadzor ter potek izvajanja,
- prilagoditev izvajanja v teku glede na rezultate opravil,
- arhiviranje rezultatov izvajanja ter samodejno obveščanje,
- samodejno proženje procesa ob zaznavanju posodobitev podatkovnih izvorov,
- možnost nadaljevanja izvajanja procesa od točke prekinitve.

Potreba po uporabniškem poseganju se pojavi v primerih, kot so:

- nepredvidene prekinitve glavnih opravil: izpad omrežja ali strežnikov, nedostopnost izvornih sistemov, nenačrtovane spremembe podatkovnih struktur ipd.;
- neuspešnih zaključnih opravil;
- nezaznanih posodobitev podatkov izvornih sistemov, ki se zgodijo izven nadziranega datumskega obdobja;
- potrebe po spremembah procesne logike opravil za obdelavo podatkov.

V praksi se izkaže, da uporabnik najpogosteje posreduje v primeru prekinitve izvajanja glavnega opravila ter ko neko opravilo za obdelavo podatkov zaključi izvajanje z napakami.

V primeru prekinitve glavnega opravila med izvajanjem uporabniku ni posredovano obvestilo o zaključku izvajanja – neznano stanje. Za reševanje tega primera sledimo naslednjemu postopku:

- preverjanje stanja glavnega procesa: začetno obvestilo vsebuje identifikacijsko številko procesa, dodeljeno glavnemu opravilu s strani operacijskega sistema – preveri se, če proces še obstaja;
- ponovno proženje oziroma nadaljevanje izvajanja: uporabnik se v tem koraku odloči, ali želi nadaljevati izvajanje od točke prekinitve ali ponovno sproži celoten proces:
  - nadaljevanje izvajanja: v nadzorni tabeli (angl. Job Monitoring) uporabnik preveri, koliko opravil s seznama začetnega obvestila se je uspešno izvršilo pred prekinitvijo – nato glavnemu opravilu (za ročno proženje) ustrezno nastavi parametre (določi zbirko neizvedenih opravil in datumsko obdobje) ter ga sproži;
  - ponovno proženje: začetno obvestilo vsebuje vrednosti parametrov prekinjenega glavnega opravila – uporabnik nastavi iste parametre ter ročno sproži glavno opravilo.

V drugem primeru, če se je neko opravilo za obdelavo podatkov neuspešno zaključilo z izvajanjem (ali je bilo posledično izključeno iz izvajanja s strani glavnega opravila), lahko uporabnik na podlagi končnega obvestila hitro določi, kje in zakaj je prišlo do napake. Za reševanje tega primera sledimo naslednjemu postopku:

- opredelitev in odprava napake: obvestilo uporabniku prikaže rezultat celotnega izvajanja ter posreduje dnevnike neuspešno zaključenih opravil – iz dnevnikov je hitro razvidno, do kakšne napake, ki jo mora uporabnik odpraviti, je prišlo;
- ponovno proženje: uporabnik glavnemu opravilu (za ročno proženje) ustrezno nastavi parametre (določi zbirko opravil za ponovno izvajanje in datumsko obdobje) ter ga sproži.

Poglavje 8

Sklepne ugotovitve

Avtomatizacija postopka podatkovne integracije je podprta z ogrodjem ETL, ki se ga danes uporablja kot standardni postopek pri implementacijah. V praksi se je izkazalo, da je funkcionalnost ogrodja primerljiva z drugimi obstoječimi rešitvami ter se od njih razlikuje po načinu uporabe in ne predstavlja samostojnega modula ali orodja programske opreme kot pri drugih rešitvah in s tem ne vpliva na ceno njihovih licenc – cenovno ugodnejša rešitev.

Z nadgradnjo se lahko ogrodje prilagodi z dodatno funkcionalnostjo glede na želje strank, s čimer povečamo njegovo fleksibilnost. Tako se implementacije med seboj do določene mere razlikujejo, vse pa sledijo istemu načinu razvoja in vsaka je podprta z osnovnimi karakteristikami ogrodja.

V zaključku lahko povzamemo, da ogrodje predstavlja preprost način za reševanje problematike avtomatizacije procesov. Uporabnikom je popolnoma transparentno ter ponuja centraliziran nadzor nad izvajanjem procesov in proženja celotnega postopka.

Slike

Literatura

[1] B. Inmon (1998) „Data Mart Does Not Equal Data Warehouse”,
Dostopno na: http://www.information-management.com/infodirect/19991120/1675-1.html

[2] GoliInfo „ETL (Extract-Transform-Load) ”,
Dostopno na: http://www.dataintegration.info/etl

[3] GoliInfo „Data Integration”,
Dostopno na: http://www.dataintegration.info/data-integration

[4] J. Ulrych (2011) „Introduction to ETL and Data Integration”,
Dostopno na: http://www.slideshare.net/cloveretl/introduction-to-etl-and-data-integration

[5] J. Serra (2011) „Methods for populating a data warehouse”,
Dostopno na: http://www.jamesserra.com/archive/2011/08/methods-for-populating-a-data-warehouse/

[6] RSRIT (2014) „Automatically detect data errors and inconsistencies through ETL Tools”,
Dostopno na: http://rsrit.com/blog/2014/08/10/automatically-detect-data-errors-inconsistencies-etl-tools/

[7] Wikipedia „Data Mart”,
Dostopno na: http://www.wikipedia.org/wiki/Data_mart

[8] Wikipedia „Dimension Table”,
Dostopno na: http://en.wikipedia.org/wiki/Dimension_table

[9] Wikipedia „Fact Table”,
Dostopno na: http://en.wikipedia.org/wiki/Fact_table

[10] Wikipedia „SAS software”,
Dostopno na: http://en.wikipedia.org/wiki/SAS_software

[11] D, Mauri „Temporal Snapshot Fact Table”,
Dostopno na: http://www.slideshare.net/davidemauri/temporal-snapshot-fact-tables

[12] Nenit „Understanding Star and Snowflake Schemas”,
Dostopno na: http://www.nenit.net/pages/schemas.aspx

[13] R. Kimball, M. Ross (2013) „The Data Warehouse Toolkit, 3rd Edition”,
The Definitive Guide to Dimensional Modeling.

[14] M. Schneider, D. Bennett, C. Robison (2011) „Understanding the Anatomy of a SAS Deployment”,
What's in My Server Soup?
Dostopno na: http://support.sas.com/resources/papers/proceedings11/363-2011.pdf

[15] S. Sayers (2008) „From Tiers to Intelligence:”,
The SAS Enterprise Intelligence Platform and What It's All About
Dostopno na: http://www2.sas.com/proceedings/forum2008/051-2008.pdf

[16] SAS Institute „SAS Base”,
http://support.sas.com/software/products/base/

[17] SAS Institute „SAS 9.2 Language Reference: Concepts, Second Edition”,
Dostopno na: http://support.sas.com/documentation/cdl/en/lrcon/62955/HTML/default/vie

[18] SAS Institute „Basic and Intermediate SAS”,
Dostopno na: http://web.utk.edu/sas/OnlineTutor/1.2/en/60476/m1/m1_33.htm

[19] SAS Institute „SAS 9.1 Macro Language”,
dostopno na: http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_91/base_macro_

[20] SAS Institute „SAS Data Integration Studio”,
Dostopno na: http://support.sas.com/software/products/etls/

[21] SAS Institute „SAS Data Integration Studio 4.8 User's GuideSAS",
Dostopno na: http://support.sas.com/documentation/cdl/en/etlug/67323/PDF/default/etlug.pdf

[22] SAS Institute (2009) „SAS Data Integration Studio Fast Track",
SAS Training Course Notes

[23] SAS Institute (2009) „SAS Programming 1 Essentials",
SAS Training Course Notes

[24] SAS Institute (2006) „ETL Performance Tuning Tips",
Dostopno na: http://support.sas.com/resources/papers/ETLperformance07.pdf

[25] E. Hunley (2002) „SAS Data Quality – A Technology Overview",
Dostopno na: http://www2.sas.com/proceedings/sugi29/09929.pdf

[26] UCLA: Statistical Consulting Group (2007) „Introduction to SAS",
Dostopno na: http://www.ats.ucla.edu/stat/sas/library/SASLang_os.htm

[27] UCLA: Statistical Consulting Group (2007) „SAS Macros Introduction",
Dostopno na: http://www.ats.ucla.edu/stat/sas/seminars/sas_macros_introduction/

[28] S. Sharma „ETL Tool Comparison",
Dostopno na: dwhnotes.com//wp-content/uploads/2009/04/etl_tool-comparision.xls

[29] S. Sharma „List Of ETL Tools",
Dostopno na: http://dwhnotes.com/data-integrator/list-of-etl-tools