STUDIA INFORMATICA 2010
Volume 31 Number 3 (91)

Grzegorz NOWAK
Politechnika Łódzka, Katedra Informatyki Stosowanej

GRAFICZNA ANALIZA PROGRAMÓW SAS

Streszczenie. Artykuł opisuje propozycję narzędzia ułatwiającego analizę programów tworzonych w systemie SAS (4GL). W artykule zaproponowano wykorzystanie systemowego logu, generowanego przez SAS dla każdego wykonywanego programu, w celu automatycznego tworzenia odpowiadającego mu diagramu przetwarzania. Tworzone w postaci wektorowej diagramy mogą być wykorzystane w dokumentacji i diagnostyce programów SAS.

Słowa kluczowe: SAS 4GL, analiza logu SAS, reprezentacja graficzna

GRAPHICAL ANALYSIS OF SAS PROGRAMS

Summary. The article describes an analysis tool for SAS 4GL programs. The paper proposes the use of the system log, generated by SAS for each executed program to automatically create a corresponding diagram of the processing. Created in the scalable form diagrams can be used for documentation and diagnostic purposes.

Keywords: SAS 4GL, log SAS analysis, graphical representation

1. Wstęp

Rozbudowane przetwarzania realizowane w języku programowania systemu SAS [1], mające na celu przygotowanie danych do przeprowadzania analiz i raportowania, zawierają dużą ilość tworzonych lub wykorzystywanych tablic i plików. Często liczba tych elementów jest na tyle duża, że pełna kontrola nad procesem staje się od pewnego momentu trudna i niezbędne staje się użycie narzędzia wizualizującego proces mapowania pomiędzy źródłem danych a ich docelową postacią. Naturalnie, dokumentacja techniczna przetwarzania jest bardzo pomocna i zwykle niezbędna. Jednak zwykle koszty jej wytworzenia (czas, kompetencje, systematyczność) powodują, że jest ona zaniedbywana, a często w ogóle nie istnieje.

Nawet dysponując dokumentacją techniczną, problem czytelności kodu jest dużym wyzwaniem, np. w przypadku skomplikowanych przetwarzań lub modyfikacji programów napisanych dużo wcześniej, czy w przypadku koniecznej ingerencji w kod napisany przez kogoś innego, np. w procesach ETL zasilania hurtowni danych oprogramowanych w systemie SAS, co podkreślono w [2],

Analiza kodu źródłowego i dokumentacji technicznej wymaga sporego nakładu sił, kompetencji, a przede wszystkim czasu. Niniejsza propozycja ma na celu automatyczne dostarczenie odpowiedniej informacji w takiej formie, aby ułatwić i przyśpieszyć analizę tworzonych lub istniejących programów. Przeprowadzana jest ona na podstawie logu wykonywanego programu SAS i może być uzupełniona analizą kodu źródłowego. Log jest źródłem wielu bardzo cennych informacji na temat wykonywanego czy wykonanego wcześniej programu, zwłaszcza w przypadku uruchomienia programu z odpowiednimi opcjami, zwiększającymi zawartość informacji zapisywanej do logu (np. MPRINT, MLOGIC, SYSLOG czy FULLSTIMER). Informacje z logu są analizowane i przetwarzane do postaci graficznej, bardziej przyjaznej i ułatwiającej interpretację programu.

Otrzymany model graficzny zawiera m. in. diagram przepływu danych, uwzględniający szczegółowy rysunek plików i tablic tworzonych i/lub wykorzystywanych w czasie przetwarzania oraz hierarchiczność wywołań makr, a także informację czasową dot. długości przetwarzania w poszczególnych etapach, liczbę zmiennych i obserwacji w tabelach, długość rekordów i ilość linii w plikach. Ponadto, model graficzny może być uzupełniony informacją tekstową, zawierającą:

- pełną inwentaryzację elementów składowych przetwarzania w postaci tablicy/listingu używanych plików, tablic, makrozmiennych, bibliotek;
- elementy tekstowej dokumentacji, o ile wprowadzono w odpowiedni sposób przy komentowaniu kodu źródłowego.

2. Podstawowe elementy przetwarzania i ich reprezentacja graficzna

Analiza logu oparta jest na założeniu, że podstawowymi elementami składowymi przetwarzania SAS są pliki i tabele (wszelkie raporty np. _webout czy html traktowane są jako pliki). W przypadku plików istotnymi dla celów diagnostycznych cechami są: ścieżka dostępu, nazwa, minimalna i maksymalna długość rekordów, ilość linii, wielkość pliku. Dla tabel są to odpowiednio: biblioteka (nazwa referencyjna i fizyczna ścieżka do katalogu), nazwa, liczba obserwacji (wierszy) oraz zmiennych (kolumn).

Ze względu na sposób dostępu pliki i tablice podzielono na:

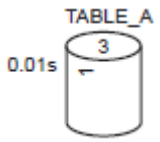
- elementy źródłowe, dostarczają danych dla innych elementów (plików/tabel) i nie są tworzone w ramach rozpatrywanego przetwarzania.

- elementy docelowe, są tworzone z innych elementów w procesie przetwarzania.
- elementy przejściowe, są tworzone i jednocześnie stanowią źródło informacji dla innych elementów.

Przyjęto graficzne reprezentacje dla tablic i plików. Elementy te rozmieszczane są w węzłach diagramu odpowiednio do miejsca wystąpienia w procesie przetwarzania.

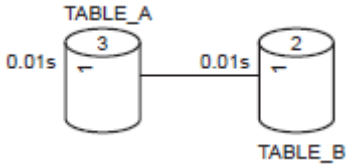
2.1. Tablice SAS

Tablice przedstawiane są na diagramie w formie jak na rys. 1, zamieszczonym poniżej, wraz z ilością linii/obserwacji oraz ilością zmiennych.

Przykładowy kod źródłowy SAS	Log odpowiadający uruchomionemu kodowi.
data table_a; z=**1**; y=**2**; x=**3**; **run**;	46818 data table_a; 46819 z=1; 46820 y=2; 46821 x=3; 46822 run; NOTE: The data set WORK.TABLE_A has 1 observations and 3 variables. NOTE: DATA statement used (Total process time): real time 0.03 seconds cpu time 0.01 seconds
Reprezentacja graficzna	
	

Rys. 1. Tworzenie tablicy w SAS: kod, log i reprezentacja graficzna
Fig. 1. Table creation In SAS: code, log and graphical representation

Tablice będące źródłem danych dla innych tablic pozostają z nimi w relacji zależności reprezentowanych liniami. Dla przejrzystości diagramu linie nie są zakończone strzałkami, jednak należy je traktować kierunkowo: tzn. element na lewym krańcu linii jest elementem źródłowym, a element na prawym krańcu linii jest elementem docelowym (rys. 2). I tak: elementy stanowiące źródło informacji dla innych elementów posiadać będą tylko linie wychodzące. Podobnie linie przychodzące będą posiadały jedynie te elementy, które są tworzone w ramach przetwarzania z danych pochodzących z innych tabel/plików. Tablice nieposiadające jako źródła danych innej tablicy lub pliku są tworzone z wirtualnych danych, np. makrozmiennych lub danych typu „cards", umieszczonych w kodzie.

Przykładowy kod źródłowy SAS	Log odpowiadający uruchomionemu kodowi.
data table_a; z=**1**; y=**2**; x=**3**; **run**; **data** table_b; set table_a; keep x y; **run**;	46818 data table_a; 46819 z=1; 46820 y=2; 46821 x=3; 46822 run; NOTE: The data set WORK.TABLE_A has 1 observations and 3 variables. NOTE: DATA statement used (Total process time): real time 0.03 seconds cpu time 0.01 seconds 46823 46824 data table_b; 46825 set table_a; 46826 keep x y; 46827 run; NOTE: There were 1 observations read from the data set WORK.TABLE_A. NOTE: The data set WORK.TABLE_B has 1 observations and 2 variables. NOTE: DATA statement used (Total process time): real time 0.02 seconds cpu time 0.01 seconds
Reprezentacja graficzna	
	

Rys. 2. Tablice źródłowa i docelowa: kod, log i reprezentacja graficzna
Fig. 2. Source and target tables: code, log and graphical representation

Elementy przejściowe, tworzone z danych pochodzących z innych elementów i jednocześnie będące źródłem informacji dla innych elementów, zawierać będą zarówno linie dochodzące, jak i odchodzące.

2.2. Pliki

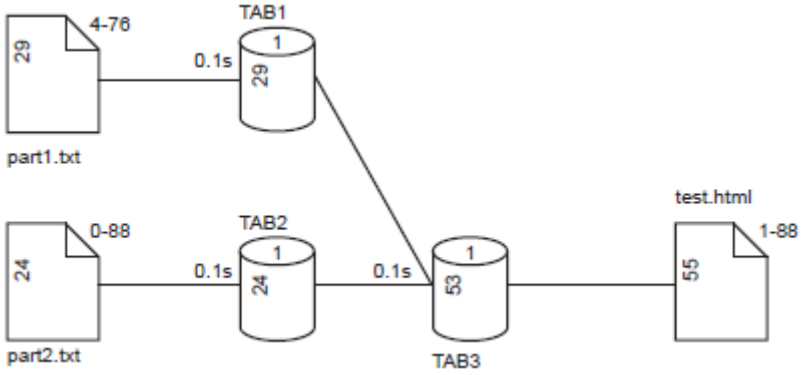
Podobnie jak w przypadku tabel, pliki posiadają swoją reprezentację graficzną, przedstawioną na diagramie poniżej wraz z podaniem ilości linii oraz długości rekordów. Zamieszczony poniżej diagram (rys. 3) zawiera pliki i tabele. Pokazano przykładowy kod i odpowiadające mu elementy graficzne.

Kod źródłowy SAS	Log

```
FILENAME indata1 "C:/tmp/part1.txt";
FILENAME indata2 "C:/tmp/part2.txt";
FILENAME htmlfile "C:/tmp/test.html";
data tab1;
       length linia $500;
       infile indata1 linesize=32767 lrecl=500 mis-
   sover end=koniec;
       input ;
       linia=_infile_;
run;
data tab2;
       length linia $500;
       infile indata2 linesize=32767 lrecl=500 mis-
   sover end=koniec;
       input ;
       linia=_infile_;
run;
data tab3; set tab1 tab2; run;
data _null_;
 file htmlfile linesize=32767 lrecl=500 ;
set tab3 end=koniec;
if _n_=1 then put '<html><body>';
put linia;
if koniec then put '</body></html>';
run;
```

Prezentację logu pominięto ze względu na jego długość.

Reprezentacja graficzna



Rys. 3. Tworzenie pliku w SAS: kod, log i reprezentacja graficzna
Fig. 3. File creation In SAS: code, log and graphical representation

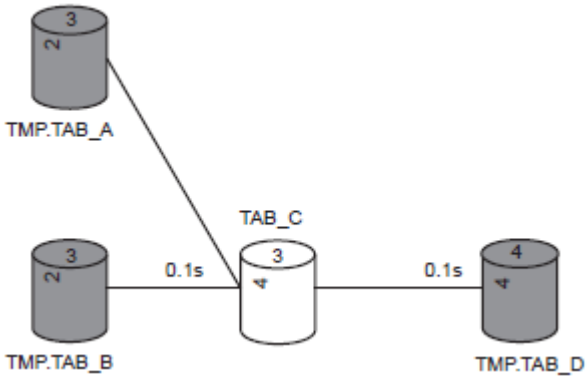
Rysunek 3 przedstawia przetwarzanie, w którym informacje pochodzące z plików part1.txt i part2.txt zapisywane są do dwóch tabel: odpowiednio tab1 i tab2, które są następnie łączone w tablicę tab3 i na podstawie tej ostatniej generowany jest plik test.html.

Na diagramie przedstawione są podstawowe informacje dotyczące tabel i plików: nazwa, liczba zmiennych tabeli lub minimalną i maksymalną długość rekordów dla plików, liczba obserwacji lub liczba linii dla plików oraz czas przetwarzania potrzebny do utworzenia tabeli lub pliku. Precyzję wyświetlania czasu można dowolnie regulować, domyślnie przyjęto ją na poziomie 0,1s (wówczas wszystkie wartości mniejsze od tego progu przedstawiane są jako równe zero). Dla czytelności rysunku nazwy tabel w jednym poziomie diagramu umieszczane są na przemian: pod tabelą i nad tabelą. Takie rozmieszczenie nazw jest ważne, zwłaszcza

w przypadku długich nazw tabel, które umieszczone obok siebie mogłyby się nakładać na rysunku.

2.3. Bibioteki (work/libname)

W przetwarzaniu SAS tablice mogą być tworzone w bibliotece tymczasowej (WORK) bądź też w bibliotece zadeklarowanej, posiadającej nazwę i lokalizację na dysku. Oba rodzaje tablic mają różną reprezentację graficzną: przyjęto, że tablice w WORK są przedstawiane jako przezroczyste (mniej ważne), a tablice w bibliotekach stałych (ang. *libnames*) przedstawiane są w postaci zacieniowanej (zwykle są one istotniejsze, gdyż zachowywane są na dysku, po zakończeniu sesji SAS). Również ich nazwy są przedstawiane odmiennie: tablice utworzone w bibliotece tymczasowej WORK posiadają jednoczłonową nazwę, a tablice tworzone w stałej, zadeklarowanej bibliotece posiadają dwuczłonową nazwę w postaci: *libname.name*.

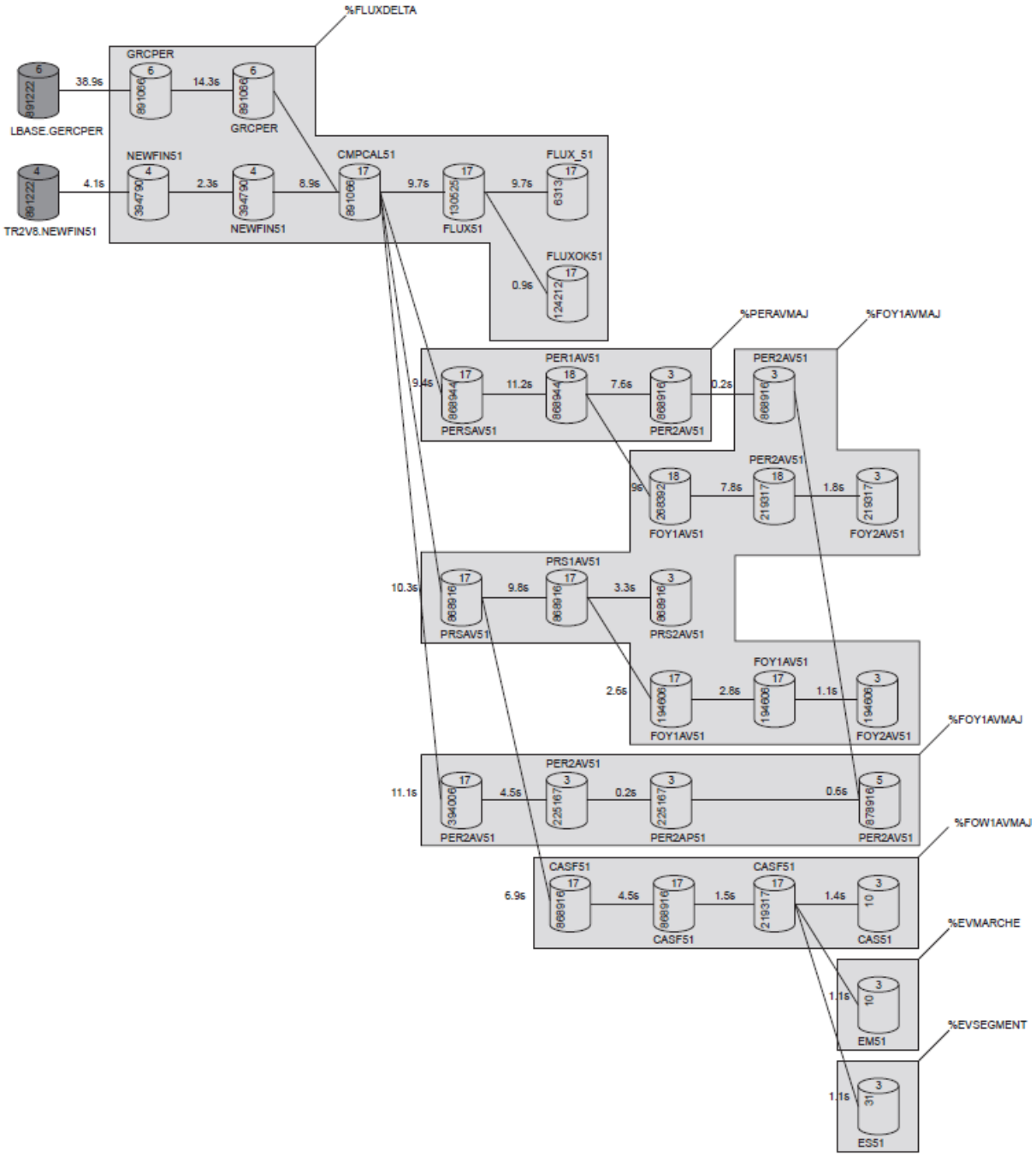


Rys. 4. Diagram z tablicami w bibliotekach TMP i WORK
Fig. 4. Diagram with tables in libraries TMP and WORK

Powyższy diagram (rys. 4) przedstawia dwie wejściowe tablice: tab_a oraz tab_b, znajdujące się w libname TMP, które są źródłem informacji dla tablicy tab_c, tworzonej w bibliotece tymczasowej WORK oraz wyjściową tablicę tab_d, również utworzoną w bibliotece TMP, na podstawie danych pochodzących z tab_c. Tablica tab_c jest w tym przypadku tablicą przejściową.

3. Makra w programach SAS

Jeżeli przetwarzanie SAS'owe zawiera makra, mogą być one również pokazane na diagramie, co pokazano na rys. 5.



Rys. 5. Diagram przykładowego przetwarzania zawierającego makra
Fig. 5. Diagram of an example program with macros

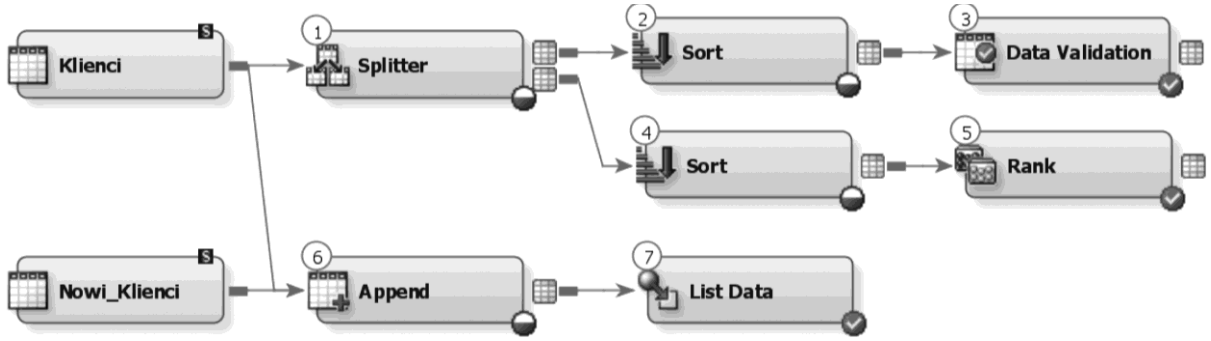
Wszystkie elementy tworzone w ramach danego makro umieszczane są na wspólnej, zacieniowanej powierzchni z podaniem nazwy makro. Odpowiedni algorytm rozmieszczania elementów zapewnia, że powierzchnie makr są spójne i rozłączne (jedno makro nie może składać się z rozłącznych powierzchni) i nie nakładają się na siebie (dotyczy to makr na tym samym poziomie zagnieżdżenia). Jeżeli makro wykonywane jest kilka razy, na diagramie może wystąpić wielokrotnie. W przypadku makr zagnieżdżonych możliwe jest pokazanie wielu poziomów zagnieżdżenia. Jeżeli wybrano opcję pokazywania kilku poziomów makr zagnieżdżonych, w przeciwieństwie do pierwszego poziomu makr, kolejne są pokazywane na

diagramie w postaci zamkniętych linii przerywanych (bez wypełnienia). Pogarsza to jednak czytelność diagramu i dlatego ta opcja jest domyślnie wyłączona.

4. Porównanie do SAS DI oraz SAS Enterprise Guide

Spośród dostępnych narzędzi SAS, dwoma które pozwalają na reprezentację graficzną przetwarzania są „SAS Data Integration" oraz „SAS Enterprise Guide".

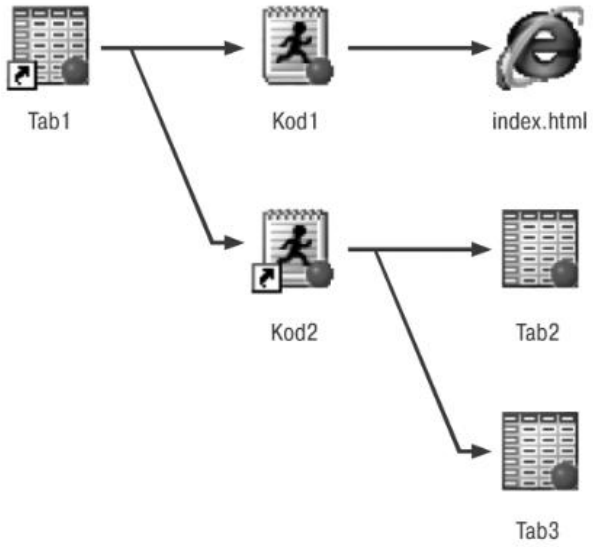
SAS Data Integration [2], [3], to pakiet rozwiązań, zapewniający pełne spektrum potrzeb w zakresie integracji danych. Jednym z elementów pakietu jest interfejs graficzny dla projektantów procesów przetwarzania danych, którego zadaniem jest usprawnienie implementacji procesu integracji danych.



Rys. 6. Diagram przetwarzania w SAS Data Studio
Fig. 6. Diagram of an example program in SAS DI Studio

Interfejs graficzny SAS DI pozwala na zaprogramowanie przetwarzania z wykorzystaniem standardowych, najczęściej używanych elementów składowych. Graficzna reprezentacja przetwarzań, którego przykład przedstawiono na rys. 6, pomaga w zrozumieniu jego przebiegu i daje również możliwość oceny statusu przetwarzania na bieżąco w trakcie jego wykonywania. W odniesieniu do zaproponowanego w niniejszym artykule rozwiązania (kod → log → diagram), SAS DI umożliwia generowanie kodu SAS na podstawie rysowanego graficznie schematu przetwarzania (diagram → kod). Jest to zatem podejście odwrotne, mimo wielu zbieżności w zakresie graficznego modelowania przetwarzań.

Z kolei SAS Enterprise Guide to środowisko wraz z zestawem narzędzi ułatwiającym pracę nad projektami obejmującymi zakres od integracji danych po tworzenie elastycznych analiz i raportów [4]. Jednym z elementów Enterprise Guide jest „Project Designer", który pozwala na graficzną interpretację przetwarzania, na wizualizację organizacji: przetwarzania, kodu oraz danych powiązanych z projektem.



Rys. 7. Diagram przykładowego przetwarzania w SAS Enterprise Guide
Fig. 7. Diagram of an example program in SAS Enterprise Guide

SAS Enterprise Guide pozwala – podobnie jak zaproponowane w niniejszym artykule rozwiązanie – na tworzenie graficznego diagramu przetwarzania na podstawie kodu programu. Jednak logika diagramu i sposób reprezentacji przetwarzania są zupełnie odmienne. Interpretacja graficzna SAS Enterprise Guide zakłada przedstawienie innych informacji, niż proponowane narzędzie i zorientowana jest na ułatwienie zrozumienia przetwarzania SAS dla analityków niebędących programistami. Istotna różnica polega tu na wykorzystaniu w diagramie elementów różnorodnej natury, gdzie każdy typ pliku ma swoją indywidualną reprezentację graficzną i gdzie kod programu może być składnikiem diagramu, co przedstawiono na rys. 7. Natomiast zaproponowane w artykule podejście ukierunkowane jest na programistów i diagnostykę skomplikowanych przetwarzań. Diagramy składają się w tym przypadku wyłącznie z tabel i plików oraz łączących je relacji, bez graficznego rozróżniania typów plików.

5. Podsumowanie

Zaproponowane narzędzie pozwala na analizę dowolnego logu SAS Base/Macro (nawet niepełnego, np. zakończonego błędem) bądź też wybranej części logu. W analizie logu automatycznie rozpoznawane i pomijane są komentarze (kod SAS umieszczony w komentarzach nie jest brany pod uwagę). Ponadto, narzędzie uwzględnia hierarchię wywołań makr i zagnieżdżonych programów. Diagramy tego typu mogą być wykorzystane w takich zakresach zastosowań, jak:

- diagnostyka programów SAS BASE/MACRO;
- dokumentacja techniczna;
- graficzna reprezentacja przetwarzania (przepływ danych);
- tworzenie mapy zależności *(impact analysis)*;
- diagnostyka błędów;
- śledzenie poprawności ilości obserwacji lub zmiennych.

Diagramy tworzone są w postaci wektorowej (postscript), co umożliwia dowolne skalowanie rysunku bez utraty jego jakości. Systematyczne tworzenie diagramów z wykonywanych programów oraz ich analiza przez programistę wpływa również korzystnie na sam styl programowania, który staje się bardziej usystematyzowany, spójny i czytelny.

BIBLIOGRAFIA

1. Dec Z.: Wprowadzenie do systemu SAS. Wydawnictwo Edition 2000, Kraków 1997.
2. Grasse, D., Nelson, G.: Base SAS vs. Data Integration Studio: Understanding ETL and the SAS tools used to support it. SAS Users Group International Conference, San Francisco 2006.
3. Olinger C., Weeks T.: The Ins and Outs of SAS Data Integration Studio. SAS Global Forum Proceedings, Orlando 2007.
4. Fecht M., Dhillon R.: A SAS Programmer's Guide to SAS Enterprise Guide. SAS Global Forum Proceedings, Orlando 2007.
5. Slaughter S., Delwiche L.: Writing Code in SAS Enterprise Guide. Western Users of SAS Software, Annual Conference Proceedings, Universal City, 2008.

Recenzent: Dr inż. Henryk Josiński

Wpłynęło do Redakcji 6 lipca 2010 r.

Abstract

The article describes an analysis tool for SAS 4GL programs (Base/Macro). The paper proposes the use of the system log, generated by SAS for each executed program to automatically create a corresponding diagram of the processing. Graphical model contains data flow diagram with files and tables used and created within the process. It takes into consideration

the hierarchy of macro procedures. The information about creation time of each element (file/table) is included one the diagram. Diagram contains also a number of variables and observations for tables and respectively a record length and line number for files. Diagrams created in the scalable form can be used for documentation and diagnostic purposes.

Adres

Grzegorz NOWAK: Politechnika Łódzka, Katedra Informatyki Stosowanej,
ul. Stefanowskiego 18/22, 90-924 Łódź, Polska, g.nowak@kis.p.lodz.pl.