

11 1979

informatyka

Język symulacyjny dla

W Instytucie Informatyki Uniwersytetu Jagiellońskiego opracowany został język GODYS-5, przeznaczony do symulacji komputerowej dynamicznych systemów ciągłych ze zdarzeniami dyskretnymi, oraz jego procesor dla komputerów serii ODRA 1300.

Język GODYS-5 nadaje się do symulacji systemów zawierających zarówno elementy ciągłe (takie jak integratory), jak też zmieniające skokowo stan (przerzutniki, komutatory, detektory przekroczeń alarmowych). Podstawą sporządzania programu w tym języku jest zapis modelu matematycznego badanego obiektu w postaci układu równań różniczkowo-logicznych, schematu blokowego lub grafu funkcyjnego. Zasadniczą częścią programu jest opis tego modelu, odwzorowywany w procesie translacji w model cyfrowy. Na modelu można przeprowadzać eksperymenty symulacyjne, związane na przykład ze zmianą wartości parametrów systemu lub symulatora, których wyniki podbudowane odpowiednią teorią pozwalają wnioskować o własnościach obiektu. Wyniki symulacji mogą być prezentowane w postaci przejrzystych wydawnictw (tabel i wykresów sporządzanych na drukarce wierszowej), mogą też być przetwarzane (na przykład przez programy analizy widmowej, statystycznej i inne).

Podjęcie prac nad stworzeniem języka GODYS-5 było wynikiem zapotrzebowania szeregu zespołów badawczych na język symulacyjny o wspomnianym przeznaczeniu. W pracach tych nie miały znaczenia miały doświadczenia autorów uzyskane w trakcie realizacji prostych języków symulacyjnych bazujących na grafach funkcyjnych (GODYS-1, 2, 3 [1, 3]), a w szczególności w trakcie realizacji wzorcowego języka symulacyjnego GODYS-4 [4].

Przystępując do projektu języka GODYS-5 przyjęto następujące założenia:

- język powinien być prosty i zrozumiały, jednocześnie powinien mieć duże możliwości wyrażania pojęć w zakresie przewidywanych zastosowań
- język i jego procesor powinny dać się łatwo modyfikować
- proces translacji powinien być oparty na metodach formalnych
- procesor języka powinien efektywnie pracować w typowej konfiguracji systemu ODRA 1300 z PAO 32 K słów i z pamięciami taśmowymi lub dyskowymi.

Konsekwencją kilkumiesięcznej realizacji projektu było opracowanie języka o strukturze frazowej, łatwego do opanowania i prostego w użytkowaniu, oraz napisanie jego procesora w formie dwóch programów (translator i system wykonawczy). Programy te zostały napisane strukturalnie w językach FORTRAN IV (95%) i PLAN (5%). Pozwala to na stosunkowo łatwą ich modyfikację w niewielkim tylko stopniu zmniejszając szybkość realizacji eksperymentu symulacyjnego.

Program w języku GODYS-5 składa się z dwóch części [5]: 1) opisu modelu symulowanego systemu i 2) sekwencji dyrektyw sterujących eksperymentami przeprowadzanymi na modelu.

W procesie translacji opis modelu jest przetwarzany na równoważny program w języku maszyny abstrakcyjnej SIM/AGH (zdefiniowanej na potrzeby projektu). Maszyna ta implementowana jest przez interpreter, którego pracą sterują wspomniane dyrektywy. Generowane w trakcie eksperymentu symulacyjnego wyniki gromadzone są w pamięci masowej, a następnie opracowane w sposób określony dyrektywami wyjścia (PRINT, PRPLOT, GRAPH, PLOTXY).

Opis modelu składa się z:

- sekcji deklaracji
- właściwego opisu modelu.

W sekcji deklaracji deklarowane są parametry modelu (parametry proste i tablice), którym następnie, w eksperymentach symulacyjnych, mogą być nadawane wartości. Deklarowane są tam również identyfikatory — synonimy stałych używanych w opisie modelu oraz identyfikatory zmiennych obserwowanych w procesie symulacji.

Na właściwy opis modelu składa się sekwencja instrukcji opisu modelu, ujęta w nawiasy syntaktyczne BEGIN-END. Instrukcje opisu modelu mają postać instrukcji podstawienia, przy czym:

- zmienne występujące po lewej stronie operatora podstawienia muszą być unikalne, tzn. mogą być zdefiniowane tylko raz
- instrukcje opisu modelu nie muszą być wykonywane w kolejności zapisu.

Kolejność wykonywania operacji obliczalną, określaną jest przez translator na poziomie języka maszyny SIM/AGH.

Wyrażenie pojawiające się w instrukcji opisu modelu jest rozszerzeniem standardowego wyrażenia arytmetycznego; zbiór dopuszczalnych operacji powiększono bowiem o operatory logiczne i relacyjne, wprowadzono też funkcje o wartościach i argumentach logicznych. Wspomniane operatory i funkcje dokonują przy tym pewnej interpretacji logicznej wartości rzeczywistych.

Język GODYS-5 dysponuje zbiorem 52 operacji standardowych i pozwala na zdefiniowanie do pięciu operacji opcjonalnych, zapisywanych na przykład w języku FORTRAN. Wśród operacji standardowych można wyróżnić kilka grup:

- operacje z pamięcią (inercyjne), wykonywane przez
 - integratory (normalny, z modyfikacją, z ograniczeniem wyjścia, z ustawieniem warunku początkowego)
 - transmitancje pierwszego i drugiego rzędu
 - przerzutniki typu JK i D



Dr inż. Jerzy KRÓL jest absolwentem kierunku „automatyka”, specjalność „technika cyfrowa” (AGH-1970), stopień doktora uzyskał na Wydziale Automatyki i Informatyki Politechniki Śląskiej (1974). Od 1975 roku pracuje w Instytucie Informatyki UJ na stanowisku adiunkta, pełniąc obowiązki zastępcy dyrektora Instytutu i kierownika Zakładu Podstaw Informatyki. Specjalizuje się w zakresie metod translacji języków programowania i systemów operacyjnych.



Mgr Jacek KURAS jest absolwentem matematyki Uniwersytetu Jagiellońskiego (1974 r.). Obecnie pracuje na stanowisku starszego asystenta w Instytucie Informatyki UJ. Zajmuje się zagadnieniami związanymi z systemami operacyjnymi i procesami współbieżnymi.

komputerów ODRA 1300

- licznik
- histereza typu „luz mechaniczny”
- ekstrapolator zerowego rzędu
- ekstrapolator algebraiczny, tzw. blok izolacyjny, umożliwiający wprowadzenie definiowanych pętli algebraicznych
- generatory liczb losowych
- operacje specjalne (diody, tyrystor, układ antyrównoległy dioda—tyrystor)
- operacje z historią
 - różniczkowanie
 - transmitancja typu $(As+1)/(Bs+1)$
 - opóźnienie jedno- i wielokrotowe
 - detektor zera
 - generatory pulsów i impulsów
 - komutatory
- operacje natychmiastowe
 - matematyczne, w tym funkcje ciągle jednej i dwóch zmiennych zadane tablicami
 - funkcje specjalne, używane w symulacji systemów dynamicznych (ogranicznik, funkcja charakterystyczna na odcinku domkniętym, funkcja pochylniowa, funkcje typu skoku jednostkowego, kwantyzator, przełącznik bez histerezy, przełączniki sterowane, funkcja schodkowa zadana tablicą, detektor przekroczeń alarmowych).

Przykładowy program-opis modelu przedstawiono poniżej.

```
GODYS-5      COMPILATION BY #XGDS      V1.0      06/06/78

0      MODEL BANBRANG
1      PARAMT X10,X20,A,B,C,D
2      PREPARE X1,X2,SIGNX2,U
3      BEGIN
4      X1= INTEG(X2;X10)
5      X2= INTEG(U-X1;X20)
6      SIGNX2= RELAY(X2;A,B,-A,-B)
7      U= RELAY (X1+X2- SIGNX2+LOG(CABS(1+X2+SIGNX2)) ;
      A,-B,-A,B)
      END
```

Jak wspomniano, program jest przetwarzany w równoważny program w języku maszyny abstrakcyjnej, stanowiący podstawowe wejście do systemu wykonawczego. Pracą tego systemu sterują dyrektywy symulacji. Umożliwiają one m.in. zainicjowanie pojedynczego eksperymentu symulacyjnego o charakterze autonomicznym (dyrektywa EXECUTE) lub kontynuację poprzedniego eksperymentu (dyrektywa CONTINUE). Dla takiego eksperymentu można ustalić parametry symulacji (metodę całkowania numerycznego, krok całkowania itp.), warunki składowania programu w przypadku wystąpienia szczególnych zdarzeń (akcja operatora, zakończenie czasu symulacji, itp.), a także spowodować, że dany eksperyment będzie inicjowany

nawet w przypadku, gdy w poprzednim był błąd wykonania. Możliwe jest też bieżące śledzenie wartości zmiennych modelu na określonym odcinku czasu.

Aktualnie system wykonawczy dysponuje sześcioma procedurami całkowania numerycznego, w tym zmiennokrotową i przyspieszania wykładniczego dla całkowania układów równań sztywnych (o skrajnie różnych wartościach własnych).

Wartości parametrów modelu można ustalać dyrektywami DATA. Dyrektywy wyjścia powodują wyprowadzanie w określonej formie (tabela, wykres wartości sporządzone na drukarce wierszowej, płaszczyzna stanu) podzbiór zmiennych obserwowanych, których wartości zarejestrowane zostały w pamięci masowej, w trakcie eksperymentu symulacyjnego. Skalowanie wykresów, dokonywane jest automatycznie, z uwzględnieniem optymalnego (w sensie czytelności) doboru współczynnika skali lub też skala ustalana jest na podstawie podanych przez użytkownika wartości granicznych danej zmiennej. Inne dyrektywy pozwalają na składowanie i wznowianie programu i na komunikację systemu wykonawczego (poprzez plik w pamięci masowej) z innym programem, na przykład z programem optymalizacji parametrycznej.

Zasadniczymi elementami translatora i systemu wykonawczego są przetworniki sterowane składnią i bazujące na gramatykach precedensyjnych. Pozwoliło to na uzyskanie przejrzystej struktury programowej systemu oraz dobrej diagnostyki i lokalizacji błędów.

Translator jest dwuprzebiegowy. W pierwszym przebiegu generowany jest lokalnie zoptymalizowany program w języku maszyny abstrakcyjnej. W drugim przebiegu następuje wyznaczenie sekwencji obliczalnej operacji i utworzenie programu wynikowego. Program ten poprzez plik pamięci masowej przekazywany jest do systemu wykonawczego. W systemie tym zainicjowanie eksperymentu symulacyjnego poprzedzone jest intensywną kontrolą poprawności parametrów modelu i symulacji. Pozwoliło to na zminimalizowanie kontroli błędów na etapie interpretacji i tym samym zapewniło jej dużą szybkość.

Kompilator i system wykonawczy zaopatrzone są obficie w środki uruchamiania, takie jak wydruk mapy identyfikatorów, symbolicznej mapy programu wynikowego, śladu wykonania i inne, co pozwala na łatwe wykrycie i lokalizację ewentualnych błędów.

Język GODYS-5 użytkowany jest od wiosny 1978 r. w wielu zespołach — zarówno dla celów badawczych, jak i dydaktycznych (dla tych celów procesor języka rozpoznawczym jest nieodpłatnie). Uzyskane doświadczenia pozwalają sądzić, że język ten jest prostym i efektywnym narzędziem do symulacji szerokiej klasy systemów dynamicznych ciągłych i ciągłych ze zdarzeniami dyskretnymi. Należy też podkreślić, że analiza wspomnianych doświadczeń wpłynęła na ostateczną formę implementacji języka GODYS-4 [4] i nadała jej szczególnie wysokie walory użytkowe.

LITERATURA:

- [1] Jakubowski R., Król J.: Implementation of the simulation language based on functional graphs. „Podstawy Sterowania” t. 2, 1972
- [2] Król J.: Symulacja cyfrowa złożonych systemów elektromechanicznych. Rozprawa doktorska. Politechnika Śląska, Gliwice 1974
- [3] Praca zbiorowa: GODYS-3 — język do symulacji złożonych systemów dynamicznych bazujących na grafach funkcyjnych. Dokumentacja Projektowa — „Systemy Sterowania”. PETROINFORM, Kraków 1974
- [4] Praca zbiorowa: GODYS-4 — język do symulacji ciągłych układów dynamicznych. Cz. I. Raport roboczy II UJ, Kraków 1977
- [5] Praca zbiorowa: GODYS-5 — język do symulacji ciągłych ze zdarzeniami dyskretnymi. Raport roboczy II UJ, Kraków 1978



Mgr Jacek LEMBAS jest absolwentem matematyki Uniwersytetu Jagiellońskiego (1974 r.). Obecnie pracuje na stanowisku starszego asystenta w Instytucie Informatyki UJ. Zajmuje się problemami związanymi z automatyczną konstrukcją translatorów.