

# ZASTOSOWANIE JĘZYKA GRIP W TECHNICZNYM PRZYGOTOWANIU PRODUKCJI

Cezary GRABOWIK, Witold JANIK

**Streszczenie:** W artykule przedstawiono krótką charakterystykę języka programowania GRIP umożliwiającego opracowanie procedur programowych wspomagających m.in. użytkownika systemu NX 3/4/5/6 w procesie technicznego przygotowania produkcji w szczególności w zakresie modelowania 3D konstrukcji oraz technologicznego przygotowania produkcji w zakresie programowania obrabiarek NC. Przykład aplikacji programowej SHAFT-CAD wspomagającej modelowanie elementów obrotowo-symetrycznych został przedstawiony w treści artykułu.

**Słowa kluczowe:** konstrukcyjne obiekty elementarne, GRIP, NX 6.0.

## 1. Wprowadzenie

Ze względu na konieczność sprostania wymaganiom współczesnego rynku celowe wydaje się wdrożenie programowego wspomaganie konstruowania oraz modelowania 3D w środowisku zaawansowanych systemów CAD/CAM takich, jak CATIA czy NX. Zastosowanie programowego wspomaganie konstruowania umożliwia:

- usprawnienie i przyspieszenie procesu projektowania,
- szybsze tworzenie rodzin konstrukcji,
- zmniejszenie ryzyka wystąpienia błędu konstrukcyjnego na etapie modelowania konstrukcji wytworu (poprzez wprowadzenie procedur automatycznego doboru wartości obiektów elementarnych z norm przedmiotowych).

Biorąc pod uwagę powyższe należy zastanowić się jaki powinien być zakres wdrożenia programowego wspomaganie procesu konstruowania i do jakich celów będzie ono stosowane. Jako przykład wskazać można program wspomagający proces konstruowania wałów-zębników z zastosowaniem opracowanej uprzednio biblioteki obiektów elementarnych. Programowe wspomaganie procesów konstruowania oraz modelowania 3D konstrukcji we współczesnych zintegrowanych systemach CAD/CAM możliwe jest głównie dzięki zastosowaniu mechanizmów OPEN/API co oznacza możliwość programowania w środowisku systemu CAD/CAM z zastosowaniem języków wysokiego poziomu takich jak C/C++, C#, Java czy Visual Basic. W odróżnieniu od innych systemów CAD/CAM system NX umożliwia programowanie z zastosowaniem języka wewnętrznego wysokiego poziomu nazwanego od pierwszych liter słów GRIP (ang. Graphics Interactive Programming). Z zastosowaniem języka GRIP możliwe jest:

- a) tworzenie zapisu geometrii dwuwymiarowej:
  - zorientowanej na geometrię,
  - zorientowanej na zapis konstrukcji,
- b) tworzenie zapisu geometrii trójwymiarowej:
  - tworzenie obiektów trójwymiarowych,
  - przeprowadzenie operacji boolowskich,

- tworzenie brył w oparciu o płaszczyzny ograniczające,
- c) transformacje (macierze transformacji),
- d) interaktywną relację z użytkownikiem,
  - tworzenie okien dialogowych wyboru opcji,
  - tworzenie okien dialogowych z polami edycyjnymi,
  - identyfikacja obiektów dwu i trójwymiarowych,
  - realizacja funkcji alternatywnych w razie popełnienia błędu przez użytkownika,
- e) przeprowadzanie obliczeń i analizy danych,
- f) przeprowadzanie operacji na plikach:
  - tworzenie plików parametrycznych,
  - zapis i odczyt danych pliku parametrycznego,
  - automatyczne uruchomienie podprogramów.

## 2. SHAFT-CAD program wspomagający konstruowanie oraz modelowane 3D wałków i zębników

Program SHAFT-CAD napisany został w języku programowania wysokiego poziomu GRIP [1]. Aplikacje napisane w języku GRIP (G<sup>R</sup>aphics Interactive Programming [4]) są edytowane, kompilowane i przekształcane w pliki wykonywalne z zastosowaniem środowiska GRADE (G<sup>R</sup>IP Advanced Development Environment [13]). Okno główne środowiska GRADE przedstawione zostało na rys. 1. Na rys. 2 przedstawiono typy plików obsługiwanych przez środowisko GRADE.

```

GRip Advanced Development Environment V2.0
-----

1) Edit                               6) send Output to [CRT]
2) Compile                             7) comPile listing [ALL]
3) Link                                8) change ediTor [notepad]
4) change Directory                    9) grade Batch
5) liSt directory                       0) turn Menu on/off

                                     q) QUIT

DIR = C:\Documents and Settings\
Enter option: [1]

```

Rys. 1. Środowisko GRADE: 1 – edycja, 2 – kompilacja, 3 – przekształcenie w plik wykonywalny, 4 – zmiana ścieżki dostępu do katalogu domyślnego, 5 – zwrócenie listy katalogów, 6 – zrzut ekranu w wybranej formie, 7 – sposób przedstawienia kompilacji, 8 – zmiana edytora pliku źródłowego, 9 – pakiet opcji wielo-procesowych GRADE (np. jednoczesne wykonanie kompilacji, przekształcenie w plik wykonywalny i uruchomienie programu), 0 – włączenie/wyłączenie menu głównego środowiska GRADE

Kolejność tworzenia programu z zastosowaniem języka GRIP jest następująca (patrz.: Rys. 2.):

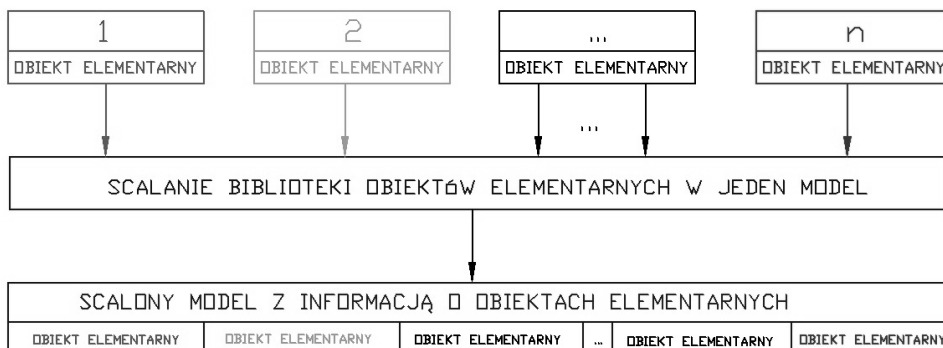
- a) należy utworzyć źródłowy plik tekstowy z zastosowaniem dowolnego edytora tekstu np. Notatnik systemu Windows. W celu usprawnienia procesu tworzenia kodu źródłowego – pliku tekstowego zasadne wydaje się użycie edytora tekstu umożliwiającego „kolorowanie składni” lub przynajmniej udostępniającego funkcję numeracji linii kodu co jest szczególnie przydatne na etapie modyfikacji kodu źródłowego programu w przypadku wystąpienia błędów na etapie kompilacji, linkowania bądź eksploatacji oprogramowania,
- b) utworzony plik z kodem źródłowym – plik z rozszerzeniem \*.grs (ang. grip source code) należy skompilować stosując opcję środowiska GRADE „Compile”, w wyniku czego kod źródłowy programu kompilowany jest do tzw. kodu obiektowego (grip executive code) tj. pliku o rozszerzeniu \*.gri,
- c) skompilowany plik kodu obiektowego \*.gri należy przekształcić w plik kodu wykonywalnego (grip executive code) tj. plik o rozszerzeniu \*.grx z zastosowaniem opcji „Link” środowiska GRADE,
- d) uzyskany plik kodu wykonywalnego \*.grx może następnie być uruchomiony w środowisku systemu NX lub w trybie wsadowym – tryb BATCH.



Rys. 2. Zestawienie typów pliku obsługiwanych przez środowisko GRADE

## 2.1. Biblioteka obiektów elementarnych

Działanie programu SHAFT-CAD opiera się na zastosowaniu metody konstrukcyjnych obiektów elementarnych, tj. obiektów o pewnej specyficznej strukturze opisującej postać konstrukcyjną wyróżnionej cechy konstrukcyjnej, dla której układ wymiarów może być dowolnie modyfikowany. Zbiór konstrukcyjnych obiektów elementarnych tworzy otwartą strukturę, która może być modyfikowana w zależności od zmieniających się potrzeb. W trakcie działania programu tworzone są konstrukcyjne obiekty elementarne, które w końcowym etapie modelowania konstrukcji wytworu scalane są z zastosowaniem operacji boole’a w spójny model. Proces użytkowania opracowanej biblioteki obiektów elementarnych przedstawiono na Rys. 3.

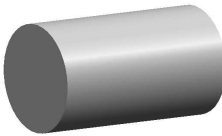

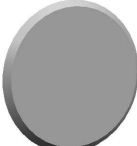




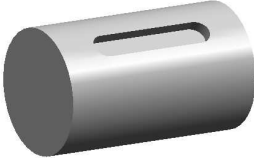
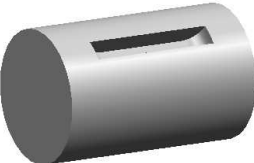
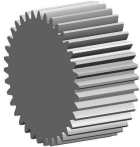
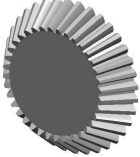
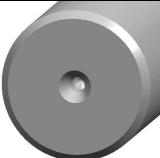
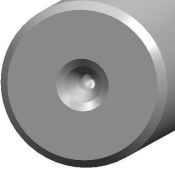
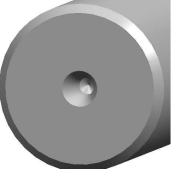
Rys. 3. Modelowanie konstrukcji z zastosowaniem biblioteki obiektów elementarnych

## 2.2. Rodzaje obiektów elementarnych oraz sposób ich zapisu

W programie SHAFT-CAD konstrukcja wytworu modelowana jest w kierunku od lewej do prawej strony w sposób przyrostowy. Zastawienie dostępnych na obecnym etapie rozwoju oprogramowania konstrukcyjnych obiektów elementarnych zestawiono w tabeli 1.

Tab.1. Zestawienie wybranych konstrukcyjnych obiektów elementarnych

RODZAJ OBIEKTU	POSTAĆ GEOMETRYCZNA	PARAMETRY OBIEKTU
Stożek wałka		SW, średnica, długość, długość całkowita wałka/zębnika (na danym etapie konstruowania)
Ścięcie lewe		SL, średnica, szerokość, kąt, długość całkowita wałka/zębnika (na danym etapie konstruowania)
Ścięcie prawe		SP, średnica, szerokość, kąt, długość całkowita wałka/zębnika (na danym etapie konstruowania)
Promień lewy		PL, promień, średnica podstawowa, długość całkowita wałka/zębnika (na danym etapie konstruowania)

Promień prawy		PP, promień, średnica następnego stopnia, długość całkowita wałka/zębnika (na tym etapie konstruowania)
Wpust palcowy		WP, szerokość rowka, długość rowka, średnica stopnia, głębokość rowka, odległość symetrycznego środka, długość całkowita wałka/zębnika (na tym etapie konstruowania)
Wpust czółenkowy		WC, szerokość rowka, długość rowka, średnica stopnia, głębokość rowka, średnica frezu, odległość symetrycznego środka, długość całkowita wałka/zębnika (na tym etapie konstruowania)
Wieniec walcowy		ZC, moduł, liczba zębów, długość zębnika, długość całkowita wałka/zębnika (na tym etapie konstruowania)
Wieniec stożkowy		ZS, moduł, liczba zębów, kąt pochylecia linii zęba, współczynnik szerokości, długość całkowita wałka/zębnika (na tym etapie konstruowania)
Nakiełek zwykły czoło lewe i prawe wałka/zębnika (nad-wymiarowe)		NK, rodzaj nakiełka, wymiary wg normy PN-83/M-02499 dotyczące nawiertaka (60°)
Nakiełek chroniony czoło lewe i prawe wałka/zębnika		NK, rodzaj nakiełka, wymiary wg normy PN-83/M-02499 dotyczące nawiertaka (60°)
Nakiełek łukowy czoło lewe i prawe wałka/zębnika		NK, rodzaj nakiełka, wymiary wg normy PN-83/M-02499 dotyczące nawiertaka (60°)

Program SHAFT – CAD jest programem o budowie modułowo – hierarchicznej. Moduł geometryczny składa się z dwóch modułów: konstrukcyjnego i odczytu danych z pliku. Z zastosowaniem tych modułów możliwe jest wykonanie działań związanych z:

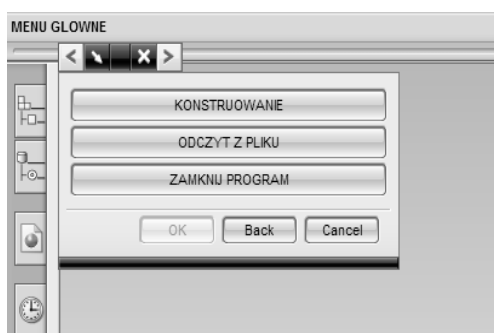
- a) inicjalizacją zmiennych i ich wartości,
- b) inicjalizacją łańcuchów alfanumerycznych,
- c) inicjalizacją obiektów geometrycznych,
- d) dokonywaniem obliczeń,
- e) tworzeniem geometrii dwuwymiarowej,
- f) tworzeniem geometrii trójwymiarowej,
- g) modyfikacją geometrii,
- h) modyfikacją ustawień z zastosowaniem globalnych parametrów dostępu,
- i) modyfikacją ustawień z zastosowaniem geometrycznych parametrów dostępu,
- j) realizacją zadań logicznych,
- k) interaktywną komunikację z użytkownikiem,
- l) logistyką między etykietami sekwencji programowych,
- m) reakcją na błędy powstające w czasie realizacji funkcji,
- n) realizacją kombinacji funkcji.

Strukturę programu SHAFT–CAD wraz z wyróżnionymi przepływami danych przedstawiono na rys. 4.

### 2.3. Opis działania programu SHAFT – CAD

Modelowanie wałków i zębników w programie odbywa się według reguły jednostronnego kierunku modelowania (od lewej do prawej). Ten sposób konstruowania odzwierciedlony jest w procesie zapisu oraz odczytu danych z pliku parametrycznego. Do tej zasady odwołują się nazwy obiektów elementarnych np. promień prawy, nakiełek lewy, ścięcie prawe itp. Na rys. 5–8 przedstawiono przykład modelowania zębника stożkowego.

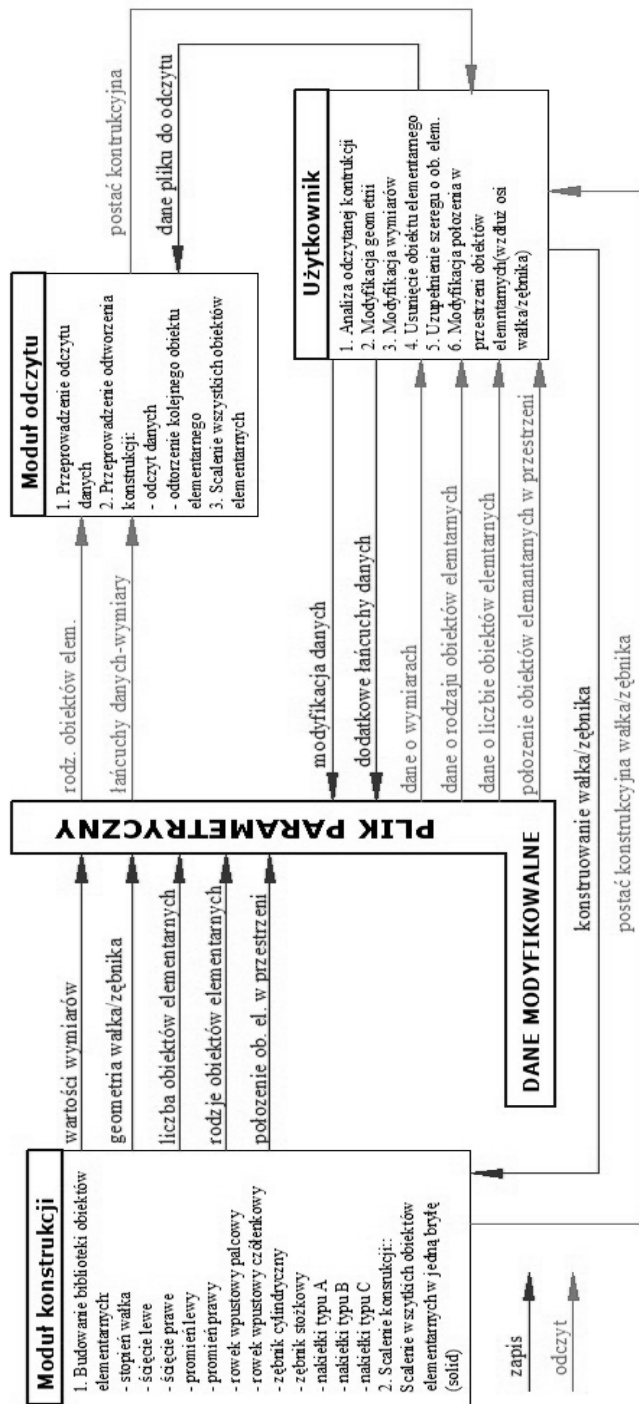
Rys. 5 przedstawia okno główne umożliwiające wczytanie danych z pliku parametrycznego, w przypadku konieczności wczytania danych uprzednio modelowanego wytworu lub przejście do modelowania nowego.



Rys. 5. Okno główne programu SHAFT–CAD

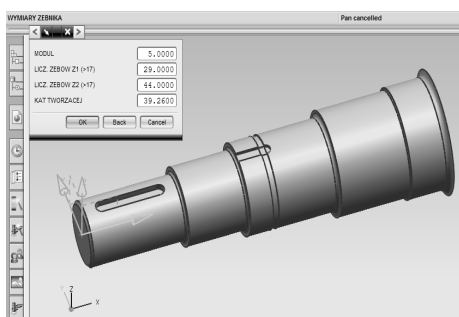


Rys. 6. Menu wyboru obiektu elementarnego

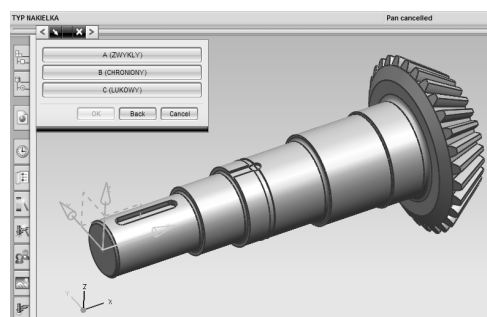


Rys. 4. Struktura programu SHAFT-CAD, przepływ danych w systemie

Na rys. 6 przedstawiono fragment okna umożliwiającego wybór jednego z zaimplementowanych w programie obiektów elementarnych. Po utworzeniu obiektu elementarnego następuje powrót do menu wyboru obiektu elementarnego, celem umożliwienia dokonania wyboru kolejnego obiektu elementarnego do wstawienia. Należy pamiętać że obiekty elementarne typu nakiełek należy wstawić na końcu procesu modelowania wałka/zębnika celem ich poprawnego wykonania. Scalenie konstrukcji kończy proces konstruowania, opcję scalenia modelu należy wybrać gdy jesteśmy całkowicie pewni o kompletności naszego modelu (ma to szczególne znaczenie praktyczne, gdyż po scaleniu konstrukcji w spójny model tracimy możliwość odwołania się do poszczególnych obiektów elementarnych tworzących model).



Rys. 7. Modelowanie wieńca stożkowego



Rys. 8. Modelowanie nakiełka

### 3. Podsumowanie

W artykule przedstawiono przykład aplikacji programowej SHAFT-CAD napisanej w języku wewnętrznym GRIP środowiska systemu NX 5.0. Ze względu na kompatybilność wstecz oraz w przód aplikacji napisanych w języku GRIP możliwe jest ich uruchomienie w różnych wersjach systemu NX. Poprawność działania opracowanej aplikacji potwierdzona została dla wersji 3, 4, 5 oraz 6. Zastosowanie metody konstrukcyjnych obiektów elementarnych oraz zapis struktury obiektów w postaci modułów programowych umożliwia ich wielokrotne zastosowanie w procesie modelowania konstrukcji w sposób zbliżony do zastosowania technik stricte obiektowych (C++, Java). Wadą języka GRIP i co za tym idzie aplikacji napisanej z jego zastosowaniem jest fakt, iż aplikacje te działają w zasadzie wyłącznie w trybie interaktywnym charakteryzującym się tym że bardzo trudne bądź nie możliwe jest pozyskanie informacji o geometrii modelu uzyskanej na drodze modelowania „tradycyjnego” nieprogramowego. Pomimo tego zasadne wydaje się opracowanie programu typu SHAFT-CAD do wspomaganie procesu modelowania konstrukcji z zastosowaniem biblioteki konstrukcyjnych obiektów elementarnych opracowanej dla wyróżnionej grupy cech konstrukcyjnych ze względu na fakt znacznego skrócenia czasu modelowania konstrukcji. Przy odpowiednio opracowanej strukturze zbioru konstrukcyjnych obiektów elementarnych proces modelowania konstrukcji w szczególności w przypadku przenoszenia modelu z dokumentacji 2D do modelu 3D zajmuje czas rzędu kilku minut. Dodatkowo na uwagę zasługuje fakt łatwej przyswajalności języka co powoduje, iż programista w bardzo krótkim czasie potrafi



efektywnie posługiwać językiem GRIP. Prezentowana praca naukowa finansowana jest ze środków na naukę w latach 2006-2009, jako projekt badawczy Nr N115 081 31/1906.

### **Literatura**

1. Janik W.: Wspomaganie technicznego przygotowania produkcji z zastosowaniem techniki obiektów elementarnych. Praca magisterska, Gliwice 2008.
2. Grabowik C., Knosala R.: The method of knowledge representation for a CAPP system. Journal of Materials Processing Technology, Elsevier 133 (2003) ss. 90-98.
3. Grabowik C., Kalinowski K., Monica Z.: Integration of the CAD/CAPP/PPC. Journal of Materials Processing Technology, Elsevier 164-165 (2005) ss. 1358-1368.
4. UGS Corporation.: Grip fundamentals – student guide, MT13010 – NX3, Plano, Texas, 2005.
5. Grabowik C., Skołud B., Kalinowski K., Krenczyk D.: Integracja technicznego i organizacyjnego przygotowania produkcji z zastosowaniem technik obiektowych. XVI Konferencja nt „METODY I ŚRODKI PROJEKTOWANIA WSPOMAGANEGO KOMPUTEROWO”, Nałęczów, październik 2007.

Dr inż. Cezary GRABOWIK

Mgr. inż. Witold JANIK

Instytut Automatykacji Procesów Technologicznych

i Zintegrowanych Systemów Wytwarzania

Politechnika Śląska

44-100 Gliwice, ul. Konarskiego 18a

e-mail: [cezary.grabowik@polsl.pl](mailto:cezary.grabowik@polsl.pl)

[witold.janik@polsl.pl](mailto:witold.janik@polsl.pl)