

Tadeusz Szuba
Akademia Górniczo-Hutnicza
Instytut Informatyki

AUTOMATYCZNA SYNTEZA PROGRAMU DLA INTELIGENTNYCH MASZYN ROBOCZYCH

Streszczenie. W referacie zostały przedstawione doświadczenia i wyniki prac nad problemem tworzenia systemów oprogramowania dla automatycznej syntezy programu pracy inteligentnych maszyn roboczych. Jako obiekt doświadczeń wybrano: N.C. inteligentną frezarkę o 6-stopniach swobody, przyszłościową 2-manipulatorową obrabiarkę oraz inteligentną koparkę. Koncepcja oprogramowania oparta jest na wielojęzycznym podejściu z wykorzystaniem PROLOGu jako czynnika łączącego.

1. Postawienie zagadnienia:

Automatyczna synteza programu jest działem sztucznej inteligencji, w którym wykorzystując wszystkie opracowane dotychczas metody rozwiązywania problemów, konstrukcji inteligentnego softwaru itp. buduje się systemy oprogramowania, które:

1/ w oparciu o definicję sceny procesu technologicznego,

2/ definicję stanu początkowego i docelowego /końcowego/ procesu

całkowicie samodzielnie znajdują sekwencje operacji technologicznych przekształcających stan początkowy w docelowy, a dalej:

albo prezentują to rozwiązanie operatorowi do wglądu /jest to wtedy model pracy CAD/CAM/Expert/,

albo kierują to rozwiązanie wprost do urządzeń wykonawczych /jest to wtedy model pracy "inteligentna maszyna"/.

Powyższe tryby pracy mogą w siebie płynnie przechodzić w zależności od sposobu współpracy człowiek/maszyna. Prezentowane badania dotyczą wybranej specyficznej klasy procesów technologicznych, dla których zbudowanie systemów automatycznej syntezy programu ma szczególne znaczenie. Są to:

- proces technologiczny realizowany przez uniwersalną obrabiarkę/frezarkę o 6 stopniach swobody i magazynku narzędzi,

- proces technologiczny dla hipotetycznej 2-manipulatorowej obrabiarki /też z magazynkiem narzędzi/,

- proces technologiczny dla inteligentnej, autonomicznej, gąsiennicowej koparki.

Prezentowane badania zostały w większości opublikowane w ubiegłych latach na terenie RPN. [4], [5], [6], [7], [8].

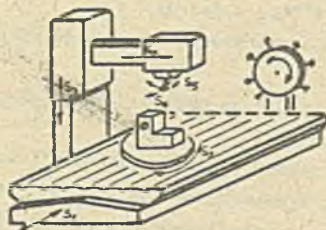
Prace są finansowane w temacie CPBP 02.13 przez IPPT PAN Warszawa, pod kierownictwem prof. A. Borkowskiego.

Pojęcie sceny procesu

Scena procesu technologicznego jest tutaj zdefiniowana jako 3-D zgeometryzowane środowisko, zawierające zarówno bryły realne, jak i wirtualne /pozorne/ niezbędne dla procesu technologicznego. Tak więc na scenie znajdują się:

- urządzenia traktowane jako robot, aproksymowane zespołem brył;
- przeszkody, takie jak: elementy stołu obrabiarki, powierzchnia, po której robot się porusza, dmy, drzewa etc. Są one także aproksymowane bryłami,
- obiekty reprezentujące cel pracy, tj.: "duch przedmiotu docelowego" tkwiący wewnątrz bryły surowego metalu, kontur przyszłego wykopu nakłożony na prófil terenu. Są to bryły 3-D w sensie geometrycznym,
- bryły powstałe w wyniku transformacji geometrycznych sceny lub' pozorne bryły narzucane przez wymagania technologiczne. Np. koparka po wykryciu bagna tworzy nad nią bryłę kolizyjną. Chcąc uprościć ruchy frezu, zamykamy go w kopertę 3-D, którą transformujemy.

Ilustracją sceny dla omawianych powyżej procesów technologicznych są rysunki 1, 2, 3.



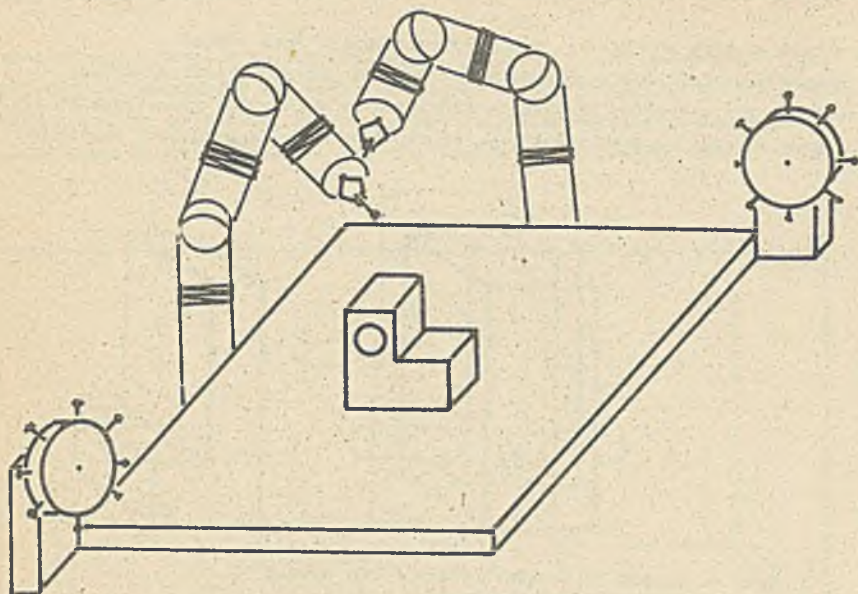
Rys. 1. Scena dla N.C. obrabiarki.
Fig. 1. The scene for N.C. machine tool.

Kolejne rysunki /4, 5, 6/ pokazują przykłady najbardziej charakterystycznych operacji geometrycznych na scenie. Są to /rys. 4./ metoda opisu obiektu początkowego i docelowego w zagadnieniu obróbki, generowania nowej przeszkody blokującej dostęp do bagna /rys. 5./, oraz transformacja sceny, która pozwala na wyodrębnienie obiektów metodą liczenia równowagi, a dalej na utworzeniu równoważnego planu 2-D dla programu planującego bezkolizyjny ruch koparki /rys. 6./.

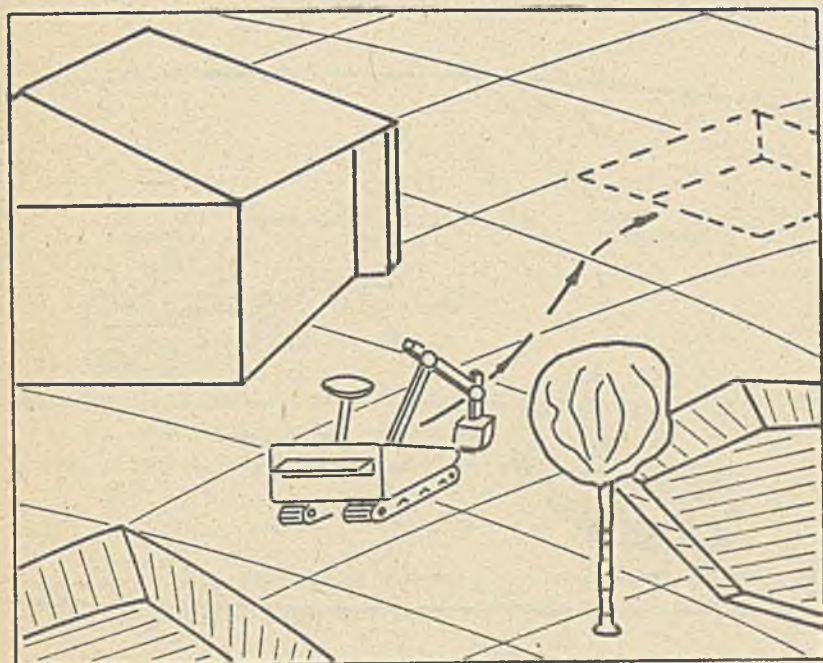
Technologiczna robotyzacja sceny

W ramach sceny, pod kątem potrzeb procesu technologicznego, podane są deterministyczne:

I/ mobilność robota jako elementu sceny, tj.



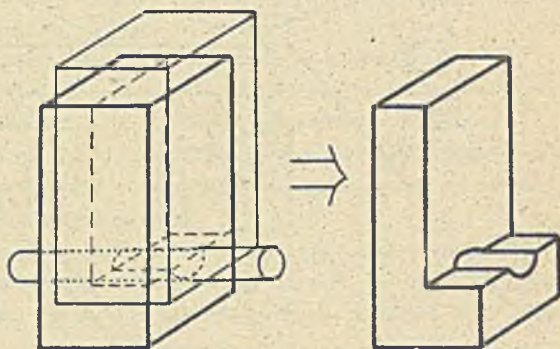
Rys. 2. Scena dla przyszłościowej 2-manipulatorowej obrabiarki.
Fig. 2. The scene for future machine tool with 2 manipulators.



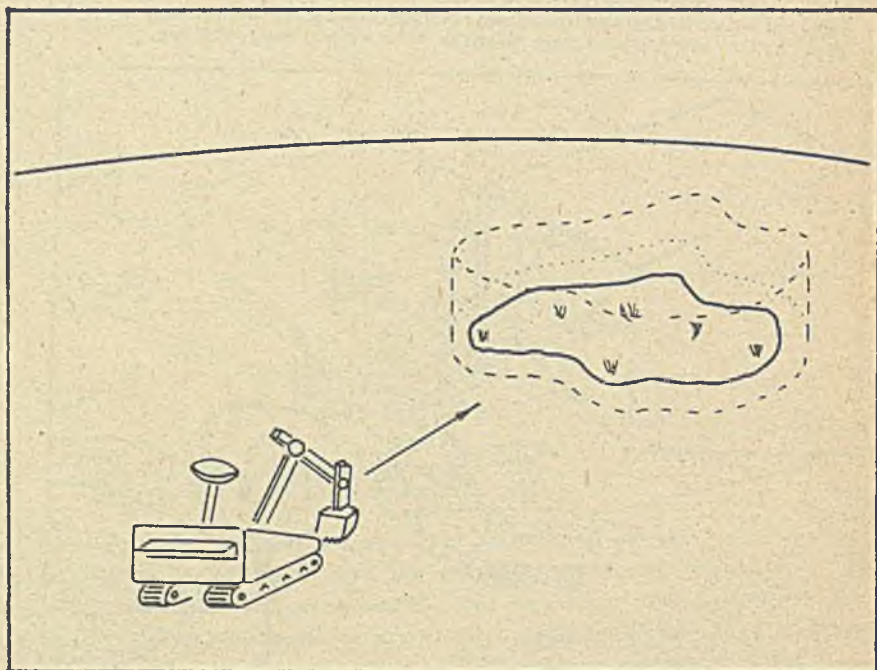
Rys. 3. Scena dla inteligentnej koparki.
Fig. 3. The scene for intelligent, autonomous digger.

- stopnie swobody obrabiarki, dwu robotów, koparki jako robota +
- zespół procedur liczących rezultaty ruchów w danych stopniach swobody.

II/ technologiczne możliwości urządzenia/roboty na scenie, tj. geometrycznie zdefiniowane warunki stosowalności tzw. standardów technologicznych i rezultaty

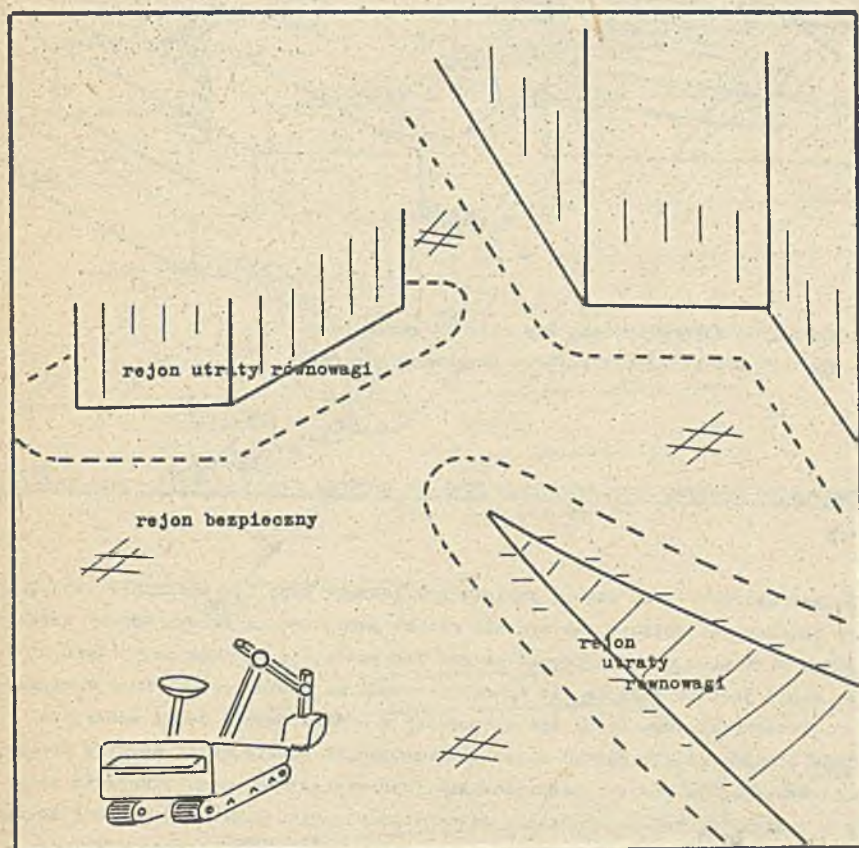


Rys. 4. Metoda opisu obrabianego bloku metalu.
Fig. 4. The workpiece description idea



Rys. 5. Problem bagna.
Fig. 5. The boggy region transformation.

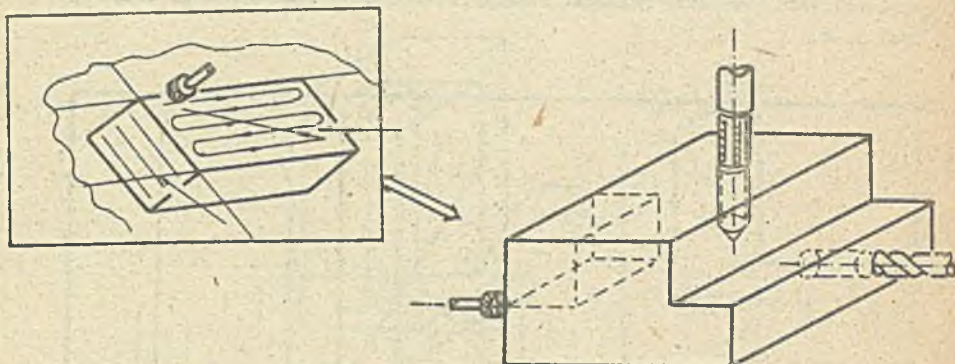
zastosowania, opisane w kategoriach geometrycznych, np. jak ustawić narzędzie przed blokiem metalu, aby móc wywiercić zadany otwór, jak należy ustawić koparkę względem konturu przyszłego wykopu, aby móc rozpocząć realizację standardowej bibliotecznego procedury kopania. Pojęcie standardu technologicznego ilustruje rys. 7, 8.



Rys. 6. Transformacja sceny do postaci 2-D.

Fig. 6. The scene transformation to 2-D plan.

Prezentowana tutaj tzw. inteligentna koparka jako rodzaj robota, jest aktualnie tematem prac zespołu [8], którym kieruję w ramach CPBP O2.13/4.8.



Rys. 7. Standardy technologiczne dla obróbki skrawaniem.

Fig. 7. The library standards for the machining process.

2. Konstrukcja systemu rozwiązującego problem syntezy ciągu operacji technologicznych,

Najpierw operator za pomocą pakietu graficznego typu CAD definiuje kształt obiektu docelowego, opisuje, jak wygląda obiekt początkowy i nasuwa "duch" kształtu docelowego na kontur obiektu początkowego. Tak postępuje się dla sceny typu obrabiarzka. Jeśli jest to koparka, to operator nanosi na kontur sceny "duch konturu wykopu". Towarzyszy temu także dialog maszyny z operatorem na temat sceny. Cel definiuje konfigurację, a raczej zbiór alternatywnych konfiguracji maszyny potrzebnych do wywiercenia otworu, nagwintowania, wyfrezowania wpustu, wykopania rowu. Znając z drugiej strony konfigurację maszyny na początku można już budować drzewo wariantów akcji, tak aby od stanu początkowego przejść do docelowego. Tak więc proces rozpada się na podcykle złożone z: ruchy manewrowe, ruchy robocze, ruchy manewrowe Wszystkie omawiane systemy były i są budowane w PROLOGU, który zapewnia tworzenie i utrzymanie drzew wariantów, a także przeszukiwanie ich według zadanej strategii. Budowa drzew jest jeszcze względnie prosta. Każda propozycja ruchu manewrowego lub technologicznego /odpowiadająca wygenerowaniu łuku i węzła/ pociąga za sobą BARDZO ŻMUDNE OBLICZENIA typu numerycznego, tj.:

- 1/ czy ruch jest celowy /eliminacja ruchów bezsensownych/, jeśli tak, to dokąd należy ruch wykonać, w jaki sposób /parametry/. Każdemu stopniowi swobody odpowiada procedura napisana w PASCALU + FORTRANie dająca odpowiedź Tak/Nie i ew. w jaki sposób /parametry/;
- 2/ jak wygląda nowa geometryczna konfiguracja sceny,

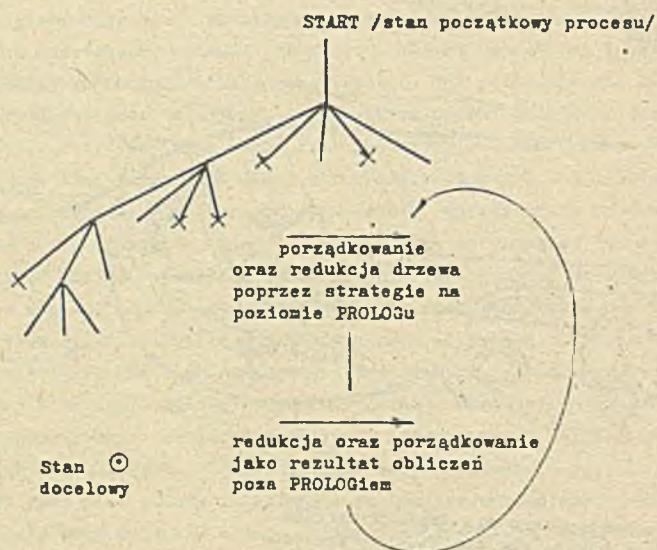
3/ czy ruch nie powoduje kolizji, utraty równowagi, kolizji technologicznej. Jeśli tak, to uzyskiwana jest informacja, gdzie i jak kolizja ta nastąpiła.

W praktyce 90% mocy obliczeniowej idzie na analizę proponowanego ruchu. Obliczenia te mogą być b. łatwe do zrównoleglenia przy użyciu, np. procesora macierzowego.

W rezultacie takich obliczeń drzewo wariantów ruchów jest obcięte o większość węzłów - co daje częściowe odsunięcie problemu eksplozji kombinatorycznej. Mimo to na poziomie PROLOGu muszą być wbudowane jeszcze strategie /maczelna/, np.:

- współpraca Master-Slave dla 2 robotów,
- strategie obchodzenia przeszkód przez koparkę.

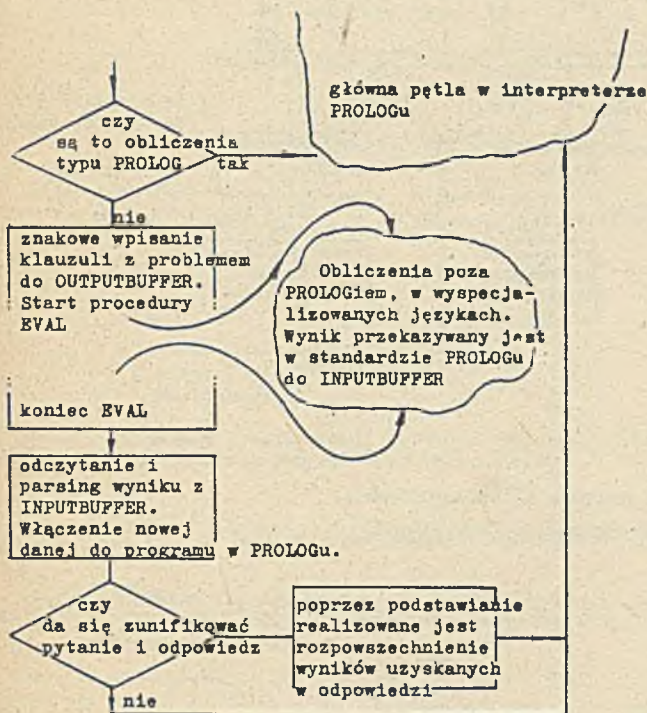
A w praktyce z punktu widzenia software'u oba istniejące już doświadczalne systemy automatycznej syntezy programu zostały zrealizowane w sposób następujący [5], [6], [7], [9]. Funkcje nadrzędne zimplementowane są na poziomie PROLOGu, przebudowanego do tego celu /tzw. Process Control PROLOG/. Przebudowano tzw. II UW PROLOG /Bien J., S., Kluźniak P., Szpakowicz S./ poprzez dodanie interfejsów do innych języków programowania oraz poprzez wprowadzenie specyficznego, zorientowanego na rozpatrywaną klasę procesów technologicznych, inteligentnego systemu odświeżania pamięci z nieaktualnych /niepotrzebnych faktów wniosków^{*)}. Program główny syntetyzujący program przejścia od stanu początkowego do docelowego napisany w c. PROLOGu generował pełne drzewo wariantów operacji technologicznych i pomocniczych. Rozgałęzienia tego drzewa były porządkowane i zredukowane natychmiast w oparciu o strategię rozwiązywania problemów zimplementowaną na poziomie PROLOGu /np. logiczne uporządkowanie operacji, algorytm współpracy manipulatorów itp./. Ilustruje to rys. 9. Wyselekcjonowane w ten sposób



Rys. 9. Zasada generacji drzewa wariantów

Fig. 9. The trees of variants generation process

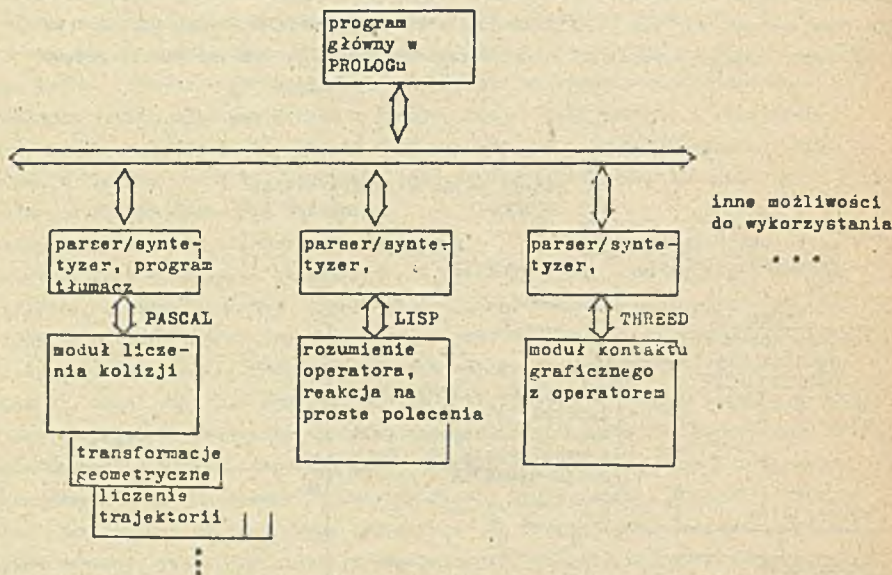
propozycje ruchów były wyliczane dokładnie poprzez procedury zrealizowane w PASCALu celem numerycznego wywnioskowania pełnych konsekwencji geometrycznych ruchu proponowanego. Realizowane to było za pomocą interfejsu przedstawionego skrótkowo na rys. 10. Rezultaty te służyły do zasadniczego obcinania drzewa wariantów poprzez



Rys. 10. Organizacja interfejsu w Process Control PROLOG.
Fig. 10. The interface organization in Process Control PROLOG.

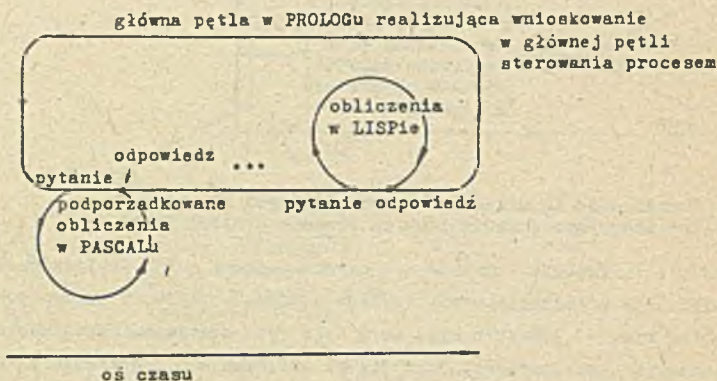
przeanalizowanie /w PROLOGu/ wyników, np. obrotu koparki o 30° /wyliczonych w PASCALu/ Ciekawe wyniki daje przeanalizowanie takiego systemu obliczeń z innego punktu widzenia podanego na rys. 11. Okazało się, że w tego typu systemach oprogramowania można b. prosto podać, jak problemy rozpadają się na obliczeniowo niezależne podproblemy /z wariantami też niezależnymi/, jaki mają one charakter i jakie jest ich powiązanie /drzewo wariantów/. Ogólnie system z rys. 11 pracuje w trybie Question-Answer przedstawionym na rys. 12. Analizując rysunki 11 i 12 widać, że dla inteligentnych maszyn tego typu najdogodniejszy i najbardziej obiecujący jest model sieciowy komputera.

*/ Był to także temat pracy dyplomowej p. mgr inż. J Grabkiej-Chrząstkowskiej realizowanej pod moim kierunkiem.



Rys. 11. Wielojęzykowa organizacja oprogramowania.

Fig. 11. Multi-language software organization.



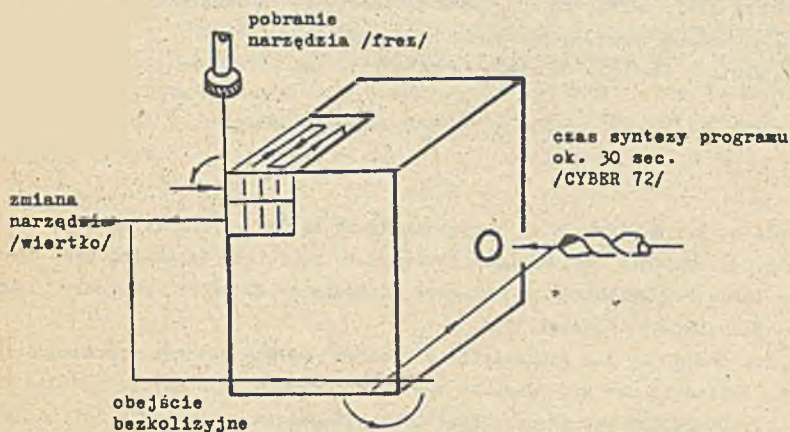
Rys. 12. Powiązanie oprogramowania poprzez Question-Answer

Fig. 12. Question-Answer system.

gdzie komputer jest układowo rozproszony w maszynie oraz występuje rozproszenie wyspecjalizowanych obliczeń w sieci.

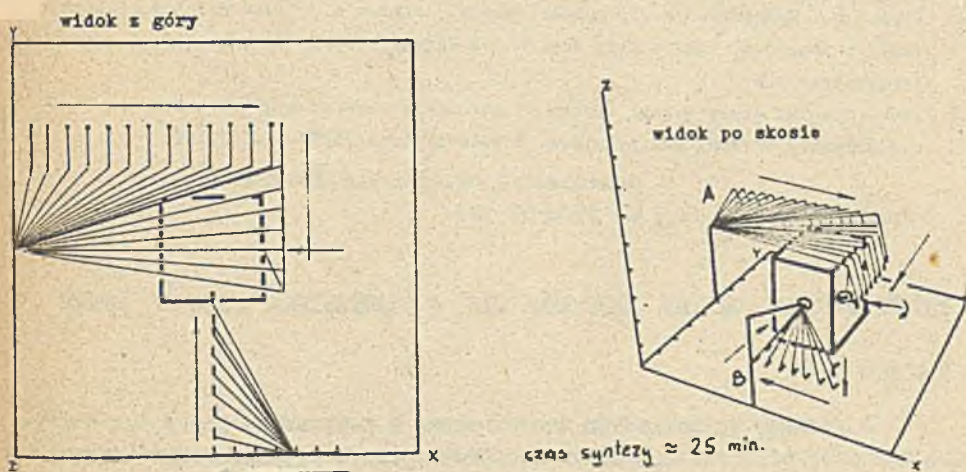
3. Uzyskane rezultaty i wnioski.

Dla obrabiarki i systemu z 2 manipulatorami udało się zbudować doświadczalne systemy automatycznej syntezy programu. Przykłady rozwiązań podaje rys. 13, 14.



Rys. 13. Przykłady zrealizowanej automatycznej syntezy programu.
Fig. 13. The automatic program synthesis example.

Jest to przerysowany i uproszczony rysunek z plottera.



Rys. 14. Przykład automatycznej syntezy programu dla 2 robotów.
Fig. 14. The automatic program synthesis example for two robots system.

Dla koparki system tego typu jest dopiero opracowywany. Generalnie trzeba powiedzieć, że dla tego typu systemów potrzebne są komputery o b. dużej mocy obliczeniowej rzędu 150 MIPS'ów. Wymagają dużej mocy obliczeniowej nawet okrojone, uproszczone wersje symulujące pracę obrabiarki w trybie CAD/CAM. Aktualnie w oparciu o zbudowaną przez doc. Krzysztofa Zielińskiego sieć UMLAN-2 opracowywane są założenia dla komputera tego typu, uwzględniające rozproszony model obliczeń. Jak się wydaje, jest to realnie możliwe w oparciu o dostępne na rynku karty z systemami mikroprocesorowymi.

LITERATURA

- [1] Clocksin W. F., Mellish C. S.: Programming in PROLOG. Springer Verlag. 1981.
- [2] Dauthins A., Geradin M.: Advanced software in robotics. North-Holland. 1984.
- [3] Lozano-Perez T.: Automatic planning of manipulator transfer movements. IEEE Trans. Vol. SMC-11, October 1981.
- [4] Szuba T., Gościński A.: Application of theorem proving methods for automatic program synthesis for N.C. machine tools. Angewandte Informatik. 5/1981.
- [5] Szuba T.: PC-PROLOG for process control applications. Angewandte Informatik. 4/1984.
- [6] Szuba T.: Automatic program synthesis system for N.C. machine tools based on PC-PROLOG. Angewandte Informatik. 6/1984.
- [7] Szuba T.: PROLOG as a real-time language for process control. Angewandte Informatik. 9/1984.
- [8] Byrski W., Karczmarszczuk J., Lubaszewski W., Stanek W., Szuba T.: Intelligent fully autonomous, numerically controlled digger. Report Version 1.0. Instytut Informatyki AGH.
- [9] Szuba T.: The experimental system of automatic program synthesis for two concurrently working manipulators. Robote Systeme /RPN/, /w druku/.

Recenzent: Prof. dr inż. H. Kowalowski

Wpłynęło do Redakcji do 1988-04-30.

АВТОМАТИЧЕСКИЙ СИНТЕЗ ПРОГРАММ ДЛЯ ИНТЕЛЛИГЕНТНЫХ РАБОЧИХ МАШИН

Резюме

В докладе представлены исследования и результаты работ над системами для автоматического синтеза программы работы интеллигентных машин. Для экспериментов были выбраны следующие машины: станок с 6 степенями свободы, оудущий станок оборудован двумя манипуляторами, интеллигентный экскаватор. Программная идея состоит в многоязыковом подходе, где язык ПРОЛОГ использован как объединяющий фактор.

AUTOMATIC PROGRAM SYNTHESIS FOR INTELLIGENT MACHINES

S u m m a r y

Paper presents researches and results on automatic program synthesis for intelligent machines. As the target machines were selected: N.C. machine tool with 6 degrees of freedom, future machine tool equipped with 2 robots, intelligent autonomous digger. The software idea is based on the so called multi-programming-language approach. PROLOG is used as the binding agent.