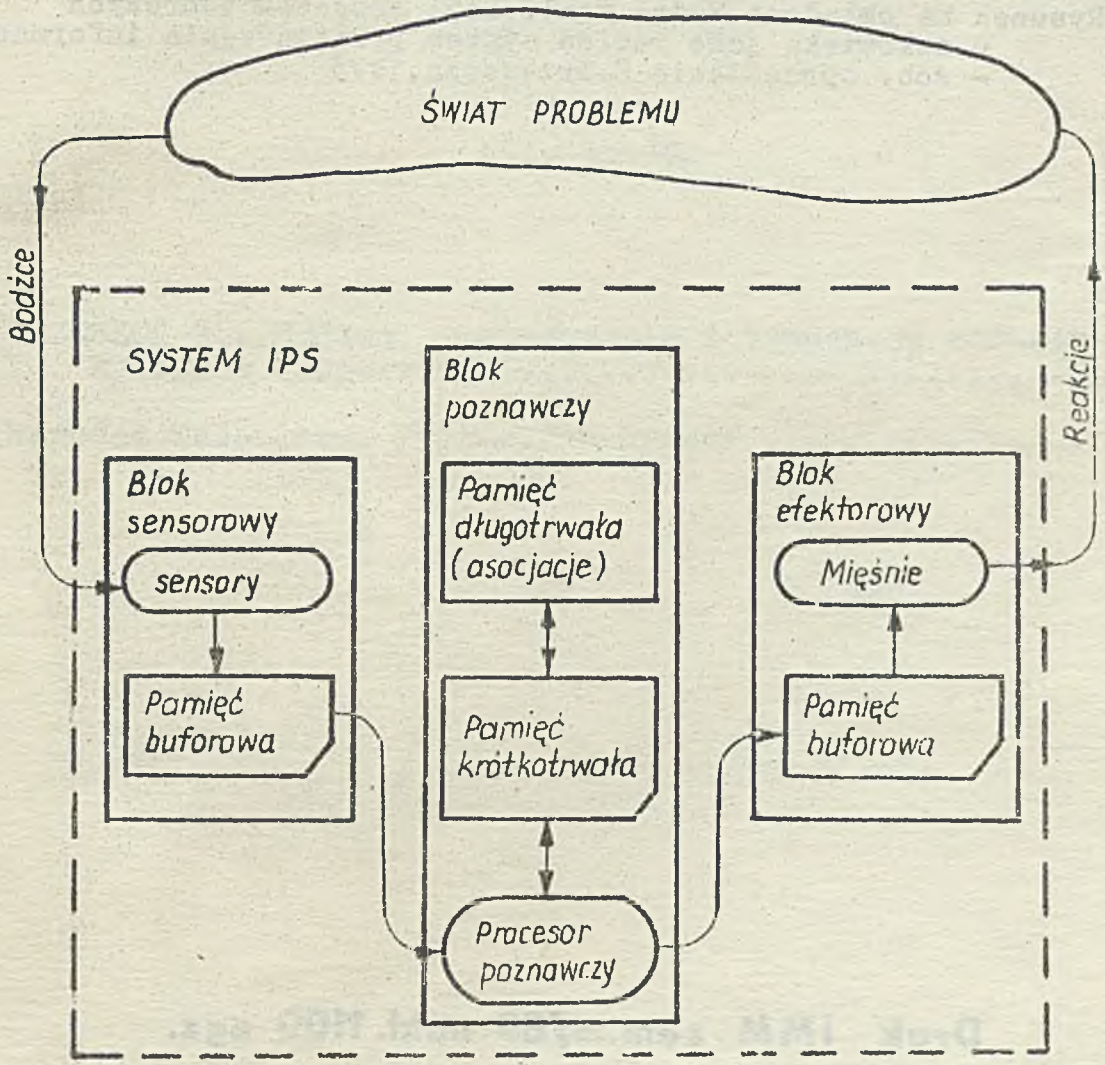
P. 3057/89

techniki komputerowe

4 '89

BIULETYN INFORMACYJNY



INSTYTUT MASZYN MATEMATYCZNYCH
BRANŻOWY OŚRODEK INTE

Rysunek na okładce: Model realizacji procesów twórczych
u człowieka jako pewien system przetwarzania informacji
- zob. opracowanie Z.Wrzeszcza, s.3

Druk IMM zam.26/89 nakł.1100 egz.

TECHNIKI KOMPUTEROWE

Rok XXVII Nr 4 1989

Spis treści

str.

WRZESZCZ Z.: Budowa, zastosowania i tendencje rozwoju systemów eksperckich 3

Nowości Techniczne /oprac.Jan Ryżko/ 57

DWUMIESIĘCZNIK

Wydaje:

INSTYTUT MASZYN MATEMATYCZNYCH
ul. Krzywickiego 34 02-078 WARSZAWA tel. 28-37-29

BRANŻOWY OŚRODEK INFORMACJI NAUKOWEJ, TECHNICZNEJ I EKONOMICZNEJ

Komitet Redakcyjny:

dr inż. Stanisława BONKOWICZ-SITTAUER (redaktor naczelny), mgr Hanna DROZDOWSKA-STRZEMIŃSKA (sekretarz redakcji), mgr inż. Zdzisław GROCHOWSKI, mgr inż. Jan KLIMOWICZ, dr inż. Piotr PERKOWSKI, mgr inż. Romuald SYNAK

doc. dr inż. Z. WRZESZCZ
Instytut Maszyn Matematycznych

Budowa, zastosowania i tendencje rozwoju systemów eksperckich

* Wprowadzenie

Celem niniejszej pracy jest zaprezentowanie systemu eksperckiego; poznamy więc jego budowę, sferę aplikacji oraz główne kierunki rozwoju w krajach, w których te badania są zaawansowane.

System ekspercki jest pewnym programem komputerowym przeznaczonym do automatycznego rozwiązywania problemów trudnych, których rozwiązywanie jest czasochłonne i wymaga wiedzy specjalistycznej, umykającej ujęciu proceduralnemu. Zbudowanie systemu eksperckiego jest równoważne nauczeniu komputera myślenia kategoriami specjalisty z określonej dziedziny.

Systemy eksperckie są pierwszym praktycznym wynikiem badań w dziedzinie sztucznej inteligencji. Początkowo prace nad sztucznym intelektem skupiały się w ośrodkach uniwersyteckich, obecnie prace te podjęto również w przemyśle, a firmy komputerowe IBM i DEC w USA oraz FUJITSU w Japonii należały do pierwszych. Wzrost zainteresowania sztucznym intelektem przez przemysł i administrację państwową znalazł wyraz w programach prac nad tzw. maszynami piątej generacji. Programy takie powstały najpierw w Japonii a następnie w krajach europejskich /W. Brytania, Francja/. Podobne inicjatywy można również odnotować w krajach RWPG.

W początkowym okresie prac nad sztucznym intelektem poszukiwano rozwiązań ogólnych, kładąc nacisk na stworzenie silnego mechanizmu wnioskującego, który pozwoliłby rozwiązywać problemy nawet przy na ogół słabej wiedzy przedmiotowej. Uzyskane tu rezultaty /GPS - General Problem Solver-Newell, Simon,

Shaw, 1957 r./ nie były zadowalające. W następnych latach zmieniono podejście zamieniając rolę obu czynników: mechanizm wnioskujący znacznie uproszczono zaś wiedzę o problemie wybitnie powiększono. Powstał w ten spobób system z bazą wiedzy /IPS - Newell, Simon [2] / stanowiący archetyp dla różnych postaci systemów, związanych z aplikacjami, takimi jak:

- maszynowe tłumaczenie z języka naturalnego,
- analiza i generacja zadań języka naturalnego,
- robotyka,
- analiza obrazów,
- organizacja wielkich baz danych,
- rozwiązywanie problemów matematycznych,
- rozwiązywanie problemów szachowych,
- systemy eksperckie.

Najbardziej rozwijają się obecnie systemy eksperckie. W krajach takich jak Stany Zjednoczone AP, Japonia, W. Brytania powstało wiele firm - producentów komputerowych, które skierowały na rynek swoje produkty w postaci narzędzi do tworzenia systemów eksperckich. Powstały też pierwsze przemysłowe systemy aplikacyjne. Czy systemy eksperckie /ich technologia i zastosowanie/ rozwiną się tak, jak to sobie wyobrażają badacze danej dziedziny, czy nie podzielą losu wielu innych głośnych swego czasu nowości takich, jak układy ze złączem JosephSona, diody tunelowe, pamięci bębenkowe, pamięci drutowe czy maszyny hybrydowe, by wymienić tylko kilka przykładów. Próba odpowiedzi na to pytanie wymaga przede wszystkim dokładnego poznania natury systemu eksperckiego.

* Pojęcia podstawowe

System ekspercki - jak się okaże - jest pewną formą systemu z bazą wiedzy, który z kolei jest konceptualną konstrukcją sztucznego intelektu. Zachodzi zatem potrzeba ustalenia znaczenia występujących tu pojęć, takich jak:

- sztuczna inteligencja, sztuczny intelekt,
- system z bazą wiedzy,
- wiedza i jej reprezentacja.

Sztuczna inteligencja, sztuczny intelekt

W badaniach nad sztuczną inteligencją można wyróżnić dwa podejścia do problemu: w obu występuje umysł ludzki i maszyna licząca /komputer/.

W pierwszym podejściu umysł ludzki jest obiektem badanym zaś maszyna - narzędziem badawczym. W drugim natomiast - komputer jest obiektem badań, zaś umysł ludzki - modelem zachowań komputera.

Podejście pierwsze skupia badaczy z dziedziny psychologii poznawczej i lingwistów, którzy poszukują wyjaśnień dla zachowań intelektu - procesów myślenia u człowieka, natomiast komputer jest tu narzędziem obliczeniowym pozwalającym badać /symulować/ procesy myślenia człowieka.

W podejściu drugim obiektem badań jest komputer zaś umysł ludzki jest modelem, wzorcem do naśladowania. Ten kierunek badań, słuszniej nazywany badaniami nad sztucznym intelektem, stawia sobie za cel stworzenie bardziej inteligentnych komputerów, a procesy myślenia twórczego u człowieka są wzorem dla zastosowań komputera. Z tym kierunkiem badań nad sztuczną inteligencją wiążą się badania nad systemami z bazą wiedzy, w tym nad systemami eksperckimi.

Przyjmując procesy myślenia u człowieka jako model zachowań komputera, zwrócono uwagę przede wszystkim na procesy kreatywne [1] , a w tym na procesy rozwiązywania problemów.

Możemy więc powiedzieć, że pod pojęciem sztucznego intelektu widzimy komputer emulujący procesy kreatywne człowieka w zakresie z góry wyznaczonym.

Model procesów twórczych u człowieka

Ważnym rodzajem procesu twórczego jest rozwiązywanie problemów. Mówimy o problemie, gdy pragnąc czegoś nie wiemy, jak to osiągnąć. Przedmiot naszych pragnień może mieć postać materialną /np. jabłko na wysokim drzewie/ lub abstrakcyjną /np. dowód twierdzenia/. Podejmując działalność prowadzącą do

osiągnięcia celu wykonujemy wiele akcji, z których część ma charakter fizyczny /np. przemieszczenie się osobnika rozwiązującego problem, branie do ręki przedmiotów, przemieszczanie przedmiotów itd./, część zaś odnosi się do organów percepcji osobnika /patrzenie, słuchanie/; trzeci rodzaj to akcje umysłu, a więc porównywanie elementów sytuacji, pamiętanie tych elementów, pamiętanie relacji między elementami itd.

Podjęcie akcji trzeciego rodzaju może być poprzedzone przez akcje drugiego rodzaju: osobnik rozwiązujący problem musi przyjąć do wiadomości sytuację początkową, zapamiętać tę sytuację, rozpoznać problem i podjąć decyzję o działaniu prowadzącym do celu, oddziałać na sytuację problemową tak, aby ten cel osiągnąć lub do niego się przybliżyć.

Powyższy opis pozwala nam wyróżnić pewne istotne zdolności systemu inteligentnego. Są to:

- odbiór bodźców otoczenia,
- wnioskowanie,
- interpretowanie relacji między faktami /obiektami itp/.
- interpretowanie znaczenia faktu,
- rozpoznanie prawdy stwierdzeń,
- uczenie się,
- korygowanie swoich działań na podstawie przeszłych doświadczeń,
- adaptacja do nowej sytuacji.

Powstaje więc pytanie, jak zbudowany jest system o takich zdolnościach?

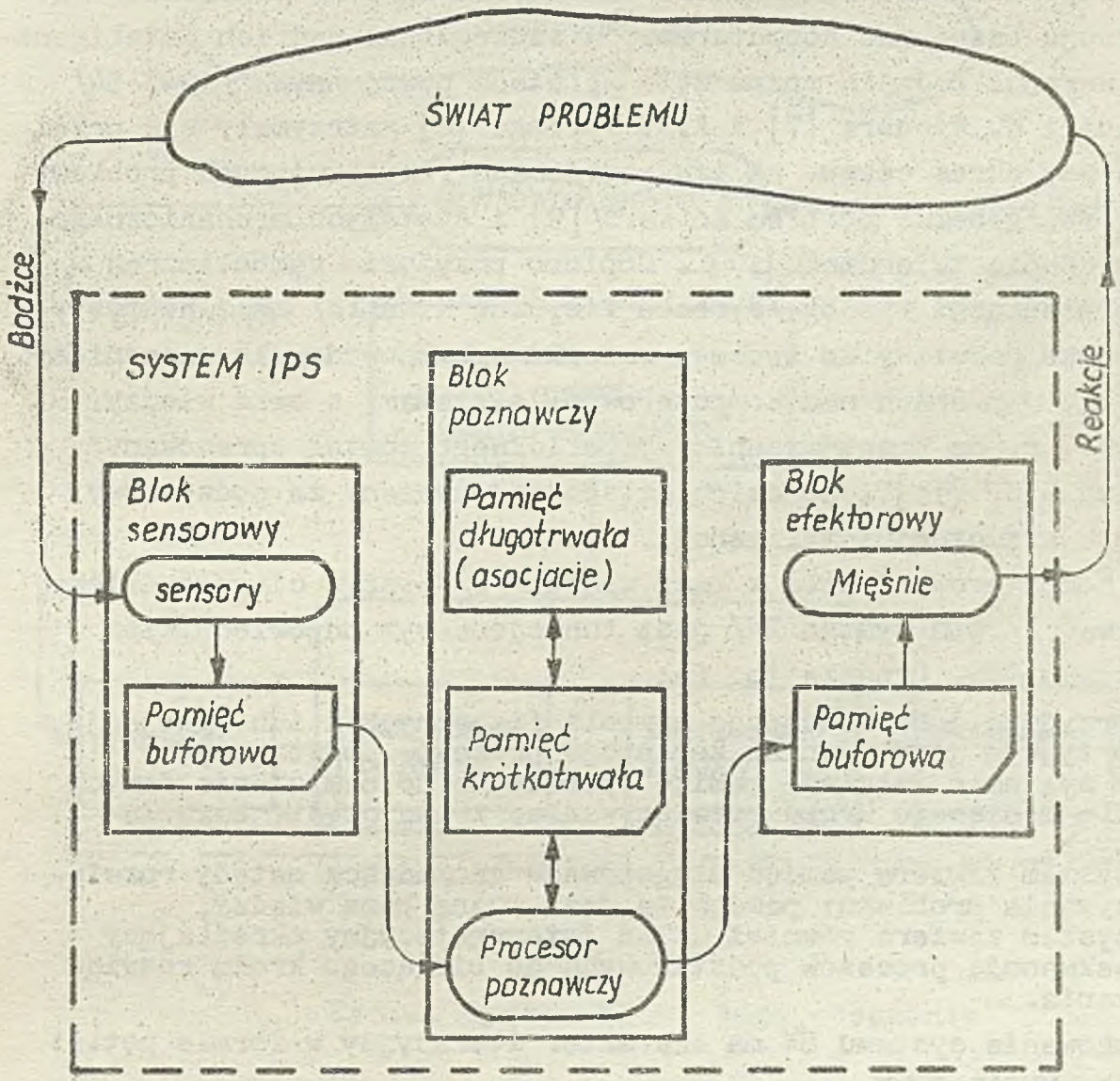
Zgodnie z hipotezą A. Newella i H. Simona [2] człowiek rozwiązując problem, działa jak pewien system przetwarzania informacji, /intelligent processing system - IPS/. System IPS zawiera następujące składniki [3] :

- receptory bodźców zewnętrznych,
- zbiór symboli fizycznych - każdy element tego zbioru jest reprezentantem obiektu ze świata problemu; mogą to być obiekty fizyczne i relacje między tymi obiektami podporządkowane określonym procesom fizycznym lub obiekty abstrakcyjne i związane z nimi zależności: symbol fizyczny jest odbiorem obiektu ze świata problemu, powstający w pamięci IPS w wyniku odebranego bodźca,
- symbole fizyczne wraz z relacjami tworzą pewne asocjacje pamiętane przez pamięć długotrwałą,
- stały i ograniczony zbiór procesów podstawowych przeznaczonych do
  - kreacji nowych symboli fizycznych,

- porównywania symboli fizycznych,
- modyfikacji symboli fizycznych,
- wprowadzania nowych asocjacji,
- kierowania asocjacji do otoczenia systemu,

- pamięć krótkotrwałą do reprezentatywnego ujęcia problemu i pamiętania bieżącego,
- pamięć długotrwałą do pamiętania asocjacji fizycznych - metod rozwiązywania problemu,
- układ interpretacyjny określający sekwencję procesów podstawowych.

Reasumując, obraz systemu IPS można przedstawić jak na rys. 1.



Rys. 1. Model realizacji procesów twórczych u człowieka jako pewien system przetwarzania informacji

osiągnięcia celu wykonujemy wiele akcji, z których część ma charakter fizyczny /np. przemieszczenie się osobnika rozwiązującego problem, branie do ręki przedmiotów, przemieszczanie przedmiotów itd./, część zaś odnosi się do organów percepcji osobnika /patrzenie, słuchanie/; trzeci rodzaj to akcje umysłu, a więc porównywanie elementów sytuacji, pamiętanie tych elementów, pamiętanie relacji między elementami itd.

Podjęcie akcji trzeciego rodzaju może być poprzedzone przez akcje drugiego rodzaju: osobnik rozwiązujący problem musi przyjąć do wiadomości sytuację początkową, zapamiętać tę sytuację, rozpoznać problem i podjąć decyzję o działaniu prowadzącym do celu, oddziałać na sytuację problemową tak, aby ten cel osiągnąć lub do niego się przybliżyć.

Powyższy opis pozwala nam wyróżnić pewne istotne zdolności systemu inteligentnego. Są to:

- odbiór bodźców otoczenia,
- wnioskowanie,
- interpretowanie relacji między faktami /obiektami itp/.
- interpretowanie znaczenia faktu,
- rozpoznanie prawdy stwierdzeń,
- uczenie się,
- korygowanie swoich działań na podstawie przeszłych doświadczeń,
- adaptacja do nowej sytuacji.

Powstaje więc pytanie, jak zbudowany jest system o takich zdolnościach?

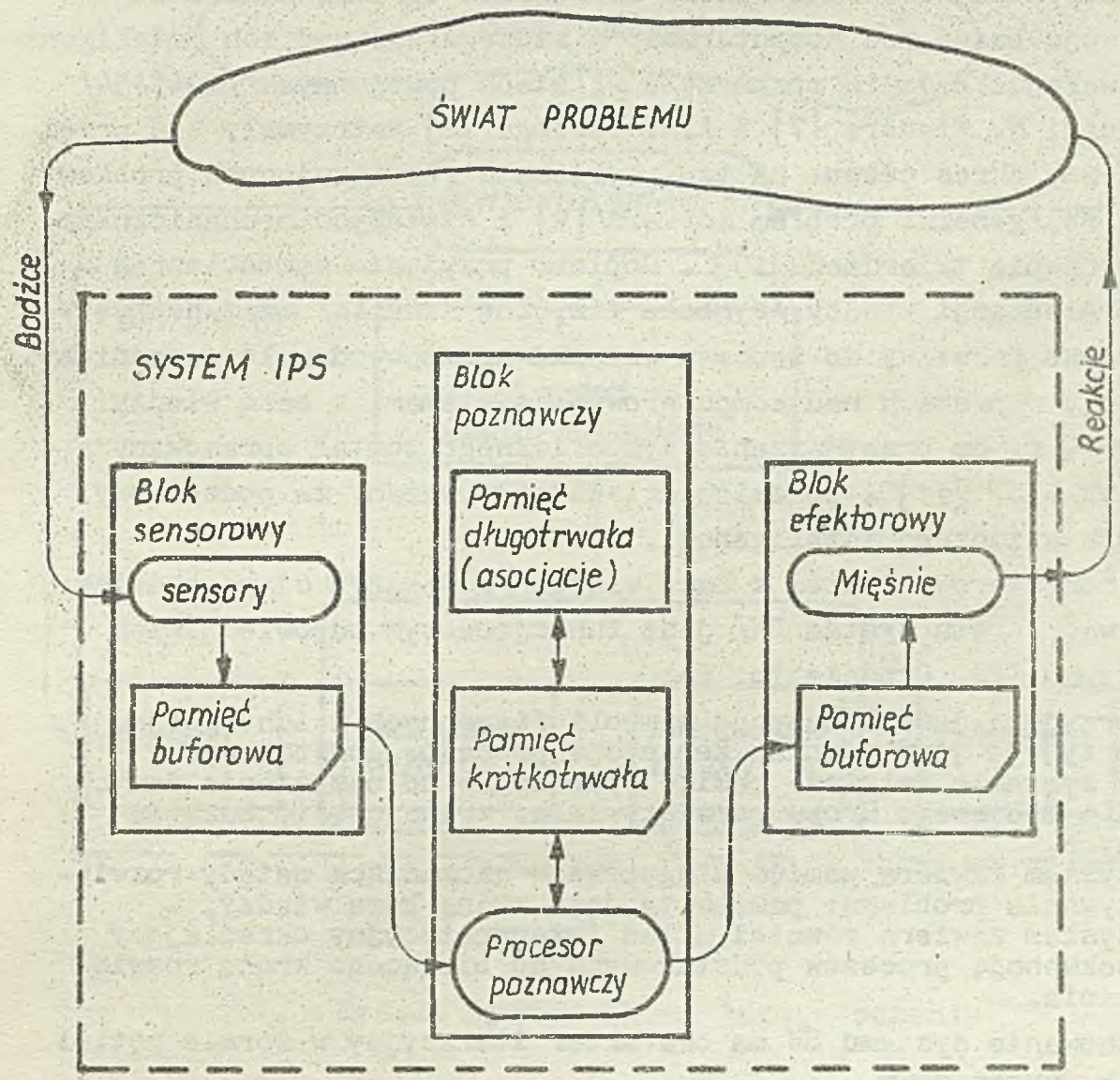
Zgodnie z hipotezą A. Newella i H. Simona [2] człowiek rozwiązując problem, działa jak pewien system przetwarzania informacji, /intelligent processing system - IPS/. System IPS zawiera następujące składniki [3] :

- receptory bodźców zewnętrznych,
- zbiór symboli fizycznych - każdy element tego zbioru jest reprezentantem obiektu ze świata problemu; mogą to być obiekty fizyczne i relacje między tymi obiektami podporządkowane określonym procesom fizycznym lub obiekty abstrakcyjne i związane z nimi zależności: symbol fizyczny jest odbiorem obiektu ze świata problemu, powstający w pamięci IPS w wyniku odebranego bodźca,
- symbole fizyczne wraz z relacjami tworzą pewne asocjacje pamiętane przez pamięć długotrwałą,
- stały i ograniczony zbiór procesów podstawowych przeznaczonych do
  - kreacji nowych symboli fizycznych,

- porównywania symboli fizycznych,
- modyfikacji symboli fizycznych,
- wprowadzania nowych asocjacji,
- kierowania asocjacji do otoczenia systemu,

- pamięć krótkotrwałą do reprezentatywnego ujęcia problemu i pamiętania bieżącego,
- pamięć długotrwałą do pamiętania asocjacji fizycznych - metod rozwiązywania problemu,
- układ interpretacyjny określający sekwencję procesów podstawowych.

Reasumując, obraz systemu IPS można przedstawić jak na rys. 1.



Rys. 1. Model realizacji procesów twórczych u człowieka jako pewien system przetwarzania informacji

Zaproponowany przez Newella i Simona model mechanizmów myślenia twórczego u człowieka został zaakceptowany przez środowisko nauk poznawczych i przyczynił się do znacznego postępu w badaniach czynności myślenia u człowieka [4, 5, 6].

System z bazą wiedzy

Hipoteza Newella i Simona ma również ogromne znaczenie dla rozwoju badań nad komputerami, a szczególnie nad ich inteligentną wersją. Badania rozpoczęte w latach powojennych /1945-54/ pracami N. Wienera [7] i A.M. Turinga [8] zatrzymały się przez znaczny okres czasu na tzw. systemach rozwiązujących problemy ogólne /general problem solvers/ [9] i systemych mechanicznego dowodzenia twierdzeń [10]. Dopiero przyjęcie symbolicznej reprezentacji wiedzy /symbole fizyczne Newella/ oraz heurystyki jako podstawy do procesu wnioskowego spowodowało zasadniczy postęp w pracach nad komputerowymi systemami z bazą wiedzy. W 1960 r. do przetwarzania symbolicznego został opracowany język LISP [11], po dzień dzisiejszy uważany za podstawowy język sztucznej inteligencji.

Komputerowy system z bazą wiedzy /w dalszym ciągu będziemy używać skrótu system BW/ jest funkcjonalnym odpowiednikiem systemu IPS. Oznacza to, że:

- przyjęta jest koncepcja symboli fizycznych i ich asocjacji,
- przyjęta jest również koncepcja procesów podstawowych,
- w systemie istnieje pamięć operacyjna do pamiętania danych dla bieżącego kroku rozwiązywania, zwana często bazą danych,
- system zawiera pamięć długotrwałą gromadzącą metody rozwiązywania problemu: pamięć ta jest zwana bazą wiedzy,
- system zawiera również układ interpretacyjny określający sekwencję procesów podstawowych do bieżącego kroku rozwiązania.

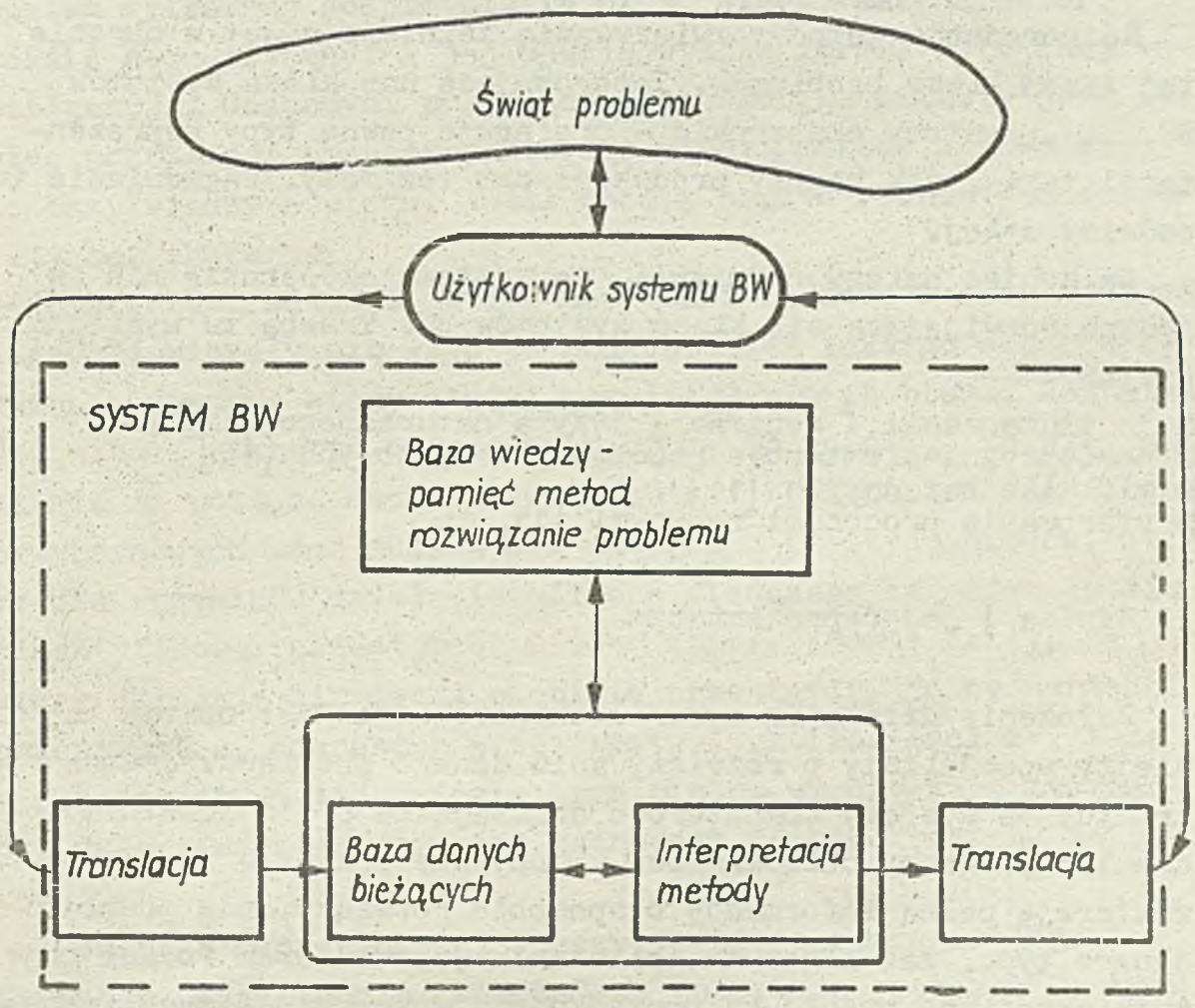
Zachowanie systemu BW ma charakter iteracyjny w formie pętli:

(1) WYBOR CELU

(2) WYBOR METODY ROZWIĄZANIA,
ZASTOSOWANIE WYBRANEJ METODY

(3) OCENA WYNIKU ROZWIĄZANIA,
POWROT DO (1)

W miejsce bloku percepcji i bloku efektorowego, w systemie BW wprowadza się blok translacji wejściowej - tłumaczenie z języka użytkownika systemu BW na język wewnętrzny systemu BW i blok translacji wyjściowej - tłumaczenie w kierunku przeciwnym. Strukturę systemu BW pokazujemy na rys. 2.



Rys.2. System BW i jego otoczenie

Osobnego omówienia wymagają metody rozwiązywania problemu. Bez ich przedstawienia działanie systemu jest niezrozumiałe. Jednak metody rozwiązywania są dość zróżnicowane i zależą ściśle od typu reprezentacji wiedzy, zaś typy reprezentacji wiedzy, mimo że odpowiadają paradygmatowi symbolu fizycznego, są zależne od specyfiki problemu.

Różnorodność metod rozwiązywania istnieje nawet w obrębie tej samej klasy problemów. Interesująca nas klasa systemów BW, tzw. systemy eksperckie - preferuje pewne typy reprezentacji takie, jak reguły produkcji czy też ramy. Zagadnienie to omówimy dalej.

Na koniec należy nadmienić, że systemy eksperckie nie są jedyną rozwijającą się klasą systemów BW. Trzeba tu wymienić systemy:

- do tłumaczenia i generacji języka naturalnego [12],
- percepcyjno-efektorowe robotów przemysłowych [13],
- wielkich baz danych [14],
- sterowania procesami realnymi [15, 16].

Wiedza i jej reprezentacja

Założenie działania systemu eksperckiego jest oparte na wiedzy specjalisty o rozwiązywaniu danego problemu. Jednak tradycyjne systemy komputerowe działają na pozór zupełnie podobnie: programy obliczeniowe wykonywane przez te komputery zawierają pełną informację o sposobie rozwiązywania problemu danego typu, zaś algorytm obliczenia jest zbiorem formuł objętych określoną procedurą porządkującą wykonywanie tych formuł. Procedura algorytmiczna powoduje, że rozwiązywanie problemu w maszynie tradycyjnej jest wykonywane przez skończoną lub nawet określoną liczbę kroków i nie wiąże się z metodą prób i błędów. Metoda jest precyzyjna, a rozwiązanie musi istnieć zawsze i osiągać założoną /przez środki/ precyzję.

Wiedza o problemie w systemie BW /w tym w systemie eksperckim/ nie ma takiego charakteru, wiedza ta ma charakter heurystyczny, opiera się na prostych regułach i tzw. chwytach, często niepewnych. Problemy, które zawierają niepewność, niejednoznaczność, a ich rozwiązywanie nie może być wyrażone w formie

algorytmu matematycznego, kwalifikują się jako przedmiot działania maszyn sztucznej inteligencji, a nie komputerów klasycznych. Chcąc uzmysłowić różnicę pomiędzy wiedzą pierwszego i drugiego typu wyobraźmy sobie problem smażenia kotleta. Przepis można znaleźć w każdej książce kucharskiej: jest to wiedza podlegająca algorytmizacji. Jednak usmażenie smacznego kotleta wymaga doświadczenia, popartego dobrym gustem wytrawnej gospodyni. Ta część wiedzy ma charakter ściśle heurystyczny, a jej ujęcie wymaga formuł sztucznej inteligencji. Gospodyni przyrządzająca smaczne kotlety jest więc przykładem eksperta, którego wiedzę możemy wprowadzić do bazy wiedzy systemu, jeśli chcemy smażyć równie dobre kotlety bez eksperta.

Pierwsze trudności, które napotkali badacze sztucznej inteligencji wiązały się więc ze znalezieniem takiego sposobu wyrażenia wiedzy, który ułatwi rozwiązywanie problemu. Rozpatrując wiedzę eksperta często dochodzimy do wniosku, że wiedza ta nie ma postaci skomplikowanych idei o głębokich związkach przyczynowych lecz składa się na ogół z prostych bloków. Weźmy dla przykładu działanie oficera śledczego starającego się zidentyfikować przestępcę. Oficer zbiera proste fakty, na które składają się cechy osobiste przestępcy, cechy narzędzi przestępstwa, charakter przestępstwa, okoliczności związane z przestępstwem itd. Liczba tych bloków wiedzy może być bardzo duża i mogą między nimi zachodzić różnego rodzaju zależności.

Mówimy o reprezentacji wiedzy, a nie o reprezentacji informacji, gdyż termin "wiedza" ma znaczenie szersze. O ile w konwencjonalnych bazach danych gromadzimy dane prostych typów /np. liczby, sznury, wartości booleowskie/, to w badaniach nad sztucznym intelektem możemy posługiwać się bardziej złożoną informacją w postaci procesów, procedur, akcji, związków przyczynowych, związków czasowych, motywacji, celów, przekonań i innych konstrukcji pojęciowych związanych z tzw. zdrowym rozsądkiem*/, których zapis w postaci konwencjonal-

---

* "kotlet należy kłaść na patelnię a nie patelnię na kotlet"

nej struktury danych bywa bardzo trudny.

Feigenbaum w swojej pracy [17] wprowadza pewne kategorie, pod kątem których należy wiedzę rozpatrywać. Są to mianowicie:

- obiekt, obiekty: każdy system reprezentacji musi mieć możność zakodowania informacji o właściwościach obiektów fizycznych i konceptualnych,
- zdarzenia: /events/ kategoria obejmuje akcje i zdarzenia zachodzące w świecie; pozwala określić w sposób ogólny element czasu; ukazuje związki przyczynowo-skutkowe,
- wykonanie, realizacja /performance/: kategoria zawiera informację o tym, jak wykonać określone zadanie, np.: jak prowadzić samochód, sformułować zdanie w określonym języku, jak przeprowadzić dowód twierdzenia,
- meta-wiedza /meta-knowledge/: jest to wiedza o wiedzy podanej reprezentacji, o jej rozmiarze i pochodzeniu, o niezawodności tej wiedzy, o względnym znaczeniu niektórych faktów i o ocenie działania nas samych jako procesorów poznawczych; w systemach z bazą wiedzy w formie reguł produkcji meta-wiedza jest wyrażana w postaci reguł decydujących o użyciu innych reguł, np. kwestia pierwszeństwa reguł.

Innym ważnym aspektem istotnym dla reprezentacji wiedzy jest posługiwanie się wiedzą. Sposoby operowania wiedzą mogą być różne, stosownie do systemu aplikacyjnego /np. system diagnostyczny z bazą danych, system ekspercki sterujący, system wizyjny robota/. Jednak w każdym wypadku można wyróżnić trzy etapy posługiwania się wiedzą. Są to: akwizycja wiedzy, odzyskiwanie /retrieval/, wnioskowanie /reasoning/.

Akwizycja polega na wprowadzeniu nowej informacji do systemu BW. Można tu mówić o wprowadzeniu wiedzy pierwotnej, /np. w formie faktów/ jak też i o akwizycji wiedzy na wyższym poziomie - przez rozszerzanie wiedzy już istniejącej w bazie. Ten typ akwizycji przypomina funkcję uczenia się człowieka.

Odzyskiwanie określonego faktu lub zależności z bazy wiedzy wymaga sortowania tej bazy, a jej zawartość może być bardzo duża. Odzyskiwanie podobne jest do funkcji przypominania, chociaż człowiek dokonuje tej czynności instynktownie i bardzo efektywnie.

Wnioskowanie w systemie BW może być realizowane z zastosowaniem różnych narzędzi, o różnej skuteczności. Najwcześniej stosowane wnioskowanie o charakterze formalnym było realizowane na podstawie reguły logicznego rachunku zdań. Następnie sposób ten rozwinął R. Kowalski [18] do postaci rachunku lo-

gicznego specjalnie przystosowanego do automatycznego rozwiązywania problemów. Następny rodzaj techniki wnioskowania opiera się na regułach produkcji, które osobno omówimy w rozdziale "Zasady działania systemu eksperckiego". Jest to sposób szeroko stosowany w praktyce systemów BW. Wnioskowanie przez analogię oraz wnioskowanie przez generalizację i abstrahowanie to kolejne sposoby stosowane przez człowieka. Sposoby te, należące do wnioskowania typu indukcyjnego, są trudne do implementacji maszynowej. Sposoby wnioskowania zostały tu wymienione w porządku ich skuteczności.

* Cechy ogólne systemu eksperckiego

Wiemy już, że system ekspercki jest szczególną formą systemu BW. Jego specyfika polega na tym, że system ten odwzorowuje zachowanie eksperta rozwiązującego problemy określonej klasy. Ekspert dysponuje dwoma rodzajami wiedzy [19, 20, 21]:

- wiedzą ogólnie dostępną, na którą składają się formuły, teorie, normy, ustalenia projektowe i realizacyjne oraz inne podobne treści dostępne przez publikacje, dokumenty, nauczanie;
- wiedzę specjalistyczną wyniesioną z doświadczenia, mającą charakter nieformalny, heurystyczny.

Otóż system ekspercki zawiera w swojej pamięci oba te rodzaje wiedzy, a wiedza heurystyczna ma tu znaczenie zasadnicze dla procesu rozwiązywania.

System elspercki gromadzi wiedzę w formie tzw. symboli fizycznych i ich asocjacji, podobnie jak jego prototyp - system BW. Jednak metody poszukiwania rozwiązania przez system ekspercki nie polegają na prostym przeszukiwaniu przestrzeni rozwiązań, co przy bardziej złożonych problemach jest nieefektywne - heurystyczna wiedza eksperta wybitnie przyspiesza proces rozwiązania.

Rodzaje zastosowań

Systemy eksperckie zostały opracowane i zastosowane w praktyce do zagadnień takich jak:

- analiza /np. interpretacja danych [22]/,
- diagnostyka /określenie uszkodzeń w systemach technicznych [23], określenie przyczyny chorób [15]/,
- ciągła interpretacja sygnałów / intensywna terapia [24]/,
- planowanie / programowanie eksperymentu w genetyce molekularnej /[25],
- projektowanie / konstruowanie urządzeń elektronicznych /[26],
- nauczanie [27],

Powyższe wyliczenie nie wyczerpuje możliwości aplikacyjnych systemu eksperckiego [28, 16, 29].

Odwzorowanie funkcji eksperta [30, 31, 32, 19, 33]

Zakładamy, że system ekspercki odwzorowuje zachowanie człowieka - eksperta określonej działalności. Oznacza to, że system ekspercki powinien naśladować wszystkie te funkcje, które ekspert realizuje rozwiązując problem z danej dziedziny. Rozpatrzmy więc te funkcje.

/1/ Wnioskowanie. Fakty dotyczące problemu oraz wiedza o metodach rozwiązywania podobnych problemów służą ekspertowi w procesie tworzenia rozwiązania konkretnego problemu.

Funkcja ta ma znaczenie podstawowe, lecz ekspert musi realizować również inne, pomocnicze funkcje.

/2/ Żądanie i przyjmowanie nowych danych o problemie.

/3/ Wyjaśnienie swojego postępowania w procesie wnioskowym.

Oprócz funkcji /1/, /2/, /3/ ekspert może realizować:

/4/ dynamiczną reorganizację bazy wiedzy stosownie do specyfiki problemu,

/5/ łamanie przyjętych reguł wnioskowych,

/6/ określenie wcześniejsze związków między konkretnymi problemami z wiedzą eksperta: problem może być podejmowany jedynie wówczas, gdy mieści się w profilu wiedzy eksperta,

/7/ określenie rozwiązań przybliżonych wtedy, gdy problem mieści się na granicy możliwości wiedzy eksperta.

Implementacja funkcji /1/ w systemie eksperckim może mieć różne formy, zależnie od przyjętej metody reprezentacji wiedzy. Funkcja /2/ stanowi ważne uzupełnienie dla funkcji /1/, zwłaszcza na etapie budowy systemu prototypowego, a także w wypadku rozszerzeń możliwości użytkowych systemu eksperckiego. Funkcja /3/ ma charakter pomocniczy, jednak może mieć znaczny wpływ na prawidłowość reakcji użytkownika w dialogu z systemem za pomocą funkcji /2/ i /3/.

Współczesna realizacja systemów eksperckich nie uwzględnia

funkcji /4/ - /7/, bądź też uwzględnia je jedynie częściowo. Przyczyna leży jak sądzimy w niedostatecznej znajomości podstaw ich implementacji.

Zróżnicowanie implementacji funkcji eksperta może wynikać również z rodzaju zastosowania, na przykład systemy eksperckie wspomagające decyzję dotyczącą zachowania urządzeń w czasie realnym wymagają dodatkowo funkcji:

/8/ przyjmowania sygnałów z procesów świata realnego,
/9/ wysyłania sygnałów do procesów świata realnego.

✻ ZASADY DZIAŁANIA SYSTEMU EKSPERCKIEGO

Działanie systemu eksperckiego silnie wiąże się z wybraną techniką reprezentacji wiedzy. Do najbardziej popularnych technik należą:

- reprezentacja wiedzy o problemie jako pewien system, w którym baza jest zbiorem reguł produkcji,
- pewne integralne ujęcie zwane ramą,
- sieci semantyczne.

Na czele stawiamy reprezentację w formie systemu reguł produkcji /system BWP / ze względu na prostotę jego koncepcji a przez to łatwość programowania. Opisy pozostałych technik można znaleźć m.in. w opracowaniu "Podstawy implementacji systemów eksperckich..." Z. Wrzeszcza i in. [34] . System BWP jest konsekwencją systemu IPS Newella i Simona; z systemem IPS zaznajomiliśmy się w rozdziale "Pojęcia podstawowe".

Reguła produkcji i schemat podstawowy

Podstawową /i jedyną/ strukturą reprezentacji wiedzy w systemie BWP jest reguła produkcji

jeśli < warunek > to < akcja >

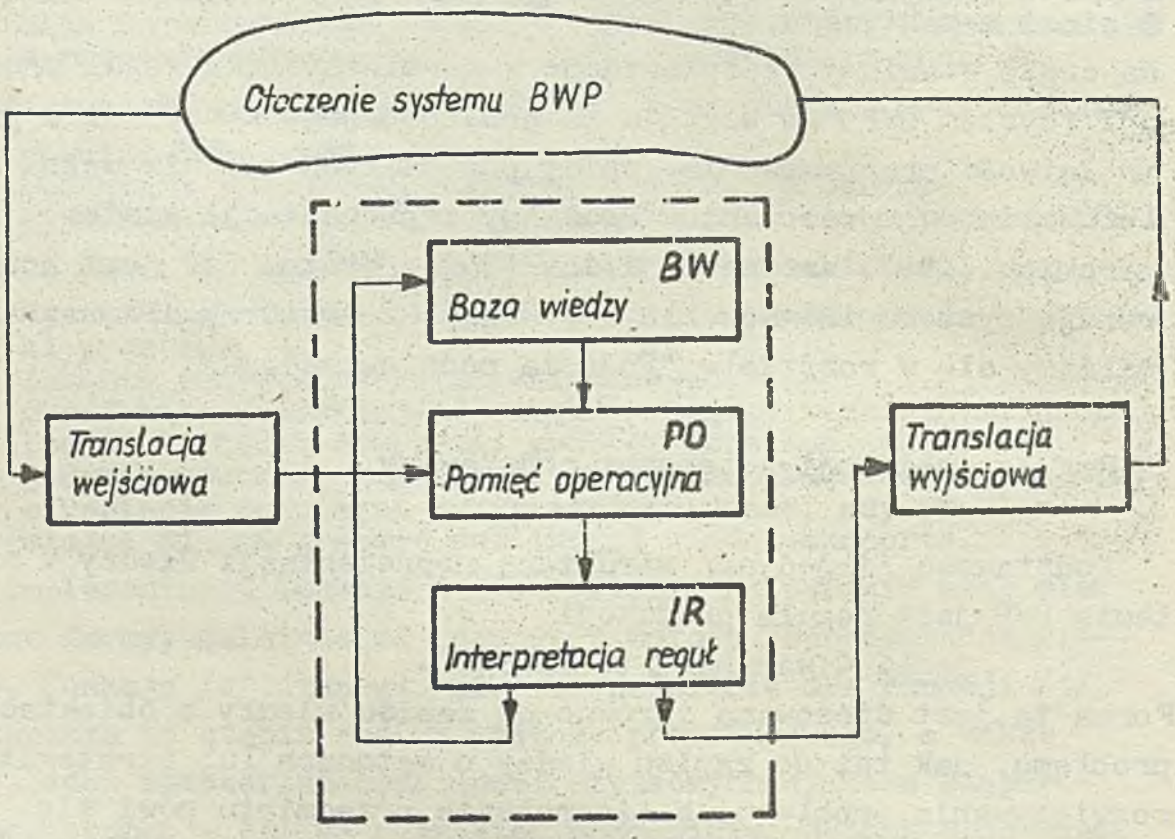
Forma ta jest stosowana zarówno do zapisu wiedzy o obiektach problemu, jak też do zapisu wiedzy o metodach lub strategii rozwiązywania problemu. W literaturze przedmiotu mówi się często o lewej i prawej stronie reguły produkcji. Można też spotkać inne rodzaje wyrażeń przypisywanych lewej i prawej stronie reguł, lecz najczęściej spotykane przypadki to:

< warunek >,< akcja >
< przesłanka >,< wniosek >
< sytuacja >,< akcja >

Zarówno lewa jak i prawa strona reguły może zawierać jedno lub więcej zdań zespolonych łącznikiem logicznym. A oto przykłady z życia codziennego:

Przykład 1		Przykład 2	
jeśli	ból_głowy	jeśli	jest_pochmurno
i	drapanie_w_gardle	i	barometr_spada
i	siąkanie_nosem	i	zła_prognoza_w_TV
to	weź_aspirynę	to	weź_parasol

Zadziałanie reguły nie jest badane przez oderwane badanie prawdziwości lewej strony lecz w kontekście stanu zawartego w tzw. pamięci operacyjnej, która jest jednym z trzech głównych układów w systemie BWP, przedstawionym na rys. 3.



Rys. 3. Schemat działania systemu BWP

PO: Pamięć operacyjna - zawiera wybór struktur danych reprezentujących stan systemu BWP równoważny stanowi rozwiązania

danego problemu.

BW:- Baza wiedzy - pamięć zawierająca zbiór struktur danych reprezentujących reguły produkcji dla danego problemu.

IR: Interpretator reguł - układ, który odnosi reguły do zawartości pamięci operacyjnej i realizuje wyrażenie prawej strony reguły. Układ ten pełni w systemie funkcję podobną do systemu operacyjnego w maszynie. IR cyklicznie śledzi reguły, których lewa strona jest dopasowana /matching/ do elementów PO. W każdym cyklu układ IR wybiera regułę do realizacji.

Wzór, stałe i zmienne wzoru

Dopasowanie do wzoru należy do skutecznych technik poszukiwania rozwiązań problemów [12] i było zastosowane w licznych aplikacjach np. program SIR [35] - odwzorowanie języka naturalnego w formę łatwą do interpretacji przez komputer, przeszukiwanie sieci semantycznej [36] , języka PROLOG [37] .

Ogólnie mówiąc metoda dopasowania do wzoru polega na poszukiwaniu takich elementów zawartych w bazie wiedzy, które pasują do określonego wzoru. Jako wzór traktowana jest lewa strona reguły produkcji / w pewnych wypadkach może to być strona prawa / . Mówimy o stałej wzoru lub zmiennej wzoru /pattern constant, pattern variable/. Przykładami stałych były elementy występujące w regułach z przykładu 1 i 2.

Zmienne wzoru mają zasadnicze znaczenie dla sprawnego działania systemu BWP. Powróćmy do przykładu o pogodzie. Dana jest następująca reguła

jeśli pogoda X
i prognoza Y
to samopoczucie Y

Gdy w pamięci operacyjnej PO znajduje się:

pogoda zła
prognoza dobra

to przy czynności dopasowania w pierwszym elemencie wzoru zmiennej X dowiązujemy wartość: "zła" zaś w drugim elemencie zmiennej Y wartość "dobra". Te same wartości jednak przyjmują zmienne prawej strony: w tym wypadku zmienna Y w elemencie samopoczucie. Tak więc reguła odnosi się do człowieka o uspo-

sobieniu optymistycznym. Pesymista natomiast to

jeśli pogoda X
i prognoza Y
to samopoczucie X

Istotą obu przykładów jest to, że wiążąc określoną wartość z bazy z określonym elementem wzoru lewej strony, czynimy to również wszędzie tam, gdzie symbol ten występuje, jednak tylko w danym kroku rozwiązania.

Funkcje wykonawcze i przykład działania systemu BWP

Oprócz podstawowej funkcji dopasowanie do wzoru, układ IR musi realizować liczne funkcje pomocnicze umożliwiające systemowi BWP działanie automatyczne.

Weźmy pod uwagę następującą regułę

Reguła 10: jesli (zadanie pisz-litery)
i NIE-ISTNIEJE (wykonano x)
to WYKONAJ (pisz X)
i DODAJ (wykonano X)

Reguła 10 ma już bogatą formę. Mianowicie występują w niej zmienne X, funkcja NIE-ISTNIEJE ( ), funkcja WYKONAJ ( ) itd. Są to elementy umożliwiające działanie systemu BWP.

Ogólnie biorąc, w regułach systemu BWP będą występować

- zmienne , które można powiązać z jednym z elementów w PO
- funkcje działające na pamięć PO, tj.
  DODAJ ( ) która dodaje do PO element określony przez argument funkcji,
  USUN ( ) która usuwa element z PO,
  NIE ISTNIEJE ( ) , która wprowadza istnienie w PO określonego elementu
- funkcje sprzęgu z otoczeniem systemu BWP:
  WPROWADŹ( ), która wprowadza do PO element
  WYKONAJ ( ), która kieruje do otoczenia systemu określone polecenie do wykonania

Rozpatrzmy działanie systemu BWP zawierającego trzy reguły:

Reguła 1: jeśli (proces wydruk - operacji)
i (operacja X)
i NIE-ISTNIEJE (wykonano X)
to WYKONAJ (drukuj X)
i DODAJ (wykonano X)

Reguła 2: jeśli NIE-ISTNIEJE (proces wydruk-operacji)
to WPROWADŹ (proces wydruk-operacji)

Reguła 3: jeśli (proces wydruk-operacji)
to USUŃ (proces wydruk-operacji)
i WYKONAJ (drukuj zakończono wydruk)

/1/ W pamięci operacyjnej systemu znajdują się następujące elementy:

(operacja sprawdź - temperaturę)
(operacja włącz - pompę)

Działanie systemu BWP przedstawia się teraz następująco.

/2/ Po uruchomieniu systemu układ IR pobudza regułę 1, a nie znajdując zgodności z żadnym z elementów w PO przechodzi do reguły 2. Reguła 2 zostaje pobudzona. Nie udaje się jednak pobudzić żadnej innej reguły więc lista reguł pobudzonych zawiera tylko regułę 2. Układ IR powoduje wykonanie prawej strony reguły 2, tj. wprowadza do PO nowy element i wprowadza regułę 2 w stan pasywny. Teraz PO zawiera następujące elementy:

(proces wydruk - operacji)
(operacja sprawdź - temperaturę)
(operacja włącz - pompę)

/3/ Po wykonaniu prawej strony reguły aktywnej, a następnie po wprowadzeniu jej w stan pasywny /dotyczy to wszystkich reguł aktywnych, jeśli ich liczba była większa niż jeden/ układ IR powraca do reguły 1. Została pobudzona reguła 1 i reguła 3 zaś reguła 2 pozostaje pasywna. Układ IR powoduje więc wykonanie prawej strony reguły 1 przypisując zmiennej X drugi element PO. Wtedy na ekranie monitora zobaczymy napis

(operacja sprawdź - temperaturę)

zaś w PO dojdzie nowy element i zawartość będzie się przedstawiać następująco:

(wykonano sprawdź-temperaturę)
(proces wydruk-operacji)
(operacja sprawdź-temperaturę)
(operacja włącz-pompę)

/4/ Po następnym kroku otrzymamy na monitorze:

(operacja włącz-pompę)

zaś w pamięci operacyjnej znajdzie się:

(wykonano włącz-pompę)
(wykonano sprawdź-temperaturę)
(proces wydruk-operacji)
(operacja sprawdź-temperaturę)
(operacja włącz-pompę)

/5/ Następny cykl nie spowoduje już aktywacji reguły 1 uruchomiona więc będzie reguła 3, która usunie z PO wyrażenie:

(proces wydruk-operacji)

stwarzając tym samym warunki do zatrzymania układu IR.

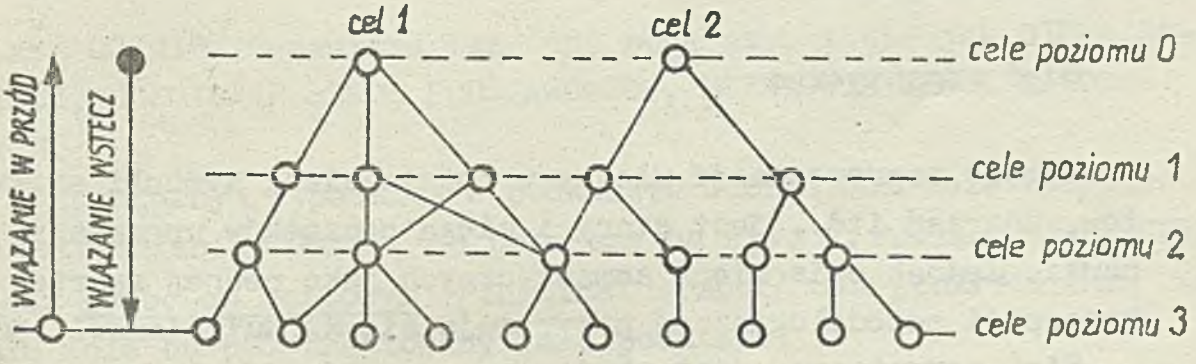
Powinno narzucać się pytanie: dlaczego obok działania reguły 1 nie działa reguła 3, przecież zostanie ona również pobudzona. Dotykamy tu ważnego mechanizmu działania systemu BWP a mianowicie rozstrzyganie sytuacji konfliktowej /conflict resolution/: w każdym cyklu IR może być pobudzona więcej niż jedna reguła. Można przyjąć zasadę jednoczesnego wyzwolenia prawych stron reguł aktywnych, lecz czyni się tak bardzo rzadko. Zamiast tego przyjmuje się określoną strategię rozwiązywania sytuacji konfliktowej, a zbudowany na tej podstawie mechanizm jest częścią układu IR: mechanizm ten wybiera jedną z reguł ze zbioru konfliktowego. Przyjęta w naszym przykładzie strategia polega na tym, że wybierana jest ta reguła ze zbioru konfliktowego, której lewa strona jest nadzbiorem dla pozostałych; oznacza to zarazem regułę najmocniejszą. Strategia wyboru reguły najmocniejszej nie jest strategią jedyną.

System BWP ma zachowanie typu sekwencyjnego, chociaż

nie ma układów takich, jak licznik instrukcji występujących w klasycznej maszynie sekwencyjnej. Stawiając do wykonania kilka zadań nie musimy wcale pisać formuł w rodzaju "wykonaj najpierw zadanie 1, potem 2, potem 3 itd". Natomiast porządek wykonania można wytworzyć przez określoną strategię rozstrzygania sytuacji konfliktowej. Dobór odpowiednich strategii rozstrzygania konfliktu ma więc pierwszorzędne znaczenie dla działania systemów BWP, a w tym systemów eksperckich. Problematyka budowy układów realizacji strategii rozstrzygania konfliktów skupia nadal wielu badaczy w dziedzinie sztucznego intelektu [12].

Sterowanie procesem wnioskowym

Działanie maszyny sztucznej inteligencji, tj. realizacja procesu wnioskowego jest poddane określonej strategii. Na czoło wysuwa się tu strategia związana z kierunkiem prowadzenia wywodu: mówi się mianowicie o "wiązaniu w przód" lub o "wiązaniu wstecz" / forward chaining, backward chaining/. Na rys. 4 przedstawiono pewne drzewo problemowe.



Rys. 4. Przykład drzewa problemowego

Można tu wyróżnić dwa cele końcowe, tj. znajdujące się na poziomie 0 drzewa. Kolejne poziomy 1,2,3 skupiają podcele, zaś żebra grafu podają ich powiązania. A więc np. cel 2 ma dwa podcele - cele poziomu 1, każdy z celów poziomu 1 ma po dwa podcele na poziomie 2, zaś cele poziomu 2 wymagają realizacji

7 podcelów z poziomu 3.

Jeśli w danym problemie cele końcowe są zdefiniowane i występują z reguły w małej liczbie to wybierana jest wtedy strategia wiązania wstecz. Jeśli na odwrót - liczba akceptowalnych rozwiązań jest duża lub też cel musi być dopiero skonstruowany to wybierana jest strategia wiązania w przód.

Przy wiązaniu w przód analizowana jest najpierw lewa strona reguły wnioskowej zaś przy wiązaniu wstecz - strona prawa. Systemy oparte na wiązaniu w przód są czasami nazywane systemami sterowanymi przez dane zaś systemy z wiązaniem wstecz - systemami sterowanymi przez cele; na ogół te ostatnie są uważane za systemy prostsze.

* Inne metody formułowania systemu eksperckiego

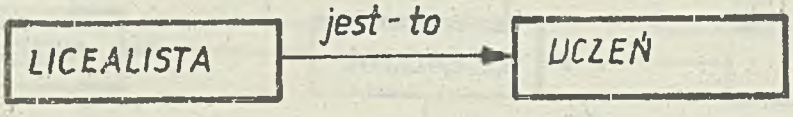
Problemy cechujące się dobrze zdefiniowaną, hierarchiczną strukturą mogą być, z korzyścią dla użytkownika systemu eksperckiego, dogodnie formułowane w postaci sieci semantycznej lub ramy. Reprezentacje te odwzorowują problem w sposób integralny i są szczególnie przydatne w problemach związanych z konstruowaniem.

Sieć semantyczna

Sposób reprezentacji wiedzy w formie sieci symboli obiektów, zdarzeń itd., jest stary i sięga początków uprawiania nauki. Jednak opis sieci semantycznych jako pewnej zwartej koncepcji metodologicznej przypisuje się R. Quillianowi [36]

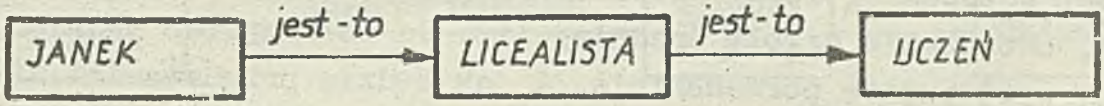
Sieć semantyczna jest zbudowana z węzłów i połączeń między nimi.

- Węzły reprezentują obiekty, koncepcje, sytuacje i są graficznie przedstawione w postaci pudełek, okręgów lub innych figur zamkniętych.
- Połączenia reprezentują zależności pomiędzy węzłami, a ich forma graficzna to strzałki łączące węzły.



Rys. 5. Reprezentacja stwierdzenia "każdy licealista jest uczniem"

Rozpatrzmy następujący przykład /rys. 5/, w którym chcemy stwierdzić, że: "wszyscy licealiści są uczniami" /lub też, "każdy licealista jest uczniem" / . Zwróćmy też uwagę, że reprezentacja, jak na rys. 6 ma charakter hierarchicznej klasyfikacji, co implikuje, że LICEALIŚCI są podklasą klasy bardziej ogólnej, mianowicie klasy UCZNIOWIE.



Rys. 6. Sieć semantyczna reprezentująca fakt, że Janek jest pewnym konkretnym licealistą

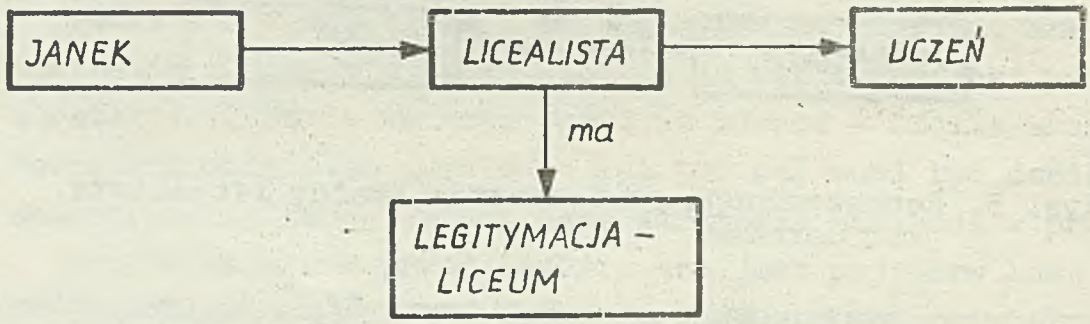
Klasa UCZNIOWIE zawiera też inne podklasy, takie jak GIMNAZJALIŚCI, UCZNIOWIE SZKOŁ PODSTAWOWYCH, UCZNIOWIE SZKOŁ ZAWODOWYCH.

Szczególnym wypadkiem w podklasie LICEALIŚCI może być licealista imieniem Janek, co pokażemy na rys. 7. Powiemy, że JANEK jest symbolem końcowym /terminal symbol/ dla podkreślenia, że nie może on być traktowany jako podklasa.

Sieć semantyczna z rys. 6 powstała przez połączenie dwóch faktów, tj. - licealista jest uczniem i Janek jest licealistą. Ten nowy zapis dostarcza nowego faktu: Janek jest uczniem.

Przykład ten ilustruje bardzo ważną własność sieci semantycznych, tj. dziedziczenie. Własność dziedziczenia umożliwia w systemie opartym na sieciach semantycznych realizację procesu wnioskowego.

Rozpatrzmy przykład ilustrujący dziedziczenie cech obiektu.



Rys. 7. Reprezentacja cechy umożliwiającej wyróżnienie licealisty przez legitymację liceum

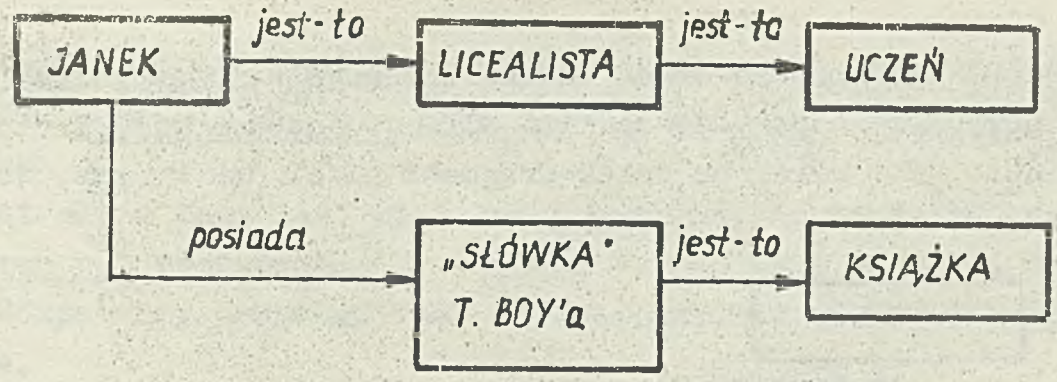
Wprowadzimy w tym celu relację ma. / has/ i zapiszemy /rys. 7./ "Licealista ma legitymację licealną". Możemy w ten sposób rozszerzyć cechy licealisty, wprowadzając przeciętny wiek, ukończenie szkoły średniej itp. Są to wszystko jednak cechy występujące permanentnie. A jak będzie przedstawiać się sieć z cechami czasowymi? Wprowadzamy w tym celu nowy rodzaj połączenia POSIADA. Załóżmy, że nasz licealista Janek ma do wykonania zadania z języka polskiego, musi zaznajomić się z określoną książką. W tym celu Janek udaje się do biblioteki i wypożycza potrzebną książkę na czas określony. Zwróćmy uwagę, że reprezentacja faktu posiadania przez Janka książki w sposób jak to pokazuje rys. 8 jest niewłaściwa, gdyż reakcja POSIADA ma charakter czasowy i Janek najpierw książki nie posiadał, potem wszedł w jej posiadanie i po jakimś czasie znów nie będzie jej posiadał.

Mamy tu przypadek reprezentacji pewnego zdarzenia dotyczącego wejścia w posiadanie czasowe:

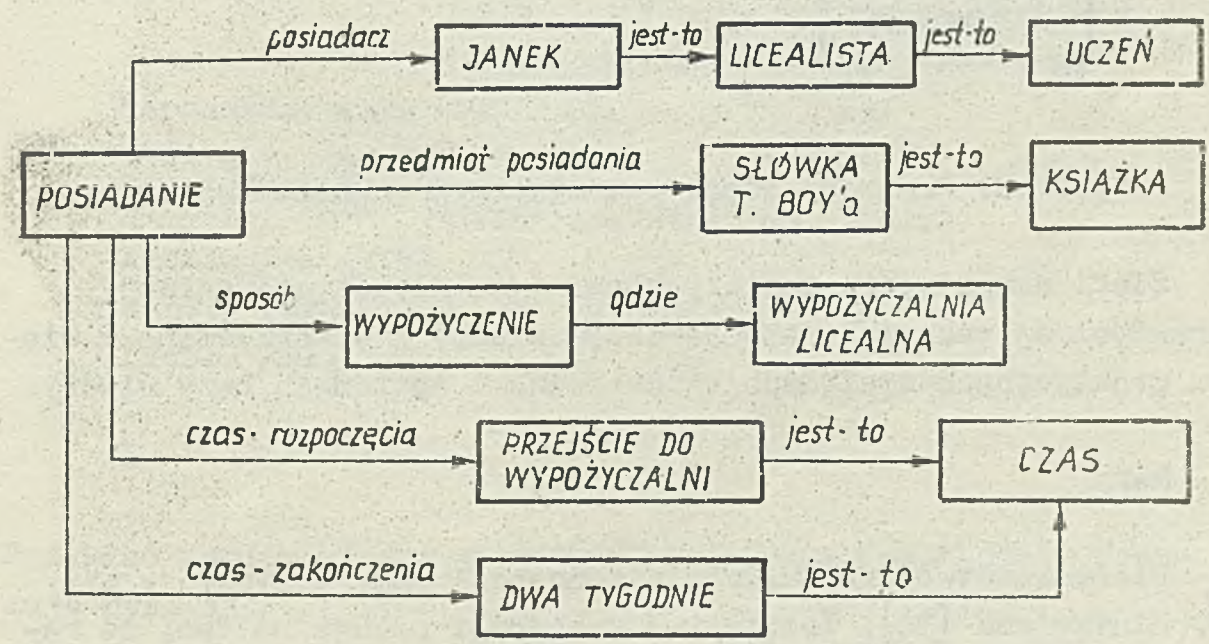
- posiadaczem jest Janek,
- przedmiotem posiadanym jest książka "SŁOWKA",
- czas rozpoczęcia posiadania: przyjście do wypożyczalni,
- czas - zakończenia posiadania: 2 tygodnie.

Całą tę sytuację przedstawia rys. 9.

Jak więc widzimy, konstruowanie schematów reprezentacji przez sieć semantyczną jest na ogół proste i przejrzyste. Korzystanie z zapisów także nie przedstawia większych trudności. Pokażemy to na podstawie zgłoszenia do pewnej bazy,



Rys. 8. Nieprawidłowa reprezentacja faktu czasowego posiadania przedmiotu

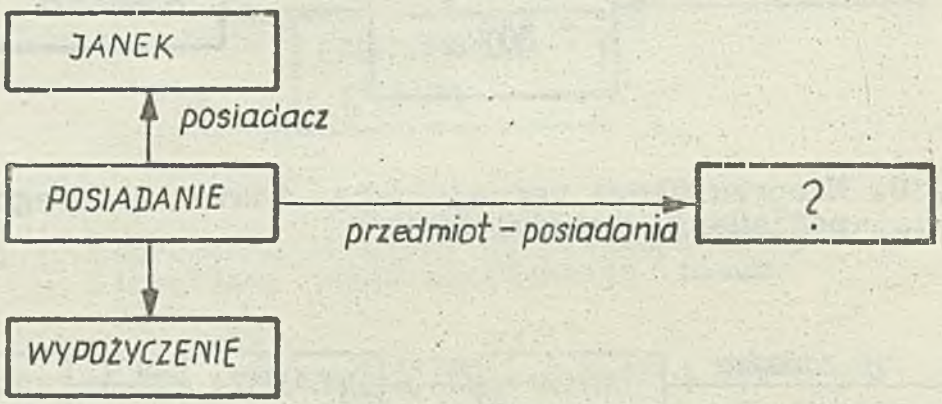


Rys. 9. Reprezentacja zdarzenia wejścia w posiadanie książki na czas określony

w której wiedza ma formę sieci semantycznej.

Pytanie takie formułuje się przez podanie fragmentu sieci. Odnalezienie odpowiedzi polega na poszukaniu w sieci semantycznej takiego wzoru, który pasuje do zapisu pytania. Z chwilą, gdy zostanie odnaleziony w bazie wzór zgodny z pytaniem, system zarządzania bazą wiedzy wysyła sygnał "TAK" oraz bada implikacje tego stwierdzenia. Badanie to polega na prześle-

dzeniu połączeń prowadzonych od znalezionego pasującego wzoru. Jeśli powiedzmy chcemy dowiedzieć się, co Janek posiadł? to pytanie to jest wyrażone przez fragment sieci jak na rys. 10.



Rys. 10. Fragment sieci semantycznej reprezentujący pytanie, co Janek posiadł?

Sieci semantyczne zostały rozwinięte do postaci pełnej, praktycznej techniki reprezentacji wiedzy i wykorzystane w wielu praktycznych systemach eksperckich i systemach bazy wiedzy.

Ramy

Pierwszeństwo sformułowania koncepcji ramy przypisuje się M. Minsky'emu [38]. Istota tej koncepcji polega na tym, że rama zawiera pewną istotną wiedzę o obiekcie, scenie, zdarzeniu itd. U podstaw leży tu spostrzeżenie, że nie tylko obiekty, ale też sytuacje i zdarzenia mają często charakter stereotypowy i mogą być brane pod uwagę jako cele przy planowaniu działań. Przykładem takiego stereotypu może być np. piłkarz, boisko piłkarskie, mecz piłkarski itd., a także znany nam już licealista, klasa, lekcja itd.

Rama gromadzi zbiory opisów artykułów, a każdy z takich opisów jest ujęty osobną działką /slot/. W ramie wyróżnia się dwa poziomy działek:

- poziom wyższy gromadzi działki z atrybutami nie ulegającymi zmianie, są tu atrybuty obiektów ogólnych klas, często też poziom ten zawiera wskaźniki do superklas,

- poziom niższy zawiera działki, które muszą być wypełnione danymi dotyczącymi sytuacji szczególnych: działki takie mogą określać warunki dołączenia ram bardziej specjalistycznych.

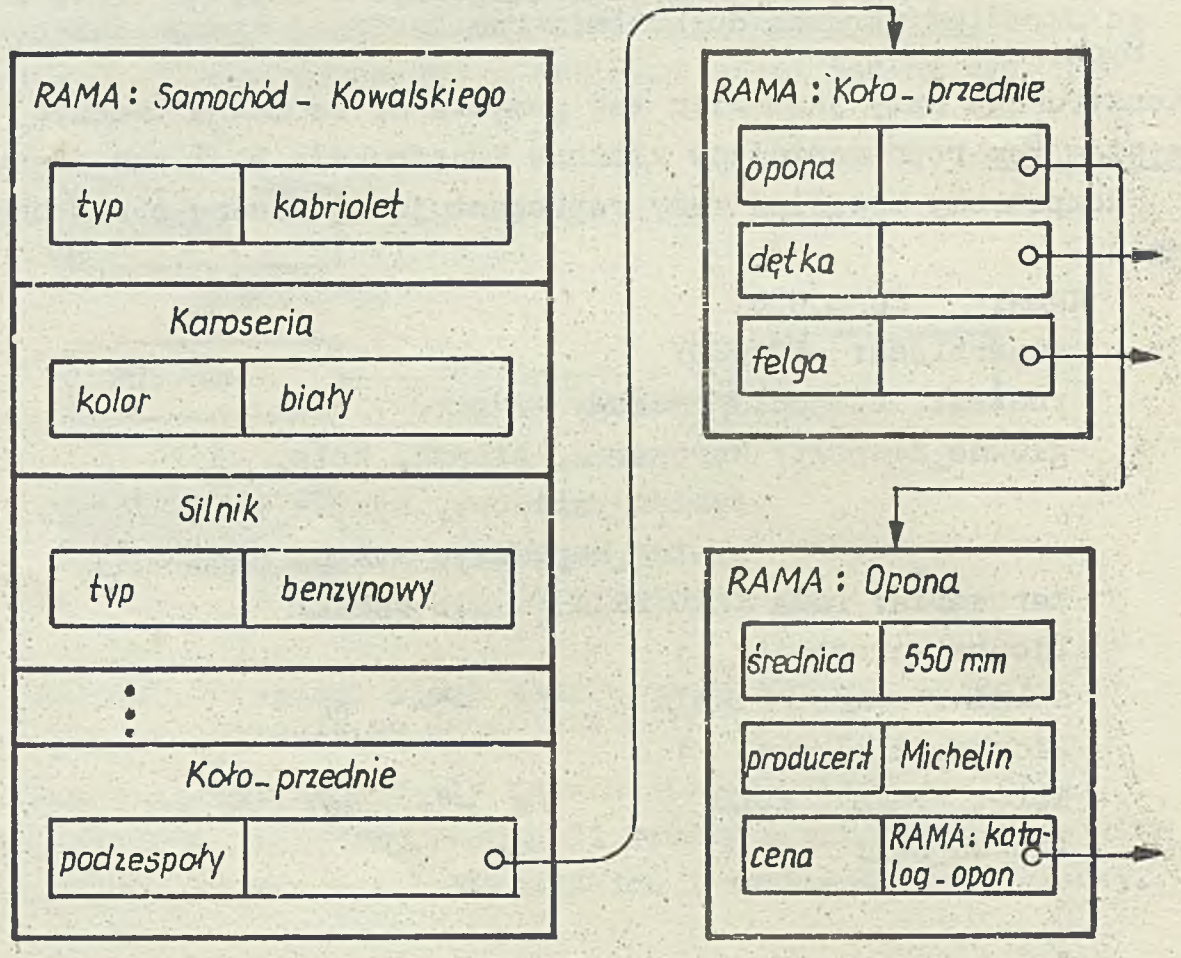
Konstrukcja ramy jest więc tak pomyślana, że można tworzyć system ram reprezentujący złożone konstrukcje bądź też sceny.

Rozpatrzmy przykład ramy reprezentującej wiedzę o samochodzie.

```
Rama:      SAMOCHOD
 superklasa:  POJAZDY
 rodzaj:  samochód_osobowy
 główne_zespoły: karoseria, silnik, koło,
                 system_paliwowy, system_elektryczny,
                 system_napędowy, system_hamulcowy
 karoseria: rama KAROSERIA /* sub-rama*/
 liczba_karoserii:  1
 silnik:  rama  SILNIK      /*  -"-   */
 liczba_silników:   1
 koło:  rama   KOŁO         /*  -"-   */
 liczba_kół:   4
 .
 .
 .
```

Przykład zapisu wiedzy o pewnym konkretnym samochodzie pokazuje rys. 11.

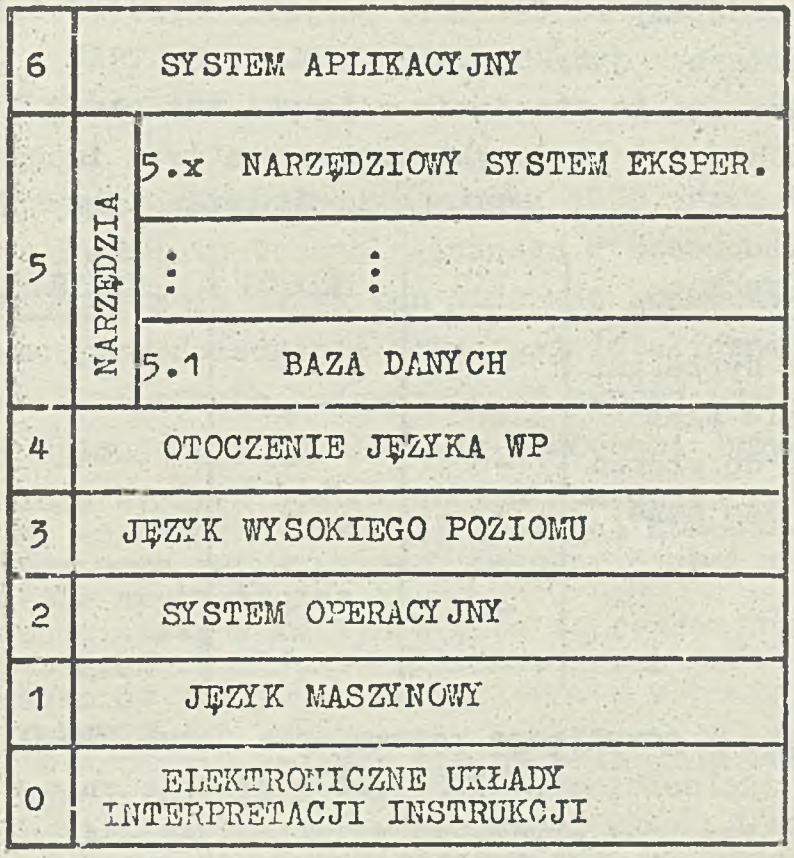
Ramy mogą doskonale służyć jako systemy informacyjne z bazą wiedzy.



Rys. 11. Reprezentacja wiedzy za pośrednictwem systemu ram, na przykładzie samochodu

✱ Język implementacji

W rozdziałach poprzednich omówiliśmy sposoby reprezentacji wiedzy i prowadzenia wywodu, co pozwala nam opisać nieformalnie projektowany system ekspercki. Aby zapis ten mógł być przydatny w praktyce, musimy przetłumaczyć ową nieformalnie zapisaną koncepcję systemu eksperckiego na jakiś język interpretowany przez nasz komputer. Względy praktyczne, tj. czas opracowania programu, niezawodność programu, dokumentacja, możliwość łatwej rewizji programu, przenośność programu, nakazują zastosować jeden z języków wysokiego poziomu /rys. 12/.



Rys. 12. Poziomy oprogramowania mikrokomputera profesjonalnego

Dokonując wyboru typu języka wysokiego poziomu bierzemy pod uwagę następujące zdolności języka:

- łatwa manipulacja strukturami symbolicznymi,
- przetwarzanie list symboli i ich połączeń jako struktur podstawowych,
- dopasowanie do wzorca - jedna z podstawowych i masowo występujących operacji układu interpretacji,
- możliwość przekazywania procedur jako danych i tworzenia procedur w trakcie procesu wnioskowego,
- możliwość zapisania systemu w postaci zbioru małych, zrozumiałych dla twórcy modułów,
- możliwość formułowania struktur danych przez użytkownika,
- możliwość formułowania struktur sterujących,
- możliwość komponowania procedur i danych,
- zmienne bez typu,
- rekursja.

Ponadto język implementacyjny dla SE, powinien dysponować pewnym otoczeniem programistycznym, które umożliwi [39] automatyczną alokację pamięci oraz komunikację "użytkownik-system".

W tabeli 1 podajemy zestawienie powyższych zdolności w odniesieniu do czterech języków:LISP, PROLOG, FORTRAN i C, z których dwa pierwsze to standardowe języki sztucznej inteligencji zaś FORTRAN i C to języki tradycyjne typu imperatywnego.

Tabela 1. Zdolności języków

Zdolność języka	LISP	PROLOG	FORTRAN	C
Manipulacje symbolami	+	+	-	-
Przetwarzanie list	+	+	-	-
Dopasowanie do wzorca	-	+	-	-
Procedury jako dane	+	+	-	+
Zamienne bez typu	+	+	-	-
Rekursja	+	+	-	+
Procedury	+	+	+	+

Jak to wynika z powyższego zestawienia język PROLOG dysponuje wszystkimi zdolnościami, LISP jest gorszy jedynie w jednym punkcie. Dlatego też obydwa te języki uznaje się obecnie za bazowe języki programowania systemów sztucznej inteligencji, dla nich też są budowane specjalne maszyny, tzw. maszyny LISP [40] i maszyny PROLOG [41] .

Język LISP jest pierwszym językiem sztucznej inteligencji; do niedawna był językiem głównym, dominującym w tej dziedzinie. W chwili ogłoszenia programu prac nad maszynami V generacji w Japonii i związanie tego programu z rozwojem języka PROLOG sytuacja się nieco zmieniła, choć nadal w USA język LISP jest językiem podstawowym w inżynierii wiedzy.

O wyborze języka może też decydować podejście do problemu: jeśli rozwiązaniu problemu chcemy nadać kształt struktury funkcjonalnej to wybieramy LISP, jeśli natomiast łatwiejsze rozwiązanie widzimy w postaci pewnej deklaracji logicznej to bardziej odpowiedni do tego celu będzie PROLOG [37], [42], [43], [44] .

Z wymienionych powodów opowiadamy się za LISP-em, dlatego też, a także ze względu na ograniczoną objętość pracy, przedstawiamy tu jedynie LISP, a i to wstępnie. Bliższe zaznajomienie z językiem LISP umożliwiają następujące prace [45] , [46] ,

[47], [48].

Własności języka LISP

Język LISP został sformułowany w roku 1958 przez J. McCarthy'ego z Instytutu Technologicznego w Massachussets, USA. Od chwili powstania język ten stał się podstawowym narzędziem badawczym w dziedzinie sztucznej inteligencji w uniwersytetach i ośrodkach badawczych. W swojej pracy z 1978 r. [47] McCarthy opisuje główne elementy koncepcji języka LISP.

/a/ Przetwarzanie wyrażeń symbolicznych /bardziej niż liczb/; oznacza to, że binarna zawartość pamięci komputera i jego rejestrów może reprezentować dowolny symbol, a nie tylko symbole arytmetyczne,
/b/ przetwarzanie list; w maszynie dane są reprezentowane w postaci powiązanych struktur listowych zaś przed ich wprowadzeniem do maszyny - na papierze - w postaci list wielopoziomowych,
/c/ struktura sterowania zasadza się na składaniu funkcji do postaci funkcji zagnieżdżonych,
/d/ rekursja jest sposobem formułowania procesów i problemów,
/e/ programy LISP-owe są reprezentowane tak jak inne dane, to znaczy: wewnętrznie w postaci powiązanych struktur listowych, zewnętrznie - w postaci list wielopoziomowych.
/f/ funkcja EVAL, również zapisana w języku LISP, służy jako interpreter dla LISP-u oraz jako formalna definicja języka.

W języku LISP istnieje zaledwie kilka funkcji podstawowych. Wszystkie inne funkcje są definiowane za pomocą funkcji podstawowych. Wymienione tu własności języka powodują, że można łatwo tworzyć środowisko programistyczne dostosowane do problemu. Tym też należy tłumaczyć, że powstało wiele mutacji języka LISP.

Język LISP nie jest językiem skomplikowanym, a jego reguły syntaktyczne i semantyczne można opanować na ogół w krótkim czasie. A oto niektóre cechy języka LISP.

Lista. Listę jest podstawową strukturą języka, listę elementów niezbędnych do budowy układu elektronicznego zapiszemy następująco:

(tranzystor opornik 1 opornik 2 dioda)

Elementy tej listy mogą być symbolami, liczbami lub listami:

(spis (1 tranzystor)(2 opornik1)(3 opornik2)(4 dioda))

Powyższa lista zawiera 5 elementów, z których jeden jest symbolem, pozostałe 4 zaś są listami dwuelementowymi, z których każda zawiera liczbę i symbol.
Lista może nie zawierać żadnego elementu, co zapiszemy jako

( )

Taka pusta lista jest określana terminem nil.
Atom. Drugim po liście, podstawowym wyrażeniem symbolicznym jest atom. Pod pojęciem atomu rozumie się tu sznur złożony z liter, cyfr lub znaków przestankowych /z wyjątkiem nawiasów i pojedynczego cudzysłowu/. Przykłady

Jan
32
opornik3
+
składnik-drugi

Atomy mają niezwykle ważne znaczenie dla programów sztucznej inteligencji. To atomy właśnie mogą służyć do zapisu symboli fizycznych z modelu A. Newella /rozdział "Pojęcia podstawowe"/
Funkcje. Programy w języku LISP składają się z funkcji. Funkcja działa na pewnej liczbie argumentów - manipuluje wartościami tych argumentów w określony sposób, a następnie zwraca obliczoną wartość. Przykładem może być funkcja dodawania PLUS:

(PLUS 2 3)
5

Liczby 2 i 3 są argumentami funkcji PLUS zaś liczba 5 jest obliczoną wartością funkcji.

Ponieważ programy i dane korzystają z tej samej struktury listowej, zachodzi potrzeba wprowadzenia pewnych reguł określania wartości wyrażeń symbolicznych.

/1/ Jeśli wyrażenie jest listą to pierwszy od lewej element listy jest traktowany jak nazwa funkcji zaś pozostałe elementy listy jak argumenty tej funkcji. Jeśli taka funkcja nie została zdefiniowana to zgłoszony zostanie błąd.
/2/ Wszystkie liczby są oceniane zgodnie ze swą wartością.
/3/ Symbol specjalny T ma przypisaną wartość prawdy zaś NIL wartość fałszu.
/4/ Wszystkie inne symbole, bez względu na to czy występują samodzielnie, czy też jako elementy listy, są traktowane jak zmienne i są oceniane celem określenia ich wartości.

Wyjątkiem od reguły /1/ będzie ewaluacja wyrażeń zawierających funkcję QUOTE.

Funkcja QUOTE. Argumentem tej funkcji może być atom lub lista. Funkcja QUOTE zwraca wartość swego argumentu, na przykład

```
(QUOTE  JAN )
 JAN
```

Inny przykład:

```
(QUOTE (1  ADAM  JAN))
(1   ADAM  JAN )
```

Sens funkcji QUOTE polega na wyłączeniu ewaluacji wewnątrz listy, jak w drugim z powyższych przykładów. W wielu systemach stosuje się skrócony zapis funkcji QUOTE a mianowicie '(x y z) w miejsce (QUOTE (X Y Z))

Funkcja CAR. Argumentem tej funkcji jest lista nie pusta. Funkcja CAR wybiera pierwszy element z tej listy i przekazuje jako swoją wartość, na przykład

```
(CAR '(A  B  C))
 A
```

Kolejny przykład

```
(CAR '((A)  (B)  (C)))
(A )
```

Funkcja CDR. Argumentem podobnie jak poprzednio jest lista nie pusta. Funkcja bierze wszystkie elementy tej listy z wyjątkiem elementu pierwszego, tworzy z tych elementów listę i zwraca jako swoją wartość. Na przykład

```
(CDR     '(A   B   C))
(B  C)
```

Kolejny przykład

```
(CDR     '((A) (B) (C)))
((B)   (C))
```

Funkcja CONS. Funkcja ta ma dwa argumenty, z których pierwszy może być dowolnym wyrażeniem zaś drugi musi być listą. Funkcja wprowadza pierwszy argument jako pierwszy element listy reprezentującej drugi argument. Wartością funkcji CONS jest nowa lista. Przykład

```
(CONS   'A    '(B     C))
(A   B   C)
```

Funkcje predykatowe. LISP dysponuje kompletem funkcji predykatowych. W wyniku ewaluacji wyrażenia funkcja predykatowa zwraca wartość fałszu lub wartość prawdy. Wartość fałszu jest reprezentowana przez NIL, zaś wartość prawdy może mieć następujące reprezentacje:

- symbol T
- atom
- lista

A oto opisy ważniejszych funkcji predykatowych.

Predykat ATOM. Jest to funkcja lub lista. Funkcja zwraca T jeśli argument jest atomem. Przykłady:

```
(ATOM 'A)
 T
(ATOM '(A B C))
 NIL
```

Predykat ZEROP. Argumentem jest numerał. Zwracana wartość funkcji T, jeśli argument jest zerem, gdy różny od zera - wartość funkcji jest NIL

```
(ZEROP  0)
 T
(ZEROP  44)
 NIL
```

Predykat GRATERP. Argumentem jest lista liczb. Wartość funkcji T, jeśli liczby w liście argumentu są ustawione w porządku malejącym zaś wartość NIL w innych wypadkach. Przykłady

```
(GRATERP '(6 5 4 3))
 T
(GRATERP '(1 2 3 6))
 NIL
```

Funkcja COND. Jest to lispowska wersja wyrażenia warunkowego. Postać ogólna tej funkcji jest następująca:

```
(COND (warunek1 wyrażenie1)
      (warunek2 wyrażenie2)
                .
                .
                .
      (warunekN wyrażenieN))
```

Zarówno część warunkowa, jak też część wyrażenia mogą być dowolnymi wyrażeniami symbolicznymi. Funkcja COND ocenia wartości warunków w podanym porządku do chwili aż wartość warunku będzie różna od NIL. W tym wypadku następuje ewaluacja wyrażenia, tj. prawej części argumentu z warunkiem prawdziwym; wynik tej ewaluacji jest zwracany jako wartość funkcji COND. Weźmy przykład

```
(COND   (EQUAL  A 1) 'JEDEN)
        (EQUAL  A 2) 'DWA )
        (EQUAL  A 3) 'TRZY )
        (T  CYFRA-NIE-ZNANA))
```

Jest to program badający wartość zmiennej A. Funkcja wypisuje JEDEN, DWA lub TRZY jeśli A jest równa odpowiednio 1, 2 lub 3. W innych wypadkach zwracaną wartością jest napis CYFRA-NIE-ZNANA.

Funkcja SETQ. Jest to funkcja dwuargumentowa. Pierwszy z argumentów jest symbolem zaś drugi - dowolnym wyrażeniem symbolicznym. Funkcja zwraca wartość drugiego argumentu. Ważniejszy jest jednak efekt uboczny funkcji a mianowicie przypisanie symbolowi /będącemu pierwszym argumentem/ wartości w postaci drugiego argumentu. Przykład

```
(SETQ   A   2)
  2
```

Jeśli teraz zastosujemy

```
(EQUAL  A  2)
  T
```

to otrzymaliśmy w wyniku prawdę.

Funkcja DEFUN. Funkcja ta umożliwia tworzenie nowych funkcji. Jako wynik swojego działania funkcja DEFUN zwraca nazwę funkcji definiowanej. Ogólna postać funkcji jest następująca:

```
(DEFUN  nazwa-funkcji ( lista-argumentów )
                      ( ciało-funkcji ))
```

Rozpatrzmy przykład

```
(DEFUN  TRZECI-ELEMENT (X)(CAR (CDR ( CDR  X))))
```

Dokonajmy podstawienia na zmienną X

```
(SETQ  X  '(A  B  C))
```

Obecnie wywołanie nowej funkcji da następujący wynik:

```
(TRZECI-ELEMENT  X)
C
```

Ten sam wynik otrzymamy jeśli wywołanie funkcji sformułujemy następująco:

```
(TRZECI-ELEMENT '(A  B  C))
C
```

Lista (A B C) stała się aktualną wartością argumentu X.

* Tworzenie systemu eksperckiego

Zamieszczone niżej uwagi na temat tworzenia systemu eksperckiego dotyczą jedynie wstępnych kroków. Umiejętność budowy systemu, jak w wielu innych wypadkach programowania, można pozyskać jedynie przez opanowanie konkretnego języka i techniki programowania. Dodatkową trudność stanowi formułowanie bazy wiedzy.

Narzędzia programistyczne

Narzędziem zasadniczym jest LISP wraz z pewnym środowiskiem programowania. Dostępne na rynkach zachodnich środki można podzielić na trzy grupy /ze względu na ich zdolność rozwiązywania problemu/:

- pierwsza grupa to narzędzia, które pozwalają realizować małe systemy na komputerze personalnym, posługujące się bazą wiedzy, o rozmiarze do 500 reguł,
- grupa druga służy do budowy systemów dużych lecz wyspecjalizowanych, narzędzia te wymagają do realizacji maszyny klasy LISP, mają zdolność tworzenia od 500 do kilku tysięcy reguł, działają w jednym trybie rozwiązywania problemu,
- grupa trzecia skupia narzędzia dla systemów wielkich, o zdolności od 500 do kilku tysięcy reguł, narzędzia takie pozwalają stosować różne techniki rozwiązywania problemu, wymagają jednak dużych komputerów lub rozwiniętych komputerów LISP.

Jest oczywiste, że początkujący budowniczy systemów ekspertkich rozpocznie od systemów małych. Do tego celu w pełni wystarczy zestaw programistyczny GCLISP, którego cena w 1986 r. wynosiła 495 dolarów USA. Zestaw GCLISP zawiera:

Tab.2. Narzędziowe programy SE firm zachodnich

FIRMA	NAZWA PROGRAMU	TYP SPRZĘTU	CENA USD	LICZBA REGUŁ /max/	WSPÓŁ-CZYNNIK UFNOŚCI	JĘZYK IMPLEMEN-TACYJNY
Human Edge Software	Expert Ease Expert Edge	IBM PC AT/XT	695 795	255 500	NIE TAK	UCSD PASCAL
KDS Inc.	KDS	IBM PC	695	4 00	NIE	Asembler 8C86
General Research	TIMM	PC XT,AT	9500	500	TAK	FORTRAN 77
RADIAN	Rule Master	PC XT, AT	1000 15000	150/200	TAK	C
Level 5 Research	Insight 1 Insight	IBM PC VICTOR 900 IBM PC	 95 485	625 2000	TAK TAK	PASCAL PASCAL
EXSYS	EXSYS	IBM PC	395	700	TAK	C
TEKNOWLEDGE	M1	IBM PC	5000	200	TAK	PROLOG
TEXAS INSTRUMENTS	PC	IBM PC TI-Prof.	950	300	TAK	IQ LISP
ARTELLIGENCE	OPS5+	IBM PC	3000	1500	NIE	C
EXPERTELLIGENCE	Exper OPS5	MacIntosh	325	500	NIE	Expert Lisp
AION	AIOM ADS	IBM PC	7000			Asembler
Neuron Data	NEXPERT	MacIntosh	5000	1000 2000	NIE	Asembler C

- pomoc dialogową do nauki języka LISP,
- edytor tekstowy, narzędzia korekty programu,
- narzędzia wspomagania programisty,
- interpreter ponad 400 prymitywów funkcyjnych,
- automatyczne zarządzanie pamięcią.

GCLISP jest przystosowany do mikrokomputera typu PC XT/AT. Dokładniejsze charakterystyki GCLISP można znaleźć w pracy "Podstawy implementacji systemów eksperckich" Z. Wrzeszcza i in. [34].

Prostym i o połowę tańszym zestawem jest TLCLisp firmy The Lisp Co.; zestaw ten jest wyposażony dodatkowo w pewne środki dla grafiki, za to nie ma dialogowej pomocy do nauki języka.

Zestawy Lispu dają twórcy systemu eksperckiego ogromne możliwości realizacyjne, nie dostarczają jednak żadnej pomocy w czynnościach związanych z reprezentacją wiedzy. W tym zakresie bardziej pomocne są narzędzia zwane językami inżynierii wiedzy.

Obecnie można spotkać się z dwoma rodzajami języków inżynierii wiedzy /języki IW/, tj.: systemy z pustą bazą wiedzy i języki IW ogólnego przeznaczenia.
Pierwszy rodzaj to po prostu dedykowane systemy eksperckie z opróżnioną bazą wiedzy. W literaturze anglojęzycznej znane są jako expert system shells. Przykładem może tu być język E-MYCIN, który powstał po usunięciu treści bazy wiedzy, z zachowaniem jednak starego układu IR [5C]. Inne przykłady systemów IW z pustą bazą wiedzy to INSIGHT2+ [53] i M.1 [54].

Przykładem języka IW ogólnego przeznaczenia są języki grupy OPS [51], [52].

Specjalista dziedziny i inżynier wiedzy

W procesie twcrzenia systemu eksperckiego inżynier wiedzy musi zaznajomić się z każdym aspektem problemu, który zamierza się poddać systemowi eksperckiemu do opanowania. Inżynier wiedzy nie może tego dokonać samodzielnie /przynajmniej w sensownym czasie/, musi korzystać z pomocy specjalisty z danej dziedziny /eksperta/. Jeśli na przykład mamy do czynienia z prob-

lemem medycznym to ekspertem będzie lekarz-specjalista. Współpraca inżyniera wiedzy i eksperta dziedziny problemu polega przede wszystkim na sformułowaniu reguł wnioskowych, które wypełnią bazę wiedzy.

Wbrew pozorom niełatwo jest znaleźć specjalistę, który bez trudności może udostępnić swą wiedzę do zapisu. Bardzo często ekspert wykonuje swą czynność automatycznie, bez planowania. Przetłumaczenie nie artykułowanych wiadomości o wykonywaniu zadania na zbiór reguł może więc zająć wiele czasu. Obecnie istnieją już narzędzia typu "system z pustą bazą wiedzy", które umożliwiają ekspertowi formułowanie reguł za pomocą przykładu. Zamiast przymuszać do wypełniania klauzulami lewej i prawej strony reguły komputer umożliwia ekspertowi zidentyfikowanie problemu oraz określenie pewnej liczby poprawnych rozwiązań. Informacje te pozwalają na sformułowanie reguł przez system. Zapisane reguły system prezentuje ekspertowi, który sprawdza te reguły i zatwierdza. W takiej sytuacji rola inżyniera wiedzy schodzi na plan drugi.

* Zastosowania

Zastosowanie metod i techniki sztucznej inteligencji /SI/ to jeden z sektorów rynku komputerowego o intensywnym wskaźniku wzrostu. Wskaźnik ten dla rynku USA i W. Brytanii w 1987 r. ocenia się na 100 do 300 procent [55] . Dowodzi to dojrzewania technologii oraz wzrostu inwestowania w sprzęt, oprogramowanie i szkolenie użytkownika. Technologia SI przestała zamykać się wyłącznie w sferze badawczej.

Aplikacje SI są wynikiem prac badawczych, których celem było zbudowanie oprogramowania komputerowego i związanego z tym sprzętu, do realizacji zadań wykonywanych zazwyczaj przez człowieka i zarazem takich zadań, które nie są łatwe dla tradycyjnego ujęcia w formie procedur obliczeniowych; są to zadania ze znaczną przewagą przetwarzania symbolicznego w stosunku do przetwarzania numerycznego. Zbiór takich zadań jest ogromny. Można w nim wyznaczyć działy zależnie

od charakteru warstwy pośredniej modelu współpracy "system BW - otoczenie". Będą to mianowicie

- komputerowe systemy wizji dźwięku i dotyku,
- ruchome roboty,
- systemy eksperckie.

W latach osiemdziesiątych dział systemów eksperckich ma znaczenie dominujące.

Pierwsze realizacje

Dwa spośród znacznej już liczby zbudowanych systemów eksperckich mają znaczenie pionierskie, są to DENDRAL i MYCIN.

DENDRAL [17]

W 1964 r. zdobywca nagrody Nobla w chemii J. Lederberg opracował algorytm pod nazwą DENDRAL umożliwiający predykcję struktury molekularnej na podstawie znajomości analizy spektroskopowej molekuły nieznanej. Algorytm DENDRAL po przyjęciu danych spektroskopowych wymienia wszystkie możliwe struktury molekularne wynikające z tych danych.

W 1965 r. powstała grupa w składzie J. Lederberg, E. Feigenbaum i B. Buchanan, która podjęła próbę budowy programu DENDRAL w postaci heurystycznej. Idea heurystycznego DENDRAL-a polegała na poszerzeniu możliwości algorytmu o zespół reguł heurystycznych stosowanych przez ekspertów - chemików. Reguły te wprowadzały ograniczenia /więzi/ na enumerację molekuł z danych spektrometrycznych. W wyniku tych ograniczeń nastąpiła redukcja przestrzeni rozwiązań, przez odcinanie niektórych gałęzi przestrzeni generowej przez algorytm DENDRAL-a.

Niezmiernie trudne i czasochłonne okazało się uzyskiwanie wiadomości o heurystyce rozwiązań od ekspertów chemików; proces ten wyznaczył czas budowy całego systemu DENDRAL /zajęło to około 15 osobolat/. Dla E. Feigenbauma, szefa projektu, był to jednak okres bardzo ważny - w tym czasie sformułowano zasadnicze postulaty budowy systemów z bazą wiedzy w postaci reguł systemów eksperckich.

Program DENDRAL przyjmuje dane w formie histogramu zawierającego pary złożone z liczby masowej i natężenia; są to odczy-

ty ze spektrografu masowego. Do tak sformułowanego wejścia program stosuje heurystyki i tworzy wstępny plan sugerujący pewien rodzaj struktur molekularnych zasługujących na rozpatrzenie. Działając w ramach więzi nałożonych przez powyższy plan wstępny, algorytm DENDRAL generuje tylko te struktury, które nie mają składników wyeliminowanych wcześniej przez więzi. Na wyjściu pojawiają się tylko struktury spełniające wymagania i są one reprezentowane przez dane spektrometryczne.

Postać heurystyki DENDRAL-a to reguła produkcji. A oto przykład reguły:

Jeśli X1 + Y2 = M + 28
i X1 - 28 osiąga wysoką wartość
i X2 - 28 osiąga wysoką wartość
i X1 lub X2 osiąga wysoką wartość
to molekuła zawiera grupę ketonową.

gdzie X1 i X2 są wartościami szczytowymi widma.

Jak widzimy reguła ma specyficzny charakter i taki też charakter ma cały system. Ponadto siłą systemu DENDRAL jest systematyczność działania, przeszukiwana jest bardzo skrupulatnie cała przestrzeń możliwości z wyjątkiem tych odciętych przez więzi. W sumie, DENDRAL nie przewyższa wiedzą eksperta-chemika lecz może daną robotę wykonać staranniej, a więc lepiej.

DENDRAL i jego następca METADENDRAL, znalazły zastosowanie w laboratoriach chemicznych uniwersytetu w Stanford /Stany Zjednoczone AP/ i podobno pracują tam po dzień dzisiejszy. Mają one jednak szczególne znaczenie dla dalszego rozwoju dziedziny pod nazwą systemy eksperckie.

MYCIN

Drugim niezwykle popularnym przykładem systemu eksperckiego jest system MYCIN. System MYCIN jest przeznaczony do wspomagania diagnozy lekarskiej i zaleceń terapeutycznych w zakresie chorób spowodowanych infekcją krwi. Pracę nad systemem rozpoczęto w 1972 r. w formie pracy doktorskiej E. Shortliffa

w uniwersytecie w Stanford /USA/ [21] .

Działanie systemu MYCIN zasadza się na systemie BWP opisanym w rozdziale "Zasady działania systemu eksperckiego". Przykład reguły produkcji przedstawia się następująco:

jeśli 1 zabarwienie organizmu badanego metodą Grama jest negatywne
i 2 morfologia organizmu jest pałeczkowa
i 3 organizm jest zdolny żyć bez tlenu /anaerobic/
to istnieje przeświadczenie /0,7/, że organizm jest bakteroidem.

Zadaniem systemu MYCIN jest dostarczenie lekarzowi eksperckiej diagnozy i porady terapeutycznej w sposób szybki i dogodny. Zdarza się przecież często, że lekarz musi dokonać szybkiego osądu, często na podstawie danych niekompletnych lub wieloznacznych; pośpiech jest tu podyktowany chęcią ratowania ludzkiego życia. A oto jak wygląda typowy dialog systemu MYCIN z lekarzem - użytkownikiem; wypowiedź użytkownika jest podkreślona.

/1/ Proszę o imię i nazwisko pacjenta
Jan Kowalski
/2/ Płeć:
M
/3/ Wiek
37
/4/ Czy jest dostępna informacja mikrobiologiczna dotycząca prawdopodobnej infekcji u J. Kowalskiego?
Tak
/5/ Czy są dane identyfikujące organizm-1
Brak danych
/6/ Czy Organizm-1 ma charakter pałeczki?
Tak
/7/ Jakie jest zabarwienie organizmu w metodzie Grama?
Negatywne
/8/ Czy istnieją kultury dostarczające organizmy, których rozpoznanie jest możliwe?
Tak

W podobnej formie dialog może przebiegać osiągając liczbę kilkudziesięciu pytań-odpowiedzi w wyniku czego MYCIN dostarcza diagnozę:

INFEKCJA MA CHARAKTER BAKTERYJNY

ELEMENT-1 : E. COLI
ELEMENT-2 : SALMONELLA
ELEMENT-3 : KLEBSIELLA-PNEUMONIAE
ELEMENT-4 : PROTEUS-NON-MIRABILIS

Infekcję mógł więc spowodować dowolny z czterech wymienionych organizmów. Powstaje więc problem, jakie przepisać leki aby poprawić stan chorego unikając szkodliwych oddziaływań spowodowanych stosowaniem wielu leków. MYCIN w tym wypadku zaleca:

PREFEROWANA TERAPIA

- przeciw organizmom /1/, /2/, /4/: GENTAMICIN
  ilość: .....
- przeciw organizmowi /3/: CHLORAMPHENICOL
  ilość: .....

W sumie o systemie MYCIN możemy powiedzieć, że:

- działa na poziomie specjalisty
- wnioskowania na podstawie danych niepewnych są istotną cechą systemu
- opracowany został program pomocniczy pod nazwą TEIRESIAS służący do akwizycji wiedzy.

System MYCIN odegrał wielką rolę w rozwoju badań nad systemami eksperckimi; mimo swojej popularności MYCIN nie był przedmiotem handlu ani też nie znalazł się w wyposażeniu licznych szpitali, choćby w USA. Jest tu kilka przyczyn:

1° zbyt wąska specjalizacja systemu,
2° odczucie przeciętnego lekarza o zbędności takiego narzędzia, obawa przed oskarżeniem o brak kompetencji,
3° obawa przed ewentualnymi błędami systemu; jeśli błędna terapia spowoduje stan krytyczny chorego, to kto za to odpowiada?
4° z drugiej znów strony systemy sztucznej inteligencji
   - nie podlegają zmiennym stanom psychicznym,
   - nie męczą się,
   - są systematyczne.

Charakterystyka rynku systemów eksperckich

Charakterystykę tę ograniczymy do rynku amerykańskiego, gdyż tylko stąd dane niezbędne są względnie łatwo dostępne. Bierzemy pod uwagę okres po roku 1980. Wiąże się to m.in. z wejściem mikrokomputerów na rynki światowe. Na ogół niska cena maszyny przy znacznej mocy obliczeniowej i jednocześnie łatwość manipulowania nośnikami zbiorów /dyski elastyczne, dyski twarde typu "Winchester"/ spowodowała niezwykły wzrost liczby użytkowników. W ślad za tym w USA powstały liczne firmy produkujące oprogramowanie narzędziowe. Główny trzon tych prac polegał na adaptacji programów pisanych na duże maszyny i przeniesienie ich na mikrokomputery. Tym sposobem narzędzia CAD /Computer Aided Design/ weszły wreszcie do szerokiej praktyki /biura projektowe, konstrukcyjne przemysłu/. Podobnie postąpiono z systemami eksperckimi SI. Rynek technologii SI w początku lat osiemdziesiątych dostarcza głównie narzędzia i języki do tworzenia systemów eksperckich. Rozpiętość cen jest znaczna, jak to widać z tabeli 1. Tabela ta zawiera tylko pewien wybór i nie wyczerpuje informacji o wszystkich dostępnych produktach. Więcej informacji na ten temat można znaleźć w pracy Watermana [50].

Okres po 1985 r. charakteryzuje się przede wszystkim znacznym rozwojem technologii mikroukładów, nośników magnetycznych, wydruku i wskaźników optycznych. Zaowocowało to ogromnym wzrostem mocy obliczeniowej mikrokomputera przy jednoczesnym spadku jego ceny. Mikrokomputer jest więc nadal głównym /choć nie jedynym/ środkiem realizacji systemów eksperckich. Wzrost mocy obliczeniowej mikrokomputera pozwolił na poprawę mechanizmów umożliwiających odwoływanie się do różnych języków, do bazy danych, do specjalistycznych pakietów itp. Tak wzmocnione mechanizmy systemowe pozwalają tworzyć systemy eksperckie o nowym paradygmacie aplikacyjnym w stosunku do klasycznego systemu doradczego.

Równocześnie ze wzrostem możliwości systemowych wielkie firmy w rodzaju komputerowej firmy IBM, firmy lotniczej Boeing, firmy samochodowej Ford /aby wymienić tylko kilku

potentatów/ powołały do życia specjalne zespoły SI, których zadaniem jest stałe śledzenie rozwoju SI, ciągła obserwacja problematyki własmej firmy i podejmowanie aplikacji systemów eksperckich dla tej problematyki. Jest to główna z przyczyn, dla których aplikacje SE nie są dość widoczne na rynku komputerowym. Aplikacje systemów eksperckich przypominają więc górę lodową - część widoczna jest bardzo mała.

Potrzeba specjalizacji wiedzy systemu eksperckiego jest, jak to wyjaśnialiśmy wcześniej cechą "przyrodzoną" systemu eksperckiego. Dlatego podział klasyfikacyjny tych systemów ze względu na różne sfery działalności wydaje się naturalny, chociaż w obrębie danej sfery zachodzi dalsza specjalizacja i podział. Tym niemniej mówi się o systemach eksperckich w działalności przemysłowej, handlu, nauczaniu, służbie zdrowia, obronności.

Zastosowanie w przemyśle

Widoki na rozwój zastosowań SE są chyba największe. Składają się na to następujące czynniki:

- duże fundusze na rozwój prac,
- bogactwo zadań realizowanych tradycyjnie przez człowieka,
- silne tendencje do mechanizacji pracy,
- trwały nacisk rynku na poprawę parametrów wyrobu i jego wskaźników techniczno-ekonomicznych,
- działanie konkurencji mające na celu poprawę narzędzi technologicznych,
- wejście w integralnie zautomatyzowane wytwarzanie /CIM, MAP/TOP/.

Obecne stosowanie systemów eksperckich to głównie działalność związana z planowaniem lub klasyfikacją. Na tym paradygmacie buduje się różnego rodzaju systemy doradcze, np.:

- projektowania,
- specyfikacji materiałowej,
- optymalizacji parametrów produkcyjnych,
- wykrywania uszkodzeń,
- wspomagania przy naprawach,
- rozpisywania szczegółowych zadań produkcyjnych /tzw. przygotowanie produkcji/,
- przygotowania planu szkoleń.

Oczywiście zróżnicowanie nastąpi ze względu na typ produkcji, w takim stopniu, że np. system ekspercki wspomaganie napraw w firmie "Kasprzak" nie będzie nadawać się dla firmy "Pafawag", ale obydwa te systemy można wykonać na bazie tego samego systemu narzędziowego /np. INSIGTH 2 +/[53] .

Problematykę systemów eksperckich dla przemysłu obszernie omówiono w specjalnym wydaniu czasopisma ROBOTICS [57] .

* Kierunki rozwoju systemów eksperckich

Reprezentacja wiedzy głębokiej

Uwagę twórców systemów eksperckich zwrócona jest przede wszystkim na udoskonalenie mechanizmów systemu. O ile pierwsze systemy /np. Dendral, Mycin/ gromadziły wiedzę dotyczącą zewnętrznych obiektów, to głęboka reprezentacja za pomocą struktur w formie ram lub skryptów umożliwia wnioskowanie przez analogię lub też za pomocą abstrakcji. Reprezentacja taka umożliwia opisanie zarówno obiektu jak też procesów, w których ten obiekt uczestniczy. Ramy, sieci semantyczne lub skrypty umożliwiają organizowanie obiektów i ich realizacji w jednostki złożone zawierające wyróżnialne obiekty np. silnik samochodowy, SE z wiedzą zewnętrzną /powierzchowną/ są pewną projekcją systemów z wiedzą głęboką.

Akwizycja wiedzy

Sposoby uzyskiwania wiedzy są następujące:

- przez omówienie,
- przez analogię,
- przez obserwację i doświadczenie,
- przez wnioskowanie z zależności strukturalnych.

Ponieważ zdobywanie wiedzy od eksperta /przez omówienie/ jest zajęciem wybitnie pracochłonnym, istnieje potrzeba wprowadzenia narzędzi do zdobywania wiedzy, stanowiących część systemu. Obecne wyniki w tym względzie to pewna forma maszynowego uczenia reguł na podstawie przykładów. SE następnie wytwarza poprawne reguły zgodne z przyjętą syntaktyką.

Znanym przykładem jest tu program ID3 /napisany w PASCAL-u!/ przez R. Quinlana [58].

Systemy wielopoziomowe

Z doświadczeń badawczych z systemami eksperckimi wynika, że proste SE z wiedzą powierzchniową, mimo uzupełniania tej wiedzy, szybko ulegają degradacji, tracą pierwotne znaczenie aplikacyjne.

Aby uniknąć tej degradacji SE muszą być wzbogacone w różnorodne typy reprezentacji wiedzy: wyznacznikami tej reprezentacji jest różna głębokość, różna specyfika, różny stopień opisowości. Stąd też nazwa SE wielopoziomowe.
Przykładem może tu być SE STEAMER [50] dysponujący mechanizmami do:

- graficznej reprezentacji obiektów dla określonej dziedziny
- reprezentacji przez ramy
- manipulacji procedurami
- gromadzenia stwierdzeń /baza twierdzeń/
- numerycznej symulacji cyfrowej

Przewiduje się, że budowa takich systemów zajmie najbliższe 5-10 lat.

Niezbędne środki obliczeniowe

Rolę języków symbolicznych omówiliśmy już wcześniej. Warto jednak powtórzyć pewne wymagane cechy ogólne języka SI /SE/.
Język SE i jego środowisko powinny zawierać

- interpreter wyrażeń na bieżącą,
- prymitywy przetwarzania list i wyrażeń symbolicznych,
- mechanizm stopniowej kompilacji.

Optymalny jest tu LISP z odpowiednim środowiskiem.
Programy SE są bardzo pamięciochłonne. Biorąc dodatkowo wymagania z poprzedniego punktu dla systemu wielopoziomowego okaże się dość szybko, że rozwiązywanie bardziej złożonych problemów wykracza poza możliwości komputera personalnego; potrzebne tu będzie komputerowe stanowisko robocze o zdolnościach zbliżonych do tych, które stosuje się w rozpoznawaniu obrazów czy też przy projektowaniu struktur VLSI.

Pewne dodatkowe dane o rozwoju SE może dostarczyć tabela 3 ujmująca rozwój prac nad sztuczną inteligencją.

Tabela 3. Główne wyznaczniki rozwoju prac nad sztuczną inteligencją

OKRES	WAŻNIEJSZE DZIAŁANIA I WYNIKI
Lata przedwojenne /korzenie badań/	. Prace nad logiką formalną . Prace w zakresie psychologii poznawczej
Lata 1945-1954 /Przygotowania teoretyczne do badań nad sztucznym intelektem/	. Rozwój konstrukcji komputerów . Praca A.M.Turinga: "Computing Machinery and Intelligence" . Praca H. Simona: "Administrative Behavior" . Praca N. Wienera: "Cybernetics"
Lata 1955-1960 /Rozpoczęcie prac doświadczalnych nad sztucznym intelektem/	. Komputery dostępne na rynkach: USA - IBM 704, 709, 7094 w POLSCE - ZAM 2 . Opracowanie języka IPL-I przez Newella i Simona . Podjęcie prac nad systemem General Problem Solver /GPS/ przez Newella, Shawa i Simona
Lata 1961-1970 /Poszukiwanie rozwiązań ogólnych/	. Praca A. Newella i H. Simona: "Human Problem Solving" . Opracowanie przez J. McCarthy-ego języka do przetwarzania struktur listowych LISP . Opracowanie metodologii poszukiwania rozwiązań zadowalających /spełniających ograniczenia/ . Heurystyka i jej znaczenie w metodologii rozwiązywania

OKRES	WAŻNIEJSZE DZIAŁANIA I WYNIKI
	problemów • Zagadnienia robotyki • Opracowanie w Stanford systemu DENDRAL
1971-1980 /Pierwsze systemy wyspecjalizowane/	• Opracowanie /Uniwersytet w Stanford/ systemu eksperckiego MYCINE będącego komputerowym doradcą lekarza • Opracowanie systemu do analizy i generacji zdań języka naturalnego HEARSAY /Uniwersytet Carnegie-Mellon/ • Opracowanie narzędziowego systemu eksperckiego - języka do formułowania systemów ekspercckich - EMYCIN • Opracowanie języka programowania logicznego PROLOG /R. Kowalski - W. Brytania; A. Colmerauer - Francja/
1981 - /Znaczny wzrost nakładów na badania i scalanie przedsięwzięć/	• Rozpoczęcie prac nad komputerami V generacji /Moto-Oka, Japonia/ • Opracowanie programu ALVEY /W. Brytania/ i ESPRIT /Francja/ dotyczącego maszyn V generacji • Opracowanie programów rozwoju i intelektualizacji komputerów w ramach SM EMC i JS EMC w krajach RWPG

OKRES	WAŻNIEJSZE DZIAŁANIA I WYNIKI
	• Zespolenie firm amerykańskich /U.S.is Microelectronics Computer Technology Corp. /MCC// w programie rozwoju mikroelektroniki i technologii komputerowej • Powstanie w USA licznych firm pracujących nad narzędziami SI i ich aplikacją

✱ Podsumowanie

Systemy eksperckie są nowoczesnymi narzędziami komputerowymi, które mogą wybitnie zwiększyć wydajność działalności człowieka niemal w każdej dziedzinie. Jak każde nowe narzędzie, SE wymagają społecznej akceptacji. U podstaw wszelkich trudności leży zazwyczaj brak wiedzy o naturze nowego narzędzia, tak jest też z systemami eksperckimi: zupełnie nowe podejście do programowania /formułowania/ systemu wymaga zerwania z koncepcjami klasycznymi. Ważnym więc krokiem w przedsięwzięciu wdrażania systemów eksperckich do praktyki jest wprowadzenie SE jako przedmiotu nauczania do średnich i wyższych szkół technicznych.

Niezbędne jest też zorganizowanie w kraju kilku ośrodków naukowo-badawczych o ustalonym programie badawczym, którego celem będzie:

a opanowanie języków i systemów narzędziowych SE,
b adaptacja narzędzi programistycznych pod potrzeby problematyki wybranych branż,
c tworzenie nowych, oryginalnych narzędzi,
d szkolenie ludzi z przemysłu, służby zdrowia, szkolnictwa i innych wybranych branż w użytkowaniu systemów narzędziowych oraz w tworzeniu baz wiedzy i baz danych dla wyspecjalizowanych SE,
e wykonywanie ekspertyz dotyczących zamierzonej budowy SE, konsultacje w zakresie formułowania problemów itp.
f prowadzenie oryginalnych prac o charakterze podstawowym

zarówno w zakresie mechanizmów działania SE, jak też rozpoznawania nowych dziedzin i metod stosowania.

Obecny stan prac w kraju przedstawia się następująco. W ośrodkach takich jak IMM, IPI PAN, IPPT PAN, Politechnika Śląska powstawały małe grupy badawcze działające głównie na poziomie a. Często, po krótkim okresie działania /1 rok/ grupy te rozpadały się wobec braku finansowania pracy badawczej. Istniejące prace w zakresie SE są pozbawione wszelkiej koordynacji naukowej, brak też działań inspirujących środowisko naukowe ze strony powołanych do tego celu urzędów /np. koordynatorów centralnych programów badawczych/. Istniejący stan badań nad SE jest wyznaczony raczej chęciami i środkami samych badaczy, co zdecydowanie nie wystarcza do osiągnięcia wyższego poziomu prac badawczych i wdrożenia wyników do praktyki. Taki stan rzeczy powinien szybko zmienić się jeśli nie chcemy oddać pola już na samym wstępie. Systemy eksperckie są na początku swej drogi rozwoju. Początkowe prace badawcze i wdrożeniowe nie wymagają jeszcze nadmiernych środków obliczeniowych - wiele prac może być wykonywana za pomocą profesjonalnych mikrokomputerów, dostępnych przecież w kraju. Brak natomiast języków i narzędzi sztucznej inteligencji oraz umiejętności w posługiwaniu się nimi przez potencjalnych budowniczych i użytkowników systemów eksperckich.

Jest najwyższy czas do podjęcia stosownych kroków organizacyjnych na rzecz systematycznych prac badawczych i wdrożeniowych nowej technologii.

LITERATURA

[1] Leibnitz G.W.: Nowe rozważania dotyczące rozumu ludzkiego. /tłum. z franc./ PWN Warszawa 1955.

[2] Newell A., Simon H.A.: Human Problem Solving. Prentice-Hall, Inc., Englewood Cliffs 1972.

[3] Newell A., Simon H.A.: Computer reience as empirical inquiry: Symbols and search. Comm. ACM, 1976 t. 19 nr 3 s. 113-126.

[4] Loftus G.R., Loftus E.P.: Human memory: The processing of information. New York 1976.

[5] Lindsay P.P., Norman D.A.: Procesy przetwarzania informacji u człowieka, wprowadzenie do psychologii / tłum. z angielskiego /. PWN Warszawa 1984.

[6] Anderson J.R., Bower G.H.: Human Associative Memory. Holt, New York 1973.

[7] Wiener N.: Cybernetics. Sci. Americain, 1948 t. 179.

[8] Turing A.M.: Computing machinery and intelligence. Mind, 1950 t. 59 s. 433-450.

[9] Newell A., Simon H.A.: GPS, a program that simulates human thought. W: Feigenbaum zob.poz 17.

[10] Chang C.L., Lee R.C.T.: Symbolic logic and Mechanical Theorem Proving. Academic Press, New York 1973.

[11] McCarthy J.: History of LISP. SIGPLAN Notices, t. 13 s. 217-223.

[12] Waterman D.A., Hayes-Roth F.: /Eds./ Pattern Directed Inference Systems. Academic Press, New York 1978.

[13] Brady M.: Artificial intelligence and robots. Art. Intell. 1985 t. 26 nr 1 s. 79-121.

[14] Nilsson N.J.: Principles of Artificial Intelligence. Palo Alto 1980.

[15] Shortliffe E.H.: Computer-based Medical Consultations: MYCINE. Am. Elsevier, New York 1976.

[16] Wrzeszcz Z.: A model of complex job control computer systems. Prace naukowo-badawcze Instytutu Maszyn Matematycznych, I 1986 R. 28 s. 3-29.

[17] Feigenbaum E.A. et al.: The Handbook of Artificial Intelligence, Vol. 1-3, Heuristech Press, W. Kaufman, Inc., Stanford SA, 1981.

[18] Kowalski R., Logic programming. w: Informatic: Processing 83, R.E.A. Hason /Ed/. North-Holland 1983 s. 133-145.

[19] McDermott J.: R1: A Rule-based configurer of computer system Art. Intell. 1982 t. 19 nr 1 s. 39-88.

[20] Hayes-Roth et al. /Eds./: Building Expert Systems, Addison-Wesley, London 1983.

[21] Burns N.A. et al.: The portable inference engines: fiting significant expertise into small systems. IBM System Journal 1986 t. 25 nr 2 s. 256-283.

[22] Buchanan B.G., Feigenbaum E.A.: DENDRAL and Meta-DENDRAL: Their application dimension. Art. Intell. 1978 t. 11 s. 5-24.

[23] Gottschalk G.R., Vandcon R.M.: A rule-based system to diagnose malfuctioning computer peripherals. Hewlett-Packard Journal 1986 Nov. s. 48-53.

[24] Fagen J.M.: VM: Representing time-dependet relations in a medical setting. Stanford Univ., Stanford, June 1980.

[25] Stefik J., Planning with constraints. Art. Intell. 1981 t. 10 nr 2 s. 111-140.

[26] Marks K.M., Goser K.E.: An expert system shell for standerdization of VLSI process data base and knowledge base. Microprocessing and Microprogramming 1987 t. 21 s. 523-530.

[27] Sleeman D., Brown J.S.: Intelligent Tutoring Systems, Academic Press, London 1982.

[28] Taylor J.H., Frederick D.K.: An expert system architecture for computer-aided control engineering. Proc. IEEE 1984 t. 72 nr 1 s. 1795-1805.

[29] Parker R.: An expert for every office. Computer Design 1983 s. 57-46

[30] Stefik M. et al.: The organization of expert systems, a tutorial. AI 1982 t. 18 nr 2 s. 135-173.

[31] Sowa J.F.: Conceptual Structures, Addison-Wesley, Reading 1984.

[32] Clancy W.J.: The epistemology of a rule-based expert system- a framework for explanation. AI 1983 t. 20 nr 3 s. 215-251.

[33] Harman P., King D.: Expert Systems - Artificial Intelligence in Business, J. Wiley, New York 1985.

[34] Wrzeszcz Z. i in.: Podstawy implementacji systemów eksperckich na mikrokomputerze. Prace naukowo-badawcze IMM 1987 t. 29 nr 1.

[35] Rich E.: Artificial Intelligence. McGraw-Hill, New York 1983.

[36] Quillian M.R.: Semantic Memory. Bolt Beranek Newman, Cambridge, Ma., 1966.

[37] Bratko I.: PROLOG Programming for Artificial Intelligence. Addison-Wesley Wokingham 1986.

[38] Minsky M.: The Psychology of Computer Vision. P.H. Winston /Ed./, McGraw-Hill, New York 1975.

[39] Corlett R.A., Features of artificial intelligence languages and their environments. Software Engineering Journal 1986 t. 1 nr 1 s. 159-164.

[40] Sheil B.: Family of personal LISP machines speeds AI program development, Electronics 1983 t. 3 nov. s. 153-156.

[41] Manuel T.: LISP and PROLOG machines are proliferating. Electronics 1983 t. 3 nov. s. 132-137.

[42] Colmerauer A.: PROLOG in ten figures, Comm, ACM, 1985 t. 28 dec. s. 1296-1310

[43] Turbo PROLOG - The Natural Language of Artificial Intelligence, Borland International, Inc., Scots Valey, CA. 1986.

[44] Clocksin W.F., Mellish C.S.: Programming in PROLOG, Springer Verlag, Berlin 1984.

[45] Winston P.H., Horn B.K.P.: LISP, Addison-Wesley, Reading, Ma., 1984.

[46] Wilensky R.: Common LISPcraft, W.W. Norton New York 1986.

[47] McCarthy J.: History of LISP. SIGPLAN Notices, t. 13 s. 217-223.

[48] Steele G.L. Jr.: Common LISP-The Language, Digital Press, Burlington, Ma. 1984.

[49] Yuasa T., Hagiya M.: Introduction to Common LISP, Academic Press, Boston 1986.

[50] Waterman D. .: A Guide to Expert Systems. Addison-Wesley, Reading /MA/ 1986.
[51] Forgy C.L.: OPS5 User's Manual. Carnegie Mellon University 1981.
[52] Forgy C.L.: The OPS83 Report. Cornegie-Mellon University
[53] INSIGHT2, Level 5 Research, Inc., Indialantic Florida
[54] M.I. Technoledge, Inc., Palo Alto, California
[55] Badźmirowski K.: Systemy ekspertowe. PIE, 1988 /opr. wewn/.
[56] Ennals J.R.: Artificial Intelligence - State of the Art Report. Pergamon Infotech. Ltd., Oxford 1987.
[57] Robotics /ISSN 0167-8493/, 1986 t. 2 nr 4
[58] Michalski R.S. et al.: An Artificial Intelligence Approach. TIOGA Publ. Co., Los Altos 1983.

NOWOŚCI TECHNICZNE

Pierwsze mikrokomputery francuskie zgodne z PS 2

Firma Normerel z Granville oferuje rodzinę mikrokomputerów NS, stanowiących odpowiedniki PS/2. Oferta ta pojawiła się 2 miesiące po ogłoszeniu pierwszych europejskich odpowiedników, jakimi były brytyjskie Apricot.

Rodzina ta obejmuje cztery modele. Najmniejszy z nich to NS30, zgodny z PS 30, przeznaczony dla użytkowników, którzy przechodzą z PC na PS. Zbudowany na mikroprocesorze 8086 o częstotliwości zegara 10 MHz, posiada magistralę standardową typu AT. Na karcie głównej mieści się pamięć operacyjna 640 Kbajty, sterownik dysku 3,5 cala o pojemności od 20 do 84 Mbajtów o sprzęgu SCSI, sterownik dysku elastycznego 3,5 lub 5,25 cala o pojemności od 360 K do 1,44 Mbajta i sterownik kart CGA, MDA, Hercules, EGA i VGA. Podstawowa cena wynosi 18,4 tys. franków.

Następny model NS 50 jest zgodny z IBM 8550 i zbudowany na mikroprocesorze 80286 z częstotliwością zegara 16 MHz. Posiada on pamięć operacyjną 1 Mbajt, którą można rozszerzyć do 4 Mbajtów. Firma opracowała własną wersję systemu BIOS, który jest wielojęzyczny /można wybrać francuski, angielski, niemiecki i hiszpański/. Konfiguracja podstawowa z dyskiem o pojemności 40 Mbajtów i sprzęgiem CSSI oraz ekranem kolorowym o przekątnej 14 cali i klawiaturą kosztuje 32400 franków. Pośredni model, NS65 wykorzystuje mikroprocesor 80386 SX o częstotliwości zegara 16 MHz. Podobnie jak w NS50 wykorzystane są tu podzespoły produkcji firmy Chips & Technologies. Za 35 400 franków oferowana jest konfiguracja podstawowa zbliżona do NS50.

Wreszcie model największy NS 70 zbudowany na mikroprocesorze 80386 o częstotliwości zegara 20 MHz, posiada pamięć operacyjną o pojemności 1 Mbajta, dysk twardy o pojemności 84 Mbajty oraz takie jak w poprzednich modelach ekran i klawiaturę. Cena wynosi 46 350 franków.

Jednocześnie Normerel oferuje nowe odpowiedniki AT z kolorowymi monitorami o przekątnej 14 cali. ATC 12 wykorzystuje mikroprocesor 80286 o częstotliwości zegara 12,5 MHz, pamięć operacyjną 640 Kbajtów, dysk 20 Mbajtów i klawiaturę o 102 przyciskach. Kosztuje 25 400 franków. Pozostałe modele zbudowane są na mikroprocesorze 80386 SX o częstotliwości zegara 16 lub 20 MHz z pamięcią operacyjną o pojemności 1 Mbajta. Największy z nich to Tower ATM 386 z dyskiem twardym o pojemności 140 Mbajtów. Ceny wahają się tu od 33 400 do 69 tysięcy franków za konfigurację podstawową.

Najmniejszy z oferowanych mikrokomputerów to Turbo ST 2 zgodny z XT, wyposażony w procesor Nec V 40 o częstotliwości zegara 8 MHz i sprzedawany za 12 600 franków w konfiguracji ze stacją dysków elastycznych o średnicy 5,25 cala i pojemności 640 Kbajtów oraz monitorem monochromatycznym o przekątnej 12 cali.

Wśród modeli przenośnych należy wyróżnić MT 286 zbudowany na procesorze 80286 o częstotliwości zegara 12 MHz z pamięcią operacyjną o pojemności 1 Mbajta, zawierający dysk twardy o pojemności 20 Mbajtów i czytnik dyskietek o pojemności 1,44 Mbajta oraz ekran ciekłokrystaliczny. Jest on wyposażony w baterię, zapewniającą 3-godzinną pracę bez zasilania. Cena wynosi 28 tys. franków.

Oferowane są też monitory o przekątnej 28 cali Mega 70 /za 15 tys. franków i Vega 70 /za 16 000/ zawierające płaskie kineskopy FST /firmy Philips/ działające zarówno w systemie EGA, jak i VGA.

Normerel zawarł szereg kontraktów z takimi firmami, jak Xios Système, Sedri /Société d'Etude de Développement et de Recherche Industrielle/, Rank Xerox i Memorex Telex na dostawę poszczególnych wyrobów.

W Granville powstał nowy ośrodek, gdzie opracowywane są prototypy i serie próbne. Firma potwierdziła dążenie do opracowania niektórych własnych układów scalonych opartych na technologii C-Mos i wymiarze podstawowym 1,5 µm. Wytwarzany ma być zestaw układów dla odpowiedników PS/2.

Około 80% dochodów firmy pochodzi z eksportu. W roku 1988 obroty wzrosły o 64% w stosunku do roku ubiegłego i osiągnęły 320 miliony franków, a w roku bieżącym powinny przekroczyć pół miliarda. Natomiast jeśli chodzi o ilość, to w r. 1988 wytworzono 35 tysięcy komputerów, a w roku 1989 wielkość ta powinna się podwoić.

Minis a. Micros nr 314/89

Intel we Francji

Czołowa firma mikroprocesorowa Intel podpisała kontrakt na sprzedaż swych wyrobów we Francji ze spółką Almex, należącą do grupy Sonepar Electronique. Jest to czwarty dystrybutor Intela na rynku francuskim po Takelec Airtronic, Métrologie i Jermyn- Generim, przy czym nadal utrzymana jest zasada posługiwania się niewielką ilością sprawnych firm współpracujących. Kontrakt dotyczy elementów elektronicznych i narzędzi wspomagających.

Almex ma już doświadczenie w sprzedaży półprzewodników przez wykwalifikowany personel, przy czym ponad połowa sieci dystrybucyjnej tej firmy dotyczyła regionu paryskiego. Obroty wynosiły 130 milionów franków, z czego elementy aktywne stanowiły 74%, bierne 18%, a resztę systemy. Oferowano wyroby nowoczesnej technologii specjalizując się szczególnie w pamięciach stałych typu EPROM, układach EPLD, mikrosterownikach, mikroprocesorach i układach peryferyjnych wg norm EISA.

Przewiduje się zmiany w strukturze przedsiębiorstwa przez utworzenie trzech stanowisk inżynieryjnych do zastosowań, z których dwa związane są z wyrobami Intela, oraz czterech stanowisk techniczno-handlowych. Ponadto Almex, przy pomocy Sonepar powiększa moje możliwości w zakresie narzędzi programowych i zastosowań. Opracowane w przedsiębiorstwie oprogramowanie systemu IBM 38, zapewniające zarządzanie firmą w czasie rzeczywistym, pozwala na dołączenie systemu do sieci europejskiej EDI. Spółka Almex utworzona została w 1971 roku i weszła do grupy Sonepar w roku 1976, z obrotami liczącymi wówczas 13 milionów franków. W roku 1988 obroty wynosiły 150 milionów, a przedsiębiorstwo posiadało swe oddziały w Paryżu, Tuluzie, Rennes i Lionie. Piąty oddział otwarty będzie w Bordeaux. Realizuje on 32 tysiące zamówień rocznie a jego zapasy, oceniane na 35 mln franków, zawierają 20 tys.pozycji. Cała grupa SONEPAR miała obroty w 1988 r. 1,11 mld franków w porównaniu do 0,88 mld w 1987 r.

Minis a. Micros nr 314/89

Szybka pamięć buforowa

Brytyjska firma Magus Electronics Ltd z Wheelock opracowała system akwizycji i przechowywania danych, który pozwala na szybkie przyjmowanie, przechowywanie i przetwarzanie przed przesłaniem danych do komputera obliczeniowego lub plotera.

Urządzenie o nazwie Data Buffer^{+}, Model DM Plus 600 zaprojektowane jest do odbierania danych binarnych w kodzie ASCII do specjalizowanej pamięci operacyjnej o pojemności 590 KB, a oprogramowanie sterujące jest dopasowane do zastosowań użytkownika. Pozwala również na dokonanie analizy matematycznej danych przed ich przesłaniem do komputera obliczeniowego. Zależnie od zastosowań analiza ta pozwala na uzyskanie danych dotyczących takich fizycznych parametrów, jak ciśnienie, przesunięcie, prędkość, moment obrotowy lub może dostarczyć informacji statystycznych i numerycznych.

System zasilany jest napięciem zmiennym 220-240 V, 50 Hz /opcjonalne ustawienie wewnętrzne 110 v/, wyposażony też jest w wewnętrzną baterię zapewniającą 10 dni ciągłej pracy. Dołączenie baterii w przypadku uszkodzenia zasilania jest natychmiastowe i automatyczne, bez żadnego ryzyka utraty danych.

London Press Service NPFB 161188

Lokalna sieć mikrokomputerowa

Brytyjska firma Madge Networks z Chulfont St Giles opracowała sieć lokalną mikrokomputerów, którą daje się łatwo sterować i rozbudowywać. Ma ona strukturę pierścieniową i działa na zasadzie przekazywania uprawnień. Może zawierać zgodne ze standardami IBM mikrokomputery lub układy peryferyjne. Tworzenie sieci następuje przez pierścieniowy węzeł dla każdego stanowiska /możliwe jest połączenie 255 takich węzłów/ lub serwera, dołączony przez stałe okablowanie w postaci skrętki lub światłowodu, albo przez pierścieniowe rozgałęźniki na cztery lub osiem stacji.

Adaptery te zgodne są ze standardowymi sieciowymi systemami operacyjnymi takich firm, jak IBM i Novell, a także z systemem sieciowym dużych maszyn SNA /System Network Architecture - architektura sieci systemów/.

Istnieją trzy typy węzłów pierścieniowych: pierwszy dla działania standardowego stanowiska pracy, drugi dla stanowiska o zwiększonej wydajności lub serwera i trzeci dla działania serwera. Te układy dostępu do pamięci nie wymagają zewnętrznego źródła zasilania.

Rozgałęźnik na cztery stanowiska może tworzyć lokalny pierścień lub rozszerzać istniejącą sieć. Może on być umieszczony na podłodze, na biurku lub na ścianie. Natomiast rozgałęźnik dla ośmiu stanowisk montowany jest w stojaku. Rozszerzenie sieci jest szybkie - dodanie nowego komputera zajmuje zaledwie 10 minut. Rozłączania i dołączania możliwe są podczas pracy sieci.

Taka sieć pierścieniowa z przekazywaniem uprawnień działa ze standardową prędkością przesyłania danych 4 Mbajty/s. Oprogramowanie jest zoptymalizowane do szybkiego działania, które może być dwukrotnie szybsze w stosunku do poprzednich systemów.

Programy zastosowaniowe dla sieci standardowych, a zwykle i programy bez specjalnych dostosowań sieciowych, mogą być realizowane na "Madge token ring" bez modyfikacji.

London Press Service NPFB: 161188

Cenne uzupełnienia systemu operacyjnego OS/2

Pod koniec 1988 r. pojawiły się standardowe i rozszerzone redakcje wersji 1.1 systemu OS/2, które obejmują dwa interesujące programy zarządzające. Pierwszy z nich to Presentation Manager występujący w redakcji standardowej. Najczęściej ocenia się go jako odpowiednik programu Windows, lecz nie wyczerpuje to jego zalet. Działanie z tym programem odbywa się na poziomie systemowym i daje dostęp zarówno do programów aplikacyjnych, jak i systemu operacyjnego. Jest on programem osłonowym /schell/, który wywoływany jest zleceniem PROTSHELL ze zbioru OS/2 CONFIG SYS. Po wystartowaniu występujemy w dwóch oknach: Task Manager i Start Programs. Są to specjalne okna systemu operacyjnego, które sterują sesją i nie mogą być usunięte. Okno Start Programs pozwala rozpocząć nowe zadania, a Task Manager pozwala przełączać się między nimi. W oknie Start Programs znajduje się lista zadań, które mogą być rozpoczęte przez wskazanie i podwójne przyciśnięcie myszą, bądź wybranie Start z menu. Na liście tej znajdują się zawsze dwa zadania: OS/2 Command Prompt i OS/2 Windowed Command Prompt. Pozwalają one na wprowadzanie szybkich komend zarówno na całym ekranie, jak i w oknach. Można również wybrać DOS Command Prompt, lecz wówczas jest się ograniczonym do jednej sesji. Dodawanie nowych programów do listy w oknie jest bardzo łatwe. Zadania mogą być grupowane w oknie Start Programs i łatwo można się przełączać z grupy do grupy.

Lista zadań zawiera również dwa kontrolne programy sterujące: PM Control Panel i PM Filing System. Pierwszy z nich pozwala użytkownikowi sterować parametrami systemu, takimi jak konfiguracja portów, barwy ekranu, rodzaje czcionek. Drugi natomiast bardzo nam ułatwia zarządzanie zbiorami. Możemy wówczas poruszać się po drzewie skorowidzów - przesyłać, kopiować i usuwać zbiory oraz części skorowidzów za pomocą myszy. Można też grupować zbiory, zmieniać ich atrybuty i sortować.

Okno Task Manager obejmuje tylko te zadania, które są właśnie obliczane. Przełączanie pomiędzy aktywnymi zadaniami może być dokonywane przez to okno lub przez wybór okna zadań myszą. Można też zamykać okna aktywnych zadań. Menu Task Managera zawiera inne rozkazy do wyłączania systemu lub ustawiania okien na ekranie. To wyłączenie może być wykorzystane do zabezpieczenia zadań aktywnych. Mogą one być ponownie wykonywane po wznowieniu pracy.

Wykorzystanie programów aplikacyjnych w Presentation Manager jest proste, natomiast trudniej jest z ich pisaniem. Lepiej tu radzą sobie programiści, którzy pracowali pod programem Windows, aniżeli zwykli użytkownicy środowiska DOS. Programy wykorzystujące funkcje jądra OS/2 dla wejścia-wyjścia /funkcje rozpoczynające się na VIO, KBO i MOV/ realizowane są szybko i bez poprawek ale nie wykorzystuje się wówczas sprzęgu graficznego.

Programy aplikacyjne izolowane są od bezpośredniego kontaktu z użytkownikami przez API /Application Program Interface - sprzęg programu aplikacyjnego/, który zapewnia wszystkim zastosowaniom dostęp do systemu, tak jak sam Presentation Manager czyni to z użytkownikami. Z punktu widzenia programu istnieją trzy rodzaje wyjścia i jeden wejścia. Program użytkowy może zapisywać tekst, obraz graficzny lub informację dialogową do przestrzeni przedstawianej /presentation space/, która przez ten program może być związana z monitorem, drukarką itp. Natomiast jedynym źródłem wejściowym do programu jest kolejka komunikatów, gdzie API posyła komunikaty do programu o działaniach użytkownika.

Najczęściej używa się jako wyjścia monitora. Obrazy i tekst mogą być wyprowadzane na dowolne urządzenie, natomiast informacja dialogowa tylko na monitor. Wszystkie wejścia do programu przechodzą przez kolejkę komunikatów. Każde okno na ekranie ma swą własną kolejkę, która zawiera informację specyficzną dla tego okna, np. gdy użytkownik zmienia wymiary okna za pomocą myszy, Presentation Manager przesyła komunikat do kolejki tego okna o tej zmianie. Podobne komunikaty posyła się przy przesuwaniu, minimalizowaniu czy usuwaniu okna. Okna mogą przesyłać komunikaty między sobą.

Ponieważ architektura oprogramowania opiera się na komunikatach - pisanie programów użytkowych w Presentation Manager jest bardzo podobne do tej czynności w Windows. Tworzenie programu użytkowego zaczyna się od określenia klasy okna i jego narysowania. Okno to musi utworzyć swą kolejkę komunikatów. Programy te mogą być pisane w języku wysokiego poziomu ponieważ duży zasób funkcji i definicji standardowych dostarcza zestaw wspomagania oprogramowania /Software Development Kit/ firmy Microsoft.

O ile Presentation Manager jest znacznym krokiem w kierunku opracowania SAA /System Application Architecture - architektura zastosowania systemów/ to głównym celem ich poczynań jest standardowy sprzęg zarówno dla użytkownika, jak i programisty, niezależnie od zastosowania i sprzętu. Prowadzą do tego usprawnienia systemu operacyjnego, które dla użytkownika są jakby jeszcze jednym zastosowaniem. Ostatnim z nich jest LAN Manager. Presentation Manager łączył OS/2 z użytkownikiem, LAN Manager łączy użytkowników ze sobą. Obiecując wielozadaniowość i niewidoczne dzielenie zasobów, OS/2 reprezentuje właściwości, które dawniej możliwe były do uzyskania tylko za pomocą dużych maszyn. Obecnie komputery osobiste wyposażone są w odpowiedni sprzęt, a kolejne kroki w dziedzinie oprogramowania umożliwiają im spełnienie tej roli.

LAN Manager oferuje sprzęt podobny, jak w Presentation Manager z hierarchicznym systemem menu. Sprzęg użytkownika dzieli się na cztery części: View menu, Message menu, Config menu i Status menu, a zarządzający systemem ma jeszcze dostęp do piątego menu - do konserwacji systemu. Użytkownik może też pobudzać LAN Manager bezpośrednio z linii rozkazów. Pozwala to nie tylko zbiorom automatycznie sterować ciągami, ale zapewnia też zgodność z innym oprogramowaniem sieciowym, jak PC-NET i MS-NET. Dołączony do dzielonych zasobów LAN Manager może przełączać się do środowiska DOS i realizować jego zastosowania.

Użytkownik nie zdaje sobie sprawy ile funkcji spełnia za niego API zapewniając sprawne działanie i pełną "przezroczystość" systemu. Jeśli zapomni on czegoś zrobić lub czyni to w złej kolejności API poprawia go, o ile jest to możliwe. Sprzęg ten wykorzystują również stanowiska pracy.

Głównym celem LAN jest dzielenie zasobów, niezależnie od tego czy są to komunikaty, zbiory, czy urządzenia fizyczne. Administrator LAN Managera musi zdecydować, kto dzieli te zasoby oraz kiedy i jak będą one dzielone. Wykorzystując zlecenie Add Share /dodaj udział/z View menu, administrator może wybierać zasoby do dzielenia, przyporządkowywać nazwę przydziału, wyznaczać sposób pobudzania i drogę jeśli jest to niezbędne, ograniczać liczbę użytkowników, którzy mają dostęp do zasobu, wyznaczyć hasło i określić, jaki zasób może być użyty.

W wypadku dzielenia drukarek lub urządzeń komunikacyjnych LAN Manager ustala kolejkę zamówień. Administrator może dopisywać do kolejki, skreślać i przekształcać ją, co pozwala mu utrzymać pełną kontrolę nad zamówieniami i dostępem do urządzeń. Może on też ustalać grupy urządzeń i przesyłać zadania do pierwszego dostępnego urządzenia w grupie.

LAN Menager ułatwia dostęp do urządzeń i ustalenia kolejności przez możliwość utworzenia kilku kolejnych zamówień na poszczególne urządzenia lub grupę urządzeń. Kolejki te, z których każda ma przyporządkowany różny poziom priorytetu,czekają na dostępne urządzenie. Zadania o identycznym priorytecie są realizowane w kolejności zgłoszenia. Zamówienia w kolejce mogą być planowane w określonym czasie.

Sieć może dzielić również przestrzeń dyskową. Administrator może wyznaczyć całe katalogi do dzielenia lub ograniczyć dostęp do określonych zbiorów. Dla zwiększenia bezpieczeństwa administrator może określić, jakie funkcje wykonuje każdy użytkownik. Jeśli kilku użytkowników ma dostęp do katalogu dzielonego dysku, to niektórzy z nich mogą tylko zapisywać zbiory do katalogu, podczas gdy inni mogą usuwać zbiory lub zmieniać ich atrybuty.

Poza odczytywaniem i zapisywaniem zbiorów, użytkownik w uzgodnieniu z administratorem może wykonywać zdalnie programy w pamięci serwera. Zbiór poprawiony przez jednego użytkownika pozostaje w dzielonym katalogu i inni mogą korzystać z niego lub go modyfikować.

LAN Manager obejmuje też usługę przesyłania komunikatów tak, że użytkownicy mogą wysyłać i odbierać komunikaty za pomocą sieci. Administrator może posyłać komunikaty do grupy użytkowników lub do każdego użytkownika sieci. Każde stanowisko robocze ma przyporządkowaną nazwę, pod którą przysyłane są komunikaty, oczkolwiek komunikaty mogą być też wysyłane według nazw komputerowych. Opcjonalne zlecenie Pop Up umożliwia natychmiastowe wyświetlenie otrzymanego komunikatu. Pojawia się on wtedy w ramce komunikatów stacji odbiorczej. Jakkolwiek LAN Manager nie obejmuje jeszcze edytora tekstu do tworzenia komunikatu, obsługa komunikatu przesyła zbiory i krótkie uwagi, a użytkownik może utworzyć komunikat na dowolnym procesorze tekstu i przekazać zbiór po sieci. Może też zapamiętywać swoje komunikaty automatycznie w dzienniku komunikatów.

Dalsze usprawnienia LAN Manager i Presentation Manager niewątpliwie wkrótce się pojawią, jednakże podstawowe elementy OS/2 mamy już do dyspozycji. Ocena ich należy do użytkowników, którzy powinni określić, na ile są one pomocne w codziennym działaniu.

Byte, październik 1988

Automatyczne przenoszenie wyrazów w 26 językach

Oprogramowanie opracowane przez firmę Computer Hyphenetion z Bradford pozwala na poprawne dzielenie i łączenie wyrazów w 26 językach przy przygotowaniu tekstu do druku. Dla każdego języka realizuje ono algorytm dzielenia wyrazów, nie pozwalając np. na podział dwóch spółgłosek tworzących jeden dźwięk. Podział określony jest zestawem standardowych reguł odpowiadających używanemu językowi, przy czym możliwe jest utworzenie listy wyjątków od tych reguł. Program ten, zwany Hyphenologist, wbudowany w istniejący program przetwarzania tekstu, zajmuje zwykle cztery do ośmiu i pół kilobajta pamięci komputera, choć znane skomplikowanie podziału w języku angielskim powoduje, że w odniesieniu do tego języka wymaga się aż 35 kilobajtów.

Oprogramowanie to, składające się z podzespołów prostych programów pisanych w języku C, może być stosowane w dowolnym systemie komputerowym. Uwzględnia ono odrębne reguły dla języka angielskiego stosowanego w USA, gdzie dzielenie wyrazów jest fonetyczne, według głosek i akcentów wymawianego słowa, podczas gdy słowa brytyjskie dzielone są na podstawie etymologii.

Wersja dla każdego języka zawiera własną bazę składającą się z wielu setek reguł. Można też łączyć dwa lub więcej języków do pracy wielojęzycznej. Zwykle podział następuje co 3-4 litery w wyrazie; taka możliwość wyboru w długich wyrazach sprawia, iż współczynnik uprzywilejowania pozwala głównemu programowi dokonać odpowiedniego wyboru.

London Press Service C8A0788

Oprogramowanie obecności i czasu pracy pracowników

Zapis czasu pracy i obecności pracowników może odbywać się automatycznie. Oprogramowanie Ganetime opracowane w firmie Krypton Computers dostosowane jest do każdego rodzaju przedsiębiorstwa z dowolną liczbą pracowników i zgodne z oprogramowaniem list płac.

System pozwala na natychmiastowe uzyskanie zgromadzonych informacji według różnych kryteriów. Można też uzyskać dodatkowe informacje jak np. terminy urlopów, co pozwoli na zachowanie

ciągłości pracy. System działa na dowolnym komputerze dostosowanym do środowiska Unix. Obecnie konwersacja z maszyną przeprowadzana jest w języku angielskim, ale przygotowywane są tłumaczenia na inne języki. System jest prosty i nie wymaga nawet podręcznika. Na ekranie pojawiają się wszystkie potrzebne wskazówki.

London Press Service NPFB:040788

Brytyjskie oprogramowanie dla sputnika Ulysses

W październiku 1990 roku ma zostać wystrzelony europejski statek kosmiczny Ulysses, który uda się w 5-letnią podróż w okolice Jowisza i Słońca. Jest to wspólne przedsięwzięcie Europejskiej Agencji Kosmicznej /European Space Agency-ESA/ i NASA.

Do sterowania krytycznymi ruchami tego statku wykorzystany będzie pakiet oprogramowania opracowany przez brytyjską firmę Science Systems Ltd z Bristolu. Steruje on dynamiką lotu ustalając położenie i orbitę statku, toteż jego praktyczne sprawdzenie nastąpi zaraz po wejściu statku na właściwą orbitę. Innym złożonym ruchem będzie dokonany, przy istniejącej grawitacji, manewr nad południowym biegunem Jowisza. Manewr ten ma wprowadzić statek na orbitę przebiegającą nad biegunami Słońca.

Kontrakt na oprogramowanie opiewa na 400 tysięcy funtów i dotyczy konwersji oryginalnego oprogramowania dynamiki lotu komputera obliczeniowego, gdy wystąpią trudności z jego podtrzymywaniem pod koniec misji. Przekształcone oprogramowanie realizowane będzie na komputerach VAX i sterować będzie położeniem i orbitą statku w czasie krytycznego manewrowania. Zespół opracowujący zrealizuje większość zadań w Bristolu, a przed startem uda się do Jet Propulsion Laboratory w Pasadenie celem doprowadzenia przygotowanego oprogramowania do systemów komputerowych tego laboratorium i ewentualnego udzielenia pomocy w pierwszych fazach misji.

Science System utworzona została przed ośmiu laty. Wyposażyła w oprogramowanie wiele statków kosmicznych m.in. GIOTTO, który został wysłany na spotkanie komety Halley'a, a także dla aparatu Hipparchos, który sporządzał mapę gwiazd. Jeden z pierwszych kontraktów firmy zawarty został z Europejskim Centrum Operacji Kosmicznych w Darmstadt /RFN/ i dotyczył prac z dziedziny telemetrii. Inne prace obejmują szeroki zakres działań: od sterowania orbitami statków kosmicznych do sterowania falami na platformach naftowych Morza Północnego.

London Press Service 05BIN0888

Urządzenie do wyszukiwania błędów komputera

Bardzo elastyczne, przenośne urządzenie do testowania komputerów o nazwie "Ferret Tester" opracowała brytyjska firma New Leaf Technology Ltd. z Cobham.

Zaprojektowane ono zostało, celem skrócenia czasu obsługi i szkolenia personelu w szerokim zakresie urządzeń peryferyjnych. Testowania dokonywać można za pomocą sprzęgu RS 232, pętli prądowej i sprzęgu równoległego z wykorzystaniem programowania pamięci stałej EPROM.

Sprawdzający dołącza po prostu Ferret do badanego sprzętu i wprowadza z klawiatury proste ciągi znaków wytwarzając testy diagnostyczne na drukarkę, ploter czy monitor ekranowy.

Oprogramowanie do testowania dowolnego wyrobu może być przygotowane na IBM PC lub zgodnym z nim komputerze dołączonym przez port RS232 do Ferreta, a następnie wprowadzone do pamięci EPROM. Łatwo można wykonać kopie tych pamięci i rozprowadzić je wśród użytkowników, co pozwoli im, przy minimalnym szkoleniu personelu testować dowolny wyrób.

Całe urządzenie ma wymiary 390 x 245 x 190 mm i waży 6,5 kg. Może być zasilane napięciem zmiennym jednofazowym 110 - 240v, 47 - 63 Hz.

London Press Service NPFB:230888

Urządzenia do oczyszczenia głowic taśmowych

Zabrudzenie głowic taśmowych powoduje 80% błędów zapisu-odczytu; głowice te mogą być oczyszczone bez uszkodzenia powierzchni za pomocą urządzenia "Streamolene", opracowanego przez Automation Facilities z Wargrave. Jest to automatyczny system oczyszczenia głowic /napędzany kasetą/, który usuwa zanieczyszczenia. Kaseta zaś napędzana jest przyciągarką w samej stacji, co eliminuje potrzebę zastosowania prętów i dźwigni. Drgające ramię wycieraczki zapewnia zwiększone tarcie gdy wkładka osiągnie środek głowicy, po czym znów

następuje zmniejszenie tarcia. Głowica nie jest więc poddawana ciągłemu naciskowi na początku każdego przesunięcia.

Ponieważ promieniowe ramię wycierające przesuwa wkładkę w tym samym kierunku co droga taśmy, zanieczyszczenia usuwane są z całej powierzchni głowicy. Wkładki czyszczące wykonane są z elastycznej, odpornej na ścieranie gąbki, która przylega ściśle do zarysów głowicy zapewniając całkowite oczyszczenie przy minimalizowaniu uderzenia na zespół głowicy.

Kaseta zatopiona w specjalnym materiale plastycznym jest zgodna ze wszystkimi stacjami typu QIC.

London Press System NPFB:130988

Projektowanie nowych lekarstw za pomocą komputera

Naukowcy z Uniwersytetu w Manchester pracują nad systemem komputerowym zdolnym do projektowania nowych lekarstw. System, który jest prawdopodobnie pierwszym tego rodzaju, wykorzystuje superkomputer wykonujący miliardy obliczeń w ciągu sekundy. Eliminuje to kosztowną metodę zgadywania, którą konwencjonalnie stosuje się przy opracowywaniu nowych lekarstw.

Naukowcy z Manchester nazwali swój system "Prometheus" i powołali specjalne przedsiębiorstwo Proteus Biotechnology, w którym wykorzystuje się prace w zakresie matematycznego modelowania wzajemnego oddziaływania na siebie lekarstw w organiźmie pacjenta.

Złożone molekuły, jakie tworzą lekarstwa, oddziałują na siebie odpowiednio do ich składu chemicznego i kształtu. O ile skład chemiczny wielu takich związków został wyjaśniony, o tyle, jak dotychczas udało się zmierzyć kształty tylko kilku z nich.

Kody komputerowe, opracowane przez dr. Barry Robsona, teoretyka biochemii, i jego kolegów w przedsiębiorstwie Proteus, pozwalają chemikom określić kształt molekuł z ich składu chemicznego. Pozwala to na komponowanie lekarstw o specjalnych własnościach "na ekranie", przy czym chemicy zmieniają skład i kształt dla osiągnięcia właściwego poziomu oddziaływania ze związkami docelowymi.

Konwencjonalne opracowywanie lekarstw wymaga tysięcy godzin prac laboratoryjnych, podczas których zmienia się własności znanych lekarstw i sprawdza wyniki. John Pool, dyrektor firmy Proteus, powiedział, że jedna godzina pracy superkomputera jest równoważna 50 tysiącom godzin prac eksperymentalnych.

System ten, nazwany "Bio-engine", będzie oferowany wielkim firmom farmaceutycznym, które ciągle szukają szybkich sposobów produkcji nowych lekarstw. Większość potencjalnych użytkowników systemu firma poszukuje w Ameryce i Japonii.

Zespół Proteus opatentował już wiele związków projektowanych przy użyciu Bio-engine. Pool powiedział, że jedna z firm farmaceutycznych, która obecnie zainteresowana jest Bio-engine, wydała prawie 2 miliony funtów na opracowanie podobnego lekarstwa, lecz bez pozytywnego rezultatu.

London Press Service 02A0988

Pierwsza olimpiada komputerowa w Londynie

Od 9 do 15 sierpnia 1989 roku w Park Lane Hotel w Londynie odbędzie się olimpiada inteligentnych gier komputerowych. Organizatorem wystawy jest David Levy, prezes Międzynarodowego Zrzeszenia Szachów Komputerowych. Zapowiedział on, że zaprezentowanych zostanie najmniej 20 gier strategicznych, w tym trzy wersje szachów. Dla każdej z gier odbędzie się odrębny turniej i w każdym z nich uczestnikami będą programy realizowane na mikroprocesorach. Ludzie uczestniczący w turnieju będą działać jedynie jak roboty do wprowadzania ruchów komputerów lub jak sędziowie w wypadku uszkodzenia urządzeń elektronicznych.

Jednocześnie eksperci z całego świata zbiorą się na towarzyszącej Olimpiadzie konferencji, podczas której będą referowane będą najnowsze teorie z dziedziny gier komputerowych. Konferencji przewodniczyć będzie prof. Tony Marsland z kanadyjskiego uniwersytetu w Albercie.

Programowanie gier strategicznych jest istotnym działem badań nad sztuczną inteligencją toteż olimpiada wraz z konferencją pomyślane są jako imprezy akcentujące te badania. Wszystkie referaty przygotowane na konferencję będą opublikowane, podobnie jak demonstrowane na olimpiadzie gry.

David Levy, międzynarodowy mistrz szachowy, organizował już konkursy komputerowe z programami naukowymi i jest światowym autorytetem w dziedzinie gier komputerowych. Szczególnie stał się sławny, gdy w 1968 roku ogłosił, że zapłaci 1000 dolarów jeśli przegra w szachy w grze z komputerem. I choć zaangażowano największe komputery, wyszedł z walki zwycięsko, mimo że w ostatnich tego typu zawodach w 1984 roku, użyto potężnego superkomputera Cray.

Jednocześnie Levy jest właścicielem przedsiębiorstwa Intelligent Chess Software, którego program zdobył mistrzostwo w mikroprocesorowych zawodach szachowych, innym programem pokonał japońskiego mistrza świata w Othello, opracował także program do gry w pokera, który mógłby prawdopodobnie zrujnować Las Vegas. Programy komputerowe do gier strategicznych, choć nie zawsze dorównać mogą mistrzom, są idealnymi przeciwnikami dla przeciętnego gracza - nie ma z nimi kłopotów, są zawsze dostępne, można zmieniać ich poziom i nie "puszą się" gdy zwyciężą. Na tym polega ich wartość handlowa.

Mimo obaw, że szachy maszynowe ograniczą grę ludzi, są one sprzedawane w milionach egzemplarzy i przyczyniły się do wzrostu zainteresowania szachami w wielu krajach. Wielka Brytania zajmuje drugie po Związku Radzieckim, miejsce w dziedzinie szachów, co w dużej mierze przypisywane jest intensywnemu rozwojowi szachów komputerowych w tym kraju. Można spodziewać się, że programy brytyjskie będą tu współzawodniczyć z konkurentami z USA, ZSRR, RFN, Holandii i Francji. Pojawiły się też inne, bardziej egzotyczne gry, jak shugi z Japonii i ziangqi z Chin. Programy z tymi grami są już na rynku w Europie i USA. Natomiast dużo do zrobienia jest w dziedzinie najtrudniejszej gry a mianowicie chińsko-japońskiej "go", znanej też pod chińską nazwą "weiqi". Oprogramowanie "go" jest w projektach. Jest ono jednak atrakcyjne dla specjalistów od sztucznej inteligencji, gdyż dla dobrej gry jest tu istotne rozpoznawanie obrazów /pattern recognition/ Pierwsze programy komputerowej gry "go" powstały na Taiwanie, w Hong-Kongu, Polsce i Jugosławii, ale Japończycy włączyli to zagadnienie do projektu Piątej Generacji.

London Press Servis JF5A588

Cena zł. 380.-

ISSN 0239-8044