

P.1879/921



9
1992

informatyka

KOLEGIUM REDAKCYJNE:

mgr Jarosław DEMINET
mgr inż. Piotr FUGLEWICZ
mgr Teresa JABŁOŃSKA
(sekretarz redakcji)
Władysław KLEPACZ
(redaktor naczelny)
dr inż. Wojciech MOKRZYCKI
mgr inż. Jan RYZKO
dr Zdzisław SZYJEWSKI
mgr Hanna WŁODARSKA –
– MARCZENKO

PRZEWODNICZĄCY RADY PROGRAMOWEJ:

Prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA:

Wydawnictwo Czasopism i Książek
Technicznych SIGMA NOT
Spółka z o.o.
ul. Biała 4
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja:

01-552 Warszawa,
Pl. Inwalidów 10, p. 104, 105
tel. 39-14-34

Materiałów nie zamówionych
redakcja nie zwraca

**W sprawach ogłoszeń
prosimy zwracać się
bezpośrednio
do Redakcji
lub
Działu Reklamy
i Marketingu
00-950 Warszawa
ul. Biała 4
telefon: 20-31-24
telefaks: 20-31-16
teleks: 814550**

W numerze:

	Strona
Sieci Inteligentne – usługi i architektura – <i>Marek Średniawa</i>	1
Symulacyjne gry decyzyjne jako narzędzie aktywnego kształcenia menedżerów – <i>Jerzy Skrzypek, Mariusz Szubra</i>	12
Strojenie mechanizmu rozumowania systemów ekspertowych z uwzględnieniem warunków niepewności – <i>Danuta Kosmowska-Miszalska</i>	17
Wpływ algorytmów tablic rozproszonych na efektywność działania systemów ekspertowych – <i>Marek Wawrzyniak, Katarzyna Stapor</i>	20
Metody analizy i projektowania systemów w praktyce – <i>Stanisław Wrycza</i>	26

W najbliższych numerach:

- Piotr Woźniak i Krzysztof Biedalok opisują metodę SuperMemo pozwalającą wielokrotnie zwiększyć szybkość uczenia się – nawet do 50 razy.
- Stanisław Szukalski i Ryszard Owczarkowski charakteryzują ekspertowy system wspomagania decyzji w planowaniu makroenergetycznym.
- Jerzy Stawicki prezentuje BB_POL – narzędzie do budowy tablicowych systemów ekspertowych i opisuje jego zastosowanie.
- Mirosława Lasek dzieli się doświadczeniami z realizacji systemu oceny sytuacji finansowo-majątkowej przedsiębiorstwa.
- Jadwiga Orylska opisuje osobiste spostrzeżenia prowadzenia dydaktyki informatyki w uczelniach rolniczych na przykładzie Akademii Rolniczej w Szczecinie.

Warunki prenumeraty

Przyjęcie prenumeraty – wyłącznie na podstawie dokonanej wpłaty na drukach dostarczanych dotychczasowym prenumeratorem przez Wydawnictwo, lub nowym – po uprzednim zgłoszeniu zapotrzebowania (pisemnie lub telefonicznie) w Zakładzie Kolportażu Wydawnictwa.

Blankiet wpłaty – powinien zawierać następujące informacje: dokładna nazwa i adres (z kodem pocztowym) zamawiającego, tytuły zamawianych czasopism, ich liczbę i okres prenumeraty.

Wpłata – zgodnie z podanymi cenami należy dokonać w banku lub w UPT na konto podane na naszym blankiecie, tj:

Państwowy Bank Kredytowy III O/Warszawa nr: 370015-1573-139-11

Prenumeratory zbiorowi – osoby prawne obowiązują blankiety „Wpłata-Zamówienie”. Cena normalna.

Prenumeratory indywidualni – osoby fizyczne obowiązują blankiety typu przekazy dla wpłat na rachunki bankowe. Cena normalna.

Prenumerata ulgowa – zgodnie z podaną ceną ulgową przysługuje wyłącznie osobom fizycznym, będącym członkami SNT, studentom i uczniom szkół zawodowych. Uczniowie szkół ogólnokształcących mogą zamówić w prenumeracie ulgowej tylko miesięcznik „Aura”

Uwaga! w podanym okresie prenumeraty można zamówić tylko po jednym egzemplarzu z każdego tytułu.

Prenumerata ze zleceniem wysyłki za granicę – cena prenumeraty ze zleceniem wysyłki za granicę jest dwukrotnie wyższa od ceny normalnej.

Należy podać dokładny adres odbiorcy za granicą.

Terminy przyjmowania prenumeraty:

– do 10 listopada na I, II, III, IV kwartał następnego roku

– do 28 lutego na II, III, IV kwartał br.

– do 31 maja na III i IV kwartał br.

– do 31 sierpnia na IV kwartał br.

Zmiany w prenumeracie, np. zmiana liczby tytułów, liczby egzemplarzy, rezygnacja z prenumeraty, można zgłaszać tylko w podanych terminach z mocą obowiązującą od następnego kwartału.

Egzemplarze archiwalne (z lat ubiegłych)

Można nabyć za gotówkę w Klubie Prasy Technicznej, Warszawa, ul. Mazowiecka 12 (tel. 26-80-16) lub zamówić pisemnie w Zakładzie Kolportażu, Dział Handlowy, 00-950 Warszawa, skr. poczt. 1004 (tel. 40-37-31), na rachunek lub za zaliczeniem pocztowym.

Informacji o prenumeracie udziela: Zakład Kolportażu Wydawnictwa SIGMA-NOT Spółka z o.o., 00-716 Warszawa, ul. Bartycza 20, skr. 1004. Telefony: 40-00-21 wewn. 293, 295, 299 lub 40-30-86, 40-35-89.

Wstępna cena jednego egzemplarza na 1992 rok: normalna – 18 000 zł, ulgowa – 13 500 zł

Wartość prenumeraty (w zł):

Normalna: kwartalna – 54 000, półroczna 108 000, roczna 216 000

Ulgowa: kwartalna – 40 500, półroczna 81 000, roczna 162 000

Uwaga: W przypadku zmiany cen w okresie objętym prenumeratą, prenumeratory zobowiązani są do dopłaty różnicy cen.



P.1877/92

Sieci Inteligentne – usługi i architektura

Przez pierwsze 90 lat historii telefonii głównym problemem było zapewnienie odpowiedniej jakości połączeń. Wkrótce po wynalezieniu telefonu w 1876 r. zdano sobie sprawę z tego, że przewody telefoniczne powinny zbiegać się w centralach zestawiających żądane połączenia. Początkowo były to centrale ręczne, obsługiwane przez telefonistki, które później zastąpiono centralami automatycznymi, początkowo elektromechanicznymi, następnie elektronicznymi, cyfrowymi, sterowanymi programowo. Aparaty telefoniczne wyposażono w tarcze, umożliwiające realizowanie lokalnych połączeń bez udziału telefonistki. Później, poczynając mniej więcej od 1964 r. umożliwiono abonentom automatyczne nawiązywanie połączeń międzymiastowych i międzynarodowych. Możliwość ta stanowiła, w pewnym sensie, ukoronowanie stulecia rozwoju telefonii.

Kolejny przełom zapoczątkowało wprowadzenie do eksploatacji w 1965 r. pierwszej centrali elektronicznej o sterowaniu programowym (SPC – *Stored Program Control*) systemu AT&T ESS nr 1. Wkrótce oprogramowanie centrali (program rzędu 100 000 wierszy w assemblerze) umożliwiło realizację zbioru dodatkowych usług, obejmujących takie udogodnienia, jak np.: wybieranie skrócone, oczekiwanie na zwolnienie, przekazywanie zgłoszeń, konferencja trójstronna, Centrex.

Pojawienie się central cyfrowych o sterowaniu programowym, cyfryzacja komutacji i transmisji, separacja kanałów użytkowych od kanałów sterujących oraz nowe media transmisyjne o bardzo dużej przepustowości, otworzyły nowe, praktycznie nieograniczone, możliwości tworzenia nowych usług. Główne tendencje rozwojowe nowych usług to:

- dążenie do uniwersalnej łączności osobistej – UPT (*Universal Personal Telecommunications*), polegającej na zapewnieniu możliwości osiągnięcia abonenta za pomocą jego numeru osobistego oraz zapewnieniu mu dostępu do sieci z zachowaniem posiadanego zakresu udogodnień, bez względu na jego aktualne miejsce pobytu; Środkami technicznymi, które pozwalają myśleć o realizacji idei UPT, są sieci abonentów ruchomych i omawiana w niniejszym artykule architektura IN;
- interaktywne usługi multimedia;
- integracja usług przekazywania informacji z jej przetwarzaniem, czyli tzw. usługi wzbogacone VAS (*Value Added Services*).

Wprowadzanie wyrafinowanych usług i udogodnień, opartych na oprogramowaniu central SPC, napotyka jednak na liczne przeszkody natury technicznej i organizacyjnej. Wiele z nich ma zbliżony charakter do problemów projektowania, implementacji i pielęgnacji dużych pakietów oprogramowania działających w rozproszonych konfiguracjach sieciowych, problemom określanych terminem „kryzysu oprogramowania”.

Po pierwsze, w przypadku gdy realizacja usług jest zlokalizowana w centrali końcowej, to powstaje problem wymiany starych central elektromechanicznych oraz zadanie modyfikacji

oprogramowania w eksploatowanych centralach SPC. Dotyczy to wszystkich central końcowych w danym kraju (np. w Stanach Zjednoczonych – ok. 20 000, w Polsce – ok. 6200). Realizacja wspomnianego zadania wymaga wielu lat. Z doświadczeń amerykańskich znany jest fakt, że w 1975 r., 10 lat po wprowadzeniu wymienionych wyżej usług dodatkowych, były one dostępne tylko dla około 1% abonentów mieszkaniowych. W 1990 r., po 25 latach, w dalszym ciągu byli abonenci, którzy nie mieli dostępu do nowych usług.

Drugim problemem jest różnorodność sprzętu pochodzącego od różnych dostawców. Implementacja tej samej usługi, w zależności od producenta, może się różnić, stwarzając dodatkowy problem dla operatora eksploatującego sieć. Z uwagi na środki materialne i finansowe zaangażowane w sieć telekomunikacyjną, jej modyfikacja może odbywać się tylko drogą ewolucyjną. Z oczywistych względów nie do przyjęcia jest rozwiązanie, które prowadziłoby do czasowego, nawet bardzo krótkiego, zawieszenia usług telefonicznych.

Po trzecie, wprowadzenie przez producenta nowej usługi dla określonej rodziny central oznacza konieczność opracowania, uruchomienia, przetestowania i wdrożenia nowej wersji oprogramowania. Z doświadczenia wiadomo, że typowy czas wdrożenia złożonego pakietu oprogramowania wynosi od pół roku do dwóch, trzech lat. Taka skala czasowa może być zbyt długa zarówno z punktu widzenia klientów, jak również firm wykonujących usługę oraz operatorów sieci (ryzyko nietrafienia w zapotrzebowanie rynku – spóźniona oferta).

Wymienione uwarunkowania sprawiają, że tradycyjne podejście do rozbudowy i poszerzania możliwości funkcjonalnych sieci, nie jest w stanie zapewnić szybkiej i elastycznej reakcji na potrzeby klientów.

Z historycznego punktu widzenia koncepcja Sieci Inteligentnej (ang. *intelligent network*) sięga czasów wprowadzenia w Stanach Zjednoczonych usługi połączenia bezpłatnego, znanej powszechnie jako usługa „800”, co było dostępne jeszcze przed rozbiem korporacji Bell na kilka mniejszych konkurujących ze sobą firm operatorskich. W 1981 r. w celu umożliwienia realizacji usługi „800” i połączeń rozliczanych za pomocą karty kredytowej, firma AT & T wprowadziła do sieci centralną bazę danych, wykorzystującą wspólnokanałowy system sygnalizacji międzycentralowej CCIS, w którym informacja sterująca połączeniem była przekazywana oddzielnym kanałem poza kanałem rozmównym. Wykorzystanie centralnej bazy danych umożliwiło szybkie upowszechnienie usługi, oparte na modyfikacji tylko central tranzytowych, bez konieczności wprowadzania zmian w centralach końcowych.

Idea Sieci Inteligentnej (*Intelligent Network* – IN) – jako sposobu rozwiązania wymienionych problemów została zapro-

ponowana w 1984 r. przez firmę Bellcore. Termin „sieć inteligentna” (IN) obejmuje koncepcję i architekturę sieci, która zapewnia jednolite, uniwersalne styki między oprogramowaniem realizującym usługi a systemami komutacyjnymi i transmisyjnymi, w sieci telekomunikacyjnej wykorzystującej sprzęt pochodzący od wielu producentów i eksploatowanej przez wielu operatorów. Koncepcja sieci IN stanowi również odzwierciedlenie osiągniętego postępu w dziedzinie systemów operacyjnych i rozproszonych baz danych oraz zainteresowania rynkiem telekomunikacyjnym ze strony firm komputerowych.

Dlaczego IN?

Zainteresowanie siecią IN wynika głównie z następujących przyczyn:

- znaczenia komercyjnego (najsilniej rozwijający się, oprócz sieci abonentów ruchomych, sektor usług telekomunikacyjnych; należy przy tym zwrócić uwagę, że możliwości funkcjonalne IN doskonale nadają się do wykorzystania w sieciach abonentów ruchomych); komercyjny sukces usług IN wynika z ich istotnego wpływu na wyniki ekonomiczne przedsiębiorstw (telemarketing, poprawa kontaktu z klientami);
- standaryzacji: europejskiej (ETSI – grupa NA6) i światowej (CCITT – grupy SGXI/WPXI/4 i SGXVIII: podstawowy zestaw usług CS-1; zalecenia serii Q.1200);
- niewielkiej liczby krajowych publikacji poświęconych sieci IN i co za tym idzie, ograniczoną wiedzę na ten temat, zarówno w środowisku telekomunikacyjnym i informatycznym, jak i wśród potencjalnych klientów – zamawiających i użytkowników.

Podstawowe cele koncepcji IN to:

- możliwość szybkiego i łatwego opracowywania i wdrażania nowych usług oraz modyfikacji usług istniejących, w ujednoczonej postaci, w całej sieci;
- wprowadzenie elastycznej architektury sieci zapewniającej:
 - możliwość dopasowywania, na bieżąco, charakterystyki i parametrów usługi do indywidualnych wymagań subskrybenta,
 - wygodne sterowanie siecią i zarządzanie usługami,
 - uniezależnienie usług od fizycznej struktury sieci;
- wprowadzenie standardowych styków sieciowych i otwartej architektury w celu poprawienia wykorzystania sieci i pobudzenia konkurencji w zakresie oferowanych usług;
- przyjęcie podejścia ewolucyjnego do modernizacji sieci telekomunikacyjnej.

Przykład

Przykład dotyczący usługi połączenia bezpłatnego poprzedzimy ogólną klasyfikacją kategorii usług z punktu widzenia informacji wykorzystywanej przez SCP przy obsłudze zgłoszenia. Wyróżniono trzy podstawowe kategorie usług:

B: wykorzystuje numer abonenta B – wywołwanego;
A + B: numery abonenta A – wywołującego oraz abonenta B – wywołwanego;

interakcyjna: numery abonenta A – wywołującego i abonenta B – wywołwanego A + B oraz dodatkowe dane przekazane w czasie interakcji przez abonenta A.

Każda z usług IN, w zależności od wersji (prosta lub złożona), może być zaliczona do jednej z wymienionych kategorii.

W przypadku kategorii usług wykorzystującej tylko numer abonenta B, interpreter programu usługi SLI (*Service Logic Interpreter*), wykonujący się w SCP, dokonuje translacji numeru wirtualnego na właściwy w danej chwili i okolicznościach numer docelowy, który jest następnie przekazywany do SSP i wykorzystywany do zestawienia finalnego połączenia.

Usługa kategorii A + B, podobnie jak poprzednia, wymaga dostępu do bazy danych, przy czym dalszy przebieg obsługi zgłoszenia jest uwarunkowany również danymi o abonencie A.

Usługa kategorii interakcyjnej umożliwia abonentowi wprowadzenie dodatkowych danych sterujących obsługą zgłoszenia. Dialog z abonentem jest wspomagany nagraniem komunikatami, zapraszającymi go do podawania wymaganej informacji (np. numeru karty kredytowej, hasła, cyfry interpretowanej jako wybór jednej z wielu możliwości itp.). Do obsługi dialogu z abonentem wykorzystuje się zespół IP (*Intelligent Peripheral*). Obsługa połączenia, wynikająca z dialogu, wymaga dodatkowych interakcji z bazą danych i sterowania zgodnego ze scenariuszem usługi. Interakcja z abonentem wymaga odpowiednich środków sygnalizacji abonenckiej. Stosowana jest sygnalizacja DTMF (wieloczęstotliwościowa) lub abonencka ISDN. Możliwe jest również oparcie dialogu na funkcji rozpoznawania głosu – pozwala to na dostęp do usług interakcyjnych również abonentom wyposażonym w konwencjonalne tarczowe aparaty telefoniczne bez DTMF.

Przykład usługi połączenia bezpłatnego

Usługa połączenia bezpłatnego została zapoczątkowana w Stanach Zjednoczonych jako tzw. usługa „800”. Po raz pierwszy udostępnił ją abonentom w 1967 r. Bell System stosując specjalne wyposażenie oraz wykorzystując centrale tranzytowe. W 1981 r. wprowadzono w Stanach Zjednoczonych wspólnokanałowy system sygnalizacji CCIS, który umożliwił wykorzystanie do realizacji usługi „800” centralnej bazy danych SCP (*Network Control Point*). Sukces komercyjny tej właśnie usługi (5 mld dolarów dochodu w 1986 r., coroczny przyrost liczby zgłoszeń o 20%) był głównym czynnikiem przyspieszającym rozwój sieci IN. Usługa połączenia bezpłatnego ma liczne zalety, zarówno dla abonentów, jak i zamawiających usługę oraz operatora sieci. Zainteresowanymi stronami mogą być również producenci sprzętu oraz niezależne firmy programistyczne opracowujące i wdrażające oprogramowanie nowych usług.

Korzyści dla abonenta:

- niepobieranie opłaty za połączenia,
- jednolity sposób dostępu na terenie całego kraju,
- ułatwienie dostępu do informacji i oferowanych towarów i usług,
- zwiększenie prawdopodobieństwa udanego załatwienia sprawy.

Zalety dla instytucji lub firmy zamawiającej usługę:

- poprawa wyników ekonomicznych przez ułatwienie i poszerzenie kontaktów z klientami,
- możliwość rejestrowania kontaktów z klientami,
- możliwość elastycznego doboru najwłaściwszego stanowiska do obsługi klienta,
- poprawa publicznego odbioru, statusu i wizerunku firmy.

Z punktu widzenia operatora główną zaletą jest zwiększenie dochodów z uwagi na wzrost liczby zgłoszeń oraz możliwość dopasowywania charakterystyki usług do indywidualnych wymagań subskrybentów.

Opis funkcjonalny usługi

Wybranie przez abonenta korzystającego z usługi połączenia bezpłatnego prefiksu identyfikującego tę usługę (np. w Stanach Zjednoczonych 800, w Niemczech 130) powoduje zestawienie połączenia do najbliższego SSP (w centrali końcowej lub tranzytowej). SSP wysyła zapytanie do bazy danych, przechowywanej w węzle SCP, w celu określenia właściwego numeru

docelowego, z którym ma być zestawione połączenie. Zapytanie z SSP jest przekazywane do SCP za pośrednictwem sieci sygnalizacyjnej SS7, wykorzystującej punkty transferowe sygnalizacji STP.

Następnie program aplikacyjny usługi w SCP dokonuje odczytu odpowiedniego rekordu z bazy danych tej usługi. Kluczem używanym przy przeszukiwaniu bazy danych jest numer podany przez abonenta wywołującego, przekazany w zapytaniu. Program aplikacyjny dokonuje konwersji numeru podanego przez abonenta na numer rzeczywisty na podstawie parametrów usługi, które zostały ustalone przez zamawiającego. Kolejnym krokiem jest przesłanie z SCP do SSP, za pośrednictwem sieci sygnalizacyjnej SS7, numeru docelowego oraz instrukcji dotyczących dalszego sposobu obsługi zgłoszenia. Do komunikacji w relacji SSP – SCP jest używany protokół TCAP.

Po odebraniu instrukcji i numeru docelowego SSP finalizuje zestawienie połączenia, kierując zestawieniem połączenia z numerem docelowym. Ostateczne ustalenie numeru docelowego może odbywać się w formie dialogu z abonentem wywołującym. Wykorzystuje się w tym celu moduł IP, który umożliwia odtworzenie zapowiedzi słownej oraz rejestrację odpowiedzi abonenta (sygnalizacja wieloczęstotliwościowa – aparat telefoniczny typu *touch-tone*). Np. „Wybierz 1, jeśli chcesz połączyć się z działem rezerwacji”; „Wybierz 2 jeśli chcesz uzyskać informację” itd.

Zamawiający usługę może wprowadzać i modyfikować jej parametry, takie jak np. uzależnienie sposobu obsługi zgłoszenia od lokalizacji abonenta wywołującego, dnia tygodnia i pory dnia oraz innych czynników. Odbywa się to przez dostęp do węzła zarządzającego SMS z terminala użytkownika. Następnie SMS realizuje transakcję do SCP, która aktualizuje bazę danych o usługach i ich użytkownikach, wykorzystywaną do realizacji połączeń.

Usługa z punktu widzenia abonenta

Z punktu widzenia abonenta usługa połączenia bezpłatnego ma następujące własności:

- jednolity sposób dostępu – prefiks (w ramach kraju, w przyszłości w ramach np. Europy czy Świata). Zwykle prefiks ten odpowiada numerowi określającemu usługę;
- użytkownik – subskrybent usługi (np. duża wielooddziałowa firma) wykorzystuje oddzielne pojedyncze numery identyfikujące konkretne aplikacje (np. informacja techniczna i serwis). Numer ten jest zwykle krótki i łatwy do zapamiętania;
- bez względu na położenie abonenta wywołującego, połączenie jest bezpłatne lub rozliczane według taryfy dla połączeń lokalnych;
- usługa w swojej rozwiniętej formie wymaga dialogu między abonentem wywołującym a siecią.

Usługa z punktu widzenia zamawiającego (subskrybenta usługi)

Numer identyfikujący usługę-aplikację jest ustalany przez operatora sieci w uzgodnieniu z zamawiającym klientem. W przypadku zmiany lokalizacji klienta numer usługi nie powinien być zmieniany.

Zamawiający usługę musi szczegółowo określić sposób obsługi napływających zgłoszeń (z jakim(i) rzeczywistymi numerami należy zestawiać połączenie), który powinien uwzględniać: lokalizację abonenta wywołującego, porę dnia, dzień tygodnia, itp. Zamawiający usługę może dodatkowo określić takie elementy,

jak np. ograniczenie zasięgu usługi (np. tylko dany region administracyjny i regiony sąsiednie), sposób wyboru drogi połączeniowej w sieci.

Dane charakteryzujące sposób obsługi zgłoszeń mogą być wprowadzane bezpośrednio przez użytkownika z terminala lub, na jego zlecenie, przez personel administracyjny operatora. Możliwość modyfikacji parametrów aplikacji jest określana angielskim terminem *user programmability*, czyli możliwością programowania aplikacji przez jej użytkownika. Zwykle proces ustalania i modyfikacji parametrów jest wspomagany przez specjalny program ułatwiający poprawne zdefiniowanie scenariusza usługi.

Fazy realizacji usługi

Realizacja usługi obejmuje następujące kroki:

Identyfikacja pochodzenia zgłoszenia. Na podstawie informacji sygnalizacyjnej lub numeru wiązki łączy przychodzących SSP ustala w przybliżeniu obszar geograficzny, z którego pochodzi zgłoszenie;

Analiza numeru. SSP i SCP dokonują analizy numeru IN wybranego przez abonenta wywołującego, a następnie interpretują odpowiedni program usługi związany z tym numerem – usługą. Wynikiem analizy jest sposób dalszej obsługi zgłoszenia;

Translacja numeru. Numer katalogowy przypisany subskrybentowi usługi IN identyfikuje zarówno usługę jak i zamawiającego. Jest to numer wirtualny, który należy zinterpretować i przetworzyć na faktyczny docelowy numer katalogowy w publicznej sieci telefonicznej. Funkcja translacji numeru IN wirtualnego na numer rzeczywisty w sieci publicznej jest dokonywana przez SSP dla usług prostych oraz przez SCP dla usług złożonych;

Kierowanie zgłoszeń. Po ustaleniu numeru docelowego, jeżeli na przebieg obsługi zgłoszenia nie wpływają funkcje zarządzania siecią, zgłoszenie jest kierowane do rzeczywistego numeru docelowego w sieci publicznej, na podstawie danych przechowywanych w SSP, lub w przypadku usług złożonych – wykorzystując dane z SCP;

Rejestracja zgłoszeń. Dla każdego zgłoszenia SSP wytwarza rekord rozliczeniowy AMA. Rekordy te są później przetwarzane w centrum rozliczeniowym operatora.

W zależności od rodzaju usługi mogą być wykonywane następujące funkcje:

- odbiór dodatkowej informacji za pomocą IP. Informacja ta może być przekazywana w postaci do 15 cyfr z zakresu 0...9, za pomocą sygnalizacji abonenckiej wieloczęstotliwościowej lub głosowo przez podanie jednej z trzech cyfr;
- analiza i translacja oparta na zmiennych. Dalszy przebieg zgłoszenia może zależeć od pory dnia, dnia tygodnia, rozkładu przydziału zgłoszeń do stanowisk obsługi lub dodatkowej informacji przekazanej przez abonenta (odebranej za pomocą IP). W zależności od złożoności usługi funkcje analizy i translacji są wykonywane przez SSP lub SCP;
- przekazywanie zapowiedzi słownych abonentom wywołującym.

Komunikaty standardowe i zindywidualizowane są przekazywane przez IP pod kontrolą SSP i SCP.

Modele architektury IN

Koncepcja Sieci Inteligentnej opiera się na wprowadzeniu do sieci telefonicznej niewielkiej liczby inteligentnych węzłów,

które przejmują na siebie przetwarzanie zgłoszeń żądających dodatkowych usług. W ten sposób podstawowa sieć transportu informacji, obejmująca transmisję i komutację, jest wzbogacana o wyspecjalizowane moduły i systemy sterujące sposobem obsługi zgłoszeń i zarządzające usługami w sieci.

Koncepcja IN wprowadza nowy podział podmiotów na rynku usług telekomunikacyjnych. Do tradycyjnego zestawu – operatorzy, producenci sprzętu i abonenci dochodzą nowe podmioty: subskrybenci usług i firmy oferujące implementację nowych usług. Subskrybenci zamawiają usługi i sprawują nadzór nad ich bieżącymi parametrami, dostosowując je do swoich indywidualnych wymagań. Bardziej zaawansowane koncepcje IN (IN/2, AIN) wprowadzają wspólny model obsługi zgłoszenia i niezależny od platformy sprzętowo-programowej uniwersalny standardowy zestaw elementów funkcjonalnych (FC – *Functional Components* wg terminologii MVI lub SIB – *Service Independent Building Blocks* wg terminologii CCITT), z których mają być konstruowane programy nowych usług. Standaryzacja styków i platformy pozwala niezależnym firmom zajmującym się projektowaniem i wytwarzaniem oprogramowania wejść na rynek usług telekomunikacyjnych. Odzwierciedleniem tego faktu jest wyróżnienie w modelu IN środowiska programistycznego SCE, służącego do opracowywania i uruchamiania nowych usług.

Architektura Sieci Inteligentnej, począwszy od oryginalnej propozycji Bellcore z 1984 r. podlega ewolucji. Kolejne jej etapy to modele architektoniczne określane oznaczeniami: IN/1, IN/2, IN/1+ oraz kolejne wersje tzw. Zaawansowanej Sieci Inteligentnej AIN (*Advanced Intelligent Network*). Modele różnią się, przede wszystkim, stopniem centralizacji sterowania, zakresem możliwości zarządzania usługami, zestawem usług i udogodnień. W Stanach Zjednoczonych prace nad IN są prowadzone w ramach tzw. Forum Producentów MVI (*Multi-Vendor interaction Forum*). W Europie standaryzacją architektury IN zajmują się CCITT (SG XI/WP XI/4) oraz ETSI (NA6). W najnowszej edycji zaleceń CCITT, Białej Księdze, znajdzie się seria zaleceń Q.1200 opisująca podstawową architekturę IN oraz podstawowy zestaw usług CS-1.

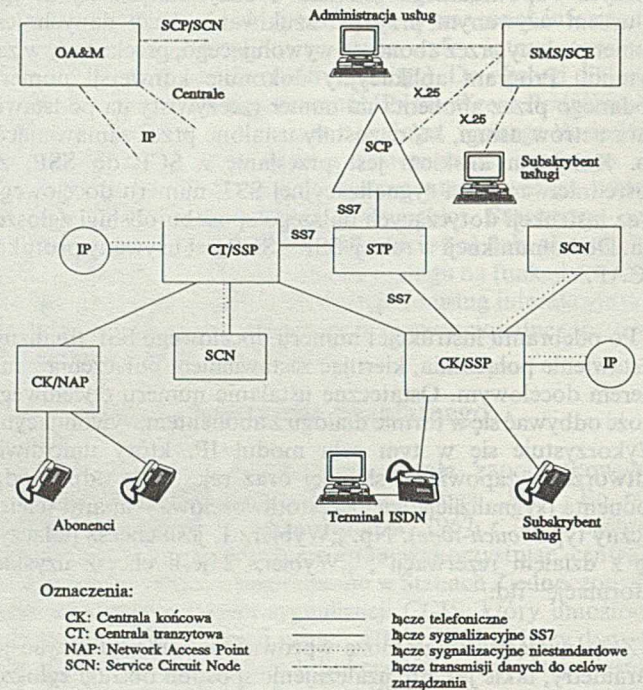
IN/1 – podstawowa architektura IN

W pierwotnej koncepcji IN/1 został zdefiniowany styk między centralą telefoniczną a scentralizowaną bazą danych, przechowującą informację o sposobie interpretacji numerów wybieranych przez abonentów.

W architekturze IN/1 rolę sieci telefonicznej ograniczono do umiejętności rozpoznawania numeru identyfikującego usługę inteligentną. Centrala wyposażona w taką umiejętność nazywa się punktem komutacji usług – SSP (*Service Switching Point*). Rozpoznany przez SSP numer, identyfikujący usługę IN, inicjuje zapytanie do bazy danych w węźle SCP, która w odpowiedzi przekazuje instrukcje określające sposób dalszej obsługi zgłoszenia. W modelu IN/1 inteligencja, związana z obsługą zgłoszeń, jest skupiona w pojedynczym, centralnym węźle – bazie danych, zwanym punktem sterowania usługą – SCP (*Service Control Point*). Innym typem węzła, wyposażonym w możliwości przetwarzania informacji jest system zarządzania usługami – SMS (*Service Management System*), za pośrednictwem którego jest możliwe programowanie SCP.

SCP i SSP wykorzystują sieciowe zespoły usługowe IP (*Intelligent Peripheral*), których zadaniem, na zlecenie SCP, jest obsługa interakcji z abonentem. SCP, SSP i IP stanowią rozproszony system czasu rzeczywistego, bezpośrednio zaangażowany w obsługę zgłoszeń. Projektowanie, definiowanie, uruchamianie i instalacja nowych usług, modyfikacja parametrów

eksploatowanych usług oraz inne funkcje zarządzania siecią IN są realizowane przez węzeł zarządzania usługami SMS (*Service Management System*) i związane z nim środowisko programistyczne wspomagające implementację usług SCE (*Service Creation Environment*). Na rys. 1 przedstawiono podstawowy model architektoniczny sieci IN.



Rys. 1. Podstawowa architektura Sieci Inteligentnej

Wadą IN/1 jest brak uniwersalności, spowodowany specjalizacją poszczególnych węzłów SCP przez przypisanie im realizacji tylko jednej określonej usługi. W związku z tym wprowadzenie nowej usługi wymaga opracowania nowego SCP. Ponadto centralizacja realizacji usług w architekturze IN/1 sprawia, że istnieje niebezpieczeństwo przeciążenia zarówno samego SCP, jak i prowadzących do niego łączy, powodując pogorszenie jakości obsługi z punktu widzenia abonentów.

Elementy

Poniżej zostaną scharakteryzowane funkcje, miejsce i role poszczególnych elementów sprzętowo-programowych sieci IN odpowiadającej modelowi IN/1. W charakterystyce uwzględniono pewne dodatkowe elementy w realizacji IN/1 proponowanej np. przez AT & T.

Punkty Komutacji Usług (SSP) są centralami rozpoznającymi przychodzące zgłoszenia żądania usług IN. Podstawowe usługi IN (wykorzystujące prosty mechanizm translacji numeru) są w całości obsługiwane przez SSP; usługi złożone wymagają współdziałania z SCP, który na żądanie SSP przekazuje odpowiednie instrukcje sterujące przetwarzaniem zgłoszenia. Do komunikacji między SCP a SSP jest wykorzystywany system sygnalizacji CCITT nr 7 (SS7). Wiadomości sygnalizacyjne między SCP a SSP są przekazywane za pośrednictwem transferowych i końcowych punktów sygnalizacyjnych (STP i SP) sieci sygnalizacyjnej SS7.

Punkty Sterowania Usługami (SCP) są scentralizowanymi systemami zarządzania bazami danych, opisującymi usługi, wykonującymi funkcje obsługi zgłoszeń dla wyrafinowanych usług IN. W SCP przechowywane są informacje o abonentach,

scenariuszach usług i ich parametrach, niezbędne do obsługi zgłoszeń żądających wyrafinowanych udogodnień.

System Zarządzania usługami (SMS) zarządza stykiem z SCP od strony operatora i subskrybentów usług, umożliwiając im dostęp do bazy danych przechowującej rekordy opisu usług. Za jego pomocą subskrybenci mogą ze swoich terminali definiować i zmieniać sposób kierowania zgłoszeń, dostosowując go do swoich indywidualnych potrzeb. Schematy kierowania zgłoszeń związane z usługami IN są tworzone, modyfikowane i ładowane do węzłów SCP przez subskrybentów usług IN, właśnie za pośrednictwem SMS. System ten umożliwia bezpośrednią kontrolę subskrybenta nad sposobem realizacji zamawianej usługi IN. SMS współpracuje z SCE przy implementacji i wprowadzaniu oprogramowania dla nowych usług. Przykładowo, usługa teległosowania jest inicjowana i kończona z SMS. SMS również dokonuje opracowania wyników głosowania i ich przekazania subskrybentowi usługi. Do zadań SMS należą także: prowadzenie dziennika działań związanych z zarządzaniem usługami, prowadzenie rozliczeń oraz funkcje utrzymaniowe.

Sieciowy zespół usługowy (IP) wspomaga realizację niektórych bardziej złożonych usług. Do funkcji wykonywanych przez IP należy np. odtwarzanie abonentowi wywołującemu zarejestrowanych uprzednio komunikatów słownych, odbieranie i rozpoznawanie cyfr przekazywanych przez abonenta głosem lub za pomocą sygnalizacji wieloczęstotliwościowej.

SCP przechowuje informacje o komunikatach słownych udostępnianych przez poszczególne IP, w związku z czym nie jest konieczne, by każdy zespół IP przechowywał wszystkie używane zapowiedzi. W przypadku niedostępności odpowiedniego komunikatu w lokalnym IP, pod nadzorem SCP jest realizowane przekazanie obsługi zgłoszenia do innego IP.

System eksploatacji i zarządzania siecią (OA & M) realizuje zadania związane z centralnym zarządzaniem, eksploatacją i utrzymaniem sieci IN central wyposażonych w moduł SSP, SCP, IP i SMS. System nadzoruje stan sieci, zbierając w tym celu i przetwarzając dane dotyczące statusów, alarmów oraz innych zdarzeń systemowych, przekazywane przez węzły SSP i SCP.

Punkty dostępu do sieci NAP (Network Access Points) umożliwiają poszerzenie dostępu do sieci IN bez konieczności wyposażania wszystkich central w funkcje SSP. NAP wykrywają zdarzenia inicjujące podjęcie działań właściwych dla rozpoznanego żądania usług IN. NAP nie mają bezpośredniej komunikacji z SCP a jedynie kierują obsługę zgłoszenia do odpowiedniego (w szczególności do najbliższego SSP).

Węzeł usługowy SCN (Service Circuit Node) jest nowym elementem architektury sieci IN, zaproponowanym przez firmę AT&T. Jest to system złożony z komputera sterującego, cyfrowego pola komutacyjnego i specjalizowanych zespołów usługowych obejmujących funkcje rozpoznawania mowy, zapowiedzi słownych, konwersji tekst-mowa, skrzynki pocztowej dla wiadomości telefaksowych, odbiornika DTMF. SCN jest umieszczany przy centrali (końcowej, tranzytowej lub na poziomie sieci nakładkowej) jako węzeł stowarzyszony, podobnie jak IP. Jednak w odróżnieniu od IP węzeł SCN dysponuje szerszymi możliwościami funkcjonalnymi i oprócz udostępniania funkcji usługowych stanowi platformę dla środowiska programistycznego SCE służącego do projektowania i realizacji nowych usług. SCN komunikuje się z centralą przez styk ISDN (dostęp podstawowy lub pierwotny).

IN/2

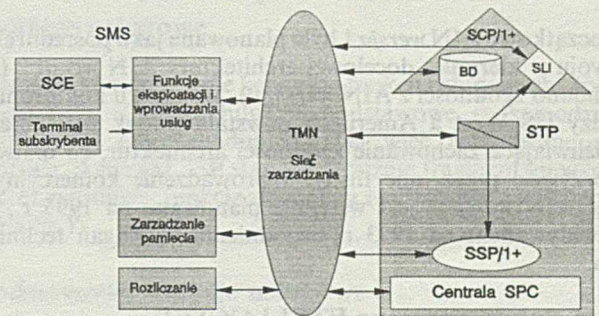
Koncepcja IN/2 została opracowana w 1986 r. w Bellcore. Uniwersalność modelu uzyskano przez wprowadzenie tzw.

interpretera logiki usługi (SLI – *Service Logic Interpreter*), zdolnego do wykonywania określonej kombinacji tzw. elementów funkcjonalnych (FC – *Functional Component*), stanowiących bibliotekę podstawowych funkcji i procedur, niezależnych od realizowanej usługi. Architektura IN/2 obejmuje również propozycję zestawu podstawowych FC.

W IN/2 przewidywano umieszczenie SLI we wszystkich centralach końcowych oraz w centralnym węźle SCP, łagodząc wady centralizacji charakterystyczne dla IN/1. Rozwiązanie takie jest kosztowne ze względu na konieczność wprowadzenia znacznych zmian w oprogramowaniu wielu central. Z tego powodu koncepcja IN/2 została zarzucona.

IN/1 +

Wspomniane wady IN/1 i nadmierny koszt modelu IN/2 doprowadziły do przyjęcia w 1988 r. kompromisowego modelu architektury nazwanego IN/1+ (rys. 2). W modelu IN/1+ interpretry SLI są umieszczone tylko w węzłach SCP. Zadania SSP zostały sprowadzone do reagowania na predefiniowane zdarzenia w przebiegu procesu komutacyjnego i przekazywania do SCP żądań obsłużenia tych zdarzeń według scenariusza określającego logikę usługi.



Rys. 2. Architektura IN/1+

Przyjęte rozwiązanie pozwoliło ograniczyć zakres i zasięg niezbędnych zmian oprogramowania central, a równocześnie złagodzić problem centralizacji. Mimo zalet architektury IN/1+, opinia co do trafności tej koncepcji jest podzielona.

NTA

Na początku 1989 r. zarysowały się dwa równoległe kierunki działań. Przedsiębiorstwa RBOC porozumiały się co do kontynuowania prac nad IN w ramach forum MVI. Celem powołania MVI było wypracowanie architektury IN właściwej z punktu widzenia najbliższej perspektywy czasowej, tzn. do 1995 r. (NTA – *Near Term Architecture*). Jednocześnie w Bellcore podjęto prace nad koncepcją tzw. architektury sieci informacyjnej (INA – *Information Network Architecture*), która umożliwiłaby po 1995 r. integrację usług telekomunikacyjnych i informacyjnych w jednolitej komunikacyjno-informacyjnej sieci przyszłości.

AIN wersja 0

MVI opracowało wiele propozycji architektury IN. Używany początkowo termin NTA został zastąpiony określeniem „Zaawansowana Sieć Inteligentna” (AIN – *Advanced Intelligent Network*) wraz z numerem wersji. Pierwsza próbna instalacja AIN wersji 0, została zrealizowana w rejonie Chicago.

ASA

Firma US West zaproponowała model architektoniczny istotnie różniący się od modelu IN/1. Uznano, że architektura sieci IN ma mieć rozproszone sterowanie, z silniej zintegrowanym systemem eksploatacji i utrzymania. Propozycją, która w zamierzeniu spełnia te postulaty jest *Adjunct Software Architecture* (ASA), przewidziana do implementacji w okresie 1991–1993 r. Koncepcja ASA polega na rozszerzeniu możliwości funkcjonalnych centrali przez wykorzystanie odrębnych fizycznie, pomocniczych komputerów współpracujących z centralą przez specjalizowane styki o dużej szybkości. Koncepcja ASA jest zgodna z AIN wersją 0. W dalszej perspektywie czasowej przyjęto projekt architektury NSSA, w której przewiduje się zintegrowanie pomocniczych funkcji z funkcjami centrali.

AIN wersja 1

Architektury ASA i NSSA miały początkowo charakter koncepcji konkurencyjnych wobec AIN, wywierając jednak wpływ na kierunek prac Bellcore. W 1990 r. firmy Ameritech i US West opracowały wspólną propozycję, w wyniku której ASA stała się praktycznie obszernym podzbiorem przygotowywanej AIN wersji 2. Główną różnicę stanowi rezygnacja w ASA ze stosowania sygnalizacji SS7 do komunikacji z SCP.

Początkowo AIN wersja 1 była planowana jako pośredni etap rozwoju w kierunku docelowej architektury AIN wersji 2 (nie zakładano zgodności z AIN wersją 0). W wyniku kompromisu między US West a Ameritech powstała jednak specyfikacja umożliwiająca zachowanie zgodności architektur, co ułatwiło jej przyjęcie przez inne firmy. Wprowadzenie komercyjnych usług AIN wersji 2 jest wstępnie planowane na 1995 r., po przewidywanym na 1993 r. uzgodnieniu wymagań technicznych.

Prace standaryzacyjne ETSI i CCITT

Omawiane dotąd prace były prowadzone przede wszystkim przez firmy amerykańskie. W Europie prace nad IN rozpoczęły się znacznie później. Prace standaryzacyjne w ETSI (grupa NA6) i CCITT (SG XI/WP XI/4) rozpoczęto w 1989 r.

Punktem wyjścia prac podjętych przez ETSI i CCITT było spostrzeżenie, że nawet najpopularniejsze usługi IN, takie jak np. połączenie bezpłatne, wirtualne sieci wydzielone czy opłata prowizyjna, są w poszczególnych krajach implementowane w odmienny sposób. Uniemożliwia to korzystanie z usług inteligentnych w relacjach międzynarodowych. Podobna sytuacja wystąpiła dla przypadku współpracy międzynarodowej sieci ISDN do chwili przyjęcia przez kraje Wspólnoty Europejskiej porozumienia (tzw. Memorandum of Understanding) co do wersji ISUP SS7 i zestawu usług dodatkowych ISDN udostępnianych w relacjach międzynarodowych. Potrzebą chwili stała się międzynarodowa standaryzacja usług IN. Kierując się głównie względami praktycznymi (użytecznością i możliwością technicznej realizacji) zdefiniowano podstawowy zestaw usług oraz funkcji usługowych tzw. CS1 (*IN Capability Set 1*).

Zestaw CCITT CS1 (tabela) ograniczono do usług realizowanych na rzecz jednego abonenta, które mogą być uruchamiane w fazie zestawiania lub rozłączania połączenia (tzw. klasa *single user, single-ended, single point of control*). Przyjęcie wspomnianego rodzaju usług ma następujące zalety:

- pokrywa szeroki repertuar usług, z których wiele potwierdziło już swoją użyteczność i atrakcyjność w realizacji tradycyjnej;
- realizacja usług opiera się na dobrze opanowanych i jasnych mechanizmach współdziałania elementów sieci. Pozwoliło to

uzgodnić specyfikację ich funkcji i interfejsów w postaci zaleceń (Biała Księga CCITT: zalecenia serii Q.1200);

- upraszcza, zarówno z punktu widzenia oferentów usług jak i producentów sprzętu, proces ewolucji sieci.

Zestaw usług CS-1

Freephone FPH
Originating Call Screening OCS
Destination Call Routing DCR
Split Charging SPL
Mass Calling MAS
Malicious Call Identification MCI
Selective Call Forward on Busy SCF
Virtual Private Network VPN
Terminating Call Screening TCS
Televoting VOT
Account Card Calling ACC
Follow-me Diversion FMD
Completion of Call to Busy Subscriber CCBS
User Defined Routing UDR
Call Forwarding CF
Security Screening SEC
Credit Card Calling CCC
Conference calling CON*
Call Rerouting Distribution CRD
Universal Personal Telecommunications UPT
Abbreviated Dialling ABD
Call Distribution CD
Premium Rate PRM
Automatic Alternative Billing AAB
Universal Access Number UAN

Zdefiniowanie podstawowego zestawu usług CS1 było podstawą do określenia zbioru tzw. SIB (*Service Independent building Block*), tj. niezależnych od usługi elementów funkcjonalnych, za pomocą których można projektować i realizować scenariusze usług.

Do zbioru SIB należą operacje takie, jak np.:

- interakcja z abonentem (*User Interaction*),
- translacja numeru w zależności od daty i pory dnia (*Translate*),
- umieszczenie w kolejce zgłoszeń oczekujących (*Queue*),
- naliczanie opłaty (*Charge*),
- sprawdzenie uprawnień (*Screen*).

Z poszczególnych SIB są budowane funkcje usługowe, a następnie z ich kombinacji konstruuje się programy usług. Programy takie są reprezentowane jako grafy decyzyjne wykorzystujące komponenty SIB i realizujące przyjęty scenariusz usługi. Projektowanie i konstruowanie programu usługi przy użyciu elementów SIB odbywa się w specjalizowanym środowisku SCE.

Sterowanie siecią IN

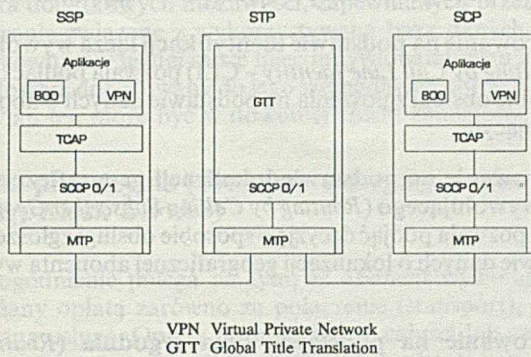
Do sterowania siecią IN są wykorzystywane: sieć sygnalizacyjna SS7 oraz protokół X.25.

Sieć sygnalizacyjna

Kluczowym elementem, niezbędnym do realizacji sieci IN, jest sieć sygnalizacyjna SS7 (rys. 3). Zadaniem SS7 jest zapewnienie wymiany informacji między systemami sterowania centrali, zestawiających krok po kroku połączenie w sieci użytkowej.

Sieć SS7 stanowi sieć komutacji pakietów, „nałożoną” logicznie na sieć użytkową. W centralach są umieszczone punkty

sygnalizacyjne SP (*Signalling Points*), które wymieniają między sobą informacje sygnalizacyjne za pośrednictwem węzłów komutacji pakietów, nazywanych Transferowymi Punktami Sygnalizacyjnymi STP (*Signalling Transfer Points*), zlokalizowanych zwykle również w centralach.



Rys. 3. Miejsce SS7 w architekturze sieci IN

W protokole SS7, w celu uzyskania struktury sieci zgodnej z modelem OSI, wyodrębniono Część Transferową Wiadomości MTP (*Message Transfer Part*), wykorzystywaną przez komunikujące się ze sobą Części Użytkowe UP (*User Parts*; wśród nich m.in. Część Użytkowa Telefoniczna – TUP oraz Część Użytkowa ISDN – ISUP). MTP realizuje funkcje mieszczące się w trzech najniższych warstwach modelu odniesienia OSI (OSI RM – *OSI Reference Model*), zaś części użytkowe realizują funkcje wyższych warstw modelu; w szczególności zadaniem MTP jest dostarczenie niezawodnego mechanizmu przekazywania informacji na zlecenie części użytkowych.

Początkowe ukierunkowanie SS7 na wymianę informacji sterujących komutacją spowodowało, że realizowana w MTP warstwa sieciowa protokołu SS7 udostępnia wyłącznie usługi sieciowe bezpołączeniowe, nazywane także usługami datagramowymi; w usłudze tego typu transfer informacji nie jest poprzedzony ustanowieniem logicznego połączenia między użytkownikami MTP. Z punktu widzenia zastosowań SS7, w szczególności sieci IN, taki mechanizm okazał się niewystarczający. W związku z tym protokół SS7 został wzbogacony o Część Sterującą Połączeniami Sygnalizacyjnymi SCCP (*Signalling Connection Control Part*), która realizuje funkcję udostępniania połączeń wirtualnych. MTP łącznie z SCCP nazywane są Częścią Usług Sieciowych NSP (*Network Service Part*) i oferują – wciąż w ramach warstw 1–3 OSI RM – zarówno bezpołączeniowe, jak i połączeniowe usługi sieciowe.

Z usług NSP korzysta ISUP oraz inne nowo wprowadzone protokoły wyższych warstw, oferujące użytkownikom usługi typu transakcyjnego: Część Usługową Pośredniczącą ISP (*Intermediate Service Part*), obejmującą funkcje warstw 4–6 OSI RM, oraz Część Zastosowań Transakcyjnych TCAP (*Transaction Capability Application Part*), świadcząca usługi warstwy 7 OSI RM procesom aplikacyjnym. Komunikacja między SSP a SCP w sieci IN opiera się na wykorzystaniu wiadomości TCAP.

Węzły stanowiące elementy składowe architektury IN, tzn. SSP, SCP, SMS, IP, VFN wykorzystują do komunikacji sieć sygnalizacyjną SS7. Realizacja usług IN wymaga, aby zaimplementowane były usługi bezpołączeniowe klasy 0 (i ewentualnie klasy 1) SCCP i MTP, na których bazuje komunikacja wykorzystująca TCAP. W przypadku dostępu abonentów ISDN jest wymagany ISUP i system sygnalizacji DSS1. Usługi, w których jest wymagana interakcja z abonentem, po zestawieniu połącze-

nia wymagają dodatkowo sygnalizacji abonenckiej (wieloczesotliwościowej lub ISDN).

Protokół X.25

Protokół X.25 jest wykorzystywany w sieciach IN do funkcji administracyjnych, eksploatacyjnych i utrzymaniowych. Stanowi platformę komunikacji między SMS/SCE a SCP oraz zapewnia dostęp do sterowania scenariuszem usługi przez subskrybenta. Jest to rozwiązanie naturalne z uwagi na powszechną dostępność sieci pakietowych (w krajach rozwiniętych). Ponadto nie wymaga to dodatkowych inwestycji w infrastrukturę komunikacyjną i pozwala wykorzystać istniejące terminale, oprogramowanie i personel. Użycie X.25 pozwala przyspieszyć implementację usług IN. Z uwagi na niekrytyczne uwarunkowania czasowe przy komunikacji z bazą danych w SCP, niezbyt wysoka efektywność X.25, w sensie szybkości przekazywania danych, nie stanowi ograniczenia.

Usługi IN

Dla zilustrowania typowego zestawu usług zostanie podana głównie implementacja usług IN przyjęta w systemie 5ESS firmy AT & T. Podobnie jak w koncepcji CS-1, wyróżniono zestaw funkcji usługowych niższego poziomu, z których są budowane pełne, oferowane klientom usługi wyższego poziomu.

Funkcje usługowe

Można wyróżnić dwie kategorie funkcji usługowych: podstawowe oraz złożone. Funkcje podstawowe są samodzielnie realizowane przez moduł SSP (są więc realizowane przez centralę, ponieważ SSP stanowi fizycznie fragment jej wyposażenia), natomiast funkcje złożone są realizowane z wykorzystaniem SCP.

Podstawowe funkcje usługowe

W odróżnieniu od funkcji złożonych, subskrybent usługi nie ma bezpośredniej kontroli nad funkcjami podstawowymi. Modyfikacja parametrów usługi może być dokonywana wyłącznie za pośrednictwem systemu zarządzania siecią. Funkcje podstawowe są wykorzystywane do realizacji podstawowych usług IN, do których należą np. podstawowe połączenia bezpłatne, podstawowe płatne służby informacyjne (opłata prowizyjna) czy usługa podziału opłaty.

Do podstawowych funkcji usługowych należą:

- podstawowa translacja numeru (*Basic Translation*) powodująca translację każdego numeru IN na pojedynczy numer docelowy;
- translacja numeru sterowana numerem wiązki łączy przychodzących (*Translation Based on Incoming Trunk Group Identity*), która sprawia, że numer IN może być przekształcony na jeden z wielu możliwych numerów docelowych, w zależności od identyfikacji wiązki łączy przychodzących;
- blokada połączeń sterowana numerem wiązki łączy przychodzących (*Call Barring Based on Incoming Trunk Group Identity*) stanowiąca mechanizm translacji analogiczny do poprzedniego, wykorzystywany do blokady połączeń przychodzących;
- procentowa parametryczna translacja numeru (*Translation Based on Percentage Allocation*) umożliwiająca translację pojedynczego numeru IN na jeden z dwóch numerów docelowych w ustalonej wcześniej proporcji (funkcja nie może być łączona ani z funkcją blokady połączeń, ani z funkcją translacji sterowanej identyfikacją wiązki przychodzącej).

Złożone funkcje usługowe

Złożone funkcje usługowe stanowią elementy służące do konstruowania zaawansowanych usług IN, takich jak np. teległosowanie, numer osobisty czy wirtualna sieć wydzielona. Możliwość łączenia ze sobą złożonych funkcji usługowych decyduje o ich przydatności i wartości do realizacji wyrafinowanych usług dopasowanych do indywidualnych wymagań klienta. Subskrybenci mogą zamawiać określone funkcje usługowe za pośrednictwem centrum zarządzania lub, uzyskując bezpośredni dostęp do rekordów opisu usługi, za pośrednictwem systemu SMS.

Klient, łącząc się ze swojego terminala z centrum zarządzania usługami SMS, może budować, zmieniać, usuwać, sprawdzać, przeglądać i planować sposób uruchamiania rekordów opisu zgłoszenia w węzłach SCP. Zbiór udogodnień oraz dane administracyjne związane z klientem usługi są zdefiniowane w odpowiadającym mu rekordzie logicznym.

Poniżej zostaną krótko omówione poszczególne złożone funkcje usługowe. W prezentacji tej przyjęto punkt widzenia klienta usługi.

- **Alternatywny numer docelowy w przypadku zajętości** (*Alternative Destination on Busy*) umożliwia klientowi usługi podanie ciągu numerów docelowych, pod które ma być kierowane zgłoszenie w przypadku natrafienia na zajętość. Subskrybent usługi może podać zarówno alternatywne numery docelowe, jak również określić ich kolejność.

- **Podział zgłoszeń** pozwala klientowi określić proporcje w jakich zgłoszenia mają być rozdzielane między dwoma lub większą liczbę numerów docelowych albo między odrębnymi sposobami obsługi. Użytkownik podaje tylko procentowy podział ruchu między wskazane punkty docelowe.

- **Blokada zgłoszeń** (*Call Barring*) pozwala użytkownikowi ograniczyć obszar dostępności usługi tylko do pewnych wybranych regionów geograficznych. Blokada może być stosowana w połączeniu z dowolną inną funkcją z omawianej grupy.

- **Licznik zgłoszeń** (*Call Counter*) zlicza zgłoszenia napływające pod wskazany numer klienta usługi. Funkcja jest wykorzystywana w usłudze teległosowania.

- **Zapytanie** (*Call Prompter*) umożliwia użytkownikowi usługi dołączanie zapowiedzi słownej z węzła IP, proszącej abonenta wywołującego o podanie dodatkowej informacji głosem lub za pomocą sygnalizacji abonenckiej wieloczęstotliwościowej.

- **Kolejkowanie zgłoszeń** (*Call Queueing*) pozwala umieszczać w kolejce zgłoszenia natrafiające na zajętość wszystkich urządzeń końcowych u subskrybenta usługi. Po wprowadzeniu do kolejki, do wywołującego abonenta jest wysyłana zapowiedź słowna informująca go, że zgłoszenie zostanie obsłużone, gdy tylko będzie wolne urządzenie końcowe.

- **Rezerwowy plan kierowania zgłoszeń** (*Command Routing*) umożliwia użytkownikowi lub personelowi zarządzającemu określenie rezerwowego planu kierowania, który może być uruchomiony w szczególnych przypadkach (np. nieoczekiwanych szczytów ruchu, klęsk żywiołowych). W danej chwili może być w użyciu tylko jeden plan kierowania. Uruchomienie rezerwowego planu kierowania może być dokonane w dowolnej chwili przez użytkownika usługi lub personel zarządzający.

- **Odpowiedź grzecznościowa** (*Courtesy response*) pozwala użytkownikowi zrealizować połączenie przez dołączenie w miej-

sce łącza abonenckiego zapowiedzi słownej z IP. Zapowiedź może mieć charakter standardowy lub może być dostosowana do indywidualnych wymagań klienta i okoliczności. Użytkownik funkcji może określić i używać różnych zapowiedzi, w zależności od przyczyny niemożności zestawienia połączenia (np. zgłoszenie poza godzinami pracy, zajętość wszystkich linii itp.).

- **Kierowanie na podstawie identyfikacji łącza wywołującego** (*Routing by Call Line Identity – CLI*) pozwala podjąć decyzję o sposobie obsługi zgłoszenia na podstawie danych o abonencie wywołującym.

- **Kierowanie na podstawie lokalizacji geograficznej abonenta wywołującego** (*Routing by Calling Subscriber Geography – CSG*) pozwala podjąć decyzję o sposobie obsługi zgłoszenia na podstawie danych o lokalizacji geograficznej abonenta wywołującego.

- **Kierowanie na podstawie dnia tygodnia** (*Routing by Day-of-Week*) pozwala użytkownikowi kierować zgłoszenie do różnych numerów docelowych lub różnie obsługiwać zgłoszenia w zależności od dnia tygodnia.

- **Kierowanie na podstawie pory dnia** (*Routing by Time-of-Day*) pozwala użytkownikowi kierować zgłoszenie do różnych numerów docelowych lub różnie obsługiwać zgłoszenia w zależności od pory dnia.

Przegląd usług

Poniżej zostaną szczegółowo omówione podstawowe usługi IN, dostępne w systemie 5ESS, oraz przykłady ich zastosowań. Ze względu na brak polskiej terminologii dla sieci IN oraz z uwagi na stosowanie określeń firmowych, przyjęto zasadę podawania również nazw oryginalnych.

Do podstawowych usług IN należą:

- usługi specjalnego trybu rozliczania zgłoszeń (*Special Charging Services*):
 - połączenie bezpłatne (*Freephone*);
 - opłata prowizyjna (*Premium Charging*);
 - podział opłaty (*Split Charging*);
- połączenie rozliczane kartą kredytową (*Calling Card – CC*);
- teległosowanie (*Televoting*);
- numer osobisty (*Personal Number*);
- obsługa fal zgłoszeń (*Mass Calling*).

Usługi te opierają się na wcześniej opisanych funkcjach podstawowych i złożonych.

Połączenie bezpłatne (*Freephone*)

Usługa połączenia bezpłatnego pozwala jej subskrybentowi odbierać zgłoszenia wolne od opłaty dla abonentów wywołujących. Subskrybent jest obciążany opłatą za zrealizowane do niego połączenia. Numery katalogowe są przydzielane użytkownikom usługi przez operatora. Abonenci korzystający z usługi muszą wybrać odpowiedni prefiks poprzedzający numer katalogowy związany z użytkownikiem połączenia bezpłatnego. Centrale końcowe kierują takie zgłoszenia do sieci IN i nie obciążają za nie abonenta wywołującego. Natomiast węzeł SSP generuje rekord wiadomości rozliczeniowej (*AMA – Automatic Message Accounting*) w celu obciążenia za usługę subskrybenta usługi.

Jak wspomniano wcześniej, istnieją dwie odmiany połączenia bezpłatnego: podstawowa i zaawansowana. W przypadku wersji podstawowej, pierwszy węzeł IN, który odebrał zgłoszenie wykonuje translację numeru wybranego przez abonenta wywołującego na rzeczywisty numer docelowy klienta usługi i następ-

nie kieruje do niego zgłoszenie. Węzeł SSP nie komunikuje się z węzłami SCP ani IP co powoduje, że abonent może korzystać tylko z tych udogodnień, które są udostępniane przez centralę.

Wersja zaawansowania usługi połączenia bezpłatnego zawiera kilka dodatkowych możliwości, zapewnianych przez oprogramowanie związane z wykorzystywaną bazą danych. Każdy z użytkowników usługi może mieć indywidualny plan kierowania zgłoszeń, dopasowany do jego szczególnych potrzeb i wymagań. Plan ten może być w dowolnej chwili zmieniony.

Oplata prowizyjna (*Premium Charging, Value Added Billing, Kiosk Service*)

Udogodnienie polega na tym, że abonent wywołujący jest obciążany opłatą zarówno za połączenie (transport), jak i za uzyskaną usługę. Oplata ta może być w całości lub częściowo przekazywana na rzecz użytkownika subskrybenta usługi. Zgłoszenia wykorzystujące tę usługę są obsługiwane przez sieć IN w ten sam sposób, jak połączenia bezpłatne, co pozwala na wykorzystanie tego samego repertuaru funkcji podstawowych i złożonych, jak w usłudze *Freephone*.

Usługa opłaty prowizyjnej jest wykorzystywana do realizacji usług wzbogaconych (ang. *value-added services*), takich jak np. telefoniczne porady podatkowe lub giełdowe).

Podział opłaty (*Split charging*)

Usługa podziału opłaty pozwala rozłożyć koszt połączeń między abonenta wywołującego a wywoływane (tj. subskrybenta usługi). Część opłaty obciążająca abonenta wywołującego jest określana przez zastosowanie taryfy sieci publicznej, właściwej dla wybranego numeru. Sieć IN przekazuje rekord rozliczeniowy AMA subskrybentowi usługi.

Połączenie rozliczane kartą kredytową (*Calling Card – CC i Commercial Credit Card – CCC*)

Usługa pozwala wprowadzić odmienny sposób rozliczania usług uniezależniając zaliczanie zgłoszeń od numeru katalogowego aparatu, z którego dokonano połączenia. Firma zarządzająca usługami udostępnia użytkownikom usługi numery kart kredytowych, którym odpowiadają konta obciążane kosztami usług telekomunikacyjnych. Numer karty kredytowej obejmuje również osobisty numer identyfikacyjny (PIN), sprawdzany przez SCP, który przechowuje listę ważnych numerów kart. Usługa jest oferowana w trzech odmianach: podstawowej, ograniczonej i zaawansowanej.

W wersji podstawowej uprawniony abonent może z dowolnego aparatu uzyskać połączenie z wybranym przez siebie numerem. Scenariusz połączenia jest następujący: abonent wybiera kod dostępu usługi CC, jest proszony o podanie numeru konta i osobistego numeru identyfikacyjnego, następnie po potwierdzeniu uprawnienia do korzystania z usługi abonent może wybrać numer, z którym chce się połączyć.

W wersji ograniczonej abonent może łączyć się tylko z jednym z góry ustalonym numerem (np. centralnym biurem firmy, w której jest on zatrudniony). Scenariusz obsługi zgłoszenia jest analogiczny jak w przypadku wersji podstawowej z tym, że po weryfikacji numeru konta i PIN automatycznie jest zestawiane połączenie z ustalonym z góry numerem.

Wersja zaawansowana pozwala zaliczać opłaty za połączenia w ciężar komercyjnych kart kredytowych.

Obsługa zgłoszeń korzystających z usługi CC wymaga odwołania do węzła SCP i użycia IP.

Teległosowanie (*Televoting*)

Usługa teległosowania pozwala wykorzystywać publiczną sieć telefoniczną do przeprowadzania badania opinii publicznej przez automatyczne zliczanie zgłoszeń napływających na ustalone numery udostępnione na czas ankiety. Skorzystanie z usługi wymaga uprzedniego ogłoszenia (w telewizji, radiu, gazetach) zestawu numerów przeznaczonego do głosowania oraz podania znaczenia poszczególnych numerów (np. w przypadku oczekiwanej odpowiedzi *Tak* lub *Nie* należy wykorzystać dwa numery; każde zestawienie połączenia z pierwszym numerem będzie traktowane jako głos za, natomiast z drugim – jako głos przeciw). Po zakończeniu ankiety wyniki jej są udostępniane subskrybentowi usługi. Dla każdego seansu teległosowania może być zastosowana specjalna taryfa. Dochód z ankietyzacji może być dzielony między firmę operatorską a subskrybenta usługi. Realizacja usługi wymaga wykorzystania opisanych wcześniej złożonych funkcji usługowych.

Numer osobisty (*Personal Number*)

Usługa numeru osobistego polega na możliwości zdalnego korzystania z mechanizmu przekazywania zgłoszeń. Usługa jest przede wszystkim przydatna osobom często zmieniającym miejsce pobytu, pozwalając im przyjmować zgłoszenia bez względu na aktualne miejsce przebywania. Po przybyciu w nowe miejsce abonent może zaktualizować numer, pod którym można go zastać.

W ramach usługi są wyróżnione dwa typy zgłoszeń: normalne zgłoszenia do abonenta korzystającego z numeru osobistego oraz zgłoszenia aktualizacyjne realizowane przez użytkownika numeru. Normalne zgłoszenia są kierowane przez sieć IN pod numer docelowy ostatnio wskazany przez użytkownika lub pod z góry ustalony numer domyślny. Zgłoszenie aktualizacyjne pozwala użytkownikowi powiadomić sieć IN o zmianie bieżącego numeru docelowego, pod który należy przekazywać napływające zgłoszenia.

Usługa stanowi namiastkę sieci abonentów ruchomych. W odróżnieniu od sieci komórkowej abonent musi sam dbać o aktualność informacji o swoim bieżącym położeniu i jest oczywiście związany z konkretnym aparatem dołączonym do sieci stałej.

Obsługa zgłoszeń do numerów osobistych wykorzystuje opisane wcześniej złożone funkcje usługowe oraz wymaga odwołania do węzła SCP i użycia IP.

Obsługa fal zgłoszeń (*Mass Calling*)

Udogodnienie obsługi fal zgłoszeń łączy w sobie funkcje usługowe sieci IN z możliwościami specjalnego kierowania ruchem oraz mechanizmami zarządzania publiczną siecią telefoniczną, pozwalając obsługiwać chwilowe szczyty ruchu. Ruch tego typu zwykle powstaje w przypadku klęsk żywiołowych czy akcji reklamowych w radiu lub telewizji (np. pierwsza osoba, która zadzwoni pod wskazany numer otrzyma nagrodę).

Przy realizacji usługi można wykorzystać funkcję usługową *Alternatywny numer docelowy w przypadku zajętości*, w postaci w której tylko pierwsze zgłoszenie jest kierowane pod podany numer w sieci publicznej, natomiast wszystkie pozostałe są obsługiwane zapowiedzią słowną. Można również wykorzystać funkcje związane z realizacją zaawansowanego połączenia bezpłatnego.

Obsługa fal zgłoszeń wykorzystuje opisane wcześniej złożone funkcje usługowe oraz wymaga odwołania do węzła SCP i użycia IP.

Wirtualna sieć wydzielona (Virtual Private Network – VPN)

Wirtualna sieć prywatna stanowi koncepcję zapewnienia własności funkcjonalnych sieci wydzielonej wykorzystując zasoby sieci publicznej. Z logicznego punktu widzenia VPN można uważać za zamknięte grupy użytkowników zrealizowane w publicznej sieci komutowanej. Zasadniczą własnością VPN jest rozpoznawanie prywatnego planu numeracji przedsiębiorstwa i jego translacja na rzeczywiste numery sieciowe. Implementacja wirtualnej sieci wydzielonej wykorzystuje oprogramowanie węzła SCP i systemu SMS.

W ramach VPN zapewnia się udogodnienia typowe dla tradycyjnych sieci wydzielonych:

- prywatny plan numeracji definiowany przez użytkownika,
- zamknięte grupy użytkowników,
- możliwość inicjowania połączeń wychodzących poza VPN i odbierania zgłoszeń spoza VPN oraz połączenia wewnątrz VPN,
- możliwość definiowania kodów upoważniających do uzyskiwania połączeń z określonymi numerami docelowymi,
- kierowanie zgłoszeń uwarunkowane porą dnia i dniem tygodnia,
- gorąca linia z numerem docelowym określonym dynamicznie (pora dnia, dzień tygodnia),
- możliwość rozliczania według specjalnych taryf,
- możliwość rekonfiguracji VPN (zapewnienie dostępu do VPN z dowolnego aparatu końcowego),
- możliwość sterowania funkcjami kierowania zgłoszeń przez użytkownika usługi.

Środowiska wspomagające projektowanie i implementację usług

Projektowanie i implementacja usług dla sieci IN wymaga posługiwania się odpowiednio do tego celu przystosowanym, wspomagającym środowiskiem programistycznym określanym w architekturze IN terminem SCE. Środowisko powinno integrować język aplikacyjny wysokiego poziomu, model koncepcyjny (np. EFSM, grafy decyzyjne, biblioteki klas obiektów), metodologię projektowania i narzędzia, pozwalające projektantowi skupić się na rozwiązywaniu zasadniczych problemów, a przesłaniając nieistotne szczegóły.

Środowisko SCE zwykle obejmuje następujący zestaw narzędzi:

- edytor języka opisu usług,
- moduły weryfikacji i symulacji usług,
- kompilatory,
- moduł umożliwiający ładowanie programów usług do węzłów SCP w sieci.

Proponowana przez CCITT koncepcja składników funkcjonalnych SIB, z których mają być budowane funkcje usługowe i wreszcie scenariusze kompletnych usług, jest dość dobrze przystosowana do użycia podejścia obiektowego. Każdy SIB może być uważany za obiekt wyposażony we własne struktury danych i metody. Poszczególne SIB mogą być ze sobą łączone w scenariusz usługi, przez traktowanie wyjść z określonych składników SIB jako wejść dla innych. Przyjęcie reprezentacji graficznej SIB (zalecenie CCITT Q.1213) jako piktogramów pozwala zrealizować środowisko SCE w postaci przyjaznej dla użytkownika.

W pewnej perspektywie czasowej przewiduje się możliwość udostępnienia, odpowiednio zabezpieczonego, środowiska SCE klientom zamawiającym usługi, aby umożliwić im budowanie i wypróbowywanie scenariuszy usług dopasowanych do indywidualnych wymagań.

Koncepcja sieci IN stanowi atrakcyjny sposób szybkiego wprowadzania nowych usług telekomunikacyjnych, zarówno dla krajów o dobrze rozwiniętej sieci telekomunikacyjnej, jak i dla krajów w tym zakresie opóźnionych. W przypadku tych drugich istnieje możliwość zbudowania w pełni cyfrowej sieci nakładkowej, zorientowanej głównie na przedsiębiorstwa. Dochód uzyskany z usług IN, bezpośrednio przez operatorów i pośrednio przez subskrybentów, może stanowić stymulator modernizacji sieci oraz rozwoju gospodarczego jej użytkowników. W krajach o dobrze rozwiniętej telekomunikacji własności sieci IN mogą być wykorzystywane do realizacji usług wzbogacanych i stanowić etap pośredni ewolucji do koncepcji uniwersalnej łączności osobistej.

Należy zwrócić uwagę, że koncepcja IN stanowi podejście do ewolucji sieci oparte na sterowaniu przez potrzeby rynku, w odróżnieniu od np. sieci ISDN, która stanowi rozwiązanie głównie uwarunkowane postępowaniem technologicznym. W tym ograniczonym sensie można uważać IN i ISDN za koncepcje przeciwstawne. Należy jednak zwrócić uwagę na to, że repertuar usług ISDN i IN ma wspólne elementy, i że terminal i dostęp ISDN nadaje się do korzystania z usług IN. Również w obu przypadkach wymagana jest sterowana siecią infrastruktura sieci sygnalizacyjnej SS7.

Odrębnym problemem jest rozpoznanie potrzeb rynku pod kątem całkowitej wielkości popytu i preferowanych usług. Usługa połączenia bezpłatnego lub numer uniwersalny wydają się być problemem bezspornym, lecz zapotrzebowanie na inne usługi powinno być poprzedzone badaniem marketingowym. Wiadomo, na przykład, że profil zainteresowań europejskich jest inny niż w Stanach Zjednoczonych. Na profil ten ma wpływ wiele czynników np. rozpowszechnienie kart kredytowych, popularność sprzedaży wysyłkowej, mobilność społeczeństwa, taryfy telekomunikacyjne.

LITERATURA

- [1] Ambrosch W.D., Maher A., Sassacer B. (Eds.): *The Intelligent Network*. Springer-Verlag, Berlin, 1989
- [2] Bauer H., Jacoby J. et al.: *Evolution of Intelligence in Switched Networks*. XIII International Switching Symposium, Stockholm, May 1990
- [3] Bright R.E., Morgan M.J., Weiss E.J.: *Service Creation in an Intelligent Network*. GLOBECOM'89, Vol. 1, pp. 137-140, November 1989
- [4] CCITT. *Robocze wersje zaleceń serii Q.1200*, Geneva, 1991
- [5] Chang H.Y., Johnson J.W., Prell E.M.: *Intelligent Networks – Benefits and Promise*. AT & T Trends in Telecommunications, Vol. 6, No. 2
- [6] Duran J.M., Visser J.: *International Standards for Intelligent Networks*. IEEE Communications Magazine, February 1992
- [7] Franx W.G., Boogert G., Kramer J.C., Marcus S.M.: *The Service Circuit Node: Intelligence Next to the Switch*. 2nd International Conference on Intelligence in Networks, Bordeaux, 1992
- [8] Husain S.S., Patruni V.M., Shores W.N., Stoops D.S., van der Veer H.: *IN Architectures*. Communications International, pp. 83-92, September 1991
- [9] Jabbari B.: *Common Channel Signalling System Number 7 for ISDN and Intelligent Networks*. Proceedings of the IEEE, Vol. 79, No. 2, pp. 155-169, February 1991
- [10] Lund R.M., Wetzel J.M.: *Toward a switch based rapid service delivery*. Communications International, pp. 38-42, September 1990
- [11] Richards P.S.: *Rapid Service Delivery and Customization in a Developing Network Infrastructure*. ITC Specialist's Seminar, Kraków, paper 1.1., 22-27 April 1991
- [12] Robrock R.B.: *The Intelligent Network – Changing the Face of Telecommunications*, w: Proceedings of the IEEE, Vol. 79, No 1, January 1991

[13] Sieci Inteligentne – Analiza zagadnień architektonicznych i funkcjonalnych. Opracowanie zbiorowe, Red. M. Średniawa, Instytut Telekomunikacji PW, Warszawa, 1991

[14] Soederberg L.: An overview of IN. Communications International, pp. 47–51, December 1990

[15] Soederberg L.: Intelligent networks. Communications International, pp. 114–119, October 1991

Konferencje

SYSTEMY OTWARTE

Biuro Informatyki Urzędu Rady Ministrów oraz Stowarzyszenie Rozwoju Systemów Otwartych organizują w dniach 26–28 października br. w Krakowie I FORUM Firm Komputerowych Europy Środkowo-Wschodniej pod nazwą „Systemy Otwarte”.

Do udziału w tej międzynarodowej imprezie zaproszono przedstawicieli instytucji rządowych oraz firm komputerowych Białorusi, Bułgarii, Czecho-Słowacji, Chorwacji, Estonii, Litwy, Polski, Rosji, Słowenii, Ukrainy i Węgier, a także reprezentantów czołowych firm międzynarodowych: Compaq, Dell, Digital, Hewlett-Packard, IBM, ICL, Lotus, Novell, SCO, Siemens Nixdorf, Soft-Tronik, Sun i UNISYS.

Patronat nad imprezą objęli: Minister Stanu – Pełnomocnik ds. Stosunków z EWG oraz Minister Łączności. Aktywny współudział i wsparcie inicjatywy obiecało Biuro DG-XIII EWG oraz węgierskie Towarzystwo Informatyczne im. von Neumana, a także następujące instytucje krajowe: Agencja Rozwoju Przemysłu S.A., Polskie Towarzystwo Informatyczne, Fundacja Rozwoju Technik Komputerowych (FRTK) oraz Telekomunikacja Polska S.A.

Podany na konferencji prasowej w dniu 4 września br. ramowy program tematyki FORUM przedstawia się następująco:

Dzień pierwszy – blok tematyczny „Świat systemów otwartych”:

- Najważniejsze przedsięwzięcia informatyczne państw uczestniczących w I FORUM;
- Strategia firm międzynarodowych w świecie systemów otwartych Digital, Hewlett-Packard, IBM, ICL, Novell;
- Stan prac nad opracowaniem krajowych edycji GOSIP;
- Komputeryzacja czy informatyzacja – myślenie w kategoriach systemów informatycznych.

Dzień drugi – blok tematyczny „Rynki informatyczne”:

- Stan rynku informatycznego w poszczególnych państwach, analiza porównawcza rynku w ocenie wybranych firm: Digital, Hewlett-Packard, IBM, ICL, Lotus, SCO, Soft-Tronik;
- Możliwości transferów programów aplikacyjnych;
- Bariery celne w transferze technologii informatycznych;
- Ochrona oprogramowania i jej wpływ na rynek licencjonowanego oprogramowania;
- Prawne wymogi kontraktów międzynarodowych;
- Formy współpracy międzynarodowej – możliwości i ograniczenia.

Dzień trzeci – blok tematyczny „Telekomunikacja”:

- Rozwój poczty elektronicznej oraz powiązania sieci ISDN z sieciami pakietowymi;
- Przyjęte rozwiązania telekomunikacyjne a rozwój zastosowań informatyki.

Dzień trzeci – blok tematyczny „Środowisko informatyczne”:

- Krajowe lobby informatyczne (stowarzyszenia, izby handlowe, kluby kapitału, fundacje);
- Szkolenie informatyczne;
- Wybór Komitetu Organizacyjnego II FORUM'93 i ustalenie programu;
- Konferencja prasowa I FORUM.

Planuje się następujące imprezy towarzyszące:

- Wystawa publikacji i wydawnictw z ofertą sprzedaży lub licencji edytorskich;
- Pokaz zastosowań multimedii;
- Wystawa firm uczestniczących w I FORUM.

Językami obrad będą: angielski, niemiecki, rosyjski i polski z tłumaczeniem kabinowym.

Zgłoszenia udziału w obradach przyjmuje do dnia 30 września br. Sekretariat I FORUM:

Centrum Promocji Informatyki – FRTK, 00-503 Warszawa, ul. Żurawia 4a, tel. 21-76-26 lub 693-59-46.

TERA Spółka z o.o.

Przedsiębiorstwo Popierania Postępu TERA Spółka z o.o. uprzejmie informuje, że posiadając kilkuletnie doświadczenie w instalacji systemów wspomagających zarządzanie

o f e r u j e :

- opracowanie projektów systemów,
- optymalny kosztowo i rozwojowo dobór sprzętu i oprogramowania,
- instalację systemu u klienta,
- bezpłatnie przez rok konserwację oprogramowania oraz serwis sprzętu wraz z doradztwem techniczno-eksploatacyjnym,
- szybką dostawę uzupełnień konfiguracji lub sprzętu mikrokomputerowego niezależnego od systemu, w tym mikrokomputerów ALR firmy Wearnes Technology.

Wszelkie dodatkowe informacje uzyskają Państwo codziennie oprócz niedziel w Biurze Handlowym

40-025 Katowice, ul. Wojewódzka 31
tel. (faks): 155-26-72, teleks: 315448 tera pl

PAMIĘTAJ! Instalacje XENIX/NOVELL/PC MOS 386 oraz serwis to nasza specjalność – ZAPRASZAMY, ponieważ czterech lat doświadczeń nigdzie nie kupisz.

0/19/90

meditronik SPÓŁKA z o.o.	 UNITED MICROELECTRONICS CORPORATION	 HEWLETT PACKARD COMPONENTS	BOURNS
• CZĘŚCI ELEKTRONICZNE	• UKŁADY PAMIĘCI	• TRANSOPTORY	• POTENCJOMETRY TRIMPOT
• KOMPUTERY PS/1, PS/2	• UKŁADY KOMPUTEROWE	• WSKAŹNIKI ŚWIETLNE	• HYBRYDY REZYSTOROWE
• Drukarki HP		• WYŚWIETLACZE LED	• REZYSTORY SUBMINIATUROWE
• INSTALACJE SIECI KOMPUTEROWYCH	• UKŁADY KOMUNIKACYJNE I KOMERCYJNE	• PRODUKTY KODÓW KRĘSKOWYCH	• BEZPIECZNIKI MULTIFUSE
		• KONTROLERY I CZUJNIKI RUCHU	• POTENCJOMETRY PRECYZYJNE
		• TECHNIKA ŚWIATŁOWODOWA	• POTENCJOMETRY PANELI CZOŁOWYCH I KODERY
		• ELEMENTY W.CZ. I MIKROFALOWE	• CEWKI I TRANSFORMATORY
		• PODZESPOŁY DO MONTAŻU POWIERZCHNIOWEGO (SMD)	• CZUJNIKI CIŚNIENIA, POŁOŻENIA I PRZYSPIESZENIA
Partnerzy handlowi: ANALOG DEVICES, INT, MOTOROLA, SAMSUNG, TELEFUNKEN i inni	PRZEDSTAWICIELSTWO	DYSTRYBUCJA	DYSTRYBUCJA
 Business Partner			
meditronik sp. z o.o.			
00-194 Warszawa, ul. Długa 4 tel. (02) 6352263, 6352264 fax (02) 6352195, tlx 816075			

Symulacyjne gry decyzyjne jako narzędzie aktywnego kształcenia menedżerów

Istnieją dziedziny, w których swobodne eksperymentowanie na funkcjonującym organizmie może być niebezpieczne, bardzo kosztowne lub po prostu niemożliwe. Jedną z tych dziedzin jest niewątpliwie zarządzanie nowoczesnymi przedsiębiorstwami. Zdobycie umiejętności menedżerskich przez bezpośrednie doświadczenie może bowiem doprowadzić do niepowodzeń o znacznych konsekwencjach finansowych.

Należy wziąć przecież pod uwagę fakt, że sytuacja menedżera jest często dużo trudniejsza niż, na przykład, inżyniera, który zwykle ma do czynienia z obiektywnymi prawami. Ekonomista natomiast napotyka często na niejednorodną i niekompletną teorię, a przecież nie istnieje laboratorium, w którym mógłby on wykonywać eksperymenty na badanym obiekcie. Ponadto struktura procesów gospodarczych jest zwykle zmienna w czasie, co wymaga odpowiednich, bieżąco dokonywanych, korekt podejmowanych decyzji. Kolejny problem stwarzają dane empiryczne, często niekompletne, natomiast obciążone trudnymi do oszacowania błędami pomiaru.

Znalezienie racjonalnego sposobu zarządzania staje się więc w ostatnich latach jednym z najważniejszych problemów ekonomii. Sposób taki powinien zapewniać przede wszystkim zgodność osiągniętych rezultatów z uprzednio wyznaczonymi celami. W każdym przypadku oznacza to konieczność zdefiniowania celów działania badanego obiektu, a następnie przełożenia ich na konkretne decyzje. Należy przy tym brać pod uwagę aktualną sytuację ekonomiczną obiektu i stopień jego samodzielności, a także przewidywane zmiany w otoczeniu. Jest rzeczą oczywistą, że wykorzystywanie w tym celu wyłącznie możliwości „nieuzbrojonego” umysłu ludzkiego nie może przynieść pożądanego rezultatu. Polegając na intuicji i doświadczeniu decydentów można zwykle ocenić, a w konsekwencji dokonać wyboru spośród najwyższej kilku wariantów postępowania, nie zapewniając przy tym możliwości uzyskania rozwiązania optymalnego. Tym bardziej, że w sytuacjach kryzysowych, gdy trzeba zdecydować się na wybór między alternatywami, większość decydentów zgodnie z prawem Rudina 7 wybiera zwykle rozwiązanie najgorsze.

Jedną z metod, które pozwalają rozwiązywać wspomniane problemy, jest budowa i aplikacja modeli symulacyjnych. Metoda ta jest szczególnie skuteczna w przypadku, gdy wykorzystuje się ją do oceny skutków wielu potencjalnych wariantów podejmowanych decyzji. Pozwala ona także na konstrukcję symulacyjnych gier decyzyjnych, które stanowią znakomite narzędzie aktywnego kształcenia menedżerów.

Symulacja a gry decyzyjne

Symulację, w rozumieniu techniki eksperymentowania, można zdefiniować jako... *badanie złożonego systemu przedmiotowego, rzeczywistego lub hipotetycznego, przez obserwowanie zmian zachodzących w upływającym czasie w dynamicznym modelu tego systemu, pod wpływem zmieniających się warunków (wewnętrznych i zewnętrznych w stosunku do systemu), odwzorowanych właściwym im modelem* [2]. Symulacja jest więc techniką modelowania, umożliwiającą szybkie generowanie rezultatów wielu wariantów decyzyjnych, stanowiąc tym samym ważny element wspomaganego procesu podejmowania decyzji. Symulacyjna gra decyzyjna jest szczególnego rodzaju sposobem wykorzystania modelu symulacyjnego. O ile bowiem w przypadku stosowania modelu symulacyjnego jako narzędzia predykcji dużą wagę przykładają do procesu testowania jego zgodności z modelowanym obiektem, to w grze decyzyjnej nieważnym zadaniem jest konstrukcja odpowiedniego scenariusza.

Scenariusz gry musi opisywać problem decyzyjny, określać kryterium oceny wyników rozgrywki oraz jej horyzont czasowy. Na jego podstawie powstaje następnie symulacyjny model decyzyjny, który ze swej natury zakładać musi udział wielu konkurujących ze sobą uczestników, odgrywających role przewidziane scenariuszem.

Typowa gra symulacyjna przebiega w kilku fazach (rys. 1). Pierwszoplanowym zadaniem uczestników jest uważne przestudiowanie scenariusza gry, a następnie skonstruowanie planu strategicznego. Stanowić on bowiem powinien podstawę do podejmowania decyzji taktycznych oraz obserwowania i oceny ich konsekwencji z punktu widzenia kryterium gry.

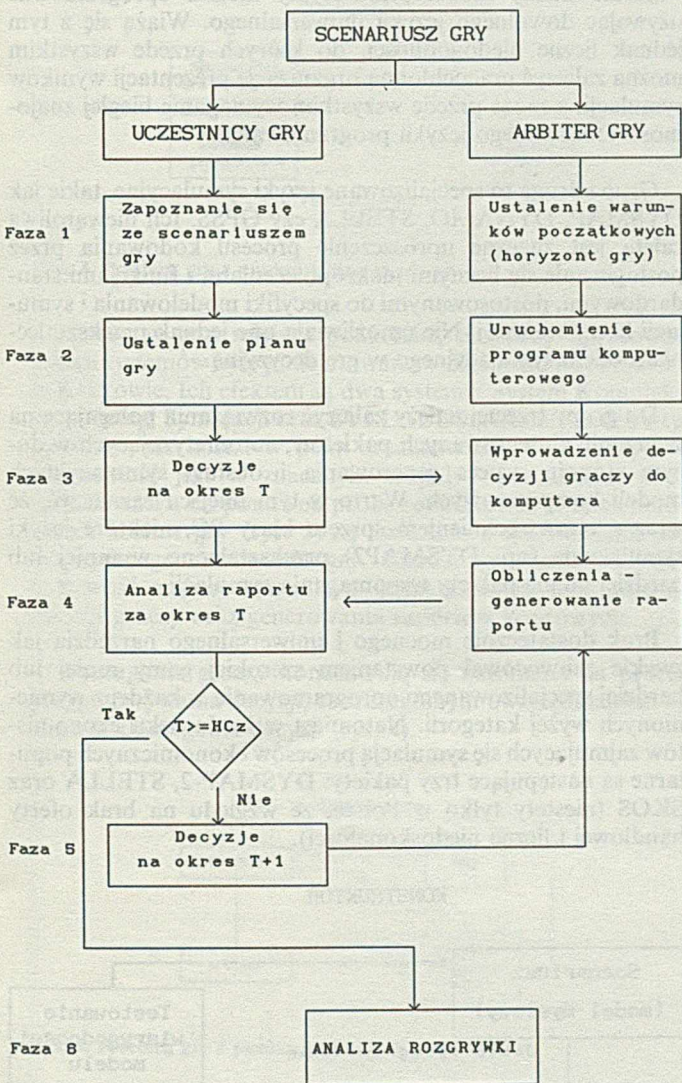
Arbiter rozgrywki czuwa natomiast nad jej przebiegiem i jest swego rodzaju pośrednikiem między modelem symulacyjnym a uczestnikami gry, sterując przepływem informacji. Jest on także odpowiedzialny za przeprowadzenie analizy przebiegu rozgrywki oraz ogłoszenie jej zwycięzców.

Rys historyczny

Symulacyjne gry decyzyjne są szeroko stosowane w renomowanych uczelniach całego świata jako element wspomagający proces dydaktyczny. Podobną sytuację zaczynamy obserwować również w Polsce. Istnieje bowiem kilka ośrodków związanych przede wszystkim ze Szkołą Symulacji Systemów Gospodarczych oraz Polskim Towarzystwem Symulacyjnym, które

starają się promować szerokie stosowanie gier symulacyjnych jako narzędzia nowoczesnego i skutecznego kształcenia menedżerów.

Historia zastosowań symulacji oraz gier decyzyjnych liczy sobie już kilkaset lat. Należy podkreślić, że aż do dwudziestego wieku, wszystkie próby stosowania gier symulacyjnych były dokonywane w trakcie doskonalenia techniki wojennej. Za pierwszą grę o charakterze militarnym uważane są przy tym szachy, w których dla osiągnięcia celu każdy z graczy dysponuje określonymi środkami i może się nimi posługiwać według ściśle określonych reguł. Gra ta przez długi okres była szeroko stosowana w procesie szkolenia przyszłych dowódców. Z czasem stała się ona bazą wyjściową dla budowy gier militarnych o bardziej złożonej konstrukcji. Informacji o tego typu grach można doszukać się nawet w starożytności [6].



Rys. 1. Fazy przebiegu gry symulacyjnej

Gry militarne zaczęły być coraz bardziej popularne w wieku siedemnastym [3]. Badano wówczas, głównie we Francji, na drodze analitycznej zdolności obronne twierdz przez poddawanie ich projektów hipotetycznemu oblężeniu, zgodnie oczywiście z ówczesnymi zasadami natarcia. Po serii pomyślnych prób, władze wojskowe Francji w tym okresie nie dopuszczały do budowy żadnej twierdzy, jeżeli jej projekt nie zawierał dziennika takiego hipotetycznego natarcia. Widać zatem na tym przykładzie, że już wtedy przykładano dużą uwagę do eksperymentów symulacyjnych i opierając się na ich wynikach.

W późniejszych latach metody symulacyjne zaczęto stosować w dydaktyce wojskowej. Pod koniec XVII wieku w Szkole Inżynierii Wojskowej w Mezieres wprowadzono do rozkładu zajęć „treningi”, w czasie których wykorzystywano gry. Na ich przykładzie tłumaczono reguły ataku i obrony pozycji. Uczniowie sporządzali relacje z przebytych ćwiczeń, w których brali udział i na podstawie tych relacji oceniano inteligencję oraz uzdolnienia przyszłych wojskowych. Relacje te były następnie opiniowane przez Ministra Wojny. Uważano, że w trakcie dwuletniego okresu szkolenia dwie symulacje oblężenia są wystarczające do wyszkolenia przyszłego sapersa w zakresie ataku i obrony fortyfikacji. Nie udokumentowano żadnego przypadku, aby w trakcie prowadzenia działań wojennych, skarżono się na kwalifikacje absolwenta tej szkoły.

W 1770 r. Oelsnitz, oficer pruski, opracował scenariusz gry symulującej wielką operację wojenną. Swoje doświadczenia przeniósł do Szkoły Rycerskiej Stanisława Augusta Poniatowskiego w Warszawie. Wykorzystując tego typu gry uczono kadetów zasad, jakimi należy się kierować w określonych sytuacjach. Sporządzono szkice sytuacji, wydawano rozkazy bojowe, uczono zasad konstruowania planów taktycznych i samodzielności. Tego typu program szkolenia uwzględniał także zajęcia w terenie oraz musztrę bojową.

W Niemczech gry wojenne zaczęły być popularne od 1812 r., kiedy to zainteresował się nimi Fryderyk Wilhelm II. Model pola walki był wykonany w piasku lub gipsie, a metalowe elementy reprezentowały piechotę, artylerię i kawalerię. Po raz pierwszy walczące strony podzielono na „czerwonych” i „niebieskich” oraz uzupełniono grę o system reguł. Problemy sporne rozstrzygał rozjemca (arbiter). W 1824 r. doświadczenia niemieckie w stosowaniu gier militarnych reprezentowano w Petersburgu na dworze Wielkiego Księcia Mikołaja.

Pod koniec XIX wieku informacje o grach wojennych dotarły na kontynent amerykański za pośrednictwem emigranta, byłego żołnierza armii bawarskiej. Spotkały się one z dużym zainteresowaniem, a w konsekwencji zaczęto konstruować własne scenariusze gier. Ze względu na to, że kontynent amerykański był w zasadzie odizolowany, jedyne zagrożenie mogło pojawić się od strony morza. Główny nacisk położono na konstruowanie gier wojennych morskich. Pierwsze gry tego typu zostały opracowane przez porucznika marynarki Williama McCarty Litte'a z US Naval War College w Newport, który za swe osiągnięcia w rozwoju tej formy szkolenia otrzymał w 1903 r. stopień komandora.

Wymieniono tu tylko kilka przykładów zastosowań symulacji w rozwiązywaniu problemów militarnych. Ponieważ zadaniem tej publikacji nie jest prezentacja historycznego tła rozwoju symulacji, a tylko zasygnalizowanie, że metoda ta jest z powodzeniem stosowana od dawna, dlatego wszystkich zainteresowanych tą problematyką odsyłamy do literatury. Warto jeszcze dodać, że wiele wniosków wynikających z przeprowadzenia gier militarnych zostało wykorzystanych podczas pierwszej i drugiej wojny światowej.

W 1956 r. na zlecenie American Management Association, dostosowano grę „Monopolog”, która dotyczyła problemów zaopatrzenia lotnictwa USA, do zagadnień gospodarczych. Gra ta była pierwszą próbą wykorzystania gier symulacyjnych w ekonomii. Jej zadaniem było wspomaganie procesów szkolenia kadry kierowniczej przedsiębiorstw. Od tego momentu notuje się coraz szersze zastosowanie symulacji do rozwiązywania problemów ekonomicznych, od prostych modeli systemów gospodarczych do gier bardziej złożonych, wykorzystywanych w procesach kształcenia kadry menedżerskiej.

Obecnie, na przykład tylko w Anglii, istnieje około 500 gier, z powodzeniem wykorzystywanych w trakcie kształcenia studentów uczelni ekonomicznych [1].

Z rozeznania autorów niniejszego artykułu wynika, że na wielu uczelniach polskich oraz w różnego typu szkołach biznesu są stosowane symulacyjne gry decyzyjne, z których jednak niewiele spełnia wymagania stawiane produktom handlowym. Uniemożliwia to praktycznie ich wykorzystanie poza miejscem powstania.

W świetle powyższych rozważań staje się oczywistym fakt znacznej popularności symulacyjnych gier decyzyjnych wśród studentów szkół menedżerskich na całym świecie. Również studenci krakowskiej Akademii Ekonomicznej od kilku już lat mają okazję zapoznać się z grami symulacyjnymi, z których najczęściej stosowane są KRAEK [5] oraz TEES w swych kolejnych wersjach [4].

Porównanie z klasycznymi metodami kształcenia menedżerów

Gry symulacyjne są z pewnością wskazanym sposobem szkolenia menedżerów, ponieważ nie pociągają za sobą ryzyka poniesienia strat, jakie mogłyby spowodować błędne decyzje podjęte i zrealizowane w rzeczywistości. W trakcie gier decyzyjnych można zdobyć doświadczenie dotyczące selekcji informacji dostarczanych graczowi, a niezbędnych w procesie podejmowania decyzji. Praca w cyklu decyzje-wyniki-decyzje-wyniki, pozwala na uczenie się na własnych błędach oraz na zdobywanie doświadczenia w zakresie pracy zespołowej. Warto tu również wskazać na element rywalizacji między uczestnikami gry, który stwarza dodatkowe motywacje do studiowania zasad racjonalnego podejmowania decyzji.

Gry decyzyjne, zwykle zawierające w swych raportach elementy merytoryczne, mogą zachęcić uczestników do studiowania takich problemów, jak analiza finansowa czy zastosowania metod ilościowych w procesie podejmowania decyzji. Wszystko to sprawia, że uczestnicy nabywają w trakcie gry określoną teoretyczną i praktyczną wiedzę. Z drugiej strony należy jednak mieć świadomość, że stopień opanowania wiedzy o systemie modelowanym i jej bezpośrednia przydatność zależy od realizmu scenariusza gry. Symulacyjne gry decyzyjne stwarzają również okazje do zdobycia umiejętności reagowania na działania prowadzone przez konkurencję, a także podejmowania decyzji w warunkach ryzyka.

Ze stosowaniem gier decyzyjnych wiąże się jednak pewne niedogodności, przede wszystkim natury finansowej. Mianowicie, wymagają one zwykle wykorzystania drogiego sprzętu komputerowego oraz kosztownego oprogramowania. Stawiają również wysokie wymagania przed arbitrami. Muszą oni bowiem posiadać znajomość wielu szczegółowych zagadnień z zakresu ekonomii oraz cechować się umiejętnością pełnienia roli doradcy uczestników rozgrywki.

Wydaje się jednak, że największą barierą, stojącą na drodze szerokiego upowszechnienia gier symulacyjnych na polskim rynku, jest brak szerokiej podaży symulacyjnych gier decyzyjnych, dostosowanych do warunków gospodarczych panujących w Polsce.

Założenia pakietu wspomagającego konstrukcję gier symulacyjnych

Konkludując należy stwierdzić, że istnieje już zapotrzebowanie

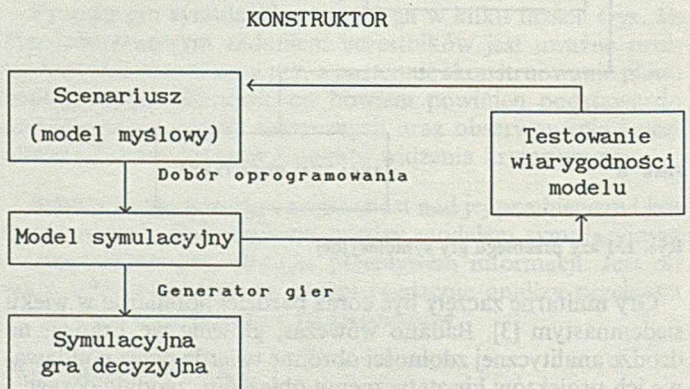
nie na silne narzędzie wspomagające zarówno sam proces budowy modelu symulacyjnego, jak i jego wykorzystanie do budowy trenażerów, jakimi są gry decyzyjne. Takim narzędziem od dawna jest komputer wyposażony w odpowiednie oprogramowanie. O ile jednak rozwój sprzętu komputerowego przekroczył najśmielsze oczekiwania sprzed lat kilku, o tyle istniejące oprogramowanie, dedykowane wyżej wymienionym zagadnieniom, w dalszym ciągu nie jest w pełni adekwatne do potrzeb.

W dotychczasowym sposobie podejścia do wykorzystania komputera jako narzędzia symulacji można wyróżnić trzy grupy rozwiązań, różniące się od siebie sposobem przekształcenia modelu zapisanego w formie równań matematycznych w program komputerowy. Do pierwszej grupy należy zaliczyć zastosowanie uniwersalnych języków programowania, takich jak FORTRAN czy Pascal. Oczywiście z formalnego punktu widzenia każdy model symulacyjny można oprogramować używając dowolnego języka uniwersalnego. Wiąże się z tym jednak liczne niedogodności, do których przede wszystkim można zaliczyć pracochłonną organizację prezentacji wyników symulacji, a może przede wszystkim wymaganie biegłej znajomości stosowanego języka programowania.

Grupa druga to specjalizowane języki symulacyjne, takie jak DYSMAP, DYNAMO, STELLA czy GPSS. Ich niewątpliwą zaletą jest znaczne uproszczenie procesu kodowania przez posługiwanie się licznymi makropoleceniami i funkcjami standardowymi, dostosowanymi do specyfiki modelowania i symulacji komputerowej. Nie umożliwiają one jednak przekształcenia modelu symulacyjnego w grę decyzyjną.

Do grupy trzeciej należy zaliczyć rozwiązania polegające na tworzeniu zintegrowanych pakietów, automatyzujących w dużym stopniu proces generowania i obsługi symulacyjnych modeli komputerowych. Warto w tym miejscu zaznaczyć, że wraz z upowszechnieniem sprzętu klasy PC, niektóre języki symulacyjne (np. DYSMAP2) przekształcono w mniej lub bardziej udane pakiety wspomagania symulacji.

Brak dostatecznie mocnego i uniwersalnego narzędzia jak zwykle zaowocował powstaniem szerokiej gamy mniej lub bardziej specjalizowanego oprogramowania w każdej z wymienionych wyżej kategorii. Natomiast w środowisku ekonomistów zajmujących się symulacją procesów ekonomicznych popularne są następujące trzy pakiety: DYSMAP-2, STELLA oraz SKOS (niestety tylko w Polsce, ze względu na brak oferty handlowej i liczne niedoskonałości).

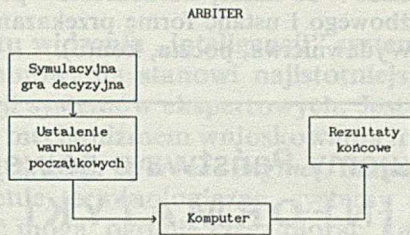


Rys. 2. Przebieg konstrukcji gry

Aby wyjaśnić, dlaczego mimo względnego bogactwa oferty programowej problem stworzenia narzędzia nadal istnieje, zwróćmy uwagę na to, że problematyka modelowania procesów

gospodarczych oraz wykorzystania gier decyzyjnych do szkolenia i podnoszenia kwalifikacji kadry menedżerskiej, angażuje ludzi o bardzo zróżnicowanym profilu zawodowym oraz zróżnicowanych wymaganiach pod adresem oprogramowania. Można przy tym wymienić trzy lub cztery kategorie osób, nierozdzielnie związanych ze stosowaniem gier symulacyjnych. Należą do nich autorzy scenariusza, programiści, arbitrzy oraz uczestnicy gry.

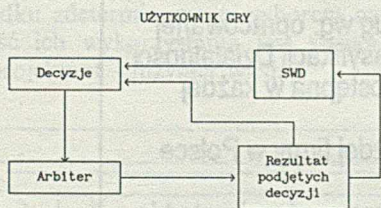
Zwykle autorem scenariusza jest specjalista, który jednocześnie nie jest zawodowym informatykiem. Z kolei specyfika konstrukcji gry decyzyjnej (rys. 2), a szczególnie procesu jej testowania, wymaga samodzielności w posługiwaniu się oprogramowaniem. Optymalnym rozwiązaniem byłoby więc posiadanie takiego pakietu oprogramowania, który umożliwiłby łatwe oprogramowanie gry autorom jej scenariusza.



Rys. 3. Przebieg gry z punktu widzenia arbitra

Należy w tym miejscu podkreślić, że próby opracowania takich systemów były podejmowane w Akademii Ekonomicznej w Krakowie. Ich efektem są dwa systemy: System Kompleksowej Obsługi Symulacji oraz SYLLABUS. Niestety żaden z tych pakietów nie został w pełni zrealizowany. Tym samym ich użyteczność jest w znacznym stopniu ograniczona, a całe zadanie czeka na ponowną próbę rozwiązania. Z kolei oczekiwania arbitra gry (rys. 3) są związane z ułatwieniami sfery praktycznego stosowania gry. Chodzi tu przede wszystkim o łatwość zmiany warunków początkowych, wprowadzania decyzji graczy oraz generowania raportów końcowych.

Wymagania graczy koncentrują się natomiast na procesie przygotowywania decyzji, ze szczególnym uwzględnieniem systemu wspomaganie decyzji (rys. 4).



Rys. 4. Przebieg gry z punktu widzenia użytkownika

W tym kontekście w niniejszym opracowaniu podjęto próbę sformułowania koncepcji systemu (pakietu) wspomaganie procesu modelowania i symulacji procesów gospodarczych przy następujących założeniach:

- pakiet jest ukierunkowany na obsługę modeli symulacyjnych ciągłych, dyskretnych i ciągłodyskretnych,
- z punktu widzenia technologii wytwarzania oprogramowania powinien powstać otwarty system informatyczny, pozwalający na rozbudowę, zmiany i dostosowywanie do zmieniających się warunków sprzętowo-programowych,
- pakiet powinien spełniać wszystkie typowe dla tej klasy systemów funkcje, wśród których za najważniejsze uznano:

- możliwość specyfikacji formalnej struktury modelu,
- możliwość specyfikacji i generowania danych potrzebnych do obliczeń symulacyjnych,
- prezentację i analizę wyników modelu,
- sterowanie procesem dokonywania eksperymentów symulacyjnych,
- zarządzanie danymi, modelami i metodami.

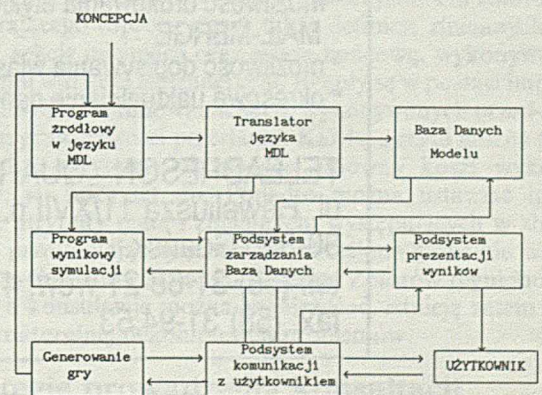
Dostępne w systemie funkcje oprogramowania powinny pozwalać na łatwy zapis równań modelu w postaci programu, kompilację kodu źródłowego na postać wynikową, dokonywanie eksperymentów z modelem oraz prezentowanie ich wyników w wybranej formie graficznej przy spełnieniu wymogu w pełni polskojęzycznej komunikacji z użytkownikiem oraz rozbudowanym systemem podpowiedzi umożliwiającym łatwe opanowanie zasad pracy z systemem osobom obeznanym z komputerem w stopniu podstawowym.



Rys. 5. Proces komputerowego wspomaganie symulacji

Komputerowa gra decyzyjna powstaje zwykle na drodze pewnego szczególnego przekształcenia modelu symulacyjnego. Przekształcenie to polega przede wszystkim na takiej reorganizacji procedur wprowadzania i wyprowadzania danych, aby wielu użytkowników mogło korzystać z tego samego modelu. System powinien być wyposażony w mechanizmy na tyle automatyzujące cały proces od strony informatycznej, aby mógł go przeprowadzić autor modelu bez potrzeby angażowania fachowców z zakresu informatyki (programistów). Dodatkowo w wersji docelowej system winien umożliwiać:

1. Generowanie różnych wariantów gier opartych na tym samym modelu.
2. Pozwalać na generowanie – stosownie do potrzeb – zarówno jedno- jak wielostanowiskowej wersji.
3. Zawierać jako integralną składową proces definiowania instrukcji gry dostępnej na stanowisku rozgrywki.
4. Wyposażać generowane gry w mechanizmy wspomaganie proces podejmowania decyzji przez uczestnika gry w stopniu



Rys. 6. Schemat koncepcji systemu informatycznego

przewidzianym przez autora (dostęp do określonych informacji, możliwość korzystania z prostego arkusza kalkulacyjnego i itp.).

* * *

Reasumując dotychczasowe rozważania, postulowany system komputerowego wspomaganie symulacji widziany oczyma głównego adresata, jakim jest specjalista z dziedziny modelowania procesów gospodarczych, winien umożliwiać realizację pojedynczego przedsięwzięcia w układzie pokazanym na rys. 5.

Wynika stąd struktura systemu informatycznego, która w największym uproszczeniu może wyglądać jak na rys. 6.

LITERATURA

- [1] Elgood C.: Handbook of Management Games. Gower Publishing Company, Aldershot, 1984
- [2] Matera A.: Symulacyjna gra kierownicza – pojęcia i struktura. Prace Szkoły Lubachów'80. TNOIK Wrocław–Gliwice, 1980
- [3] Nolewajka R.K.: Rozwój symulacyjnych gier decyzyjnych na przykładach zastosowań wojskowych od połowy XVII wieku do 1956 r. Prace Szkoły Trzebiezowice'82. TNOIK Wrocław–Gliwice, 1982
- [4] Skrzypek J., Szubra M.: Symulacyjna gra Decyzyjna TEES-1 1992 (w druku)
- [5] Wąsik B.: Komputerowa Gra Kierownicza KREAK-1, AE Kraków (praca zbiorowa) 1989
- [6] Wilson A.: The Bomb and the Computer – War gaming from Ancient Chinese Mapboard to Atomic Computer, Celacorte Press, New York, 1968
- [7] Zawiślak A.: Pułapki zarządzania. GLOB, Szczecin, 1984.

W związku z wprowadzeniem powszechnego podatku dochodowego i koniecznością informowania właściwych Urzędów Skarbowych o wszystkich wypłatach zwracamy się do Autorów wysyłających teksty do opublikowania o podawanie następujących danych:

- × nazwisko, imiona (pierwsze i drugie),
- × imiona: ojca i matki,
- × miejsce i data urodzenia,
- × numer identyfikacyjny PESEL (wpisany przez Biuro Meldunkowe do dowodu osobistego – nie mylić z numerem dowodu osobistego!),
- × dokładny adres (miejsce zameldowania),
- × adres Urzędu Skarbowego właściwego dla miejsca zamieszkania Autora (bardzo ważne – bez tej informacji nie możemy przekazać honorarium do wypłaty!).

Prosimy także – do naszej wiadomości – podać numer telefonu służbowego i ustalić formę przekazania honorarium (kasa Wydawnictwa, poczta, konto).

Informujemy Państwa o nowej cenie
INFORMATYKI
w roku przyszłym
– 25 000 złotych za egzemplarz

Kolejny przełom w dziedzinie katalogów informacji o biznesie

TELEADRESON

Kompletny komputerowy katalog ponad 500,000 firm w Polsce
Rozpowszechniany jako program na dyskietkach na komputer IBM PC

- * pełne nazwy i adresy firm, sklepów, instytucji, hurtowni, przedsiębiorstw, biur, urzędów, szkół itd.
- * numery telefonów i faxów
- * szczegółowa klasyfikacja handlu, przemysłu i usług wg. opracowanej przez naszą firmę "Rozszerzonej Europejskiej Klasyfikacji Działalności"
- * informacja o objętości 10-tomowej encyklopedii dostępna w każdej chwili na Twoim komputerze
- * w ciągu 1 minuty znajdziesz adres i telefon do każdej firmy w Polsce podając tylko jej nazwę
- * możliwość drukowania etykiet adresowych lub korespondencji w trybie MAIL-MERGE
- * możliwość dopisywania własnych adresów i innych informacji do bazy
- * okresowe uaktualnianie danych o wszystkich firmach

TELEADRESON - QUART

ul. Heweliusza 11/XVII p.

80-890 Gdańsk

tel: (58) 31-68-21 wew. 473 i 474

fax: (58) 31-94.53

01/4/92

Strojenie mechanizmu rozumowania systemów ekspertowych z uwzględnieniem warunków niepewności

Z punktu widzenia „inteligencji” systemu, mechanizm rozumowania stanowi najistotniejszy element konstrukcji systemów ekspertowych. Jest najczęściej nazywany mechanizmem wnioskowania lub maszyną wnioskującą (ang. *inference engine*). Stanowi to pewne zawężenie terminologiczne, systemy te bowiem realizować mogą, ogólnie rzecz biorąc, każdy z podstawowych sposobów rozumowania (wnioskowanie, sprawdzanie, dowodzenie i tłumaczenie).

Mechanizm rozumowania w systemie ekspertowym odpowiada za całą warstwę „myślenia” systemu użytkowego. W systemach narzędziowych, przeznaczonych do budowy systemów ekspertowych, mechanizm ten istnieje jako pewien metaalgorytm, szkielet, któremu w konkretnym systemie użytkowym należy nadać cechy, specyficzne dla procesów rozumowania w danej dziedzinie. Moc konkretnego pakietu, jako narzędzia do budowy systemów ekspertowych, zależy głównie od ogólności i elastyczności wbudowanego mechanizmu rozumowania, tzn. od tego, ile i jakie stopnie swobody przewidziano w metaalgorytmie, oraz jaka jest skala udostępnionych możliwości strojenia mechanizmu. Konieczność uchwycenia natury i specyfiki procesów myślowych z danej dziedziny powoduje, że synteza warstwy parametrów strojenia mechanizmu rozumowania jest w tworzeniu systemu ekspertowego na ogół etapem trudnym. Stopnie swobody mechanizmu rozumowania są w każdym przypadku zdeterminowane wyborem narzędzia, natomiast trafność ich wykorzystania jest sztuką, której musi dokonywać projektant – inżynier wiedzy.

Ideę działania „maszyny wnioskującej” trafnie ilustruje rysunek, zaczerpnięty z [2].

Możliwości modyfikacji mechanizmu rozumowania omówimy na przykładzie pakietu GURU, traktując go jednak głównie jako ilustrację pewnych koncepcji natury ogólnej.

Charakterystyka pakietu GURU

Zintegrowany pakiet GURU firmy Micro Data Base Systems (USA) jest w miarę uniwersalnym narzędziem do tworzenia regulowych systemów ekspertowych. Istnieje w nim dość bogaty mechanizm rozumowania, rozbudowany aparat zmiennych specyfikujących środowisko, podsystem do budowy zbiorów reguł, a ponadto następujące elementy wspomagające tworzenie bazy wiedzy:

- podsystem zarządzania relacyjną bazą danych z językiem wyszukiwania, opartym na standardzie SQL,
- moduł arkuszy obliczeniowych w standardzie LOTUS 1-2-3.

W skład pakietu wchodzi również komponenty, ułatwiające tworzenie przyjaznego interfejsu systemu z użytkownikiem:

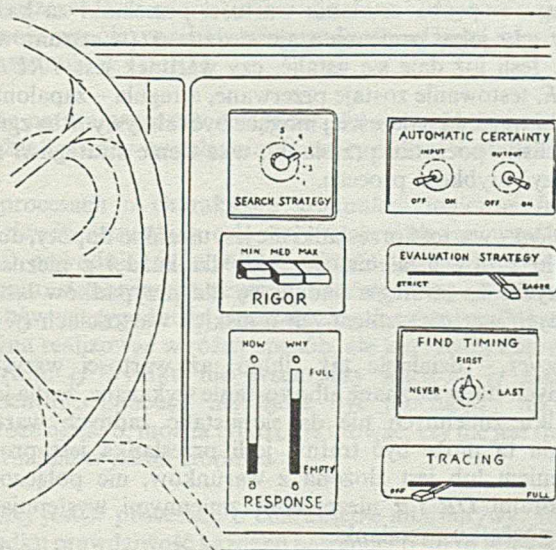
- moduł projektowania formularzy ekranowych generator raportów użytkowych,
- pakiet graficzny do przedstawiania danych z tablic i arkuszy w formie wykresów i rysunków,
- moduł śledzenia procesów rozumowania (jak? i dlaczego?).

Oprócz wymienionych elementów pakiet zawiera wygodny edytor tekstowy oraz mechanizmy językowe, umożliwiające budowanie programów strukturalnych. Wszystkie elementy pakietu są w pełni zintegrowane pod względem funkcjonalnym.

Proces konsultacji przebiega z wykorzystaniem zbioru reguł, zawierającego (oprócz reguł) także definicje zmiennych roboczych, sekcję inicjującą oraz sekcję końcową, wykonywaną po ustaleniu wartości zmiennej celu. Reguły są w postaci implikacji z możliwością definiowania akcji, wykonywanych przed rozważaniem prawdziwości przesłanki. Każdej z reguł można przypisać także atrybuty, określające umowny koszt wykonania reguły, jej priorytet itp. Ponadto można ustawiać lokalnie pewne atrybuty reguł i zmiennych, występujących w zbiorach reguł, przesłaniając tym samym określone globalnie wartości zmiennych środowiskowych systemu. Zbiorów reguł może być wiele, a konsultacje można zagnieżdżać, istnieją zatem możliwości naturalnej dekompozycji problemów.

Strategie prowadzenia konsultacji

Fundamentalne znaczenie ma wybór metody prowadzenia



procesu rozumowania. W GURU istnieją trzy zasadnicze strategie prowadzenia konsultacji:

- rozumowanie w tył (ang. *backward chaining, reverse reasoning*),
- rozumowanie w przód (ang. *forward chaining, forward reasoning*),
- rozumowanie mieszane (ang. *mixed mode chaining*).

Rozumowanie w tył, standardowo najczęściej stosowane, polega na iteracyjnym wybieraniu i wykonywaniu reguł, które w konkluzji wyznaczają wartość poszukiwanej w danym kroku zmiennej. Łańcuch przebiega zatem od zmiennej celu przez zmienne potrzebne do ustalania wartości przesłanek kolejno wybieranych reguł. Jest to postępowanie w pewnym sensie najbardziej naturalne.

Rozumowanie w przód ma charakter generowania wiedzy, dopełniającej zawartą w systemie informację. Polega na iteracyjnym wybieraniu i wykonywaniu reguł, które w danym kroku mają prawdziwą logicznie przesłankę. Podczas takiego działania, w pewnym momencie, niejako „po drodze”, można osiągnąć wartość zmiennej celu. Postępowanie to jest celowe w sytuacjach, w których z niewielkiego zasobu wiedzy staramy się uzyskać maksimum informacji, często bez sprecyzowanej zmiennej celu.

Szczególnie ciekawe wydaje się stosowanie strategii mieszanej, polegającej na przełączaniu (na jeden krok) trybu rozumowania z wstecznego na postępujące, co następuje po ustaleniu wartości zmiennej, mającej lokalnie zdefiniowany odpowiedni atrybut. Strategia ta stwarza możliwość wyciągania wniosków towarzyszących w kluczowych momentach rozumowania w tył.

Możliwości sterowania działaniem mechanizmu rozumowania

Działanie mechanizmu rozumowania w GURU zależy od wartości pewnych zmiennych środowiskowych lub analogicznych wartości, ustawianych lokalnie w zbiorach reguł. Wartości ustawione lokalnie mają priorytet wyższy od wartości zmiennych środowiskowych. Standardowo zmienne i atrybuty w zbiorach reguł mają ustawione pewne wartości domyślne. Wartości opcji w regułach można wprowadzać w momencie budowania lub modyfikowania reguł. Wartości zmiennych środowiskowych można zmieniać następującymi sposobami:

- w pliku konfiguracyjnym,
- interakcyjnie przed konsultacją,
- w programie wywołującym konsultację,
- w sekcji początkowej zbioru reguł.

Przez odpowiednie ustawienie wartości zmiennych środowiskowych można uzyskać następujące cztery zasadnicze typy oddziaływań na mechanizm rozumowania:

- 1) ustalać, na ile rygorystycznie mechanizm rozumowania ma rozważać reguły,
- 2) ustalać kolejność wyboru reguł do rozważania,
- 3) wybierać strategię obliczania wartości przesłanek reguł,
- 4) ustalać, czy i kiedy mechanizm rozumowania ma wykonywać specjalne akcje, zdefiniowane dla zmiennych typu *unknown*.

Ad (1). W procesie konsultacji można zmieniać obszar poszukiwań przez ustalanie, na ile rygorystycznie mechanizm rozumowania ma rozważać reguły w trakcie poszukiwania wartości określonej zmiennej. W ten sposób dynamicznie modyfikuje się zasięg (głębokość) poszukiwań.

Rygorystycznością rozważania reguł (ang. *reasoning rigor*) dla badania nieznanymi zmiennymi steruje się przez wybór

z następujących możliwości:

A – absolute, *C* – considerate, *M* – minimal.

Ostateczny efekt ustawienia jednej z tych wartości zależy jednocześnie od przyjętego sposobu rozumowania. Mówiąc ogólnie, wartość *A* oznacza maksymalnie głębokie, pełne poszukiwania. System nie poprzestaje np. na osiągnięciu wartości potrzebnej zmiennej, ale wykonuje w każdym kroku wszystko, co jest możliwe.

Wartość *M* oznacza przeciwną skrajność. System „najkrótszą drogą” zmierza do celu i zadowala się pierwszym wynikiem osiągniętym w danym kroku. Proces konsultacji trwa wtedy z reguły krócej, rozważaniu podlega mniejsza liczba reguł, ale wynik nie zawsze musi być ten sam, a przede wszystkim jest na ogół mniej pewny. Wartość *C*, zależnie od przyjętego trybu rozumowania, oznacza pewien rodzaj strategii pośredniej.

Ad (2). Ważnym kluczem strojenia mechanizmu rozumowania jest możliwość ustalania kolejności wyboru reguł do rozważania. Wybór następuje spośród reguł kandydujących do rozważania, tzn. takich, których konkluzje mogą ustalić lub zmienić wartość potrzebnej zmiennej (podczas rozumowania w tył), lub dla których wartości przesłanek dają się obliczyć (podczas rozumowania w przód). Istnieje sześć podstawowych kryteriów wyboru, (z których można tworzyć kryteria złożone):

F – wybór pierwszej z kandydujących reguł wg kolejności występowania w zbiorze reguł,

P – wybór reguły o najwyższym priorytecie,

C – wybór reguły o najniższym koszcie,

U – wybór reguły o najmniejszej liczbie zmiennych o wartościach nieznanymi w przesłance,

H – wybór reguły, której wykonanie daje najwyższy wskaźnik wiarygodności *cf* dla nieznanymi zmiennej,

R – pseudolosowy wybór reguły.

Ad (3). Przez wybór strategii obliczania wartości przesłanek reguł można wpływać na to, na ile szczegółowo i wyczerpująco prowadzi się poszukiwania w każdym kroku rozumowania. Lokalna zmiana tej cechy na poziomie reguły może spowodować np. przyspieszenie procesu rozumowania przez wybieranie minimalnej drogi do celu. Zależnie od postaci przesłanki ogólnie są polecane następujące strategie:

bez operatorów *OR* z operatorami *OR*

z *cf* *S* (strict) *P* (patient)

bez *cf* *S* (strict) *E* (eager)

E – *eager* – próba ustalania wartości przesłanki za każdym razem, gdy jakaś zmienna staje się znana (w rozumowaniu w tył). Jeśli już daje się ustalić, czy warunek jest *TRUE* czy *FALSE*, testowanie zostaje przerwane, a reguła – zapalona lub nie. Jeśli warunki najczęściej mogące być fałszywymi są zgrupowane blisko początku przesłanki, ustawienie strategii *E* może zwiększyć szybkość procesu.

P – *patient* – wartość przesłanki nie jest ustalona dopóty, dopóki nie było próby ustalenia wartości dla każdej z nieznanymi zmiennymi. Ta strategia nadaje się dla przypadków łączenia wspólnych wartości zmiennymi o niskich wartościach *cf*.

S – *strict* – działanie tak długo, aż wartości wszystkich zmiennymi staną się znane albo zostanie wykazane, że dla jednej lub kilku zmiennymi nie da się ustalić żadnych wartości. Strategia ta może być trafna, jeśli przesłanka jest prostym wyrażeniem lub jest złożona z warunków, nie połączonych operatorami *OR* i z nieznanymi zmiennymi występującymi blisko początku przesłanki.

Ad (4). Można także ustalać, czy i kiedy mechanizm rozumowania ma wykonywać zdefiniowane wcześniej, specjalne akcje dla znalezienia wartości zmiennych nieznanych (np. pobieranie wartości od użytkownika). Zależnie od semantyki zagadnienia, wykonanie tych akcji może być ostatnim ratunkiem lub wymuszenie pobrania wartości aktualnej z wejścia może poprzedzać proces szukania wartości wtedy, gdy informacja ta ma najwyższy priorytet. Można także chwilowo zablokować wykonanie przygotowanych akcji specjalnych.

Rozumowanie w warunkach niepewności Zmienne fuzzy

Możliwość rozumowania w warunkach niepewności w istotny sposób zwiększa zakres zastosowań systemów ekspertowych. Informacje, wprowadzane do bazy wiedzy, często mają charakter danych o określonym stopniu wiarygodności. Zarówno nasza wiedza o rzeczywistości na ogół nie jest stuprocentowa, jak i natura ludzkiego myślenia dopuszcza rozumowanie, w wyniku którego powstają hipotezy o pewnym prawdopodobieństwie, prognozy i decyzje z określonym ryzykiem itp. Różne systemy ekspertowe spełniają w różnym stopniu postulat uwzględniania wiedzy niepewnej.

W pakiecie GURU istnieje mechanizm, umożliwiający wykorzystywanie danych z określonymi współczynnikami wiarygodności *cf* (ang. *certainty factor*), prowadzenie obliczeń z ich uwzględnieniem, jak i rozumowanie na podstawie powiązań między faktami (reguł), które same mogą być słuszne jedynie do pewnego stopnia.

Często w wyniku rozumowania z wykorzystaniem współczynników wiarygodności powstaje potrzeba jednoczesnego uwzględnienia w dalszych rozważaniach różnych możliwych wartości danego obiektu, każdej o określonym poziomie wiarygodności. GURU ma aparat zmiennych typu *fuzzy* o naturze zbiorów rozmytych (Zadeh'a) [3]. Zmienne *fuzzy*, posiadając wiele możliwych wartości jednocześnie (wraz z ich współczynnikami wiarygodności), umożliwiają względnie wierne odwzorowanie rzeczywistości informacyjnej oraz równoległe prowadzenie różnych nurtów rozumowania. Klucze strojenia systemu w postaci parametrów środowiskowych umożliwiają sterowanie i kontrolę procesów obliczeniowych, obcinanie ich nadmiernego kombinatorycznego wzrostu, wybór najbardziej obiecujących ścieżek rozumowania, dynamiczne zmiany parametrów, uzależnione od poziomu wartości współczynników, warunków logicznych itp.

W dalszej części zostaną omówione nieco bliżej różne algebry obliczania współczynników wiarygodności *cf* oraz sposób ich wykorzystania w GURU.

Sama natura procesów rozumowania generuje dwa zasadniczo różne sposoby uwzględniania wiarygodności obiektów.

W procesach o charakterze koniunkcji logicznej faktów prawdziwość wszystkich członów (z określonymi współczynnikami) powoduje, że wynik przyjmuje się jako prawdziwy w stopniu, stanowiącym pewne połączenie (ang. *joint*) współczynników wiarygodności poszczególnych członów. Połączenie to można realizować w różny sposób, ale jest rzeczą naturalną, że wartość *cf* wyniku nie przekroczy wartości *cf* żadnego z elementów. Łańcuch nie może być mocniejszy niż jego najsłabsze ogniwo, można tylko dyskutować, czy nie jest słabszy i ewentualnie w jakim stopniu.

Inaczej jest w procesach o charakterze alternatywy. W tym przypadku prawdziwość każdego następnego faktu potwierdza,

wzmacnia (ang. *confirmative*) przesvědzenie o prawdziwości wyniku. Dla tej operacji na współczynnikach jest rzeczą naturalną, że wynik nie może być mniejszy od żadnego z elementów. Dyskusyjny jest jedynie fakt, czy równoległy spłot łańcuchów nie jest silniejszy od najmocniejszej jego składowej.

Te dwa działania przewijają się nieustannie w procesach rozumowania. Na przykład równoległe z wykonywaniem wyrażeń numerycznych czy tekstowych działa *joint*. W przypadku wyrażeń logicznych sprawa jest bardziej złożona. Wybór operacji na współczynnikach zależy nie tylko od operatorów logicznych, ale nawet od wartości członów wyrażenia. Np. w przypadku prawdziwości koniunkcji działa *joint*, ale w przypadku przeciwnym, wiarygodność (*cf*) fałszywości każdego następnego elementu potwierdza fałszywość wyniku, a więc działa *confirmative*. Na wynik konkluzji reguły ma wpływ połączenie (*joint*) stopnia pewności przesłanki i wiarygodności samej reguły, *joint* jest więc w pewnym sensie operacją wewnątrzregulową. W przypadku uzyskania dla zmiennej *fuzzy* z dwóch źródeł (np. w wyniku wykonania dwóch reguł) tej samej wartości, powstaje problem uzgodnienia wartości między oboma współczynnikami. Jest sprawą oczywistą, że każdy z nich wzmacnia rzetelność wyniku, zatem działa *confirmative* i w tym sensie jest to operacja międzyregulowa.

System GURU przez ustalanie atrybutów środowiskowych:

- statycznie na poziomie globalnym,
 - dynamicznie w procesie rozumowania,
 - przez lokalne przypisania do obiektów,
- pozwała wybierać spośród dostępnych w nim czterech realizacji operacji typu *joint* i czterech operacji *confirmative*, co umożliwia wybór spośród 16 algebr współczynników wiarygodności.

Operacje typu *joint*:

- 1) metoda minimum: $\min(x, y)$,
- 2) metoda iloczynowa (ang. *product*): $x * y / 100$,
- 3) metoda średniej (ang. *average*): $(\min(x, y) + x * y / 100) / 2$,
- 4) metoda Bonczek'a-Eagin'a: $(x * y / 100) * (2 - \max(x, y) / 100)$.

Najwyższe, najbardziej ryzykowne, odważne wyniki daje metoda minimum. Jest ona – w myśl zdrowego rozsądku – granicą, poza którą wartości żadnej operacji typu *joint* wyjść nie mogą. W prostych rozważaniach jest to jednocześnie najbardziej naturalna metoda i przypuszczalnie bywa często stosowana. Powstają jednak pytania, np. czy spadek wartości wyższego ze współczynników wiarygodności nie powinien mieć żadnego wpływu na wynik? W pozostałych trzech metodach wpływ ten zaznacza się wyraźnie. Najbardziej zachowawczą, rygorystyczną jest metoda iloczynowa, dająca zawsze wyniki najniższe, stanowiąca naturalną dolną granicę dla operacji typu *joint*. Wydaje się, że operacja ta powinna być stosowana w sytuacjach, w których z semantyki zastosowań wynika wymóg maksymalnego bezpieczeństwa. Pomędzy tymi granicami GURU proponuje dwie operacje kompromisowe. Jeżeli wyższy z dwu współczynników przekracza 50, operacja Bonczek'a-Eagin'a jest bardziej optymistyczna niż średnia, tzn. daje wyniki bliższe metodzie minimum. W przeciwnym przypadku operacja Bonczek'a-Eagin'a jest bardziej ostrożna oraz daje wartości bliższe iloczynowi. Punktem równowagi między metodami jest wartość 50 wyższego z argumentów.

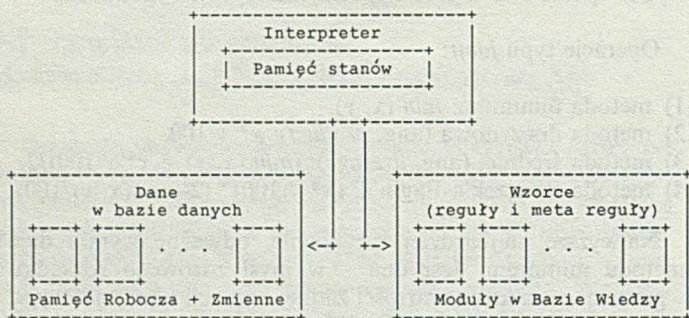
Operacje typu *confirmative*:

- 1) metoda maksimum: $\max(x, y)$,
- 2) metoda sumy probabilistycznej (ang. *probability sum*):
 $(x + y) - (x * y / 100)$,

dokończenie na s. 25

Wpływ algorytmów tablic rozproszonych na efektywność działania systemów ekspertowych

Celem wzrostu efektywności działania wielu algorytmów stosowanych w systemach ekspertowych (SE) – patrz rys. 1 – często korzysta się ze specjalnych struktur danych oraz wprowadza relację porządkującą dane. Istnieje wiele różnorodnych sposobów uporządkowania danych w pamięci roboczej interpretera. Większość algorytmów opiera się na strukturach będących często bardzo wysublimowanymi odmianami drzew. Okazuje się jednak, że efektywność wyszukiwania informacji może znacznie poprawić się, jeżeli zamiast struktur danych o dużym stopniu złożoności wprowadzić tzw. struktury nie uporządkowane zwane często mieszany (ang. *hash data structure*) [2].



Rys. 1

Podstawowy cykl pracy interpretera obejmuje:

- wybranie z bazy wiedzy reguł na podstawie aktualnego stanu pamięci roboczej (ang. *fetching*),
- porównanie aktualnego stanu z regułami aktywnymi (ang. *matching*),
- rozwiązywanie ewentualnych konfliktów (ang. *conflict resolution*) między wybranymi regułami,
- zmiana stanu pamięci roboczej i bazy wiedzy (ang. *execution*).

W powyższym cyklu wykorzystuje się wiedzę z następujących trzech źródeł:

- pamięci roboczej interpretera,
- bazy wiedzy SE,
- pamięci aktualnych stanów interpretera.

Ponieważ do najczęściej stosowanych struktur danych w technologii budowy SE należą obok list, tablice, to warto zająć się algorytmami optymalizującymi dostęp do przechowywanych w nich danych. Algorytmy mieszania danych (AMD) można stosować w organizacji efektywnego dostępu do wartości zmiennych roboczych w pamięci roboczej maszyny wnioskującej (ang. *inference engine*) lub na wyższych poziomach pracy otoczki (powłoki) SE, jak np. przechowywanie reguł. Zastosowanie

wanie algorytmów przemieszania danych służy w tym przypadku do szybkiego przeszukiwania zbioru reguł (w celu odrzucenia reguł, które na pewno nie odpowiadają aktualnemu stanowi pamięci roboczej maszyny wnioskującej) oraz do implementowania pamięci wirtualnej, w której byłoby możliwe przechowywanie i udostępnianie dużej, zawierającej ponad 100 reguł bazy wiedzy (dzięki temu można mówić o przemysłowym wykorzystaniu SE) wraz z efektywnymi algorytmami dostępu do nich (organizacja indeksowania za pomocą tablic rozproszonych). Algorytmy przemieszania danych można stosować w SE typu „powłoka” na poziomie bardzo niskim w przypadku, gdy istnieje dostęp do źródeł programów lub w dowolnym komercyjnym pakiecie typu kompilator PROLOGU (implementowanie dyskowej pamięci wirtualnej dla reguł), niezależnie od producenta. Słuszność wyboru algorytmów przemieszania danych do optymalizacji tych właśnie fragmentów SE potwierdzają kilkunastoletnie już doświadczenia eksploatacji SE. Wynika z nich jasno, że najlepsze rezultaty osiąga się, jeżeli SE działa z dużą i często dynamicznie rosnącą liczbą reguł. Siła SE polega na ilościowym przedstawieniu za pomocą wielu reguł różnych aspektów modelowanego fragmentu rzeczywistości, a nie na kilku „super” regułach uzyskanych od „super” ekspertów. Wykorzystanie AMD pozwoli na praktycznie opłacalną metodę przechowywania i szybkiego dostępu do bardzo dużych zbiorów reguł [3].

Istnieje jeszcze jedna możliwość implementowania tablic rozproszonych do samego przedstawienia reguł. Do tego celu najczęściej stosuje się odmianę drzew binarnych. Jednakże struktura drzewiasta reguły jest jedynie szkieletem, na którym jest rozpięta zawartość (istota, treść) reguły. Tkwi ona w „liściach” tego drzewa. Ponieważ praca maszyny wnioskującej polega na cyklicznym sprawdzaniu zawartości reguł, to skrócenie tej procedury znacznie przyspieszy proces wnioskowania. Z wielu powodów jest celowe połączenie struktury drzewa zrównoważonego z metodą przemieszania danych. Po pierwsze, dzięki połączeniu obu tych metod jest możliwe wykorzystanie algorytmów przemieszania danych do obsługi procedur wyszukiwania na wybranej hierarchii drzewa zrównoważonego lub wykorzystanie struktur drzewopodobnych do przedstawienia tych synonimów, które zaistniały w trakcie działania algorytmu przemieszania danych. Po drugie, algorytmy przemieszania, podobnie jak drzewa, uzyskały dostatecznie dużą popularność i okazały się niezwykle efektywnym sposobem związującym formalne przedstawienie skomplikowanych struktur symbolicznych z ich istotą w świecie zewnętrznym.

Należy zwrócić uwagę, że metoda tablic mieszanych rozwiązuje klasyczny w informatyce problem, a mianowicie jak przechowywać informacje, aby dostęp do nich był jak najszybszy. Oznacza to, że mając zbiór zapisów, każdy w postaci

unikalnego klucza K_i i części zawierającej dane D_i , żąda się, aby dostęp do dowolnie wybranego zapisu był możliwie jak najszybszy. Unikalny klucz w każdym zapisie jest po to, aby na obiekcie implementować elementarne operacje, takie jak przeglądanie, wstawianie czy usuwanie, a także operacje związane z zarządzaniem tym zbiorem, jak np. reakcja na przepelnienie. Część danych zapisu może zawierać jedno lub wiele pól dowolnego typu np. liczby, łańcuchy znaków lub daty. W praktyce takie zbiory są implementowane najczęściej jako listy wewnętrzne (drzewa) lub tablice, bądź jako zbiór w pamięci zewnętrznej komputera.

Wymóg minimalizowania czasu dostępu może być w konflikcie z innymi kryteriami wydajności systemu SE, np. z minimalizacją pamięci przechowującej dane lub program, albo też może spowodować niepożądane efekty uboczne przy niektórych operacjach obsługi zbioru zapisów. Np. listy są posortowane strukturami, które stosunkowo łatwo można przeglądać, ale operacje wstawienia są bardzo czasochłonne, gdyż zaburzając porządek w liście wymuszają potrzebę nowego posortowania elementów listy.

Reasumując można stwierdzić, że szybkość dostępu do danych oraz efektywne wykorzystanie pamięci zależą głównie od typu i struktury danych, nośnika pamięci oraz algorytmu wyszukiwania. W przypadku zbiorów danych przechowujących reguły, stawia się zwykle na pierwszy plan zminimalizowanie czasu dostępu do nich i to nawet wtedy, gdy prowadzi to do znacznej liczby operacji porównań w pamięci operacyjnej. Z kolei trzymanie reguł w postaci list lub tablicy w pamięci wewnętrznej powodują, że kryteriami efektywności stają się: minimalizacja czasu porównywania kluczy, zużycia pamięci wewnętrznej oraz całkowitego czasu przeszukiwania.

Algorytm mieszania danych (AMD)

Zagadnienia wyboru sposobu przedstawiania danych w SE są ściśle związane z operacjami, za pomocą których następuje wnioskowanie. Od trafnego wyboru struktur danych zależy efektywne działanie całego systemu. Na najniższym poziomie abstrakcji do operacji tych należą: pobranie danych, aktualizacja, włączenie nowego obiektu do struktury oraz jego wyłączenie. Podstawową operacją działania na obiekcie jest bezpośrednie udostępnienie jego wartości. Operacja ta jest nierozzerwalnie związana ze sposobem przedstawienia danych. Przez bezpośredni dostęp rozumie się przechowywanie obiektów w pamięci operacyjnej komputera, a co za tym idzie – dostęp przez adres obiektu w pamięci. W zagadnieniach związanych z przeszukiwaniem zakłada się, że wszystkie dane pamiętane w systemie są zaopatrzone w klucz jednoznacznie identyfikujący każdy obiekt. Dlatego mówiąc o udostępnieniu danych myśli się głównie o bezpośrednim dotarciu w sposób efektywny do kluczy obiektów, a nie samych danych. Dla uproszczenia zakłada się, że każdy obiekt ma jeden unikalny klucz główny (cecha obiektu). Z kolei przy dużej liczbie obiektów nawet klucze nie zawsze muszą być dostępne w sposób bezpośredni. Występuje to nawet przy średniej wielkości SE. W ten sposób powstaje hierarchia pamięci ([operacyjna, bezpośrednia] → zewnętrzna) i wynikająca z tego potrzeba implementowania pamięci wirtualnej.

Istnieją dwie szerokie klasy metod realizujących dostęp do danych na podstawie ich klucza, a mianowicie: przeszukiwanie w drzewie i algorytmy przemieszania danych. Naturalnie oba te sposoby mają swoje wady i zalety. Wybranie jednego z nich jest zadaniem trudnym i w dużym stopniu zależy od specyfiki danej aplikacji lub bazy wiedzy. Sytuacja jest tym bardziej skomplikowana, że powyższe algorytmy opierają się na odmiennych strukturach danych (drzewa na listach), zaś przemieszanie

danych korzysta z tablic. Struktury danych oparte na drzewach są bardziej elastyczne, lepiej wykorzystują pamięć (mniejsza zajętość pamięci operacyjnej), dają wiele dodatkowych możliwości, jak np. uzyskanie alfabetycznego (wg. klucza) spisu występujących w nim obiektów. Z drugiej strony same algorytmy manipulacji na danych są skomplikowane, a co za tym idzie, programy implementujące drzewa są trudniejsze w testowaniu i bardziej pracochłonne.

Tablice natomiast cechuje bardzo przejrzysta struktura algorytmu realizującego podstawowe operacje na obiektach, a także bardzo efektywny algorytm przeszukiwania.

Dlatego, jeżeli spodziewamy się, że w większości przypadków podstawową operacją będzie wyszukiwanie danych, to celowym wydaje się implementowanie struktur danych opartych na tablicach. Natomiast, jeżeli rozkład operacji podstawowych jest bardziej równomierny (żadna z nich nie dominuje w sposób jawny) lub gdy przeważać będą operacje związane ze wstawianiem i/lub usuwaniem obiektów, to z punktu widzenia efektywności działania SE warto implementować struktury listowe. Bardzo często stosuje się też przedstawienie danych w postaci struktur mieszanych.

Implementacja AMD w systemach ekspertowych typu shell

Metody związane z tablicami rozproszonymi opierają się na założeniach odmiennych od zadań przeszukiwania w drzewach. Zamiast typowej dla drzew operacji porównywania klucza wejściowego K z innymi kluczami w drzewie, próbuje się przeprowadzić na kluczu pewną transformację i uzyskać funkcję $h(K)$ wskazującą bezpośrednio na pozycję w tablicy, gdzie jest trzymany klucz K oraz związana z nim informacja. Teoretycznie jest możliwe zbudowanie takiej funkcji, która za pomocą operacji arytmetyczno-logicznych przekształci w sposób jednoznaczny dowolny klucz K w indeks do tablicy o określonym rozmiarze M . Można to zapisać w następujący sposób:

$$0 < = h(K) < = M$$

dla wszystkich kluczy K oraz, że z warunku

$$K_1 \# K_m$$

wynika

$$h(K_1) \# h(K_m)$$

W praktyce uzyskanie takiej transformacji jest bardzo trudne i wymaga dużych mocy obliczeniowych nawet dla tablic średnich rozmiarów. Metoda ta ma jeszcze jedną dodatkową wadę, że nadaje się wyłącznie do tablic statycznych o ustalonym z góry rozmiarze. Powiększenie lub zmniejszenie rozmiaru tablicy pociąga za sobą konieczność nowego skonstruowania funkcji $h(K)$ z uwzględnieniem aktualnego rozmiaru. W praktyce korzysta się z bardziej elastycznego podejścia, ponieważ rezygnuje się z wymogu jednoznaczności $h(K)$ i dopuszcza istnienie synonimów. Przez synonim rozumiemy zbiór różniących się od siebie kluczy, które mają tę samą wartość funkcji $h(K_i)$. Zjawisko takie nazywa się kolizją. W realnych aplikacjach liczba podobnych do siebie kluczy (a więc potencjalnych synonimów) może być duża (np. zmienne $ZM1, ZM2, \dots$). Dlatego od tej funkcji wymaga się jedynie, aby miała dobre własności „rozrzucające” równomiernie takie klucze po mieszanej tablicy. Jest to bardzo ważne dla zmniejszenia kolizji kluczy w tablicy.

Z powyższego wynika jasno, że w dobrym algorytmie mieszania danych występują dwa kluczowe parametry. Są to:

- wybór skutecznej, efektywnej i szybkiej funkcji transformacji kluczy, oraz
- właściwa strategia rozwiązywania konfliktów kluczy.

Wybór „dobrej” funkcji mieszającej jest zadaniem trudnym i w dużym stopniu zależy od własności kluczy, z którymi będzie pracowała tablica mieszająca. Ponieważ trudno tu sformułować uniwersalne zalecenia, temat dobrania transformanty klucza należy pozostawić twórcom konkretnego systemu typu shell.

Sposoby wstawiania synonimów do tablic mieszających

Generalnie istnieją dwa sposoby rozstrzygnięcia konfliktów. Są to metody: łańcuchów synonimów (ang. *chaining*) oraz otwartej adresacji (ang. *open addressing*).

Idea algorytmu łańcuchowania polega na dodaniu do każdego obiektu przechowywanego w tablicy jednego pola wskaźnikowego o nazwie *Link*. Pole *Link* służy do organizowania prostych list połączonych, „zakotwiczonych” w każdym elemencie tablicy rozproszonej. W listach tych są przechowywane klucze będące synonimami klucza podstawowego, dla których nastąpił konflikt w trakcie wstawiania do tablicy. Istnieją dwa sposoby przydzielenia pamięci dla tych list. Stąd wywodzą się dwa warianty algorytmów łańcuchowania. Jeżeli listy są trzymane w odrębnym obszarze pamięci, to algorytm nosi nazwę łańcuchowania zewnętrznego, zaś gdy listy są przechowywane w samej tablicy rozproszonej to mamy do czynienia z łańcuchowaniem wewnętrznym.

Przeszukiwanie w tablicy rozproszonej korzystającej z łańcuchowania zewnętrznego jest niezwykle proste. Po obliczeniu dla klucza wejściowego K indeksu i w tablicy *TABLE* (tj. $i = h(K)$) następuje przeglądanie listy synonimów „podczepionych” do i -tego elementu w *TABLE*. W przypadku szczególnie dużych tablic listy synonimów mogą być nawet przechowywane w pamięci zewnętrznej. Istnieją różne modyfikacje tego algorytmu, np. w celu zoptymalizowania dostępu do elementów na liście używa się zamiast list prostych (z jednym wskaźnikiem) struktur drzewiastych. Mimo wielu zalet metoda ta ma jedną istotną wadę, a mianowicie zbyt rozrzućnie gospodaruje pamięcią.

Dlatego w praktyce często wybiera się drugi wariant łańcuchowania. Łańcuchowanie wewnętrzne lepiej wykorzystuje pamięć operacyjną, ponieważ przechowuje powstałe w wyniku wstawiania nowych kluczy synonimy w tablicy rozproszonej. Klucze będące synonimami trafiają w aktualnie wolne w tym czasie pozycje tablicy rozproszonej. Wybierając różne sposoby znalezienia wolnej pozycji w tej tablicy uzyskujemy różne odmiany algorytmu łańcuchowania. W najprostszym przypadku stosuje się przeszukiwanie liniowe kolejnych elementów tablicy. Metoda wewnętrzna ma jedną istotną wadę, otóż przy wstawianiu nowych elementów do tablicy pojawia się tendencja „zrastania” łańcuchów synonimów. Oznacza to, że do listy synonimów dołączone zostają klucze, które nie są synonimami. Weszły one na listę synonimów ponieważ ich „właściwe” (im odpowiadające) miejsce w tablicy zajął synonim należący do innego klucza. Podsumowując można powiedzieć, że istota łańcuchowania wewnętrznego polega na rozstrzygnięciu kolizji kosztem zajęcia „obcych”, a jeszcze wolnych pozycji w tablicy przez synonimy innego elementu tablicy. Metoda otwartej adresacji, podobnie jak w poprzednim algorytmie, szuka miejsca wolnego dla powstałego synonimu wewnątrz tablicy rozproszonej. Wadą tej metody jest brak efektywnego sposobu powiększenia tablicy mieszającej w trakcie działania algorytmu oraz to, że tablica musi mieć z góry założony rozmiar. Jej zaletą jest to, że nie potrzebuje dodatkowych łączników do tworzenia list

synonimów. Oprócz tego cechuje ją dość prosty algorytm oraz duża efektywność. Szkielet tego algorytmu jest następujący:

- D1. [inicjowanie]. $i = 1$
- D2. [oblicz funkcję mieszającą]. $a = h_i(K)$
- D3. [sprawdzenie]. Jeżeli pozycja a jest „pusta”, to algorytm się kończy. Jeżeli pracujemy w trybie wstawiania możemy do tej pozycji wpisać nowy klucz. W przypadku wyszukiwania informacji z kluczem K takie zakończenie algorytmu oznacza, że nie ma go w tablicy.
- D4. [dalej]. Jeżeli pozycja a jest „zajęta” i zawiera klucz K , to algorytm kończy działanie pomyślnie (klucz został znaleziony).
- D5. [szukaj miejsca]. Jeżeli pozycja a jest „zajęta” i zawiera inny klucz K_a , to nastąpiła kolizja i klucz K jest kolejnym synonimem K_a . Wtedy $i = i + 1$ i przejście do D2.

W tym miejscu należy zwrócić uwagę, że w trakcie przeglądania kolejnych pozycji w tablicy posługujemy się ciągiem funkcji mieszających $\{h_i\}$ $i = 0$. Ich wybór stanowi również trudną procedurę. Zakładając, że mamy już skonstruowaną funkcję h_0 można zaproponować następujące postaci kolejnych elementów tego ciągu:

- A. Przeszukiwanie liniowe – najprostszy, choć nie zawsze skuteczny sposób. Wtedy $h_i(K) = h_0(K) + c*i$, gdzie c oznacza „właściwie” wybraną stałą;
- B. Przeszukiwanie z funkcją kwadratową o postaci: $h_i(K) = h_0(K) + c*i + d*i^2$, gdzie c i d oznaczają odpowiednie stałe i ;
- C. Podwójne mieszanie, metoda stosowana najczęściej. Funkcja przy i -tej kolizji ma postać: $h_i(K) = h_0(K) + i*g(K)$, gdzie $g(K)$ jest funkcją mieszającą, niezależną od $h_0(K)$.

Otwarta adresacja z podwójnym mieszaniem

Algorytm ten pozwala odszukać klucz wejściowy K w tablicy M -elementowej. Elementy tablicy oznacza się jako *TABLE* [i], gdzie i znajduje się w przedziale $\langle 0, M \rangle$. Każdy element może być dwojakiego typu: „wolny” lub „zajęty”. Element „zajęty” ma pole *KEY* [i] zawierające klucz K_i oraz inne pola przechowujące dane o obiekcie związanym z tym kluczem. W przypadku gdy klucza K nie ma w tablicy, a tablica nie jest do końca zapełniona, następuje wstawienie do tej tablicy klucza K . W tym celu oblicza się wartości dwóch funkcji „mieszających” $h_1(K)$ i $h_2(K)$. Od funkcji $h_1(K)$ wymaga się, aby generowała wartości z przedziału od 1 do $M-1$, zaś funkcja $h_2(K)$ powinna zwracać znaczenia od 1 do $M-1$, które nie powinny być dzielnikiem liczby M , np. w przypadku gdy $M = 2^m$, to $h_2(K)$ powinna przybierać wartości liczb nieparzystych z przedziału 1 do $2^m - 1$. Algorytm składa się z następujących kroków:

- D1. [pierwsze „przemieszanie”]. Należy obliczyć $i = h_2(K)$;
- D2. [pierwsza próba]. Sprawdzić, czy element *TABLE* [i] jest „wolny”. Jeżeli tak, to skok do D6. Inaczej, jeżeli *KEY* [i] = K , to koniec algorytmu;
- D3. [powtórne przemieszanie]. Obliczyć $c = h_2(K)$;
- D4. [przejście do następnego elementu]. $i = i - c$, jeżeli $i < 0$ to $i = i + M$;
- D5. [porównanie]. Gdy element *TABLE* [i] jest „pusty”, to przejść do D6. Jeżeli *KEY* [i] = K , to koniec algorytmu, w przeciwnym przypadku skok do D4;
- D6. [wstawienie]. Jeżeli $N = M - 1$, to przepełnienie tablicy i wyjście z algorytmu z komunikatem o przepełnieniu. W przeciwnym przypadku powiększyć $N = N + 1$, wstawić do *KEY* [i] = K oraz oznaczyć ten element jako „zajęty”.

Konkretna implementacja algorytmu w języku C

Pakiet # HASH # jest uniwersalnym programem umożliwiającym zorganizowanie danych w tablicach mieszanych i składa

się z trzech podprogramów służących do inicjowania (*HCREATE*), wstawiania i wyszukiwania (*HSEARCH*) oraz usuwania (*HDESTROY*) tablicy o nazwie wybranej przez użytkownika. Składnia tych funkcji jest następująca:

```
ENTRY *hsearch (item, action)
ENTRY item;
ACTION action;
int hcreate (no_of_el)
unsigned no_of_el;
void hdestroy ()
```

A oto opis poszczególnych funkcji:

HSEARCH jest podprogramem implementującym uogólniony algorytm D przeszukiwania tablicy rozproszonej (ang. *hashtable search*) opisanym w trzecim tomie „The Art of Programming” D. Knutha [2]. Funkcja ta zwraca wskaźnik elementu w tablicy rozproszonej, zawierającego szukany obiekt zwany *item*. Każdy obiekt *item* jest strukturą zdefiniowaną jako typ *ENTRY* oraz zadeklarowaną w pliku nagłówkowym. Struktura ta ma następującą budowę:

item.key – wskaźnik do unikalnego klucza określonego obiektu,
item.data – wskaźnik do łańcucha danych, opisującego zawartość obiektu lub do dowolnej struktury danych związanej z danym obiektem.

W przypadku, gdy określone obiekty opisuje struktura mająca inną budowę od domniemanej (łańcuch znakowy), należy dokonać nałożenia (redefinicji) tej struktury na wskaźnik do *item.data*. Zmienna *action* należy do wyliczalnego typu danych *ACTION* i podaje działanie, jakie należy podjąć w przypadku nieznaledzenia podanego obiektu w tablicy rozproszonej. Zmienna ta może przyjąć następujące znaczenia:

ENTER wymusza wstawienie podanego obiektu do tablicy pod odpowiadający funkcji „mieszającej” element.
FIND szuka w tablicy rozproszonej elementu zawierającego podany w wywołaniu tej funkcji obiekt.

W przypadku zwrócenia przez tę funkcję wskaźnika *NULL*, należy przyjąć, że funkcja *HSEARCH* zakończyła niepomyślnie swoje działanie (tj. nie udało się wstawić nowego elementu do tablicy z uwagi na brak wolnego miejsca lub w tablicy rozproszonej nie ma poszukiwanego obiektu).

Podstawowy algorytm funkcji *HSEARCH* korzysta z metody otwartej adresacji, połączonej z multiplikatywną funkcją „mieszającą” (ang. *open addressing with multiplicative hash function*). Jednakże za pomocą odpowiednich wartości wybranych zmiennych preprocesora można go w czasie kompilacji elastycznie zaadaptować do innych potrzeb użytkownika. Oto zmienne, wpływające na budowę kodu źródłowego tej funkcji:

- *DIV* – zamiana algorytmu funkcji mieszającej z metody mnożenia na metodę reszty z dzielenia według modulo „rozmiar tablicy” (ang. *mod nel*);

- *USRC* – umożliwienie „podpięcia” przez użytkownika własnego podprogramu porównującego klucze między dwoma obiektami (tzw. *USER SUPPLIED COMPARISON ROUTINE*). Podprogram ten musi zachowywać się analogicznie jak funkcja *strcmp* języka C;

- *CHAINED* – podmienia standardowe działanie podprogramu w przypadku kolizji na rozstrzygnięcie kolizji metodą list połączonych (ang. *linked list*). Przy skorzystaniu z tej opcji użytkownik ma następujące możliwości;

- *START* – umieszczenie nowego obiektu wstawianego do tablicy na początku listy połączonej (domyślnie wstawianie odbywa się na koniec listy);

- *SORTUP* – utrzymywanie posortowanej listy synonimów w kolejności rosnącej (ang. *ascending*);

- *SORTDOWN* – utrzymywanie posortowanej listy synonimów w kolejności malejącej (ang. *descending*);

- *HCREATE* – buduje właściwą tablicę rozproszoną przez dynamiczne zarezerwowanie dla niej dostatecznej ilości pamięci operacyjnej. Funkcja ta musi być wywołana jako pierwsza (przed korzystaniem z innych funkcji tego pakietu), gdyż inicjuje właściwy dostęp do tablicy rozproszonej. Zmienna *nel* określa przybliżony rozmiar tablicy rozproszonej. Algorytm funkcji *HCREATE* koryguje tę górną granicę w celu dopasowania rozmiaru tablicy do potrzeb związanych z własnościami algorytmu D (wielokrotność 2);

- *HDESTROY* – zwalnia pamięć operacyjną zajmowaną przez tablicę rozproszoną, czyniąc tym samym zawarte w niej dane niedostępnymi. Po jej uruchomieniu można utworzyć tablicę rozproszoną za pomocą kolejnego wywołania funkcji *HCREATE*.

```
#include <stdio.h>
typedef struct entry { char *key, *data; } ENTRY;
typedef enum { FIND, ENTER } ACTION;
#ifdef NO_PROTOTYPE
extern ENTRY *hsearch( ENTRY, ACTION );
extern int hcreate( int );
#endif
struct info {
    int age, room;
};
#define NUM_EMPL 5000
main()
{
    char string_space[NUM_EMPL*20];
    struct info info_space[NUM_EMPL];
    char *str_ptr = string_space;
    struct info *info_ptr = info_space;
    ENTRY item, *found_item, *hsearch();
    char name_to_find[30];
    int i = 0;
    (void) hcreate(NUM_EMPL);
    printf("Podaj Atrybuty pracownika (Nazwisko Wiek Pokoj) :");
    printf("-----");
    while (scanf("%s%d%d", str_ptr, &info_ptr -> age,
                &info_ptr -> room) != EOF && i++ < NUM_EMPL)
    {
        item.key = str_ptr;
        item.data = (char *)info_ptr;
        str_ptr += strlen(str_ptr) + 1;
        info_ptr++;
        (void) hsearch(item, ENTER);
    }
    printf("");
    printf("Koniec wprowadzania. Dziekuje!");
    printf("-----");
    printf("Wyszukiwanie:");
    printf("Podaj Nazwisko pracownika:");
    item.key = name_to_find;
    while (scanf("%s", item.key) != EOF)
    {
        if ((found_item = hsearch(item, FIND)) != NULL)
        {
            (void)printf("Znalazłem %s, age + %d, room = %d",
                found_item->key,
                ((struct info *)found_item->data)->age,
                ((struct info *)found_item->data)->room);
        }
        else
        {
            (void)printf("Nie znalazłem %s", name_to_find);
        }
    }
}
```

Powyższy przykład ilustruje wykorzystanie tablicy mieszanej do przechowywania i udostępniania danych o pracowniku. W pierwszej części, po założeniu tablicy następuje wpisanie z klawiatury danych o pracownikach, natomiast część druga, po podaniu klucza (nazwisko), wyszukuje w tablicy dane o tym pracowniku.

Ocena efektywności

Był to podstawowy algorytm D, dla którego trudno określić średnią liczbę prób potrzebnych do znalezienia danego klucza w tablicy. Badania teoretyczne pozwalają jedynie znaleźć przybliżoną liczbę prób korzystając ze wzoru:

$$D_{fail} = (l - a) - l \quad \text{lub} \quad D_{success} = -a^l \ln(l - a)$$

gdzie:

a – współczynnik zapelnienia tablicy rozproszonej,

D_{fail} – liczba prób, gdy przeszukiwanie tablicy było nieudane (najgorszy przypadek), tj. poszukiwanego klucza nie ma w tablicy,

$D_{success}$ – liczba prób, gdy przeszukiwanie tablicy zakończyło się pomyślnie, tj. w tablicy znaleziono dany klucz, co umożliwia pobranie związanych z tym kluczem danych (najlepszy przypadek).

Powyższy wzór dosyć dobrze odpowiada wynikom doświadczalnym i jest słuszny przy założeniu, że funkcje $h_1(K)$ i $h_2(K)$ są od siebie niezależne. Nieufność do tego algorytmu może wzbudzać fakt dużej różnicy liczby prób w obu przypadkach (D_{fail} i $D_{success}$). W przypadku implementowania tablic rozproszonych jako bazowych struktur interpretera regułą można nie obawiać się o skuteczność i efektywność tego algorytmu (szczególnie najgorszego przypadku). Dzieje się to dlatego, że zasadniczym trybem pracy interpretera jest uzgadnianie reguł, a co za tym idzie, szybkie wyszukiwanie wartości zmiennych roboczych według ich nazwy. Nazwa zmiennej roboczej pełni tutaj rolę klucza, na podstawie którego szukamy jej wartości w tablicy mieszanej. Dlatego też można śmiało powiedzieć, że w podstawowym trybie pracy interpretera liczba prób podczas szukania wartości zmiennych (kluczy) opisze się dobrze wzorem dla $D_{success}$.

Modyfikacje bazowego algorytmu

Modyfikacja algorytmu R. Brent'a polega na ustabilizowaniu na jednym poziomie, niezależnie od współczynnika zapelnienia tablicy a , liczby prób w przypadku przeszukiwania danego. Sposób do osiągnięcia tego celu polega na wykorzystaniu faktu, że dla niektórych aplikacji częstotliwość operacji wyszukiwania jest dużo większa w porównaniu do operacji wstawiania (np. interpreter reguł w SE). Stąd wniosek, że chcąc przyspieszyć pracę interpretera konieczne jest poświęcenie więcej czasu na „porządne” wstawienie nowego elementu z kluczem K (tj. nowej zmiennej), nawet za cenę potrzeby przesuwania niektórych istniejących kluczy w tablicy, po to, by zmniejszyć ogólny czas dostępu do dowolnego istniejącego w tablicy klucza. Powoduje to następujące konsekwencje w postaci zmian w algorytmie D.

Założmy, że podczas nieudanego przeszukiwania tablicy mieszanej w trybie wstawiania nowego elementu K zostały sprawdzone elementy:

$$p_0, p_1, \dots, p_{t-1}, p_t$$

gdzie

$p_j = (h_1(K) - j \cdot h_2(K)) \bmod M$ i okazało się, że element $TAB-LE[p_t]$ jest „pusty”. Jeżeli $t < l$, to postępujemy jak w bazowym algorytmie D, czyli wstawiamy klucz na pozycję p_t tj. $KEY[p_t] = K$.

W przeciwnym razie (gdy $t > l$) obliczamy nową zmienną $c_0 = h_2(K_0)$, gdzie $K_0 = KEY[p_0]$ i sprawdzamy czy jest wolne miejsce pod elementem $TAB-LE[(p_0 - c_0) \bmod M]$. Jeżeli tak jest, to przesuwamy do tego elementu wartość $TAB-LE[p_0]$, a w zwolnionym w ten sposób miejscu umieszczamy nowy klucz K ($KEY[p_0] = K$). W przeciwnym przypadku, tj. gdy element $TAB-LE[(p_0 - c_0) \bmod M]$ jest zajęty i $t > 3$, to sprawdzamy element $TAB-LE[(p_0 - 2 \cdot c_0) \bmod M]$ itd. Uogólniając można stwierdzić, że mając $c_j = h(KEY[p_j])$ i $p_j k = (p_j - k \cdot c_j) \bmod M$ oraz jeżeli pozycja $TAB-LE[p_j, k]$ jest zajęta dla wszystkich j, k takich, że $j + k < r$ i jeżeli $t > r + 1$, to sprawdzamy po kolei elementy $TAB-LE[p_0, r], TAB-LE[p_1, r - 1], \dots, TAB-LE[p_{r-1}, 1]$. Jeżeli znajdziemy „pusty” element na pozycji $p_j, r - j$, to umieszczamy w nim klucz z pozycji p_j (tj. $TAB-LE[p_j, r - j] < -TAB-LE[p_j]$), a na pozycje p_j wpisujemy nowy klucz K . Jak wykazały badania teoretyczne, poparte danymi empirycznymi, modyfikacja tego algorytmu ulepszyła go do tego stopnia, że $D_{success}$ obniżył się do 2,49 (niecałe trzy próby).

Porównanie skuteczności różnych algorytmów mieszających

Bardzo trudno powiedzieć, który z algorytmów jest najkorzystniejszy dla danej aplikacji, a jeszcze trudniej opisać to, co należy uwzględnić przy wyborze metody poszukiwania. Jednak biorąc pod uwagę kryterium minimalnej liczby prób oraz minimalnego zapotrzebowania na pamięć operacyjną, można zaproponować kilka ogólnych wskazówek (opierających się na przesłankach teoretycznych popartych wynikami praktycznych implementacji tych algorytmów (patrz tabela oraz rys. 2)).

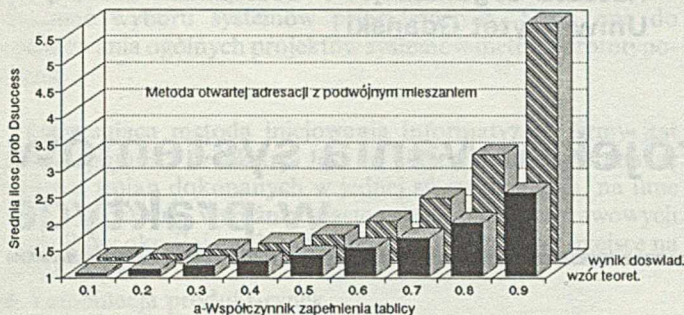
W algorytmie przeszukiwania liniowego częściej niż w innych metodach, następuje odwołanie się do tablicy (większa liczba badanych elementów – dłuższy czas przeszukiwania). Z drugiej strony, metodę tę cechuje prostota co oznacza małą zajętość zasobów procesora, a co za tym idzie, dużą szybkość pracy. Jednakże zastosowanie, wydawać by się mogło, tak mało efektywnego algorytmu, pozwala znacznie zmniejszyć liczbę prób przy wyszukiwaniu w tablicy, np. przy zapelnieniu tablicy w 90% średnia liczba prób w przypadku losowego wyszukiwania istniejącego w tablicy klucza wynosi ok. 5,5. Fatalnie wręcz wygląda sytuacja w przypadku wstawiania nowego elementu, bo w tym przypadku może dojść do 50,5 próby przed znalezieniem dla tego elementu miejsca w tablicy. Algorytm łańcuchowania synonimów jest ekonomiczny z punktu widzenia

Porównanie metod rozstrzygania kolizji

		D _{fail}	D _{success}
Metoda rozstrzygania kolizji		Przeszukiwanie nieudane	Przeszukiwanie udane
otwarte	liniowe przeszukiwanie	$1/2^* (1 + 1/(1-a))$	$1/2^* (1 + 1/(1-a)^2)$
adresowanie	podwójne mieszanie	$1/(1-a)$	$-1/a^* \ln(1-a)$
metoda	zewnetrzne	$a + e^{-a}$	$1 + a/2$
łańcuchowania	wewnetrzne	$1 + 1/4^* (e^{-2^* a} - 1 - 2^* a)$	$1 + a/4 + a/8^* (e^{-2^* a} - 1 - 2^* a)$

a – współczynnik zapelnienia tablicy

Porównanie średniej ilości prób przy udanym przeszukiwaniu



I) Brak wskaźników w tablicy, w momencie kolizji przeszukiwane są sekwencyjnie kolejne elementy tablicy dopóki nie trafi się "pusta" pozycja. Ważny jest algorytm służący do wyznaczenia kolejnych elementów tablicy, które należy sprawdzić, najoptymalniej gdy sekwencja kolejnych numerów zależy od klucza K.

II) Organizuje się M list jednokierunkowych (po jednej dla każdego elementu tablicy). Elementy tablicy mają oprócz klucza K pole Link wskazujące na początek tych list. Przy wyszukiwaniu klucza K sprawdza się kolejne elementy listy rozpoczynając od pozycji TABLE[I] gdzie $i = h(K)$.

Rys. 2

liczby przeszukiwanych elementów w tablicy, mniej jednak korzystnie wygląda z punktu widzenia zużycia pamięci. Jego wadą jest to, że wymaga dodatkowej pamięci na wskaźniki adresowe. Dlatego nie mający tych wad algorytm otwartej adresacji wydaje się być najodpowiedniejszym wyborem. Wśród nich wyróżnia się jeden algorytm nie zużywający nadmiernie pamięci i dający niezłe rezultaty, jeżeli chodzi o liczbę prób przed znalezieniem wybranego klucza w tablicy. Jest to modyfikacja słynnego algorytmu D, zaproponowana przez R. Brent'a. Pozwala ona na umieszczenie obiektów w tablicy o rozmiarze $M = N + 1$ oraz znalezienie dowolnego obiektu po średnio 2,5 próbach (przy założeniu, że obiekt znajduje się w tej tablicy), a także nie wymaga dodatkowej pamięci na łączniki adresowe. Jednakże ma jedną wadę, ponieważ w skrajnym przypadku (*worst case*) potrzebuje aż $N/2$ prób zanim wykryje, że dany klucz nie znajduje się w tablicy.

* * *

Porównując algorytmy mieszania danych z metodami przeszukiwania w drzewach, można mówić o pewnych zaletach, jakie daje wykorzystanie tablic mieszanych w zadaniach dostępu do zmiennych oraz reguł w SE. Jeżeli chodzi o szybkość pracy tych pierwszych algorytmów, ich przewaga jest widoczna dopiero przy tablicach większych ($M > 1000$). Dzieje się to dlatego, że średnia liczba przeszukiwanych obiektów w tablicach mieszanych jest nawet przy $M \rightarrow 8$ niezmienna i ograniczona z góry. Jak widać z tabeli, jedyny wpływ na liczbę prób ma współczynnik a (zapełnienia tablicy). Wadą tablic rozproszonych jest to, że nie należy bezkrytycznie wierzyć wzorom wyprowadzonym na podstawie teorii prawdopodobieństwa, ponieważ są one prawdziwe w sensie statystycznym. Liczba prób w przypadku najgorszym jest wręcz katastroficznie duża. Z tego punktu widzenia algorytmy przeszukiwania oparte np. drzewie zrównoważonym gwarantują wręcz górną granicę czasu przeszukiwania i dlatego są bezpieczne w przypadku aplikacji pracujących w czasie rzeczywistym. Z drugiej strony, oprogramowanie przeszukiwania w drzewach jest bardziej skomplikowane i czas przeszukiwania w nim danych, mimo że gwarantowany, jednak jest większy niż w przypadku tablic rozproszonych.

Pakiet #HASH#, zaprogramowany w języku C, został przygotowany w celu doświadczalnego wypróbowania i oceny przydatności technik tablic rozproszonych do obsługi podstawowych mechanizmów systemu eksperckiego typu *shell*.

Obecnie trwają prace związane z jego implementacją w medycznym systemie eksperckim, pisany w języku Turbo Pascal 6.0. Wybór tego języka jest podyktowany chęcią wykorzystania metod programowania obiektowego jako podstawowego narzędzia do tworzenia systemów eksperckich. Programowanie obiektowe pozwala bardziej elastycznie podejść do zagadnienia, zapewnia bezpieczniejszy kod wynikowy, umożliwia lepsze i wygodniejsze wykorzystanie gotowych modułów w formie bibliotek klas oraz lepiej od innych metod programowania izoluje programistę od szczegółów implementacyjnych określonych klas obiektów. Dzięki takiemu podejściu programista pracuje na wyższym poziomie abstrakcji, mając do czynienia jedynie z obiektami, klasami i metodami. Dzięki temu jego produkt jest niezawodny, a czas tworzenia systemu – maksymalnie krótki.

LITERATURA

- [1] Gries D.: Compiler construction for digital computers. John Wiley & Sons Inc. 1971
- [2] Knuth D.E.: The Art of Computer Programming, vol. 3. Addison-Wesley Publishing Company, Inc. Reading, Mass., 1974
- [3] Popow E.W.: Ekspertyjne systemy. Nauka, 1987
- [4] Sibuya M., Yamamoto T.: Wwiedzenie w systemy programowania. Mir, 1986.

Strojenie mechanizmu rozumowania systemów ekspertowych...

dokończenie ze s. 19

- 3) metoda średniej (ang. *average*):

$$(\max(x, y) + (x, y) - (x^*y/100))/2,$$
- 4) metoda Bonczek'a-Eangin'a:

$$\max(x, y) + (x^*y/100) * (1 - \max(x, y)/100).$$

Najbardziej zachowawczą i ostrożną jest metoda maksimum. Stanowi ona, zgodnie ze zdrowym rozsądkiem, naturalną dolną granicę wyników operacji typu *confirmative*. Pozostałe trzy operacje uwzględniają, w odróżnieniu od maksimum, wpływ również drugiego współczynnika na wynik operacji. Najbardziej optymistyczną, ryzykowną jest metoda sumy, stanowiąca górną granicę możliwych operacji typu *confirmative*. Pomiedzy tymi granicznymi operacjami GURU proponuje, podobnie jak dla *joint*, dwie operacje kompromisowe.

* * *

W systemie GURU istnieje ogółem ponad 100 zmiennych środowiskowych. Istotna część z nich w sposób bezpośredni lub pośredni wpływa na działanie mechanizmu rozumowania i przebieg procesów konsultacji. Omówione na przykładzie pakietu GURU mechanizmy sterowania „maszyną wnioskującą” wraz z warstwą rozumowania w warunkach niepewności, ilustrują bogactwo możliwości strojenia „myślenia maszynowego” na wzór procesów myślowych człowieka.

LITERATURA

- [1] GURU Reference Manual, vol. 1. Lafayette, Ind. MDBS Inc. 1985
- [2] Holsapple C.W., Whinston A.B.: Manager's Guide To Expert Systems Using GURU. Dow Jones-Irwin Homewood, Illinois, 1986
- [3] Zadeh L.A.: Fuzzy sets. Information and control, 8, 1965.

Przypominamy o zaprenumerowaniu
INFORMATYKI
 na przyszły rok

Metody analizy i projektowania systemów w praktyce

Metody i techniki analizy i projektowania systemów, które w pierwszej połowie lat osiemdziesiątych były przedmiotem rozważań na konferencjach naukowych znajdują obecnie coraz szersze zastosowanie praktyczne. Można mówić nawet o przyjęciu pewnych standardów z tej dziedziny w krajach o wysokim poziomie rozwoju technologii informatycznej.

Celem niniejszego artykułu jest wprowadzenie do jednodniowych prowadzonych przez Katedrę Organizacji Przetwarzania Danych Uniwersytetu Gdańskiego warsztatów (ang. *workshops*) ukierunkowanych na praktyczne zastosowanie metod i technik wstępnych faz cyklu życia systemu informatycznego, tj. planowania, analizy i projektowania. Warsztaty te opierają się na rozwiązaniu funkcjonowania przykładowego przedsiębiorstwa.

Stosownie do celów warsztatów na początku artykułu przedstawiono przegląd metod i technik tworzenia systemów informatycznych. Prezentację tę powiązano z fazami cyklu życia systemu. Szczególną uwagę zwrócono na stosunkowo mało znane w Polsce metody planowania systemu. Dalsza część artykułu zawiera szczegółowy harmonogram warsztatów.

Metody i techniki tworzenia systemów w cyklu życia systemu

Dobór i użytkowanie metod, technik i narzędzi tworzenia systemów informatycznych należy ściśle powiązać z cyklem życia systemu. Istnieją kontrowersje co do zakresu, kolejności faz oraz ich nazewnictwa w odniesieniu do procesu tworzenia systemu informatycznego (TSI). Dotychczasowe próby standaryzacyjne nie uzyskały dotąd powszechnej akceptacji. Ogólnie można przyjąć, iż cykl życia systemu obejmuje następujące fazy:

- planowanie,
- analizę,
- projektowanie,
- wdrożenie,
- użytkowanie i kontrolę.

Każdej z tych faz odpowiadają specyficzne metody i techniki, choć w niektórych przypadkach służą one kilku fazom. W szczególności dotyczy to faz analizy i projektowania.

O ile metody analizy i projektowania torują sobie w kraju powoli zrozumienie, o tyle metody planowania systemu informacyjnego są zupełnie zaniechane i niedoceniane. Tymczasem związana z nimi dziedzina określana w terminologii angiel-

skiej *Information Planning* rozwija się niezwykle dynamicznie. W konkretnych realizowanych projektach chodzi o osiągnięcie następujących dwu wzajemnie ze sobą powiązanych celów:

- stworzenie systemów informatycznych skutecznie wspomagających strategiczne cele firmy,
- osiągnięcie rzeczywistego zaangażowania (pośrednio i odpowiedzialności) kierownictwa firmy w proces TSI.

Istnieje wiele aktywizujących, heurystycznych metod identyfikowania przez kierownictwo tych obszarów działalności, których informatyzacja sprzyjać będzie rozwojowi firmy, skutecznemu wdrażaniu jej strategii rynkowej oraz zwiększaniu jej konkurencyjności. Wśród metod tych należy wymienić w pierwszym rzędzie:

- sesję Metaplan,
- Analizę istotnych czynników powodzenia,
- model spójności Broekstry.

Szczegółowa charakterystyka każdej z tych metod jest zagadnieniem samym w sobie. Warto jednak podać nawet krótkie omówienie każdej z tych metod, choćby z tego powodu, że jeszcze są one w Polsce mało znane.

Bardzo efektywną metodą planowania systemu informacyjnego jest sesja Metaplanu. Jej założenia są stosunkowo proste, lecz jak wskazuje doświadczenie, pozwala osiągnąć bardzo użyteczne rezultaty oraz wysoki stopień zaangażowania kierownictwa. W następujących po sobie „taktach”, metodą dyskusji, ustalania priorytetów i wreszcie prezentacji określa się kolejno:

- cele firmy,
 - zagrożenia,
 - działania dla osiągnięcia celów,
 - działania dla uniknięcia zagrożeń,
 - specyfikację systemów informacyjnych wspomagających osiągnięcie celów oraz zapewniających uniknięcie zagrożeń.
- Pomijamy tu szczegółowe, aktywizujące, często atrakcyjne techniki realizacji sesji Metaplan.

Kolejną metodą planowania systemów informacyjnych jest metoda nazywana: Analizą istotnych czynników powodzenia (*Critical Success factors*). Oznacza ona wyodrębnienie pięciu do ośmiu obszarów działalności gospodarczej firmy, w których pozytywny rezultat gwarantuje jej pomyślne funkcjonowanie na rynku. Metody tej używa się do wydzielenia przez kierownictwo firmy głównych obszarów zainteresowania, zapewniających ciągłość i rozwój jej funkcjonowania. Identyfikacja „Istotnych czynników powodzenia” następuje przez:

- przeprowadzenie warsztatów pozwalających określić cele i priorytety firmy,
- opracowanie i przeprowadzenie ankiety wśród kierownictwa firmy na podstawie rezultatów warsztatów,
- po zapoznaniu się kierownictwa firmy z wynikami ankiety, przeprowadzenie kolejnego warsztatu celem ustalenia ostatecznej listy „Istotnych czynników powodzenia”.

Metoda ta ma bezpośredni związek z dalszymi fazami cyklu życia systemu. W wyniku analizy scenariuszy decyzji można dokonać wyboru systemów priorytetowych. Służą one do opracowania ogólnych projektów systemów metodą prototypowania.

Interesującą metodą inicjowania informatyzacji firmy jest model spójności Broekstry. Jest on ukierunkowany na analizę wpływu zmian dokonanych w jednej sferze działania, na inne sfery. Broekstra wyróżnia następujące pięć podstawowych czynników określających funkcjonowanie firmy i jej miejsce na rynku. Są to:

- kombinacja produkt-rynek,
- technologia,
- organizacja,
- kadry,
- dominująca koalicja, czyli formalne i nieformalne grupy mające wpływ na strategię działania firmy.

Powyższe czynniki są ze sobą ściśle powiązane i zmiana któregokolwiek z nich powoduje zmiany w pozostałych. Zmiany stają się w ten sposób przewidywalne, co pozwala na rozważenie m.in. negatywnych skutków zmian. Np. można przewidzieć skutki wdrożenia systemu informatycznego (technologia) na kombinację rynek-produkt, organizację, kadry oraz dominującą koalicję. Z drugiej strony, określenie potrzeby zmian w zakresie produkowanych wyrobów i rynku docelowego identyfikuje wspomagające te zmiany systemy informatyczne.

Również w popularyzacji w Polsce metod i technik analizy i projektowania istnieje duża dysproporcja w porównaniu do bogatego dorobku światowego w tej dziedzinie. Tym niemniej pierwsze opracowania w tym względzie zostały już opublikowane ([1, 4, 5]). Dlatego w niniejszym artykule nie podjęto szerszej ich charakterystyki. Ogólnie można je podzielić na modelowanie danych oraz procesy. Typowym przykładem modelowania danych są modele związków encji, a procesów – diagramy przepływu danych. Najczęściej metody te są jednak kompromisem modelowania danych oraz procesów. Fizycznie mają one postać diagramów lub macierzy.

Spośród około setki tego typu metod i technik największe uznanie zdobyły:

- diagramy związków encji (obiekt-atrybut-związek),
- diagramy przepływu danych,
- A-grafy, D-grafy, I-grafy (podejście ISAC),
- diagramy dekompozycji funkcjonalnej,
- modelowanie danych metodą normalizacji,
- diagramy Jacksona,
- diagramy struktury,
- diagramy Nassi-Shneidermana,
- tablice i drzewa decyzyjne,
- diagramy Warniera-Orra.

Listę tę można wydłużyć. Obszerniejsze zestawienie metod, z bardziej dokładną ich charakterystyką, zawierają m.in. opracowania [2, 3, 4].

Niektóre z wyżej wymienionych metod są użyteczne w fazach wdrażania i użytkowania systemu. Należy wymienić tu przede wszystkim diagramy Jacksona i Nassi-Shneidermana w związku z programowaniem oraz prototypowanie i normalizację w odniesieniu do schematów relacyjnych baz danych. W tym względzie znaczącą rolę odgrywają języki czwartej generacji oraz pakiety CASE. Składnikami tych narzędzi są generatory kodów oraz generatory schematów relacyjnych baz danych. Użyteczną techniką stosowaną w tych fazach jest *Reverse Engineering*,

umożliwiająca generowanie dokumentacji analitycznej i projektowej na podstawie kodu źródłowego systemu.

Warsztaty metod i technik analizy oraz projektowania systemów

Artykuł niniejszy jest w zasadzie wprowadzeniem w tematykę jednodniowych warsztatów z zakresu podstawowych metod analizy i projektowania systemów. Przebieg warsztatów opiera się na analizie funkcjonowania przykładowego przedsiębiorstwa. Wykonanie poleceń umożliwiających rozwiązanie tego przypadku jest wspomagane wykładami, dyskusjami, pracą w grupach oraz odpowiednimi ćwiczeniami. Celem tych warsztatów jest opanowanie przez użytkowników, analityków i projektantów systemów metod i technik, umożliwiających definiowanie w sposób jednoznaczny potrzeb informacyjnych, a następnie realizację projektów.

Szczegółowy przebieg warsztatów obejmuje następujące punkty:

1. Zdefiniowanie celu warsztatów i spodziewane wyniki.
2. Tworzenie systemów informatycznych – wykład.
3. Prezentacja i wyjaśnienie istoty wiodącego przypadku.
4. Diagramy przepływu danych (DPD) – wykład.
5. Ćwiczenia z zakresu DPD.
6. Opracowanie diagramu kontekstowego przypadku – praca w grupach.
7. Prezentacja i weryfikacja diagramów kontekstowych przypadku – dyskusja.
8. Opracowanie diagramu zerowego przypadku – praca w grupach.
9. Prezentacja i weryfikacja diagramów zerowych przypadków – dyskusja.
10. Modelowanie związków encji (obiekt-atrybut-związek) – wykład.
11. Ćwiczenia z zakresu modelowania związków encji.
12. Opracowanie diagramu związków encji przypadku – praca w grupach.
13. Prezentacja i weryfikacja diagramów związków encji przypadku – dyskusja.
14. Projektowanie słownika-skorowidza danych – wykład.
15. Opracowanie słownika-skorowidza danych przypadku.
16. Podsumowanie warsztatów.

W trakcie realizacji warsztatów stosownie do harmonogramu, na bieżąco jest wręczana uczestnikom obszerna dokumentacja, dotycząca treści wykładów i ćwiczeń oraz zawierająca poprawne rozwiązania.

LITERATURA


- [1] Fuglewicz P.: Systemy CASE – problemy, techniki, rozwiązania, w: Komputerowe Wspomaganie Tworzenia Systemów Informatycznych, Trzecia Wiosenna Szkoła PTI, Świnoujście 1990
- [2] Martin J., McClure C.: Structured Techniques. The Basis for CASE, Prentice Hall, New York 1988
- [3] Turner W.S., Langerhorst R.P., Hice G.F., Eilers H.B., Uijtjenbroek: System Development Methodology, Pandata, North-Holland, 1988
- [4] Wrycza S.: Współczesne metodyki tworzenia systemów informatycznych zarządzania, PTC, Gdańsk 1989
- [5] Wrycza S.: Aktualne trendy komputerowo wspomaganego tworzenia systemów informatycznych, w: Komputerowe Wspomaganie Tworzenia Systemów Informatycznych, Trzecia Wiosenna Szkoła PTI, Świnoujście 1990
- [6] Wrycza S.: The Impact of CASE Tools on Teamwork of Information Systems Developers, in Finkelstein A. et al., Human Factors in Analysis and Design of Information Systems, North Holland, pp. 105–122, Amsterdam 1990
- [7] Wrycza S.: The ISAC-Driven Transition between Requirements Analysis and ER Conceptual Modelling, Information Systems, Vol. 15, pp. 603–614, No. 6 (1990).

CHCESZ BYĆ PIERWSZY

1. AURA
2. CEMENT, WAPNO, GIPS
3. CHŁODNICTWO
4. CIEPŁOWNICTWO, GRZEWNICTWO, WENTYLACJA
5. DOZÓR TECHNICZNY
6. ELEKTRONIKA
7. ELEKTRONIZACJA
8. ENERGETYKA
9. GAZ, WODA I TECHNIKA SANITARNA
10. GAZETA CUKROWNICZA
11. GOSPODARKA MIĘSNA
12. GOSPODARKA PALIWAMI I ENERGIA
13. GOSPODARKA WODNA
14. HUTNIK + WIADOMOŚCI HUTNICZE
15. INFORMATYKA
16. INŻYNIERIA I BUDOWNICTWO
17. INŻYNIERIA MATERIAŁOWA
18. INŻYNIERIA MORSKA

19. KARBO
20. MATERIAŁY BUDOWLANE
21. NOWATOR - PROBLEMY WYNALEZCZOŚCI
22. OCHRONA POWIETRZA
23. OCHRONA PRACY
24. OCHRONA PRZED KOROZJĄ
25. ODZIEŻ
26. OPAKOWANIE
27. PROBLEMY JAKOŚCI
28. PRZEGLĄD ELEKTROTECHNICZNY
29. PRZEGLĄD GASTRONOMICZNY
30. PRZEGLĄD GEODEZYJNY
31. PRZEGLĄD GÓRNICZY
32. PRZEGLĄD ODLEWNICTWA
33. PRZEGLĄD PAPIERNICZY
34. PRZEGLĄD PIEKARSKI I CUKIERNICZY
35. POLISH TECHNICAL REVIEW
36. PRZEGLĄD SKÓRZANY

37. PRZEGLĄD TECHNICZNY
38. PRZEGLĄD TELEKOMUNIKACYJNY I WIADOMOŚCI TELEKOMUNIKACYJNE
39. PRZEGLĄD WŁÓKIENNICZY I TECHNIK WŁÓKIENNICZY
40. PRZEGLĄD ZBOŻOWO-MLYNARSKI
41. PRZEMYSŁ CHEMICZNY
42. PRZEMYSŁ DRZEWNY
43. PRZEMYSŁ FERMENTACYJNY I OWOCOWO-WARZYWNY
44. PRZEMYSŁ SPOŻYWCZY
45. RADIOELEKTRONIK + AUDIO-HIFI-VIDEO
46. RUDY I METALE NIEŻELAZNE
47. SZKŁO I CERAMIKA
48. TECHNIKA ZAGRANICZNA
49. WIADOMOŚCI ELEKTROTECHNICZNE
50. WIADOMOŚCI GÓRNICZE
51. WIADOMOŚCI PRODUKCYJNE

Wydawnictwo
 SIGMA-NOT
 00-716 Warszawa
 ul. Bartycka 20

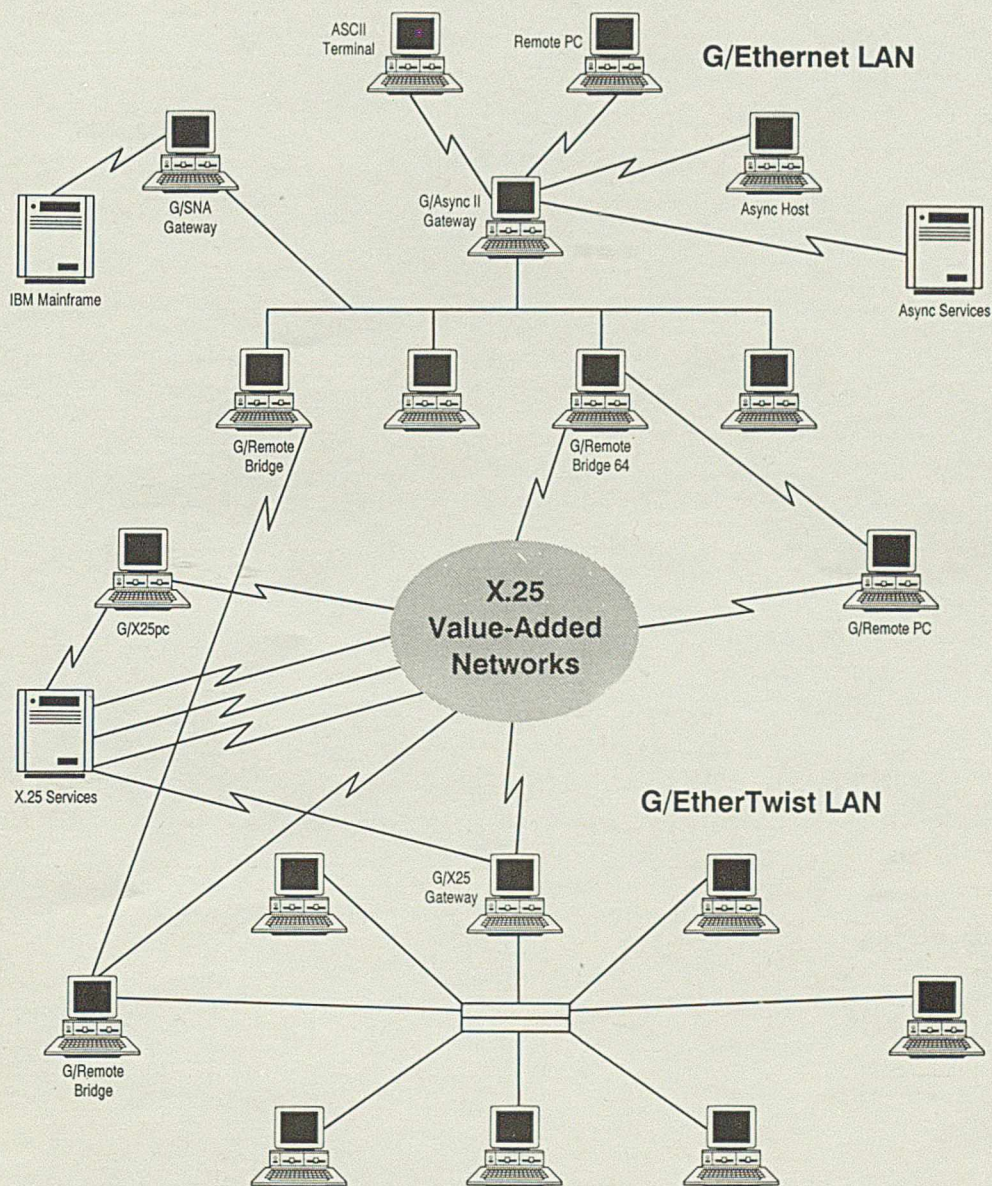
Informacje telefoniczne
 tel. 40-30-86, 40-35-89
 fax 20-31-16

PRENUMERUJ!

<p>Średniawa M.: Sieci inteligentne – usługi i architektura INFORMATYKA 1992, nr 9, s. 1 Sposób działania, stosowane architektury i standaryzacja rozwiązań oraz usługi sieci inteligentnych w telekomunikacji.</p>	<p>Średniawa M.: Intelligent networks – services and architecture INFORMATYKA 1992, No. 9, p. 1 Operation method, applied architectures and standarization, as well as services of intelligent networks in telecommunication.</p>	<p>Średniawa M.: Intelligente Netze – Dienstleistungen und Architektur INFORMATYKA 1992, Nr. 9, S. 1 Betriebsmethode, angewendete Architekturen und Normung, sowie Dienstleistungen der intelligenten Netze in Nachrichtentechnik.</p>
<p>Skrzypek J., Szubra M.: Symulacyjne gry decyzyjne jako narzędzie aktywnego kształcenia menedżerów INFORMATYKA 1992, nr 9, s. 12 Charakterystyka symulacyjnych gier decyzyjnych i ich porównanie z klasycznymi metodami kształcenia menedżerów oraz specyfikacja wymagań dla właściwego oprogramowania tych gier.</p>	<p>Skrzypek J., Szubra M.: Simulative decision games as a tool for active teaching of managers INFORMATYKA 1992, No. 9, p. 12 Characteristics of simulative decision games and their comparison with classic methods of manager teaching, as well as specification of requirements for the games adequate software.</p>	<p>Skrzypek J., Szubra M.: Simulative Entscheidungsspiele als Hilfsmittel für aktive Führungskräfteausbildung INFORMATYKA 1992, Nr. 9, S. 12 Eine Charakteristik von simulativen Entscheidungsspielen und ihre Vergleichung mit klassischen Methoden der Führungskräfteausbildung, sowie eine Spezifikation von Bedingungen der geeigneten Software für solche Spiele.</p>
<p>Kosmowska-Miszalska D.: Strojzenie mechanizmu rozumowania systemów ekspertowych z uwzględnieniem warunków niepewności INFORMATYKA 1992, nr 9, s. 17 Charakterystyka działania mechanizmu rozumowania w warunkach niepewności na przykładzie rozwiązań amerykańskiego pakietu GURU jako narzędzia do tworzenia systemów ekspertowych.</p>	<p>Kosmowska-Miszalska D.: Arranging of reasoning mechanism for expert systems with regard of uncertainty's conditions INFORMATYKA 1992, No. 9, p. 17 Characteristics of reasoning mechanism's operation in conditions of uncertainty on example of solutions of GURU, the american software package, as a tool for expert system building.</p>	<p>Kosmowska-Miszalska D.: Abstimmung des Urteilsmechanismus für Expertensysteme mit Berücksichtigung der Unsicherheitsbedingungen INFORMATYKA 1992, Nr. 9, S. 17 Eine Charakteristik von Betriebsmethode des Urteilsmechanismus bei Unsicherheitsbedingungen auf Beispiel von Lösungen des GURU, eines amerikanischen Softwarepakets, als Hilfsmittel für Expertensystemebau.</p>
<p>Wawrzyńczyk M., Stapor K.: Wpływ algorytmów tablic rozproszonych na efektywność działania systemów ekspertowych INFORMATYKA 1992, nr 9, s. 20 Charakterystyka oraz ocena skuteczności różnych metod i algorytmów porządkowania danych w systemach ekspertowych w celu zwiększenia ich efektywności.</p>	<p>Wawrzyńczyk M., Stapor K.: Influence of distributed tables algorithms on operation effectiveness of expert systems INFORMATYKA 1992, No. 9, p. 20 Characteristics and effectiveness evaluation of different methods and algorithms for data arrangement in expert systems according its effectiveness increasing.</p>	<p>Wawrzyńczyk M., Stapor K.: Einfluss von Algorithmen für zerstreute Tabellen auf Betriebseffektivität der Expertensysteme INFORMATYKA 1992, Nr. 9, S. 20 Eine Charakteristik und Beurteilung von Wirksamkeit der verschiedenen Methoden und Algorithmen des Datenordnens in Expertensystemen zwecks Steigerung ihrer Effektivität.</p>
<p>Wrycza S.: Metody analizy i projektowania systemów w praktyce INFORMATYKA 1992, nr 9, s. 26 Wprowadzenie do jednodniowych warsztatów, prowadzonych przez Katedrę Przetwarzania Danych Uniwersytetu Gdańskiego na temat praktycznego stosowania metod i technik planowania, analizy i projektowania systemu informatycznego.</p>	<p>Wrycza S.: Methods for system analysis and design in practice INFORMATYKA 1992, No. 9, p. 26 Introduction to one-day workshop, carried on by Data Processing Department of the Gdansk University, regarding practical application of methods and technics for data processing system analysis and design.</p>	<p>Wrycza S.: Methoden für Systemanalyse und -projektierung in der Praxis INFORMATYKA 1992, Nr. 9, S. 26 Eine Einführung zu eintägigen Workshops, die zum Thema der praktischen Anwendung von Methoden und Techniken der EDV-Systemplanung, -analyse und -projektierung im DV-Lehrstuhl der Danziger Universität geführt werden.</p>

Produkty sieciowe LAN i WAN

Gateway
communications, inc.



BEZKONKURENCYJNE PRODUKTY
SIECI KOMPUTEROWYCH NAGRADZANE PRZEZ:
PC MAGAZINE, LAN MAGAZINE, INFO WORLD
OFERUJE AUTORYZOWANY DYSTRYBUTOR:



MIKROB

P.W.P.T. MIKROB SP. Z O.O.

20-346 Lublin, ul. Długa 5
Tel. (0-81) 420-61, faks 415-43, teleks 643776 MIKRO PL



Na życzenie wysyłamy katalog produktów. Atrakcyjny program współpracy dla dealer'ów.

0/15/91

JUNISOFTEx Sp. z o.o.

44-100 GLIWICE ul. Konstytucji 11,
tel.-faks 31-75-10, 31-90-81 do 88 w. 250, 272, 282, teleks 036233

JUNISOFTEx – to firma z tradycjami i najdłuższymi doświadczeniami w eksploatacji wielodostępnych systemów komputerowych Novell w kraju.

JUNISOFTEx – to najstarszy w kraju wykonawca własnych wielodostępnych systemów pracujących w sieci NetWare firmy Novell.

JUNISOFTEx – to autoryzowany reseller amerykańskiej firmy Novell.

JUNISOFTEx – to dostawca zintegrowanych systemów komputerowych działających w kilkudziesięciu firmach obejmujących w każdym przedsiębiorstwie, niezależnie od formy własności i dziedziny gospodarki: finanse, majątek obrotowy, majątek trwały i nietrwały, sprzedaż, techniczne przygotowanie produkcji oraz kadry-płace.

JUNISOFTEx – to dostawca i wykonawca autoryzowanych sieci lokalnych firmy Novell.

JUNISOFTEx – to dostawca sprawdzonego sprzętu komputerowego renomowanych firm ALR, IBM, TEAM, DIGILAB, EPSON, MANNESMANN TALLY.

JUNISOFTEx – to wyłączny i autoryzowany dystrybutor komputerów i terminali firmy DIGILAB na Śląsku.

JUNISOFTEx – to dostawca i wykonawca okablowania lokalnych sieci komputerowych ARCNET, ETHERNET na kablach zwykłych i światłowodowych oraz na dwużyłowych kablach telefonicznych w przypadku stosowania oprogramowania NetWare ACCESS SERVER firmy Novell.

JUNISOFTEx – to nauczyciel, który chętnie podzieli się swoją wiedzą na organizowanych kursach w swojej szkole informatycznej – w tym nauczy Cię: podstaw informatyki, obsługi sprzętu komputerowego, eksploatacji własnych systemów informatycznych, edytorów tekstu: CHiWRITER, WordPerfect, arkuszy kalkulacyjnych QPRO v. 3.0, LOTUS 1-2-3 itp.

JUNISOFTEx – to doradca w zakresie komputerowych metod organizacji i eksploatacji systemów komputerowych.

JUNISOFTEx – to strażnik postępu i nowoczesności w Twojej firmie.

JUNISOFTEx – to gwarancja niezawodności, rzetelności i terminowości.

JUNISOFTEx – to Twój doradca i partner, któremu możesz zaufać, który Cię nigdy nie zawiedzie.

JUNISOFTEx – to partner, który Cię wysłucha i zawsze pomoże podjąć dobrą decyzję.

Jeżeli chcesz

- ✧ zastosować najnowocześniejsze systemy komputerowe oraz lokalne sieci komputerowe,
- ✧ zreorganizować, usprawnić i szybko skomputeryzować swoje przedsiębiorstwo,
- ✧ lepiej wykorzystać już istniejące w Twoim przedsiębiorstwie komputery,
- ✧ wymienić swoje niefortunnie zakupione oprogramowanie,
- ✧ pozbyć się problemów eksploatacyjnych, upadających systemów, nie działających komputerów,
- ✧ przeskoczyć załogę swojego przedsiębiorstwa,
- ✧ dobrze zainwestować swoje pieniądze i podnieść rangę swojego przedsiębiorstwa,
- ✧ w przyszłości mieć dostęp do krajowej i światowej sieci komputerowej,

TO ZGŁOŚ SIĘ DO NAS!