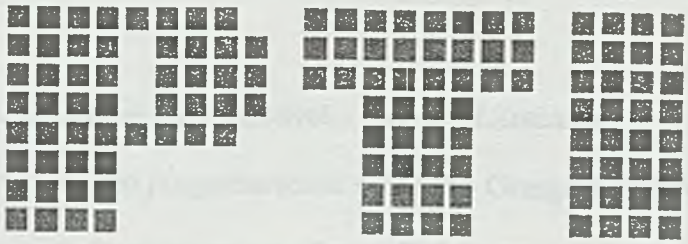
VII Górska
Szkoła PTI
Szczyrk '95
19-23.06.95r.



VII Górska Szkoła PTI

Szczyrk '95

19-23.06.1995r.

POLSKIE TOWARZYSTWO INFORMATYCZNE
ODDZIAŁ GÓRNOŚLĄSKI

Katowice 1995r.

VII Ogólnopolska
Szkoła PTI

Szczyrk '95

19-23.06.1995 r.

POLSKIE TOWARZYSTWO INFORMATYCZNE
ODDZIAŁ GÓRNOŚLĄSKI

Katowice 1995 r.

Spis Treści

1. Oprogramowanie systemów rozproszonych - Krzysztof Zieliński 1
2. Zagadnienia rozproszonego programowania w logice - Grzegorz Czajkowski 12
3. Protokół SNMP i zarządzanie sieciamu - Janusz Piela 25
4. Sieci komputerowe i metropolitarne. Sieci bez problemów - Zbigniew Skotniczy 36
5. Koncentratory Wielkich Sieci - Krzysztof Ślatała 38
6. Problemy budowy systemów multibaz danych - Marcin Szyprowski, Z. Królikowski, M. Januchta 43
7. Realizacja dużego projektu informatycznego dla PKP - Adam Lipiński 53
8. Rozproszona baza danych ORACLE - Zbigniew Pojmański 59
9. DB2 - Architektura rozproszonej, relacyjnej bazy danych - Jerzy Kruszewski 68
10. BETA Systems - product description - Jochen Arms 73
11. Moduł automatycznego tłumaczenia jako uniwersalny interfejs do bazy danych - Borys Czerniejewski 75
12. Usługi sieciowe w Internecie - Marek Miłosz 84
13. Kierunki rozwoju transmisji danych w sieciach TPSA - Jarosław Kępkiewicz, Marian Suskiewicz 94
14. Koncepcja realizacji poczty elektronicznej X.400 w TP S.A. - Sławomir Michalski, Marian Suskiewicz 99
15. Wykorzystanie metafor organizacyjnych do badania potrzeb użytkowników - Marian Kuraś, Agnieszka Zając 106
16. Organizacja sieciowych serwisów informacyjnych dla małych i dużych przedsiębiorstw - Elżbieta Miłosz, Marek Miłosz 119
17. Praktyczne aspekty budowy rozproszonej bazy danych - Andrzej Rogowski, Dariusz Koncewicz 125
18. Implementacja Data Warehouse w rozproszonym środowisku informatycznym - metodologia SAS Institute - Tomek Kuźmierz 133
19. System informatyczny w rzeczywistych warunkach istniejącego przedsiębiorstwa - Włodzimierz Adamski 138

20. Infrastruktura i usługi miejskiej sieci komputerowej środowiska śląskiego - Grzegorz Dymek, Krzysztof Nałęcki 145

21. Bazy danych państwowego monitorowania srodowiska - Leszek Bogusławski 152

22. Modelowanie i realizacja procesów gospodarczych w rozproszonych systemach baz danych - Stanisław Kędzierski 159

23. USENET w procesie dydaktycznym - Carl Ericson, Tadeusz Wilusz 168

24. Sieć transmisji danych Polskich Kolei Państwowych - Edward Mazur 177

Lista sponsorów

Bankowe Przedsiębiorstwo Telekomunikacyjne **TELBANK S.A.** W-wa,

BETA Systems Sowtware AG Berlin,

CIOG S.A. K-ce,

ComputerLand Poland W-wa,

Consortia Sp. z o.o. W-wa,

IBM Polska Sp. z o.o. W-wa,

ORACLE Polska Sp. z o.o. W-wa,

PKP Cenrtum Informatyki Kolejnictwa W-wa,

Przedsiębiorstwo Handlowo-Usługowe **VENTUS** Commputer Poznań,

Przedsiębiorstwo Techniki Bankowej **elba** Wrocław,

SAS Instytute W-wa,

SOLIDEX Ltd. Kraków,

Telekomunikacja Polska S.A.

Centrum Systemów Teleinformatycznych W-wa

Krzysztof Zieliński
Akademia Górniczo-Hutnicza

Oprogramowanie systemów rozproszonych

1. Wprowadzenie

W ostatnich latach dał się zaobserwować gwałtowny wzrost zainteresowania obliczeniami rozproszonymi. Wynika to z kilku wzajemnie uzupełniających się faktów.
Rozpowszechnienia szybkich stacji roboczych, które połączone siecią tworzą zasoby wieloprocesorowe możliwe do wykorzystania w ramach jednego zadania obliczeniowego.

- Wysokie koszty obliczeń realizowanych na superkomputerach zmusiły niektóre organizacje do poszukiwania tańszych możliwości ich realizacji przy wykorzystaniu posiadanych zasobów komputerowych.
- Duże ilości zbędnej mocy obliczeniowej, występujące na stacjach roboczych w okresie nocnym, a także podczas prac nie obciążających zbytnio procesora, jak np. edycja tekstu.
- Znaczący postęp w zakresie rozwoju oprogramowania sieciowego związany z przyjęciem standardów *de facto*, zwłaszcza na styku: oprogramowanie sieciowe -- system operacyjny.
- Ukończenie szeregu projektów badawczych z zakresu rozproszonego przetwarzania i komercjalizacja uzyskanych rezultatów.
- Rozwój systemów mutimedialnych i teleinformatyki.

Możliwości realizacji rozproszonego i równoległego przetwarzania w sieciach stacji roboczych były przez dosyć długi okres czasu ograniczone z powodu braku odpowiednich środowisk programowych. Obecnie jednak sytuacja uległa zasadniczej zmianie i użytkownik musi podjąć coraz trudniejszą decyzję odnośnie wyboru odpowiedniego narzędzia programowego.

Dokonanie trafnego wyboru utrudnia brak rozeznania w zakresie przydatności poszczególnych rozwiązań programowych, ich efektywności, sposobu użycia oraz perspektyw rozwoju.

Celem tego artykułu jest przedstawienie problematyki oprogramowania systemów rozproszonych oraz kierunków jego rozwoju w przyszłości.

2. Charakterystyka systemów rozproszonych

Obliczenia rozproszone, będące przedmiotem niniejszego opracowania dotyczą pewnego szczególnego rodzaju organizacji przetwarzania informacji w zbiorze stacji roboczych, połączonych siecią komputerową. Polega ona na wykorzystaniu takiego systemu jako jednego globalnego zasobu, stanowiącego środowisko wieloprocesorowe. Zasoby sieci komputerowej wykorzystywane w tym trybie określa się mianem metakomputera lub klastra obliczeniowego [14]. Pojęcie metakomputera jest szersze i obejmuje system sieciowy integrujący trzy składowe: zasoby obliczeniowe, pamięci masowe oraz środki wizualizacji wyników obliczeń, które są udostępniane przez jednolity interfejs użytkownika.

Zazwyczaj klastry obliczeniowe stanowią grupy stacji roboczych, połączonych szybkimi sieciami lokalnymi typu: Ethernet, FDDI lub UltraNet. Wynika to z dążenia do ograniczenia czasu komunikacji pomiędzy procesorami. Nie

oznacza to jednak, iż metakomputera nie można utworzyć w sieci zdalnej. Znane są przykłady wykorzystania komputerów kilku uniwersytetów komunikujących się w sieci Internet w celu rozwiązania jednego dużego problemu obliczeniowego.
Klastry stacji roboczych wykorzystywane są w dwóch trybach:

- **wsadowym** -- polegającym na tym, iż poszczególne programy zgłaszane do wykonania, alokowane są do poszczególnych komputerów klastra. Strategia alokacji programów uwzględnia zazwyczaj aktualne obciążenie komputerów, a sam algorytm alokacji zdań może być realizowany w sposób rozproszony bądź scentralizowany. Przykład oprogramowania organizującego pracę wsadową klastra stanowi Task Broker firmy HP lub NQS (*Network Queueing Software*) oferowany przez Thomasine Bailey Sterling Software.
- **równoległym** -- charakteryzującym się tym, że zadanie obliczeniowe dzielone jest na części, wykonywane równoległe w poszczególnych komputerach klastra. Zazwyczaj podziału zadania obliczeniowego na podzadania dokonuje się tak, aby komunikacja pomiędzy podzadaniami była możliwie rzadka oraz nie wymagała transferu zbyt długich wiadomości. Sposób organizacji obliczenia zależy od problemu obliczeniowego, algorytmu przetwarzania oraz narzędzi programowych. Środowiska programowe dla realizacji tego właśnie trybu przetwarzania stanowią przedmiot niniejszego opracowania.

Przetwarzanie wsadowe nie wymaga od programisty praktycznie żadnych dodatkowych umiejętności i nie różni się od korzystania z dużego komputera. Wykonanie programu w trybie równoległym związane jest z odpowiednią dekompozycją programu, często zmianą algorytmu obliczeniowego oraz zrozumieniem przyjętego modelu obliczeniowego. Stosowanie tego trybu przetwarzania wymaga zatem od programisty znacznie większych umiejętności, a uzyskanie wysokiej efektywności nie jest zadaniem prostym.

Istnieje szereg przyczyn, dla których klastry stacji roboczych (zwłaszcza wykorzystywane w trybie równoległym) stanowią atrakcyjną alternatywę dla dużych komputerów i superkomputerów. Należą do nich m.in.:

- Znacznie niższe koszty zakupu. Przykładowo zakup klastra stacji roboczych jest około dziesięć razy tańszy od zakupu superkomputera o podobnej mocy obliczeniowej. Dodatkowo koszty eksploatacji superkomputerów są znacznie większe niż stacji roboczych. Koszty roczne eksploatacji komputera Cray są porównywalne do ceny klastra stacji roboczych.
- Metakomputery są modularne. Wirtualny superkomputer może być budowany stopniowo w miarę dostępności środków oraz wzrastających potrzeb.
- Metakomputery stanowią ważny sposób wykorzystania środowisk stacji roboczych tradycyjnie używanych do przetwarza zadań osobistych. Pozwala to w naturalny sposób uniknąć problemów niekompatybilności oprogramowania i reprezentacji danych, które mogą mieć miejsce przy dostępie do superkomputerów.

Przetwarzanie oparte na sieciach stacji roboczych posiada jednak kilka cech niekorzystnych:

- Użytkowanie klastrów stacji roboczych wymaga większych zasobów ludzkich i finansowych oraz czasu potrzebnego dla konfiguracji systemu i zrównoleglania oprogramowania niż w przypadku eksploatacji superkomputera.
- Użytkownicy systemu zazwyczaj muszą mieć możliwość konsultowania technik równoległego przetwarzania z ekspertami z tej dziedziny.
- Szybkość komunikacji pomiędzy stacjami roboczymi jest znacznie mniejsza niż w komputerach wieloprocesorowych, co sprawia, iż klastry stacji roboczych nie

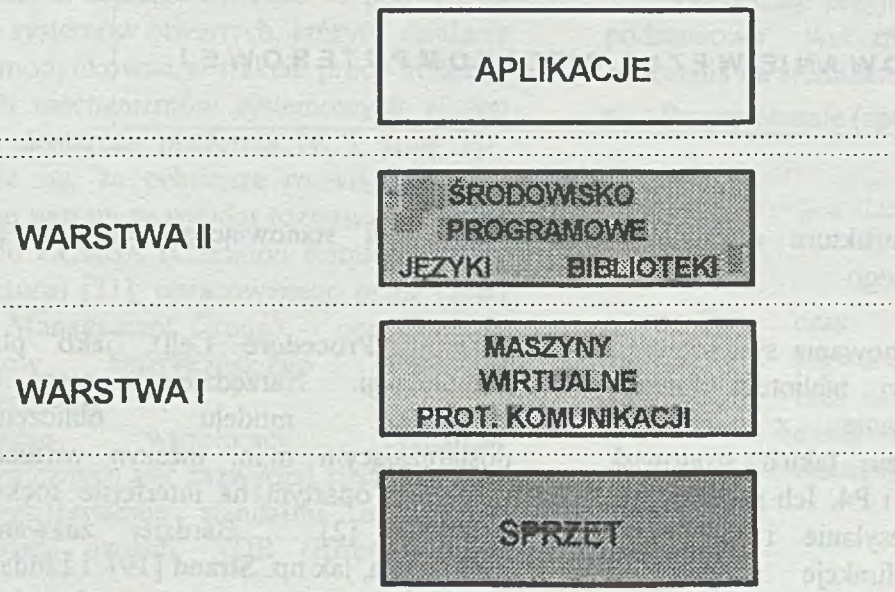
nadają się do pewnych zastosowań, np. przetwarzania transakcji.

Należy podkreślić, że modele obliczeniowe oraz środowiska programowe stosowane w równoległym trybie użytkowania klastra stacji roboczych są podobne do używanych na maszynach wieloprocesorowych (z tzw. masywną równoległością). Przewiduje się, że ten rodzaj maszyn będzie dominujący w przyszłości. Stąd niektóre firmy komputerowe oraz wielu użytkowników traktuje klastry stacji roboczych wykorzystywane w trybie równoległym jako ważny etap prowadzący do masywnego przetwarzania równoległego. Jako przykład można tu podać firmę CONVEX, oferującą tzw. Meta Serię, tj. klaster stacji roboczych firmy HP ściśle współpracujący z superkomputerami CONVEX. Rozwiązanie to poprzedza maszynę SPP *(Scalar Parallel Processor)* zapowiadaną przez tą firmę. Będzie to maszyna złożona z wielu niezależnych procesorów wraz z pamięcią, stanowiących węzły komunikujące się przez specjalny układ, zapewniający praktycznie bezpośrednią wymianę danych między nimi.

3. Budowa oprogramowania

W budowie oprogramowania do obliczeń rozproszonych i równoległych można wyróżnić dwie warstwy, pokazane na rys. 1.

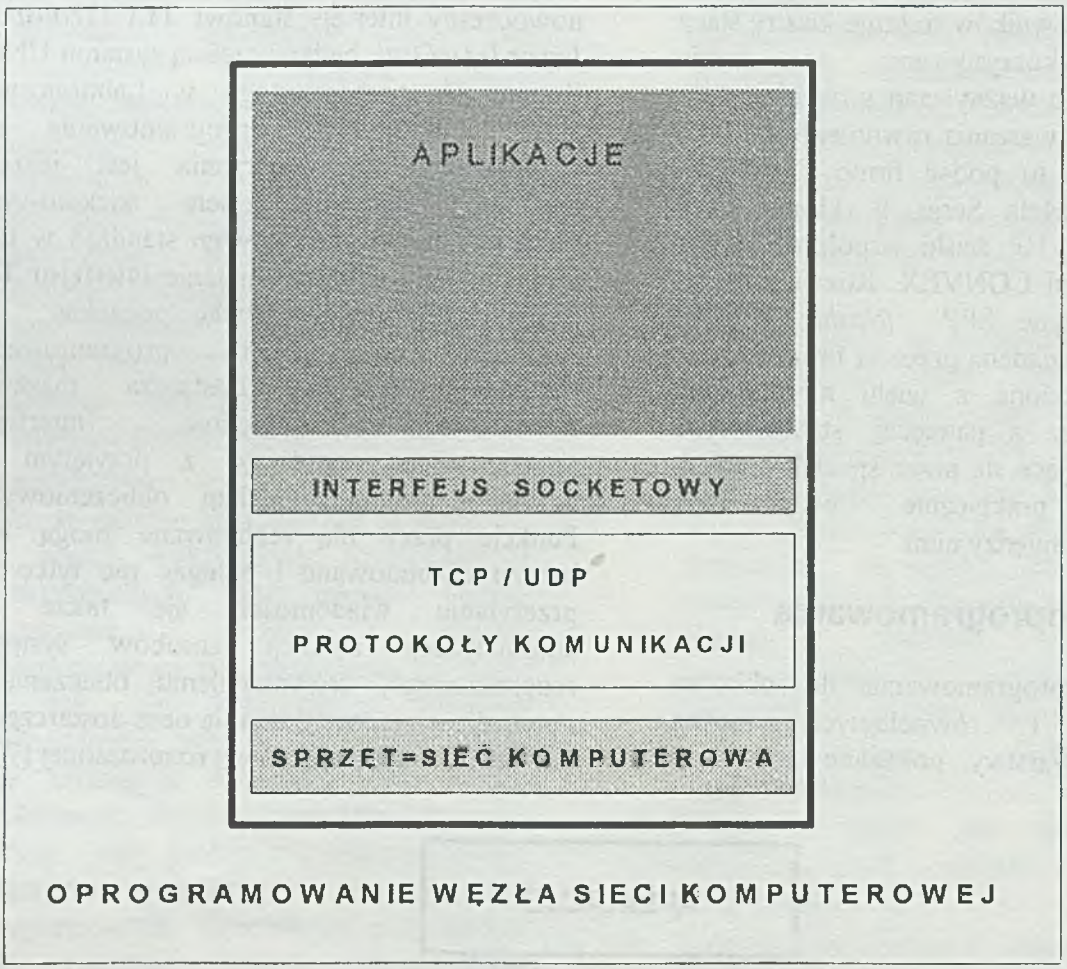
Pierwszą z nich stanowi oprogramowanie, które współpracuje bezpośrednio ze sprzętem sieciowym i interfejsy stanowiące styk pomiędzy systemem operacyjnym a siecią. Do najczęściej wykorzystywanych interfejsów tego typu należy interfejs socketowy wprowadzony w roku 1986, jako część systemu operacyjnego UNIX 4.3 BSD. Jest to styk programowy zapewniający dostęp do usług protokołu transportowego. Podobny, lecz bardziej nowoczesny interfejs stanowi TLI *(Transport Layer Interface)*, będący częścią systemu UNIX System V, opracowanego w Laboratorium AT&T. Większość oprogramowania do rozproszonego przetwarzania jest jeszcze budowana nad interfejsem socketowym, stanowiącym *de facto* pewien standard w tym zakresie. Na rozpowszechnienie interfejsu TLI należy chyba jeszcze trochę poczekać. W zależności od rodzaju oprogramowania rozważana warstwa dostarcza maszyny abstrakcyjnej, względnie interfejsu programowego zgodnego z przyjętym w warstwie drugiej modelem obliczeniowym. Funkcje przez nią realizowane mogą być bardzo rozbudowane i polegać nie tylko na przesyłaniu wiadomości, ale także na automatyzacji alokacji zasobów systemu rozproszonego, zrównolegleniu obliczenia i optymalizacji jego wykonania oraz dostarczeniu wirtualnej, dzielonej pamięci rozproszonej [7].



Rysunek 1: Warstwy oprogramowania rozproszonych systemów obliczeniowych

Druga warstwa stanowi właściwe środowisko programowe, w którym budowane są aplikacje użytkownika. Może ono posiadać formę zbioru procedur bibliotecznych lub języka programowania.

Oprogramowanie dla systemów rozproszonych i równoległych różni się bardzo sposobem konstrukcji warstwy pierwszej w sensie jej złożoności i dostępności dla użytkownika. Najprostszą budowę posiadają pakiety programowe udostępniające przesyłanie wiadomości i synchronizację procesów. Przykład typowej struktury oprogramowania dla obliczeń w sieciach stacji roboczych pokazano na rys. 2.



Rysunek 2. Typowa struktura oprogramowania węzła sieci stanowiącego element systemu rozproszonego

Od strony oprogramowania systemowego część narzędzi stanowi biblioteki funkcji, ułatwiających korzystanie z interfejsu socketowego. Przykładem takich środowisk programowych są PVM i P4. Ich podstawowe funkcje obejmują przesyłanie i odbieranie wiadomości, proste funkcje adresacji i synchronizacji procesów. Niektóre narzędzia, jak np. ANSA wykorzystują mechanizm RPC (Remote Procedure Call), jako platformę komunikacji. Narzędziem o nieco bardziej złożonym modelu obliczeniowym, dostarczającym m.in. maszyn wirtualnych i obiektów, opartym na interfejsie socketowym jest SR [2]. Bardziej zaawansowane środowiska, jak np. Strand [19] i Linda [12,1] dostarczają maszyn abstrakcyjnych, implementujących nie tylko sposób

komunikacji, ale również schemat realizacji obliczeń. Środowiska te zbudowane są odpowiednio na platformie oprogramowania ONC (Open Network Computing) firmy SUN lub NCS (Network Computing System) firmy HP. Obydwie te platformy określają dla implementowanych nad nimi „mechanizmy" komunikacyjne i synchronizacyjne.

Odrębne zagadnienie stanowi zapewnienie współbieżności przetwarzania wewnątrz procesów systemu operacyjnego. Rozwiązanie tego problemu sprowadza się do udostępnienia tak zwanych lekkich procesów oraz opracowania mechanizmów przełączania kontekstu w ramach pojedynczego procesu systemu operacyjnego. W istniejącym oprogramowaniu mechanizmy te dostarczane są przez maszyny wirtualne o różnym stopniu efektywności i sposobie implementacji. Pewną zmianę w sposobie implementacji tych funkcji wniesie zapewne system Solaris 2.x, posiadający standardowe biblioteki obsługi lekkich procesów na poziomie systemu operacyjnego, oraz nowa platforma ONC+, wykorzystująca te mechanizmy.

Duże znaczenie dla budowy systemów rozproszonych posiada zagadnienie wiązania poszczególnych składowych aplikacji. Może ono być realizowane statycznie -- na etapie kompilacji programu, względnie dynamicznie -- na etapie wykonania. Nie analizując tego bardzo ważnego zagadnienia szerzej, należy stwierdzić, iż łączenie dynamiczne pozwala na budowę systemów otwartych, których działanie można modyfikować w trakcie pracy systemu. Pewnych mechanizmów systemowych w tym zakresie dostarcza platforma NCS firmy HP. Oczekuje się, że pełniejsze rozwiązanie tego problemu nastąpi w wyniku rozpowszechnienia standardu CORBA (Common Request Broker Architecture) [31], opracowanego przez OMG (Object Management Group) -- porozumienie potentatów amerykańskiego przemysłu komputerowego.

Zagadnienie współpracy systemów heterogenicznych jest zazwyczaj rozwiązywane przy wykorzystaniu standardu zewnętrznej reprezentacji danych: XDR *(External Data Representation)*, która jest niezależna od specyfiki architektur sprzętowych.

Należy zauważyć, iż obecnie zasadnicze problemy leżą w konstrukcji pierwszej warstwy oprogramowania. Jej standaryzacja jest utrudniona wobec mnogości architektur sprzętu komputerowego. Prowadzi to do sytuacji, iż algorytm rozproszony lub równoległy musi być zaprojektowany z uwzględnieniem danej architektury i bardzo często bezpośrednio realizować alokację zasobów fizycznych. Sprawia to, że powstające oprogramowanie nie jest przenośne i wymaga od jego autorów znajomości sprzętu. Ponadto powstają duże trudności z predykcją efektywności realizacji obliczeń.

4. Ocena efektywności obliczeń rozproszonych

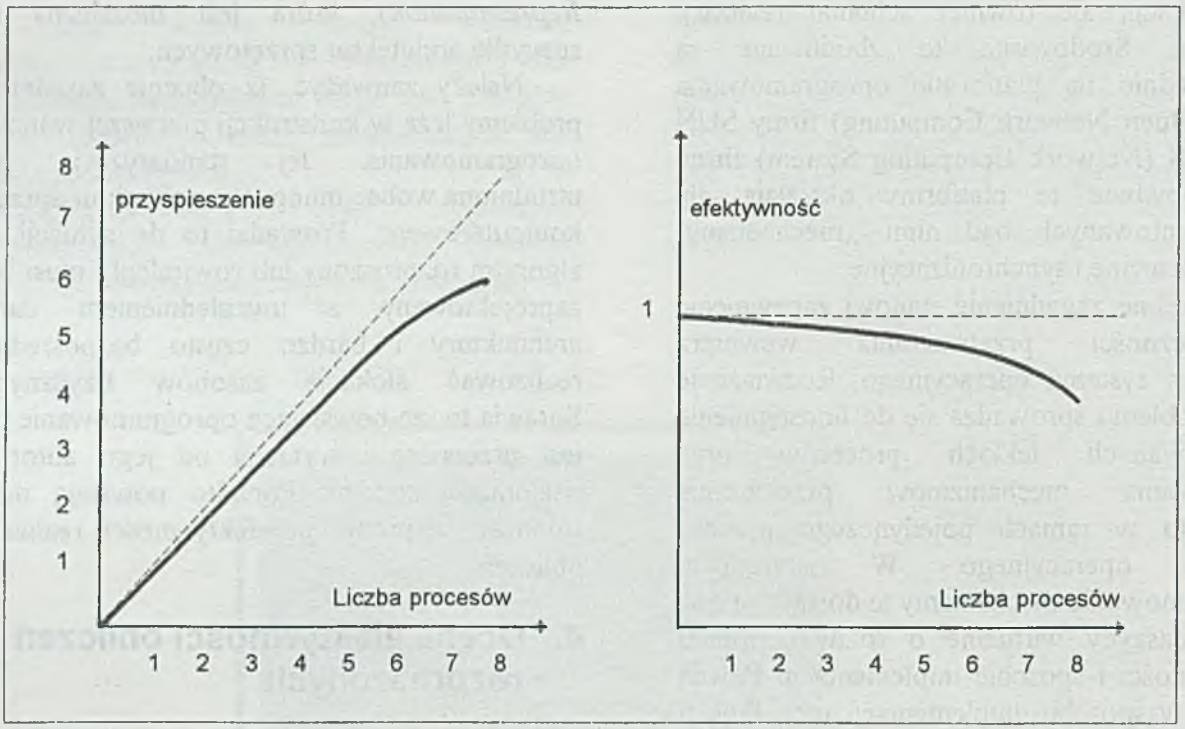
Efektywność obliczeń w systemach rozproszonych i wieloprocesorowych zależy decydująco od parametrów takich jak: liczba i szybkość procesorów oraz topologia i szybkość połączeń. Parametry te określają dwa główne zasoby systemu, tj. moc obliczeniową i przepustowość komunikacji. Konstrukcja efektywnych programów rozproszonych wymaga wypracowania kompromisu pomiędzy narzutami związanymi z komunikacją, a użytecznym wykorzystaniem procesorów dla zadań obliczeniowych.

Najczęściej przyjmuje się [21, 22] dwa podstawowe wskaźniki jakości realizacji obliczenia na architekturze wieloprocesorowej:

- Przyspieszenie (speed-up), określane jako

$$s = \frac{T_s}{T_r}$$

gdzie: T_s - czas obliczeń algorytmu sekwencyjnego,
T_r - czas obliczeń równoległego na danej architekturze wieloprocesorowej,



Rysunek 3: Typowe przebiegi przyspieszenia i efektywności obliczeń w systemie rozproszonym w funkcji liczby procesorów.

Efektywność *(efficiency)*, zdefiniowana jako

$$e = \frac{s}{N}$$

gdzie: N - liczba procesorów, na których realizowane są obliczenia.

Wzrost liczby procesorów prowadzi do zwiększenia narzutu czasowego związanego z komunikacją i sprawia, że efektywność spada, a uzyskiwane przyspieszenie rośnie wolniej niż liniowo. Zostało to pokazane na rys. 3.

Podobny efekt dotyczy również systemów wieloprocesorowych z dzieloną pamięcią. Przyczyną spadku efektywności jest w tych systemach rywalizacja w dostępie do pamięci.

W celu osiągnięcia wysokiej efektywności należy we właściwy sposób dobrać wielkość zadań obliczeniowych liczonych na poszczególnych procesorach tak, aby ani zbyt częsta komunikacja nie powodowała obniżenia efektywności, ani zbyt duże zadania wykonywane na pojedynczych procesorach nie zmniejszały potencjalnej równoległości przetwarzania. Zagadnienie to określa się doborem właściwej ziarnistości obliczenia rozproszonego. Dla każdego problemu obliczeniowego można zatem wskazać określoną liczbę procesorów, dla której uzyskuje się maksimum efektywności.

Rozważania te ilustrują spostrzeżenie dokonane w 1967 roku zwane „prawem Amdahla", stwierdzające, że narzuty komunikacyjne ograniczają możliwość wzrostu przyśpieszenia przetwarzania wraz ze wzrostem liczby procesorów do wartości określonej przez stosunek czasu obliczeń całego zadania na pojedynczym procesorze do komunikacji [21]. Narzuca to pewien teoretyczny kres możliwości uzyskania przyspieszenia obliczeń na drodze zrównoleglenia. Prawo to jest kwestionowane i podważane przez różnych badaczy, jednakże jego istnienie potwierdza

zdecydowana większość wyników praktycznych. W roku 1988 John Gustafson zademonstrował, że prawo Amdahla może być złamane przez równoczesne zwiększanie rozmiaru zadania obliczeniowego i liczby procesorów, na których jest rozwiązywane.

Uzyskanie wysokiej efektywności obliczeń w systemach rozproszonych wymaga wnikliwej analizy algorytmu przetwarzania oraz jego dostosowania do wybranej architektury sprzętowej. Bardzo pomocne w tym celu są różnego rodzaju narzędzia do wizualizacji przebiegu obliczeń. Pozwalają one na określenie rozkładu komunikacji między procesami i obciążenia poszczególnych procesorów.

5. Przegląd systemów oprogramowania

Intensywny rozwój systemów do programowania rozproszonego przejawia się m.in. w dużej liczbie oprogramowania dostępnego w sferze publicznej *(public domain)*. Świadczy to z jednej strony o licznych projektach badawczych, a z drugiej o intensywnym dążeniu do szerszej akceptacji którejś z propozycji.

Z analizy szerokiej listy istniejącego i na stale pojawiającego się nowego oprogramowania wynikają między innymi następujące wnioski:

- Istnieje duża różnorodność narzędzi udostępniających przesyłanie wiadomości. Są one często tworzone przez grupy naukowców zajmujących się określoną tematyką, np. PVM, Express - obliczenia fizyczne, TCGMPS - obliczenia chemiczne, a więc rozwój tych systemów jest stymulowany przez konkretne zastosowania. Systemy te stanowią uzupełnienie najpopularniejszych języków programowania, jak C i FORTRAN, a ich opanowanie nie wymaga szczególnego wysiłku, nawet od użytkowników nie posiadających szerszych podstaw informatycznych.

- Znacznie mniejsza grupa narzędzi pozwala na obiektową konstrukcję oprogramowania rozproszonego. Są to systemy znacznie bardziej zaawansowane, a ich konfiguracja i eksploatacja wymaga pewnego doświadczenia. Część z tych systemów, jak np. Mentat czy Charm, stanowi rozwinięcie C++. Inne, jak np. Emerald, stanowią zupełnie oryginalnie skonstruowane języki programowania, starannie zbudowane w celu zapewnienia efektywności przetwarzania w środowisku rozproszonym. Praktyczne znaczenie tych narzędzi było stosunkowo małe. Posiadały one jednak duży wpływ na rozwój nowych koncepcji przetwarzania rozproszonego. Dopiero w ostatnim okresie obserwuje się intensywny rozwój komercyjnych systemów obiektowego rozproszonego przetwarzania rozproszonego.

- Istnieje grupa narzędzi zawierająca mechanizmy deklaratywne. Należą do nich m.in. Linda, PCN i Strand. Powstanie tego oprogramowania można uznać za pierwsze kroki w kierunku stworzenia narzędzi programowych, cechujących się elegancją rozwiązań językowych i koncepcyjnych oraz pozwalających na konstrukcje programów równoległych w sposób niezależny od sprzętu. Jest to możliwe dzięki wykorzystaniu zasobów abstrakcyjnych, takich jak np. przestrzeń wirtualna (przestrzeń krotek) czy topologie wirtualne. Efektywność tych narzędzi jest jednak znacznie mniejsza niż prostych bibliotek zapewniających tylko wymianę wiadomości. Stąd zasadniczo zakres ich zastosowań jest ograniczony. Konsekwentny rozwój tej grupy środowisk powinien przynieść rozwiązanie obecnych problemów konstrukcji oprogramowania rozproszonego i równoległego. Opublikowanie [9] w 1992 przez S. Taylora i J. Fostera zasad konstrukcji PCN roku uznano za jeden z ważniejszych kroków w rozwoju oprogramowania równoległego.

- Większość pakietów programowych powstała w ramach prac prowadzonych na uniwersytetach i w laboratoriach naukowych. Wyjątkową pozycję zajmuje wśród nich Argonne National Laboratory,

współpracujące z kilkoma uniwersytetami w USA, w którym opracowano najciekawsze z omawianych środowisk.

Porównanie przedstawionych środowisk programowych można w skrócie przedstawić następująco:

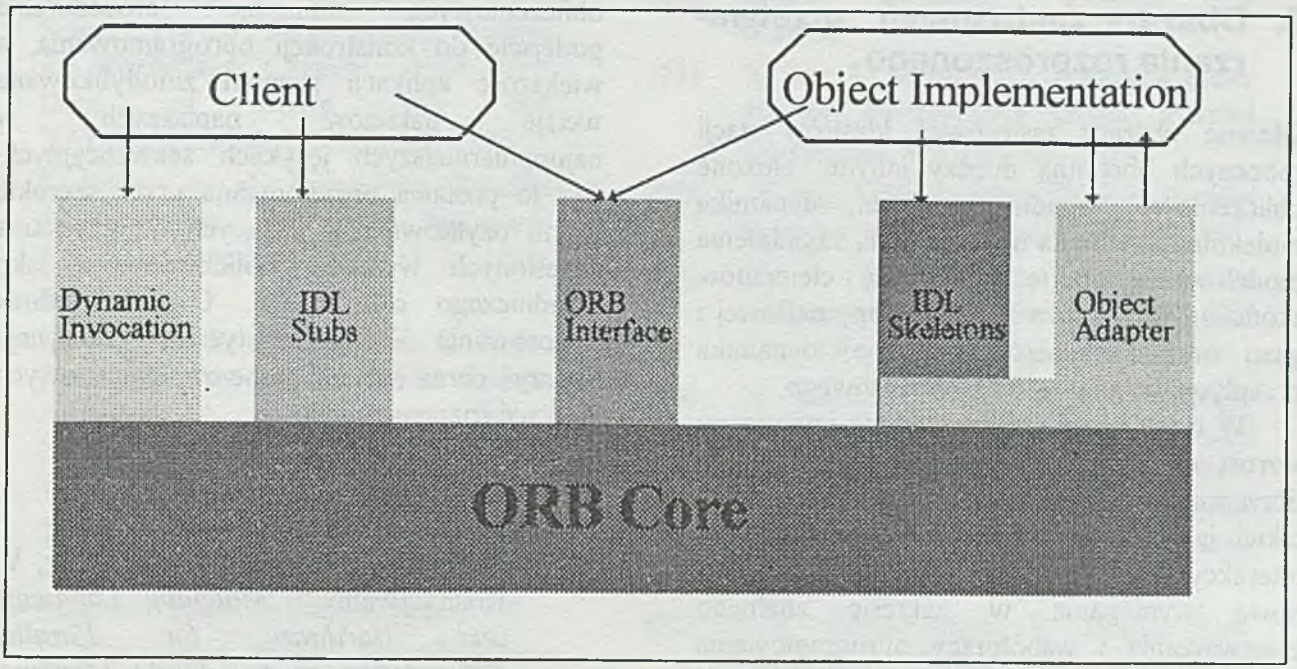
- Poszczególne środowiska programowe różnią się znacznie efektywnością transmisji danych pomiędzy procesami. Wpływ tej różnicy staje się jednak mniej znaczący przy wzroście stopnia komplikacji powiązań komunikacyjnych oraz współbieżności przesłań.
- Bardzo znaczący wpływ na efektywność obliczeń posiada stopień współbieżności realizacji zadań oraz operacji transmisji danych, w tym również sposób zarządzania buforami danych.
- Środowiska zapewniające większą potencjalną współbieżność operacji posiadają często bardzo słabe wskaźniki efektywności dla prostych testów komunikacyjnych, lecz dla złożonych zadań praktycznych osiągają bardzo dobre rezultaty. Odnosi się to w szczególności do porównania C-Lindy i PVM.
- Imperatywne środowiska programowe zapewniają znacznie większą szybkość obliczeń niż środowiska oparte o modele deklaratywne. Jest to fakt dobrze znany w odniesieniu do sekwencyjnych środowisk programowych i przenoszący się na środowiska rozproszone i równoległe. Dotyczy to w szczególności porównania Strandu 88 z PVM oraz P4.

6. Obiektowe przetwarzanie rozproszone

Wzrost zainteresowania obiektowym przetwarzaniem rozproszonym związany jest z upowszechnieniem sieci komputerowych oraz nowych aplikacji dotyczących rozproszonych baz danych, systemów sterowania, monitorowania i zarządzania oraz systemów multimedialnych itp.

Znaczący wpływ na rozwój środowisk programowych w tym obszarze posiadają prace prowadzone przez OMG (Object Management Group). OMG stanowi konsorcjum dużych firm komputerowych i wiodących producentów oprogramowania, którego zadaniem jest doprowadzenia do standaryzacji zasad budowy obiektowych aplikacji obiektowych oraz wymagań, które muszą być spełnione, aby oprogramowania produkowane przez różne firmy mogło współpracować w sieci komputerowej w ramach zdalnego przetwarzania.

Jednym z głównych osiągnięć OMG jest ustalenie standardu CORBA [31] (Common Object Request Broker Architecture). Standard ten stanowi specyfikację architektury warstwy oprogramowania, której zadanie polega na dostarczeniu obiektom mechanizmów transparentnego wywołania operacji na innych obiektach niezależnie od ich umiejscowienia w węzłach sieci komputerowej. W modelu CORBA aplikacja stanowi zbiór komunikujących się obiektów przy czym obiekty dostarczające usług (serwery) są odizolowane od obiektów żądających usług (klientów) za pomocą dobrze zdefiniowanych interfejsów. W ten sposób klienci są odizolowani od implementacji usługi, która może być wykonana przy użyciu dowolnego języka i techniki programowania. Separacja interfejsów od implementacji usług jest jedną z najważniejszych cech architektury CORBA. W celu specyfikacji interfejsów obiektów został opracowany język IDL (Interface Definition Language), którego standaryzacja i odwzorowanie w istniejące języki programowania typu C, C++, Prolog i inne, jest jednym z głównych zadań OMG.
Architektura systemu CORBA została przedstawiona schematycznie na rys. 4.



Rysunek 4: Architektura systemu CORBA

Podstawowe elementy tej architektury stanowią:

- **ORB Core** - warstwa odpowiedzialna za znalezienie obiektu serwera względnie jego uruchomienie, dostarczenie do serwera żądania wykonania usługi od klienta oraz zwrotne przesłanie odpowiedzi do klienta,
- **IDL Stub oraz IDL Skeleton** - są to moduły komunikacyjne automatycznie generowane przez kompilator języka IDL i dołączane na etapie konsolidacji aplikacji odpowiednio do kodu klienta i serwera,
- **Dynamic Invocation** - moduł dynamicznego wywołania operacji serwera przez klienta. W odróżnieniu od statycznie dołączanego modułu **IDL Stub**, moduł ten pozwala na wywołanie usługi serwera, który nie był znany na etapie kompilacji kodu klienta.
- **ORB interface** - interfejs do podstawowych operacji zapewnianych przez warstwę ORB Core, który powinien być identyczny dla wszystkich implementacji CORBA,
- **Object Adaptor** - Podstawowy interfejs wykorzystywany przez obiekt w celu dostępu do usług ORB'a. Interfejs ten może być odmienny dla różnych implementacji CORBA a nawet różnych obiektów.

Prace OMG nie dotyczą tylko poziomu komunikacji pomiędzy obiektami, lecz również standaryzacji typowych usług występujących w systemach rozproszonych. Ustalenie interfejsów tych usług pozwoli bowiem na budowę aplikacji rozproszonych przez dołączenie standardowych obiektów o ustalonych cechach funkcjonalnych i sposobie dostępu do świadczonych usług. Na obecnym etapie prac ustalono specyfikacje usługi nazewniczej, trwałości obiektów (presistency) oraz obsługi zdarzeń i cyklu życia obiektów w systemie. Przedmiotem intensywnych prac są następujące dalsze usługi: replikacji, archiwizacji, kontroli współbieżności, instalacji i aktywacji systemu, licencji, wątków, czasu, bezpieczeństwa, transakcji i wiele innych. Osiągnięcie postępu w tych pracach pozwoli na budowę aplikacji rozproszonych przy użyciu standardowych obiektów systemowych.

7. Obszary zastosowań przetwarzania rozproszonego

Główne obszary zastosowań klastrów stacji roboczych obejmują między innymi: złożone obliczeniowo zagadnienia fizyki, dynamika molekularna, chemia obliczeniowa, zagadnienia modelowania oparte o metodę elementów skończonych, analiza zasobów ropy naftowej i gazu oraz symulacje sejsmiczne, dynamika przepływów i analiza rynku finansowego.

W ostatnim okresie notuje się intensywny wzrost rozproszonych aplikacji obiektowych, dotyczących zastosowań multimedialnych takich jak video konferencje, video na żądanie, interakcyjna telewizja, itp. Aplikacje te stawiają nowe wymagania w zakresie zdalnego przetwarzania i współpracy oprogramowania różnych producentów. Uzasadnia to rozwój prac nad standaryzacją środowisk programowych oraz metod osiągania ich wzajemnej współpracy (interoperability).

W aspekcie zastosowań należy podkreślić, że najistotniejszą przyczyną, decydującą o przydatności klastrów stacji roboczych do realizacji danego typu obliczeń jest możliwość wydzielenia niezależnych podzadań, tak aby stosunkowo wolna komunikacja pomiędzy procesorami klastra nie degradowała efektywności obliczeń.
Uważa się [14], że tryb wsadowy nie narzuca ograniczeń na wielkość systemu i znane są instalacje zawierające ponad 100 maszyn. W trybie równoległym jako optymalną liczbę maszyn przyjmuje się od ośmiu do szesnastu.

Większość zastosowań wykorzystuje tryb wsadowego przetwarzania. Bardziej zaawansowani użytkownicy dążą do zrównoleglenia obliczeń poprzez ich dekompozycję i koordynację z wykorzystaniem mechanizmów przesyłania wiadomości. Obliczenia w zdecydowanej większości realizowane są w języku FORTRAN i C. Tendencja ta jest wyraźnie widoczna w tabeli zamieszczonej jako dodatek do niniejszego rozdziału i stanowi genezę powstania takich pakietów, jak: PVM, TCGMSG, P4 czy Express.

Reasumując, ze względu na liczbę przetwarzanych na świecie zadań obliczeniowych dominuje proceduralne podejście do konstrukcji oprogramowania, a większość aplikacji stanowi zmodyfikowane wersje pakietów napisanych w najpopularniejszych językach sekwencyjnych. Jest to postawa prezentowana przez szerokie grono użytkowników dążących do uzyskania określonych wyników obliczeniowych jako zasadniczego celu badań. Ostatnio jednak zastosowania teleinformatyczne zaczynają tworzyć coraz szerszą gamę zupełnie nowych aplikacji rozproszonych.

Bibliografia

[1] S. Ahuja, N. Carriero, D. Gelernter, V. Krishnaswamy, *Matching Language and Hardware for Parallel Computation in the Linda Machine*, IEEE Transactions on Computers, Vol. 37, nr 8, sierpień 1988.

[2] G.R. Andrews and R.A. Olsson. *The SR Programming Language: Concurrency in Practice* Benjamin/Cummings Publishing Company, 1992.

[3] A. Beguelin et. al., A *User's Guide to PVM Parallel Virtual Machine*, TR, Oak Ridge National Laboratory, 1991.

[4] R. Bisian, A. Forin, *Architectural Support for Multilanguage Parallel Programming on Heterogeneous Systems*, Sec. Int. Conf. on ASPLOS, 1987.

[5] R. Butler, E. Lusk, *User's Guide to the P4 Programming System*, Argonne National Laboratory, 1992, TR ANL-92/17.

[6] C.H. Cap, V. Strumpen, *High Performance Platform for Parallel Computing in a Distributed Workstation Environment*, Institut fur Informatik, Univ. Zurich, 1992, TR IFI-92-07.

[7] D.L. Cohn, W.P. Delaney, M. Tracey, *Platform for Heterogeneous Distributed Operating Systems*, Proc. of the Usenix Distributed and Multiprocessor Systems Workshop, 1989.

[8] G.C. Douglas, T. G. Mattson, M.H. Schultz, *Parallel Programming Systems for Workstation Clusters*, Yale Univ. Dept. of Comp. SC. RR YALEU/DCS/TR-975, August, 1993.

[9] I. Foster et. al. *Productive Parallel Programming: The PCN Approach*, Scientific Programming, Vol 1, 1992, p.51 - 66.

[10] M. Gajęcki, K. Jędrzejek, K. Zieliński, *Parallelization of the DBW Render Package on a Transputer Farm*, Machine GRAPHICS & VISION, Vol.1, No.4, 1993.

[11] T. Geigel, M. Pagan, *A Distributed Application for the Pharros Project*, Proc. Int. Conf. on Parallel Proc. 1988.

[12] D. Gelernter, J. Philbin, *Spending your Free Time*, Byte, maj 1990.

[13] A.S. Grimshaw *Easy-to-use Object-Oriented Parallel Processing with Mentat*, Univ. of Virginia, 1992,TR No. CS-92 32.

[14 HP Apollo 9000 Series 700, Computational Cluster Guide, Jaunary 1993.

[15] A.H. Karp, H.P. Flatt, *Measuring Parallel Processor Performance*, Communication of the ACM, May 1990 Vol.33, Number 5.

[16] J. Kobielus, *LANs Bid for Distributed Power*, Network World, Dec. 3, 1990.

[17] K.L Lorh, L. Nentwig, J. Muller, DAPHNE: *Support for Distributed Computing in Heterogeneous Environment, European Workshop Proceedings, Progress in Distributed Operating Systems and Distributed Management*, Springer-Verlag, 1989.

[18] C.M. Pancake, D. Bergmark, *Parallel Programming Languages: Do They Meet the Needs of Scientists and Engineers?*, Cornell Theory Center, CTC89TR18.

[19] Strand Software Technologies Limited: *STRAND'88 User Manual*, Buckingham Release, June 1990.

[20] *Solaris Strategy*, SunSoft TR 6/21/12

[21] X. Sun, J.L. Gustafson, *Toward a Better Parallel Performance Metric*, Parallel Computing, Vol.17, Dec. 1991.

[22] X. Sun, L.Ni, Another View on Parallel Speedup, Proc. of Supercomputing'90.

[23] *The Charm 3.2 Programming Language Manual*, Dec. 18, 1992, TR, Univ. Of Illinois at Urbana Champaign.

[24] M. Sullivan, D. Anderson, Marionette: *A System for Parallel Distributed Programming Using a Master/Slave Model*, Proc. of the Ninth International Conference on Distributed Computer Systems, 1989.

[25] C.A. Waldspurger, T. Hogg, B.A. Huberman, Spawn: *A Distributed Computational Economy*, IEEE Transaction on Software Engineering, 1992, February, Vol.18, No.2.

[26] C.B. Weinstock, *Performance and Reliability Enhancement of the Durra RuntimeEnvironment*, SEI at CMU Univ., February 1989, CMU/SEI-89-TR-8.

[27] L. Wittie, C. Maples, Merlin: *Massively Parallel Heterogeneous Computing*, Proceedings of the Int. Conference on Parallel Processing, 1989.

[28] S.A. Yemini, G.S. Goldszmidt, A.D. Stoyenko, Yi-Hsiu Wei, *CONCERT: A High-Level-Language Approach to Heterogeneous Distributed Systems*, Proc. of the Ninth International Conference on Distributed Computer Systems, 1989.

[29] K. Zieliński, G. Czajkowski, M. Gajęcki, *Parallel Programming Systems for LAN Distributed Computing*, Proceednigs of the 14th International Conference on Distributed Computer Systems, IEEE CS Press, 1994.

[30] *Draft Common Object Request Broker Architecture* Revision 1.1. OMG Report 91-12-1, OMG Inc., (1991).

Grzegorz Czajkowski
Cornell University, Ithaca

Zagadnienia rozproszonego programowania w logice

1. Wstęp

Postęp technologiczny oraz malejące ceny komputerów sprawiły, że w ciągu ostatnich kilku lat coraz popularniejsze stają się stacje robocze połączone siecią lokalną, która z kolei wchodzi w skład sieci zdalnych. Dostępność i bardzo wysoka jakość sprzętu nie jest jednakże równoważna istnieniu ogólnie zaakceptowanych i pod różnymi względami optymalnych języków i środowisk programowania rozproszonego. Mnogość oraz różnorodność rozwiązań sprawiają, że nawet eksperci nie są w stanie wymienić chociażby nazw wszystkich istniejących języków komputerowych, nie mówiąc już o ich cechach. Sytuacja ta na pewno nie zmieni się w przyszłości (niech świadczy o tym dość prowokacyjny tytuł jednego z artykułów w prasie fachowej: *The next 700 programming languages*). Niniejsza praca w pewnym sensie pogłębia istniejący bałagan, gdyż jej zasadniczą część stanowi prezentacja języka zdefiniowanego i zaimplementowanego przez autora. Język[1] oparty na modelu obliczeniowym równoległego programowania w logice oraz rozszerzenie go o nowe mechanizmy, takie jak operatory sekwencji, śledzenie systemowe czy współpraca z innymi systemami.
Bezami pracy są następujące stwierdzenia:

- równoległe programowanie w logice, a w szczególności w prezentowanym języku, pozwala na szybkie tworzenie złożonych aplikacji rozproszonych z wykorzystaniem bogatego repertuaru bardzo interesujących technik programistycznych,
- wprowadzone istotne rozszerzenia przystają do bazowego modelu równoległego programowania w logice, znacznie ułatwiając programowanie oraz umożliwiają tworzenia aplikacji hybrydowych, tzn. napisanych w różnych językach,
- możliwa jest bardzo efektywna implementacja systemu wykonującego programy w prezentowanym języku.

Spróbowano dokonać syntezy rezultatów otrzymanych w wyniku rozwoju języków równoległego programowania w logice na przestrzeni ostatniej dekady. Model obliczeń oraz składnia wzorowane są na pochodzących z drugiej połowy lat osiemdziesiątych systemach *Concurrent Prolo* oraz *Strand*. Jednakże nowe (lata 1990-94), niezmiernie ciekawe rezultaty japońskiego *Projektu Komputerowych Systemów Piątej Generacji*, wyniki pracy grup w Weizmann Institute w Izraelu, Imperial College w Londynie oraz California Institute of Technology pozwoliły na wykorzystanie bardzo efektywnych metod implementacyjnych oraz na wzbogacenie dostępnego zbioru technik programowania.

Prezentowany język i system rozproszonego programowania w logice mają wiele zalet. Model obliczeń, choć prosty i spójny, pozwala jednak na projektowanie bardzo skomplikowanych rozproszonych aplikacji o różnorodnych protokołach komunikacji między procesami. Używanie deklaratywnych języków wysokiego poziomu zmniejsza znacznie czas tworzenia aplikacji, redukując jednocześnie możliwości

[1] Nie zostanie wprowadzona nazwa dla stworzonego języka -- będzie on dalej jednoznacznie identyfikowany jako prezentowany język, opisywany język albo, o ile nie będzie to prowadziło do nieporozumień, po prostu jako język.

programistycznych pomyłek. Wewnętrzna współbieżność pozwala na traktowanie procesów jako wątków, które na pewno będą w przyszłości jedną z podstawowych metod programowania. Możliwość metaprogramowania oraz tworzenia dowolnych topologii połączeń między procesami pozwala na projektowanie ram komunikacyjnych. Dynamiczne tworzenie procesów oraz ich kontrolowane migrowanie umożliwia efektywne wykorzystanie rozproszonych zasobów obliczeniowych poprzez równoważenie obliczeń. Podział programów na moduły pozwala na wielokrotne wykorzystanie kodu oraz uniezależnienie implementacji procedury od jej intefrejsu, co jest jedną z podstaw programowania obiektowego. Interesujące są też operatory synchronizacji oraz śledzenie wykonywania danego programu przez ten sam program.

Do oryginalnego wkładu badawczego autora można zaliczyć projekt i implementację rozbudowanych operatorów sekwencji, zapewnienie wygodniejszej niż w dostępnych systemach notacji dla metaprogramowania, co znajduje odbicie w prostocie tworzenia ram, zapewnienie możliwości współpracy programów deklaratywnych z innymi systemami rozproszonego programowania.

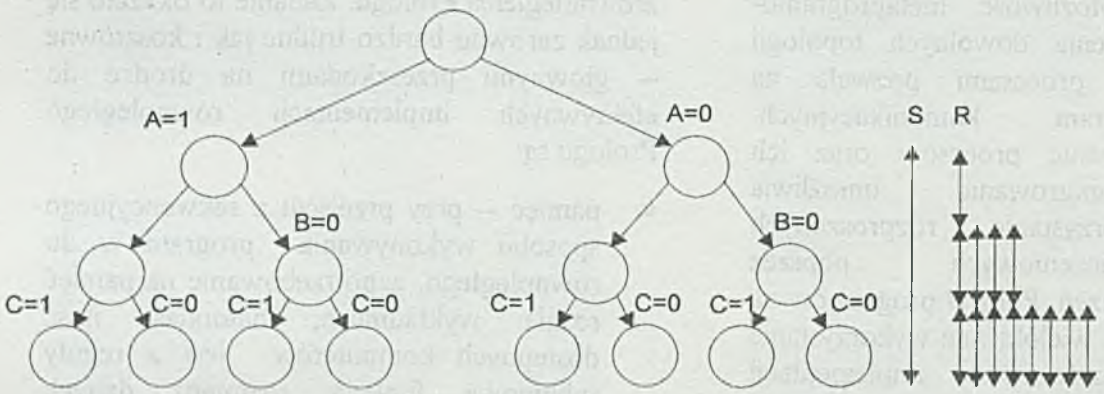
Układ artykułu jest następujący: po opisaniu ogólnego modelu współbieżnego programowania w logice zaprezentowane zostaną wybrane zagadnienia: tworzenie ram komunikacyjnych, programowanie rozproszone, topologie wirtualne połączeń między procesami, operatory sekwencji, współpraca między procesami. Artykuł podsumowany jest eksperymentalnymi danymi dotyczącymi efektywności oraz uwagami o kompilatorze i implementacji systemu. U czytelnika zakłada się podstawową wiedzę na temat programowania opartego o logikę.

2. Ogólny model obliczeniowy

Zainteresowanie równoległym programowaniem w logice wiąże się z próbami zrównoleglenia Prologu. Zadanie to okazało się jednak zarówno bardzo trudne jak i kosztowne -- głównymi przeszkodami na drodze do efektywnych implementacji równoległego Prologu są:

- pamięć -- przy przejściu z sekwencyjnego sposobu wykonywania program w do równoległego, zapotrzebowanie na pamięć rośnie wykładniczo, natomiast ilość dostępnych komputerów jest z reguły subliniową funkcją rozmiaru danych problemu. Rysunek 1 pokazuje pewien prosty program w Prologu, jego drzewo poszukiwania rozwiązań oraz wymagania pamięciowe w wykonaniu sekwencyjnym (S) i równoległym (R),
- kontekst -- różne etapy obliczania mogą być na różnych poziomach drzewa obliczeń i mieć różne wartości tych samych zmiennych,
- cięcie ("!") oraz niepowodzenie indukowane użyciem **fail** -- chcąc zdefiniować ich znaczenie identycznie jak w modelu sekwencyjnym, należałoby zapewnić odpowiednie możliwości systemowego przerywania nieukończonych etapów obliczeń,
- kolejność znajdowania rozwiązań -- jeżeli jako wynik programu sekwencyjnego zdefiniujemy nie tylko ilość i wartość rozwiązań, lecz także porządek ich znajdowania, w modelu równoległym należałoby wprowadzić kontrolę kolejności wypisywania rezultatów (to samo dotyczy **findall** i pokrewnych predykatów), co ogranicza paralelizm,
- brak ogólnych heurystyk pozwalających na efektywne rozpraszanie obliczeń (dobór optymalnej ziarnistości)

```
trzy_bity([A,B,C]) :- bit(A), bit(B), bit(C).
bit(1).
bit(0).
```



Rysunek 1 Schemat wykonania programu w Prologu.

W wyniku prac związanych z Projektem Systemów Komputerowych Piątej Generacji [1] zdefiniowano ogólny model obliczeniowy *równoległego programowania w logice, z dokonanym wyborem (parallel logic programming commited choice languages)*, różniący się od modelu Prologu, lecz umożliwiający efektywne równoległe i rozproszone implementacje. W ciągu kilku lat trwania badań wyszczególniono ponad dziesięć *podmodeli* ogólnego modelu.

Najważniejsze pojęcia dotyczące definiowania procesów i wykonywania programów w językach równoległego programowania w logice można przedstawić następująco. Dynamiczny zbiór procesów stanowi zawartość puli procesów. W wyniku redukcji procesu z jedną z definiujących go reguł mogą powstać nowe procesy. Możliwość dopasowania argumentów procesu i nagłówka reguły jest warunkiem koniecznym dla sukcesu próby redukcji. Wybór procesu z puli należy traktować jako niedeterministyczny.

Podstawową cechą komunikacji jest jednolite traktowanie kanałów komunikacyjnych jako zmiennych. Pozwala to na przekazywanie strumieni między procesami, rezydującymi zarówno na różnych Odmiennie niż w tak znanych językach jak Occam [5], proces wysyłający komunikaty nie musi znać nazwy ani położenia odbiorcy. Pozwala to na w pełni dynamiczną konfigurowalność topologii połączeń. Strumienie to pełnoprawne dane, a zatem do ich tworzenia można wykorzystywać wszystkie techniki budowania struktur. Ponadto kanał komunikacyjny zawiera wszystkie wysłane nim wiadomości. Procesy mają więc dostęp do zapisu historii obliczeń, co może być wykorzystane przy śledzeniu programów.

3. Ramy komunikacyjne

Bardzo ciekawym i nowym zagadnieniem jest tworzenie ram *(frameworks)* komunikacyjnych dla procesów. Zadanie to składa się z dwóch części: konstrukcji topologii połączeń między procesami oraz parametryzacji obliczeń nazwą procedury. Otrzymujemy w ten sposób niezależność implementacji obliczeń od połączeń między poszczególnymi procesami[4, 6], zwiększony stopień możliwości ponownego użycia tego samego kodu oraz modularność projektowania aplikacji. Przykładowo, jeżeli wiemy, że obliczenie szybkiej transformaty Fouriera można przedstawić jako statyczne drzewo binarne, można napisać program, którego argumentem jest nazwa procesu obliczającego transformatę oraz głębokość drzewa; w każdym jego węźle zostanie uruchomiony proces, mogący się komunikować z bezpośrednim przodkiem oraz swoimi potomkami. Ewentualne zmiany w algorytmie obliczania transformaty nie wpłyną na topologię — tak więc tworzenie sieci połączeń

odseparowano od specyfikacji obliczeń. Stworzony program można wykorzystywać do dowolnego problemu wymagającego ustawienia procesów w pełne, statyczne drzewo binarne. Definicja procesu tworzącego ramę może być następująca:

btree(Proc, In, Out, 0) :- V := { Proc, In, Out, _, _, _, _ }, V.

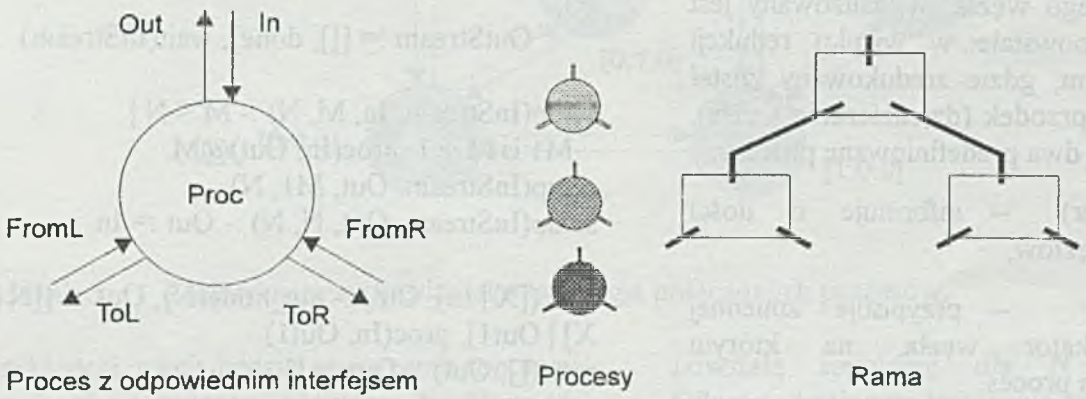
btree(Proc, In, Out, D) :- D > 0 |

D1 is D - 1, V := { Proc, In, Out, FromL, ToL, FromR, ToR }, V,

btree(Proc, ToL, FromL, D1), btree(Proc, ToR, FromR, D1).

gdzie **In, Out, ToL, FromL, ToR, FromR** są strumieniami komunikacyjnymi procesu **Proc** z przodkiem oraz potomkami w drzewie (rys. 2a). Zależnie od potrzeb, powyższy program można zmodyfikować tak, żeby przekazywał procesowi jego poziom w drzewie lub też zamykał strumienie do i od potomków dla procesów-liści (pierwsza reguła). Procesy traktowane są jako klocki wkładane w ramę, czyli szkielet komunikacyjny (w trakcie wykonywania programu topologia połączeń może się zmienić). Wykorzystanie ram daje programiście dodatkowy stopień swobody, uniezależniając tworzenie *klocków* od komunikacji. Jedynym wymogiem jest, aby proces można było wkomponować w ramę -- to znaczy, żeby posiadał odpowiednią ilość strumieni wejściowych i wyjściowych oraz żeby używał identycznego protokołu (rodzajów komunikatów) jak inne procesy współtworzące ramę (rys. 2b -- procesy mogą w różny sposób realizować swe funkcje, lecz muszą *eksportować* ten sam interfejs komunikacyjny). Zagadnienie to bliskie jest związanemu z programowaniem obiektowym problemowi zgodności interfejsów.

(a) (b)



Rysunek 2 Schemat ramy wraz z procesami.

W połączeniu z opisanymi dalej mechanizmami rozpraszania obliczeń na sieci komputerowej, wykorzystanie ram do tworzenia szkieletów połączeń między procesami staje się bardzo mocnym i atrakcyjnym narzędziem. Ramy znacznie ułatwiają tworzenie i konserwację oprogramowania oraz oddzielają w tekście programu logicznie niezależne etapy: rozmieszczane procesów i właściwe obliczenia.

4. Programowanie rozproszone

Z punktu widzenia teorii systemów operacyjnych, procesy dotychczas zaprezentowanych programów można traktować jako wątki [7]. Wykonują się one współbieżnie w tej samej przestrzeni adresowej, są bardzo lekkie (tzn. koszty ich tworzenia i przełączania są znikomo małe w stosunku do tworzenia i przełączania procesów w systemie operacyjnym), dostępne są mechanizmy

komunikacji i synchronizacji. Poniżej przedstawione zostaną sposoby programowania z użyciem wielu komputerów. *Wątkopodobne* cechy procesu zostają więc poszerzone o możliwości migracji procesów między maszynami.

Przez *węzeł* będziemy rozumieli proces systemu operacyjnego, wykonujący program zarządzania pulą procesów. Zbiór mogących komunikować się węzłów będziemy nazywali *maszyną wirtualną*. Węzły mogą znajdować się na różnych komputerach lub, w przypadku maszyny wieloprocesorowej, na jej różnych procesorach. Na danym komputerze może znajdować się więcej niż jeden węzeł. Każdy węzeł posiada unikalny identyfikator, będący liczbą całkowitą nieujemną. Węzeł, na którym rozpoczynane są obliczenia, jest węzłem zerowym.

Jedynym dostępnym dla użytkownika mechanizmem fizycznej alokacji procesów jest umieszczenie w tekście programu adnotacji **P @ Id** przy wybranym procesie P[2] gdzie **Id** jest identyfikatorem węzła na którym ma wykonywać się dany proces. Jeżeli **Id** nie określa istniejącego węzła, sygnalizowany jest błąd. Procesy powstałe w wyniku redukcji wykonują się tam, gdzie zredukowany został ich bezpośredni przodek (dziedziczenie węzła). Udostępnione są dwa predefiniowane procesy:

- nodes(Number) -- informuje o ilości wszystkich węzłów,
- my_node(Id) -- przypisuje zmiennej **Id** identyfikator węzła, na którym wywołano ten proces.

Zdecydowano nie dostarczać bezpośrednich mechanizmów pozwalających na identyfikację nazwy komputera na którym znajduje się dany węzeł. Powodem takiej decyzji jest preferowane przez autora pracy podejście do programowania w logice -- uwaga powinna być zwracana na postać algorytmu, interakcją między jego komponentami oraz odzwierciedlenie natury problemu w logicznej strukturze sieci współpracujących procesów. Korzystanie w programach z takich informacji jak sieciowe adresy komputerów nie byłoby naszym zdaniem zgodne z obrazem prezentowanego systemu jako języka wysokiego poziomu. Informacja taka nie jest potrzebna do efektywnego osiągnięcia najważniejszego celu programowania równoległego, jakim jest maksymalne przyspieszenie obliczeń w stosunku do sekwencyjnego wykonywania programów, gdyż nie da się przewidzieć obciążenia danego komputera.

Wprowadzającym przykładem niech będzie rozmieszczenie po jednym procesie na każdym z węzłów. Procesy połączone są w pierścień -- strumień wyjściowy ostatniego z nich jest strumieniem wejściowym pierwszego. Pierwszy proces wysyła listę zawierająca jego własny numer (będzie to zero). Każdy proces otrzymujący wiadomość dodaje do niej identyfikator swojego węzła.

```
go :- nodes(N), setup(InStream, OutStream, 1,
N),
      OutStream := [[], done], wait(InStream).

setup(InStream, In, M, N) :- M < N |
  M1 is M + 1, proc(In, Out)@M,
setup(InStream, Out, M1, N).
setup(InStream, Out, N, N) :- Out := In.

proc([X | In], Out) :- my_node(N), Out := [[N |
X] | Out1], proc(In, Out1).
proc([], Out) :- Out := [].
```

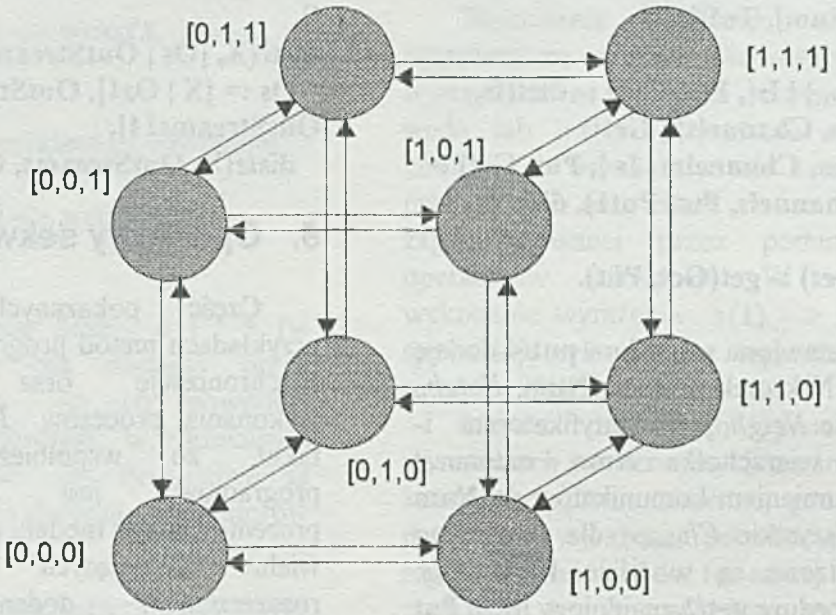
Bardzo ważne jest spostrzeżenie, że każdy poprawnie napisany program równoległy może wykonywać się na jednej maszynie (wystarczy opuścić adnotacje dotyczące alokacji) Prawdziwe jest też twierdzenie odwrotne rozproszenie obliczeń nie wpływa na poprawność programu. Inaczej mówiąc, współbieżność wykonywania obliczeń jest podstawową cechą prezentowanego modelu. Wbrew pozorom powyższe zdania nie są stwierdzeniami trywialnymi. Wśród bardzo popularnych języków przetwarzania rozproszonego istnieją na przykład takie, w

[2] Nie są nakładane żadne ograniczenia na wartość P. W szczególności, P może być zmienną, mieć postać `ModuleName:Process`, czy też być poleceniem wykonania procesu na jeszcze innej maszynie: `P1@Id1` (można w ten sposób dynamicznie rozmieszczać procesy korzystając ze znanych z języka Logo technik *grafiki żółwiowej.*)

których współbieżność wewnętrzna oraz wykonywanie równoległe wykluczają się wzajemnie (np. SCA Linda [10]).

4.1 Tworzenie topologii połączeń między procesami

W przedstawionych dotychczas przykładach zasygnalizowano problem tworzenia sieci procesów o topologii połączeń odpowiedniej dla struktury problemu. Topologią taką jest na przykład gwiazda dla modelu *farmer-worker*, czy drzewo binarne dla szybkiej transformaty Fouriera. Do tworzenia o wiele bardziej interesujących i skomplikowanych topologii niż wspomniane wystarczy rozwiązanie tylko dwóch problemów: przekazywania między procesami zmiennych reprezentujących strumienie oraz reprezentacji unikalnego identyfikatora procesu. Pierwszy problem rozwiązywany jest z reguły przez zastosowanie procesu-tablicy, do którego procesy wysyłają swoje strumienie wejściowe oraz żądania podania strumieni do procesów sąsiednich w danej topologii. Postać identyfikatorów jest zależna od charakteru problemu.



Rysunek 3. Trójwymiarowa kostka jako topologia połączonych procesów.

Przykładem niech będzie utworzenie topologii hiperkostki o zadanym wymiarze **N**. Algorytm tworzenia opisanej topologii opiera się na wykorzystaniu procesu-tablicy **dict**, z którym komunikują się pozostałe procesy. Proces **h/4** tworzy 2^N procesów **process/4**, reprezentujących wierzchołki hiperkostki, identyfikowanych **N**-elementowymi listami zawierającymi współrzędne wierzchołka. Każdy proces posiada listę **N** strumieni wejściowych oraz **N** wyjściowych, łączących go z sąsiadami. Sąsiadem procesu o identyfikatorze **Id** jest każdy proces identyfikowany jako **Id1**, taki że **Id** oraz **Id1** różnią się dokładnie na jednej pozycji. Rysunek 3 przedstawia powstałą topologię dla **N** równego 3. Odpowiedzialne za tworzenie topologii są dwa procesy: **h/4** i **dict/3**. Pierwszy z nich tworzy odpowiednią ilość procesów **process**, natomiast drugi przechowuje strumienie wejściowe do tych procesów oraz żądania znalezienia wszystkich strumieni wyjściowych z danego procesu (strumień wejściowy procesu jest strumieniem wyjściowym dokładnie jednego z jego sąsiadów). Po otrzymaniu wszystkich komunikatów następuje realizacja żądań. Znajdowanie odpowiednich strumieni możliwe jest dzięki dołączeniu do każdego komunikatu identyfikatora wysyłającego procesu.

```
hc(N) :- merger:merger(ToDict,
ToDictReqs),
  dict(ToDictReqs, [], []), h(N, N, [],
ToDict).

h(0, Dim, MyNum, ToDict) :-
  ToDict := [ put(MyNum, InStreams),
get(MyNum, OutStreams) ],
  process(Dim, MyNum, InStreams,
OutStreams).
h(N, Dim, MyNum, ToDict) :- N > 0 |
  ToDict := [ merge(ToDict1),
merge(ToDict2) ], N1 is N - 1,
  h(N1, Dim, [0 | MyNum], ToDict1), h(N1,
Dim, [1 | MyNum], ToDict2).

dict([get(Num, ) | Is], Put, Get) :- dict(Is,
Put, [get(Num, Channels) | Get]).
dict([ put(Num, Channels) | Is ], Put, Get) :-
  put(Num, Channels, Put, Put1), dict(Is,
Put1, Get).
dict([], Put, Get) :- get(Get, Put).
```

Nie przedstawiona procedura **put/4** dodaje do listy **Put** N krotek postaci {**Num**, $Neigh_i$, $Ch_{Num,i}$ }, gdzie $Neigh_i$ jest identyfikatorem **i**-tego sąsiada wierzchołka **Num**, natomiast $Ch_{Num,i}$ jest strumieniem komunikatów do **Num** od $Neigh_i$. Wszystkie $Ch_{Num,*}$, dla ustalonego Num, gromadzone są w liście **InStreams**. Podobnie, procedura **get/2** znajduje w liście **Put** wszystkie strumienie komunikatów od wierzchołka **Num** do jego sąsiadów i zbiera je w liście **OutStreams**.

Zarządzanie listami strumieni **InStreams** oraz **OutStreams** (mają one długość równą wymiarowi kostki) może zostać uproszczone przez zastąpienie ich pojedynczymi listami, odpowiednio łączonymi lub rozdzielanymi. Na przykład **InStreams** można zastąpić strumieniem **In** oraz wywołaniem procesów **multiple_merge(InStreams, InS)**:

```
multiple_merge([], Os) :- Os := [].
multiple_merge([I|Is], Os) :- Os :=
[merge(I)|Os1], multiple_merge(Is, Os1).
```

Otrzymujemy w ten sposób mulipleksowanie komunikatów od sąsiadów. Protokół rozgłoszeniowy do sąsiadów można zrealizować zastępując **OutStreams** pojedynczym strumieniem **Out** oraz wywołaniem zdefiniowanej poniżej procedury **distribute(Out, OutStreams)**. Proces **distr/3** umieszcza dany komunikat w każdym strumieniu wyjściowym oraz tworzy listę **OutStreams1** strumieni do rozgłaszania innych komunikatów.

```
distributor([X | Out], OutStreams) :-
  distr(X, OutStreams, OutStreams1),
  distributor(Out, OutStreams1).

distr(_, [], OutStreams1) :- OutStreams1 :=
[].
distr(X, [Os | OutStreams], OutStreams1) :-
  Os := [X | Os1], OutStreams1 := [Os1 |
OutStreams11],
  distr(X, OutStreams, OutStreams1).
```

5. Operatory sekwencji

Część pokazanych w poprzednich przykładach metod programowania ma na celu synchronizację oraz sekwencjonowanie wykonania procesów. Jest to konsekwencją faktu, że współbieżność wykonywania programów jest pierwotną cechą prezentowanego modelu a nie, jak w przypadku wielu istniejących środowisk [10], rozszerzeniem dodanym do języka sekwencyjnego. Zaprezentowane metody synchronizacji i sekwencjonowania wystarczają do rozwiązywania różnorodnych zagadnień programowania rozproszonego nie naruszając modelu obliczeniowego (komunikacja odbywa się jedynie przez zmienne dzielone z zachowaniem zasady jednokrotnego przypisania). Podejście takie jest eleganckie, jednak w niektórych przypadkach zapewnienie pożądanej kolejności wykonania może być pracochłonne. Na przykład wykrywanie zakończenia wykonywania obliczeń zdefiniowanych przez program o kilkuset regułach może wymagać wielu godzin pracy programisty. Dlatego też zdecydowaliśmy się na wprowadzenie *operatorów sekwencji*. Powinny być one wykorzystywane tylko wtedy, gdy jest to niezbędne, to znaczy gdy zapewnienie odpowiedniego porządku wykonania

"klasycznymi" metodami jest zbyt kosztowne lub pracochłonne [4].
Możliwe jest użycie dwóch operatorów sekwencji: **-->** i **<--**. Są one lewostronnie łączne, o identycznym priorytecie, słabszym od operatorów arytmetycznych lecz mocniejszym od przecinka. Ich argumentami mogą być nagłówki procesów, zmienne procesowe, operacje przypisania i obliczenia wartości oraz listy tych argumentów. Ponieważ oba operatory są lewostronnie łączne (podobnie jak +), można tworzyć złożone **wyrażenia sekwencji**. Dla przykładu rozważmy następujące wyrażenie:

[write(1,_),write(2,_)] --> write(3,_) <-- [write(4,_),write(5,_)]

Jest ono równoważne wyrażeniu z nawiasami

([write(1,_),write(2,_)] --> write(3,_)) <-- [write(4,_),write(5,_)]

i oznacza, że wypisanie liczb **4** i **5** (w dowolnym porządku) ma nastąpić przed wykonaniem operatora **-->**, co spowoduje wypisanie **1** i **2** (również w dowolnym porządku) a następnie **3**. Prawdziwe są następujące stwierdzenia dotyczące operatorów sekwencji:

- **L --> R** oznacza, że kompletne wykonanie wszystkich procesów powstałych w związku z **L** nastąpi przed rozpoczęciem wykonywania **R**,
- **L --> R** jest równoważne **R <-- L**[3],
- jeżeli argumentem operatora sekwencji jest lista $[A_1, .., A_n]$, to wykonanie procesów $A_1, .., A_n$ jest współbieżne, czyli równoważne redukcji pewnego procesu $q z regułą $q :- $A_1, .., A_n$,
- procesy nie związane wyrażeniami sekwencji wykonują się współbieżnie z procesami sekwencjonowanymi,
- z powodu statycznej analizy i transformacji wyrażeń sekwencji, nie mogą one stawać się wartościami zmiennych procesowych, nie można zatem ich stosować w metaprogramownaiu.

W ciele reguły może występować więcej niż jedno wyrażenie sekwencji. Mówimy, że wyrażenia te są rozłączne, gdy nie występują w nich zmienne procesowe lub gdy zmienne te mają różne nazwy. Na przykład trzy wyrażenia w poniższej regule są niezależne:

p :- a --> b --> c, c <-- x, x <-- d --> d.

Wykonania niezależnych wyrażeń mogą przeplatać się w czasie. Zauważmy, że **N**-krotne wystąpienie nazwy tego samego procesu w wielu lub jednym wyrażeniu nie powoduje konfliktu -- uruchomionych zostanie **N** kopii procesu, oczywiście w kolejności zagwarantowanej przez podane własności operatorów sekwencji. W szczególności, wykonanie wyrażenia **x(1) --> x(1) --> x(1)** spowoduje sekwencyjne wykonanie trzy razy procesu **x(1).**

Użycie zmiennych procesowych w wyrażeniach sekwencyjnych ma inne znaczenie: każde wystąpienie zmiennej procesowej **P** w dowolnym wyrażeniu sekwencyjnym w ciele reguły odnosi się do tej samej zmiennej, co powoduje, że różne wyrażenia mogą stawać się zależne. Ponadto, błędy programisty mogą być przyczyną powstawania cykli, co uniemożliwia wykonanie wyrażenia operatorowego.[4] Na przykład V --> V jest cyklem. Przykładem bardziej skomplikowanego cyklu jest powstały w poniższej regule:

q(A, B, D, E, F) :-

[A, B, p(1)] --> [D, E], E --> F, F --> [r([]), B].

[3] Zatem jeden z operatorów jest zbędny, lecz wprowadzono go dla wygody, podobnie jak minus binarny w arytmetyce, gdzie odejmowanie można zastąpić dodaniem wartości pomnożonej przez -1.

[4] Cykle wykrywa kompilator. Algorytm wykrywania oparty jest na sortowaniu topologicznym [8]. Z wszystkich występujących w regule wyrażeń sekwencji tworzone jest jedno, reprezentujące acykliczny graf zamierzonej kolejności wykonania. Niemożność stworzenia takiego grafu tożsama jest z wystąpieniem cyklu. Wspomaganie wykrywania takich zapętleń przez kompilator jest dużą zaletą implementacji operatorów sekwencji.

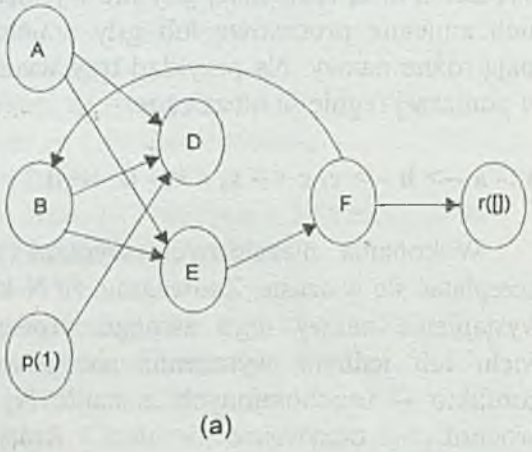
co pokazuje rysunek 4a, natomiast kolejność nałożoną na procesy w regule:

q(B, C, D, E, F, G, H) :- I := writeln(done,_),

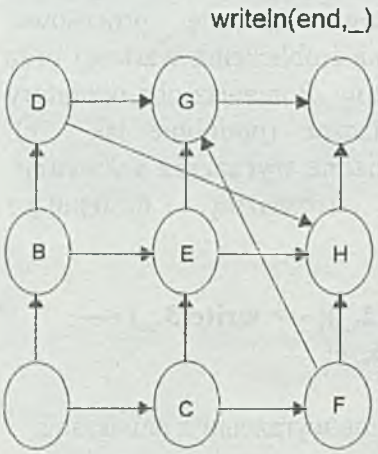
writeln(done,_) --> [B, C] --> [D, E, F] --> [G, H] --> writeln(end,_),

H <-- D, G <-- E.

przedstawia rysunek 4b.



(a)



(b)

Rysunek 4 Porządek nałożony na procesy przez operatory sekwencji.

6. Współpraca z innymi systemami

Każda platforma programistyczna posiada swoje własne cechy, wyróżniające ją spośród coraz większej liczby innych środowisk. Współpraca wielu systemów jest bardzo atrakcyjna: można osiągnąć mniejszy rozmiar ziarna paralelizmu *(task grain)* niż w danym systemie, jeżeli tylko inny to umożliwia. Wykorzystanie istniejącego kodu staje się wówczas natychmiastowe i naturalne. Specjalistyczne części obliczeń (symboliczne, funkcjonalne, numeryczne) mogą być wykonywane przez odpowiednio dobrane języki. W systemach rozproszonych odpowiednią strukturalizacją współpracy środowisk wydaje się schemat klienta-serwera. W przyjętym przez autora rozwiązaniu [2] nacisk kładziony jest na to, aby każde środowisko[5] wywoływało procedury udostępniane przez inne środowiska, wykorzystując swoje własne dostępne mechanizmy -- np. dla systemów obiektowych będzie to wywołanie metody a dla deklaratywnych komunikacja oparta o strumienie. Inaczej mówiąc, składnia języka pozostaje bez zmian. Oczywiście, dołączenie systemu do zbioru współpracujących środowisk wymaga przedtem gruntownego zapoznania się ze szczegółami implementacji i może wymagać nieraz wielu pomysłowych *sztuczek* programistycznych.

Prezentowany system staje się jednym ze współpracujących środowisk poprzez wywołanie procesu **ia:system_ia_start(From, To, Done)**[6] **.** Pierwszy i trzeci argument muszą być zmiennymi w chwili wywołania, a ich znaczenie jest następujące:

[5] Każde środowisko należy rozumieć jako *każde zmodyfikowane przez nas środowisko*. Implementacyjny opis rozszerzeń różnych systemów można znaleźć w cytowanej literaturze. W chwili obecnej do zbioru współpracujących środowisk należą: ANSA, ORBIX (implementacja standardu CORBA), Strand, PCN. Środowiska te zostały opisane m.in. w[3]

[6] Skrót **IA** pochodzi od *Interope·rability Architecture*, angielskiej nazwy dla zbioru współpracujących środowisk[9].

- **Done** -- ma wartość listy pustej, gdy operacja zakończyła się sukcesem. W przeciwnym wypadku zawiera komunikat o błędzie,
- **To** -- strumień żądań do innych środowisk. Przykładowo, **To := [ackermann(M, N, A) | To1]** spowoduje skierowanie żądania obliczenia liczby Ackermanna *(M,N)* i zwrócenia rezultatu w zmiennej **A**. Istotne jest, że nie jest potrzebna informacja, które środowisko udostępnia serwis liczenia liczb Ackermanna, natomiast ważne jest, aby istniał taki serwis. Lokalizacja odpowiednich procedur udostępnianych przez współpracujące środowiska dokonywana jest dzięki *rejestracji*-- np. twórca hybrydowej aplikacji chcąc udostępnić serwis symbolicznego różniczkowania innym środowiskom, umieszcza w strumieniu **To** komunikat **register(symb_diff/2),**
- **From** -- strumień żądań nadchodzących od innych środowisk.

Implementacja zapewnia poprawne dostarczenie rezultatu klientowi zarówno gdy serwerem jest opisywany język jak i gdy serwerem jest inne rozproszone środowisko. Dzięki **ia:system_ia_start/3** komunikowanie się z innymi środowiskami może być oparte o wszystkie dostępne techniki – w szczególności **To** może być wynikiem łączenia strumieni, a **From** można przekazać jako wejście dla dystrybutora. Użycie protokołu niekompletnych komunikatów oraz techniki procesów-tablic pozwoli na tworzenie serwerów ze stanem (*stateful servers* [7]).

7. Efektywność

Opisane metody programowania pozwalają szybko tworzyć skomplikowane aplikacje, zarówno rozproszone jak i wykonujące się na jednym komputerze. Łatwość programowania i elegancja rozwiązań są istotnymi cechami, lecz z punktu widzenia inżynierii oprogramowania najważniejsza[7] jest efektywność. Niniejszy rozdział zawiera eksperymentalne dane dotyczące szybkości przetwarzania w prezentowanym języku.

Pierwsze wyniki dotyczyć będą wykonywania programów na maszynie wirtualnej składającej się z jednego węzła. Czasy wykonania programów porównane są w tab. 1 z najszybszą dostępną komercyjnie w chwili pisania pracy implementacją języka współbieżnego programowania w logice -- Strand88.

Program	Czas obliczeń [ms]	
	prezentowany język	Strand
9-queens (dziewięć hetmanów)	15006	11320
qsort (sortowanie szybkie)	6512	6378
avl (tworzenie drzewa AVL)	11125	10398
tokenize (tokenizacja)	651	638
compile (kompilacja)	5027	5030
assemble (asemblacja)	491	469

Tabela 1 Wyniki pomiarów szybkości wykonywania programów

[7] Obok poprawności, którą łatwiej jest weryfikować w językach opartych o logikę niż np. w językach z efektami ubocznymi.

Analiza podanych wyników nasuwa kilka wniosków:

- dla programów, w których występuje dużo operacji numerycznych np. **(9-queens)** Strand jest o ok. 30% szybszy od naszego języka. Jest to konsekwencją przyjętej zasady nieuprzywilejowywania operacji arytmetycznych w języku. Dzięki temu np. krotka **(1 + 2) • 3** może zostać stworzona przez program jako wartość zmiennej **T** a następnie obliczona przez **E is T**. W Strandzie jest to niemożliwe -- zmienne występujące po prawej stronie operatora **is** mogą być ukonkretniane tylko do liczb. Podjęcie opisanej decyzji jest motywowane chęcią zapewnienia jednolitego modelu języka oraz zapewnienia jego pełnej symboliczności. Rozszerzenia pozwalające na korzystanie z procedur napisanych w C++ oraz współpracę z innymi środowiskami umożliwiają wykonanie numerycznie intensywnych partii obliczeń przez procedury innych języków,
- dla programów bez wielu operacji arytmetycznych czas wykonania programów w prezentowanym języku jest porównywalny ze Strandem (dla programów testowych różnice nie są większe od 7%). Można to uważać za znaczny sukces prezentowanej implementacji, która jest prototypowa i powstała w środowisku akademickim,
- jako trzy testowe programy **(tokenizer, compiler, assembler)** wybrano poszczególne fazy kompilatora, który został napisany w prezentowanym języku i jest używany jako część systemu. Argumentem wymienionych trzech faz był moduł **compile** (512 linii kodu). Czasy wykonania informują zatem również o rzędzie wielkości czasu kompilacji bardzo dużych modułów.

Kolejne testy mają na celu oszacowanie przydatności języka jako narzędzia efektywnego rozpraszania obliczeń. Schemat obu eksperymentów jest następujący. Trzeba policzyć 1000 zadań, z których każde jest wywołaniem procedury napisanej w C i dołączonej do języka. Czas wykonania tej procedury (skompilowanej jako program C) na komputerze typu SUN SLC wynosi 5.4 sekundy, natomiast na jednym procesorze SUN-SPARC-10 1.0 sekundy. Podczas przeprowadzania pomiarów sieć była bardzo lekko obciążona. Pierwszym parametrem jest czas wykonania pojedynczego zadania. Proces **farmer** znajduje się na węźle zerowym i przydziela zadania procesom **worker**. Każdy z nich znajduje się na innym węźle, natomiast węzły znajdują się na różnych stacjach roboczych (sieć) lub na różnych procesorach (maszyna wieloprocesorowa). Na węźle **farmer**a znajduje się **worker**.

Pomiary przeprowadzono w dwóch konfiguracjach sprzętowych: na sieci jedenastu SUN SLC (tabela 2) oraz na czteroprocesorowym SPARC-10 (tabela 3). Przyspieszenie zdefiniowano jako stosunek czasu potrzebnego do wykonania programu na jednym węźle do czasu potrzebnego na wykonanie tych samych obliczeń na danej liczbie węzłów. Osiągnięty wskaźnik przyspieszenia należy uznać za bardzo dobry i pozwalający na predykcję efektywnego skalowania obliczeń dla kilkudziesięciu węzłów, a zadowalającego nawet dla ponad stu węzłów maszyny wirtualnej.

Ilość **workerów**	Czas **T** obliczeń [s]	Przyspieszenie (5430 / T)
1	5430	1
3	1819	2.99
5	1096	4.95
7	798	6.80
9	636	8.54
10	575	9.44
11	530	10.25

Tabela 2 Efektywność równoważenia obliczeń (SLC).

Ilość **workerów**	Czas **T** obliczeń [s]	Przyspieszenie (5430 / T)
1	1015	1
2	509	1.99
3	341	2.98
4	256	3.96

Tabela 3 Efektywność równoważenia obliczeń (czteroprocesorowy SPARC-10)

Interesującym faktem jest zakończone sukcesem powtórzenie powyższych eksperymentów w środowisku, w jego skład wchodził 16 komputerów SUN SLC, 3 SPARC2 oraz jednego czteroprocesorowego SPARC-10 z trzech domen Internetu, co świadczy o możliwości wykorzystania prezentowanego systemu do obliczeń na sieciach rozległych (*wide area networks*).

Nie poruszonym dotychczas zagadnieniem, jest przenośność systemu. Został on zaprojektowany tak, aby można było go skompilować i wykorzystywać na różnych architekturach. System działa na kilku typach komputerów firmy SUN Microsystems (stacje robocze i maszyny wieloprocesorowe), a zakończone sukcesem eksperymenty, polegające na przeniesieniu implementacji na komputery IBM RS 6000 oraz na IBM PC 386 (w ostatnim wypadku, ze względu na brak wsparcia systemu operacyjnego dla komunikacji sieciowej oraz braku wielozadaniowości możliwe było uruchomienie tylko jednego węzła) potwierdzają możliwości przenośności systemu.

8. Uwagi o kompilacji i implementacji

Kompilator, dokonujący szeregu optymalizacji, napisany jest w prezentowanym języku. Symboliczność ogromnie ułatwia analizę leksykalną i syntaktyczną, deklaratywny charakter języka upraszcza zapis natomiast współbieżność pozwala na wykonywanie wielu etapów kompilacji jako współistniejących procesów (efekt *coroutines*), co powoduje że kolejne pośrednie postacie programu nie muszą być przechowywane do momentu zakończenia danego etapu lecz od razu stają się wejściem następnej fazy kompilacji. Dwa zagadnienia dotyczące kompilacji były przedmiotem szczególnej troski autora: optymalizacja wynikowego kodu oraz minimalizacja ilości niezbędnych instrukcji maszyny abstrakcyjnej (podejście typy *RISC*). Kompilator opisanego języka napisany jest w jego podzbiorze dzielonym z językiem Strand88, co pozwoliło na początkowe skompilowanie kompilatora, używając systemu Strand88. Powstało w ten sposób kilka modułów, realizujących różne fazy kompilacji. Pozwalają one na kompilowanie

dowolnych programów, a w szczególności programu samego kompilatora. Bardzo ułatwia to zmiany języka.

Implementacja oparta jest na koncepcji instrukcji maszyny abstrakcyjnej oraz na wynikach otrzymanych w trakcie projektowania nowoczesnego języka wysokiego poziomu -- PCN. Komunikacja między węzłami jest bardzo efektywna. Zaimplementowano też rozproszony algorytm odzyskiwania pamięci.
Wykorzystanie tylko jednej sterty jako obszaru danych dla puli procesów, danych programu, komunikacji międzywęzłowej oraz przechowywania wykorzystywanych modułów pozwoliło na zaproponowanie i zastosowanie szeregu optymalizacji.

9. Zakończenie

Należy stwierdzić, że osiągnięto przedstawione we wstępie cele. Otrzymane w wyniku realizacji pracy rezultaty, dotyczące wprowadzonych rozszerzeń bazowego modelu programowania w logice, wraz z cechami stworzonej praktycznej implementacji języka oraz doświadczalnymi wynikami pomiarów efektywności rozproszonego przetwarzania w prezentowanym języku, świadczą o prawdziwości postawionych na wstępie tez. Zasygnalizowane zagadnienia są tematem badań w kilku wiodących ośrodkach i w niedalekiej przyszłości należy się spodziewać wielu nowych, interesujących rezultatów.

Bibliografia

[1] Communications of the ACM, numer poświęcony Projektowi Systemów Komputerowych Piątej Generacji, Marzec, 1993

[2] G. Czajkowski , A. Uszok, K. Zieliński, Distributed Declarative Systems as Parts of Cooperating Software Environments. In *Proceedings of ICLP-94 Workshop on Integration of Declarative Paradigms*, Max-Planck Institut fur Informatik, 1994

[3] K. Zieliński (ed). *Środowiska programowania rozproszonego w sieciach komputerowych*. Wydawnictwo Księgarnia Akademicka, Kraków, 1994

[4] I. Foster and R. Olson and S. Tuecke Productive Parallel Programming: The PCN Approach, *Scientific Programming*, 1(1), 1992

[5] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1985

[6] A. Laurentowski and A. Gąsiorowska, *Ramy i ich zastosowanie do enkapsulacji interakcji między procesami,* Materiały 6-tej Międzynarodowej Wdrożeniowej Szkoły PTI, Szczyrk, Czerwiec, 1994

[7] A. Tanenbaum. *Modern Operating Systems,* Prentice-Hall International, 1992

[8] N. Wirth. *Algorytmy + Struktury Danych = Programy*. Wydawnictwo Naukowo-Techniczne, 1989.

[9] K. Zieliński, G. Czajkowski, A. Uszok. Coordination through Interoperability in Distributed Processing. In *ECOOP-94 Workshop on Coordination Languages*, 1994

[10] K. Zieliński, M. Gajęcki, G. Czajkowski. Parallel Programming Systems for LAN Distributed Computing, In *Proceedings of IEEE ICDCS-94*, 1994

Janusz Piela
ComputerLand Poland

Protokół SNMP i zarządzanie sieciami

1. Zarządzanie sieciami

Rozwój zarówno sieci lokalnych jak i sieci globalnych powoduje wzrost ich złożoności, a co za tym idzie wzrost problemów związanych z ich funkcjonowaniem. O ile małe sieci (głównie lokalne) mogą być w stosunkowo prosty sposób utrzymywane przez administratora systemu, to w przypadku większych sieci, a tym bardziej sieci rozproszonych, problem ich funkcjonowania wymaga wprowadzenia dodatkowego elementu, a mianowicie systemu zarządzania, który umożliwia zarówno monitorowanie stanu sieci, jak i oddziaływania na jej funkcjonowanie. Efektywność funkcjonowania sieci nie zależy wyłącznie od stosowanych rozwiązań sprzętowych i używanych łączy. Zależy również od możliwości zarządzania coraz bardziej złożonym środowiskiem funkcjonowania poszczególnych systemów informatycznych.

Rozproszone aplikacje i systemy sieciowe są bardziej skomplikowane do zarządzania niż tradycyjne systemy typu terminal/komputer. Są one coraz większe, zawierają różne wyposażenie sieciowe, zarówno dla sieci lokalnych jak i sieci rozległych, utrzymują aplikacje krytyczne ze względu na zadania, i dlatego też istotne jest posiadanie dobrego zarządzania. Linia podziału odnośnie tego czym jest system, a czym sieć jest płynna; dystrybucja aplikacji i danych między wieloma komputerami tworzy sieć łączącą je w integralną część ogólnego systemu logicznego.

Zarządzanie siecią jest istotnym problemem w dowolnym systemie. Problem staje się bardziej krytyczny w środowisku OSI, które pozwala na współdziałanie sieci budowanych na sprzęcie od różnych dostawców.

Architektura zarządzania siecią OSI jest podzielona na pięć kategorii usług zarządzania znanymi jako specyficzne obszary funkcjonalne zarządzania. Są nimi:

- **zarządzanie błędami**
  Jest to wykrywanie, izolowanie i poprawianie niewłaściwego działania. Obejmuje ono informowanie o błędach, rejestrowanie, śledzenie testowanie diagnostyczne i działania naprawcze.
- **zarządzanie rozliczaniem**
  Jest zdefiniowane jako określanie kosztów wykorzystania resursów i przydziałem odpowiednich opłat.
- **zarządzanie konfiguracją**
  Składa się z ułatwień do inicjowania i kończenia działania zarządzanych obiektów, ustawiania lub zmieniania ich parametrów konfiguracyjnych, zbierania informacji o stanach i związywanie nazw z zarządzanymi obiektami.
- **zarządzanie osiągami**
  Jest zdefiniowane jako oszacowanie zachowania się zarządzanych obiektów. Obejmuje to narzędzia dla gromadzenia i rejestrowania danych statystycznych.
- **zarządzanie ochroną**
  Utrzymuje zarządzanie upoważnieniami, sterowania dostępem i szyfrowaniem danych.

2. System zarządzania siecią

Różne organizacje mają różne poglądy na zarządzanie siecią i dlatego wykorzystują różne definicje tego terminu. Być może najbardziej użyteczną jest definicja zapożyczona z ogólnej definicji zarządzania.

Jest ogólnie przyjęte dzisiaj, że *zarządzanie dotyczy planowania, organizowania, monitorowania, rozliczania i sterowania działaniami i resursami*. Ta definicja może być stosowana również do zarządzania siecią. Jednak, struktury zarządzania siecią OSI i Internet skupiają się głównie na monitorowaniu, rozliczaniu i sterowaniu działaniami i resursami sieci. Pozostałe dwa aspekty zarządzania, planowanie i organizowanie nie są włączane w schemat OSI/Internet, ale są one bardzo ważnymi aspektami zarządzania siecią i konsumują większość czasu i resursów organizacji. Faktycznie, jeżeli sieć nie jest planowana i organizowana właściwie, to każda ilość monitorowania, rozliczania i sterowania jest daremna.

Ważnym celem zarządzania siecią jest utrzymywanie zintegrowanego podejścia do zarządzania siecią (lub sieciami), która zawiera komputery, pakiety oprogramowania i nośnik od wielu dostawców. Należy podkreślić kilka kluczowych punktów o zintegrowanym zarządzaniu siecią.

- Zintegrowane zarządzanie siecią jest potrzebne w celu redukcji kosztów łączenia różnych systemów.
- Ponieważ integracja jest potrzebna, zarządzanie siecią wymaga jednolitości w wymianie informacji zarządzania pomiędzy liniami produktów różnych dostawców.
- Zintegrowane zarządzanie siecią nie oznacza, że usługi są takie same dla wszystkich użytkowników. Oznacza to, że podstawowy poziom usługi zarządzania jest taki sam dla definicji osiągów sieci, rozliczania, konfiguracji, ochrony i kryteriów błędów i wymiany jednostek danych wspólnego protokołu.
- Integracja i jednolitość nie wyklucza usług z dodaną wartością w kontekście tych standardów.

Celowym byłoby stosowanie wspólnego podejścia do zarządzania sieciami Ethernet, TCP/IP, sieciami fonicznymi, sieciami FDDI, Frame Realy, ISDN i ATM. Dotychczas, większość pakietów zarządzania różniło się w tych sieciach. Kluczowym celem standardów zarządzania siecią jest rozwój zintegrowanego zbioru procedur i standardów, które stosują się w równy sposób do różnych sieci i różnych dostawców.

Zarządzanie siecią OSI jest związane z usługami i funkcjami, które są używane do zarządzania i sterowania wzajemnie połączonymi działaniami przetwarzania danych i resursów komunikacji danych. Dlatego, zarządzanie siecią OSI nie dotyczy resursów, które nie są wzajemnie połączone. Co więcej, ponieważ zarządzanie siecią OSI jest rozszerzeniem OSI, to dotyczy tylko aspektów komunikacyjnych zarządzania, a nie operacji wewnętrznych, powiedzmy, modemu lub komutatora.

W bardziej ograniczonym rozmiarze, zasady te również stosują się do standardów Internet, z wyjątkiem, że Internet nie jest traktowany jako rozszerzenie OSI. Jednak, podobnie jak w modelu OSI, standardy zarządzania siecią Internet są traktowane jako działania wzajemnie połączone. To samo jest prawdziwe dla standardów zarządzania siecią IEEE (sieci lokalne).

Standardy zarządzania siecią OSI, Internet i IEEE definiują odpowiedzialność dla *procesu zarządzania* (nazywanego systemem zarządzania siecią w produktach pewnych dostawców lub manager) i *agenta zarządzania* (również znanego jako proces agent). W najbardziej ścisłym sensie, system zarządzania siecią rzeczywiście nie zawiera nic więcej niż protokoły, które przenoszą informacje o elementach sieci tam i z powrotem, między różnymi agentami w systemie i procesem zarządzania.

Jeszcze jeden składnik jest istotny dla systemu zarządzania siecią. Jest on nazywany *bazą lub biblioteką informacji zarządzania*. Ten obiekt koncepcyjny jest bazą danych, która jest dzielona między managerami i agentami w celu dostarczania informacji o zarządzanych elementach sieciowych. Następujące notacje podsumowują te ważne koncepcje:

Obiekt zarządzany - wyposażenie sieciowe, procesy i usługi, które mogą być zarządzane.

Baza informacji zarządzania (MIB) - baza danych informacji o zarządzanych obiektach.

Manager - oprogramowanie w stacji zarządzania które zapytuje agentów; odbiera i działa na odpowiedziach agentów; ustawia zmienne MIB; archiwuje dane MIB dla analizy trendów; dostarcza graficznego interfejsu użytkownika. Dowolny manager dostosowany do protokołu może komunikować się z agentem utrzymującym ten sam protokół. W SNMP, manager przepytuje agentów o ich stany. W CMIP, agenci raportują swoje własne stany do managera.

Agent - oprogramowanie w zarządzanym urządzeniu, które pamięta, bada i zmienia zmienne MIB w odpowiedzi na żądania manager. W CMIP, agent również raportuje zmiany swojego własnego stanu i wykonuje wstępne manipulacje danymi MIB przed nadaniem ich do managera.

Protokół - prawa formatowania informacji i jej wymiany pomiędzy managerem i agentem.

Gdzie te elementy rezydują? Standardy zarządzania siecią OSI, Internet i IEEE nie wymagają aby były one umieszczone w poszczególnym miejscu w sieci. W bieżących implementacjach, oprogramowanie agenta jest zwykle umieszczane w elementach takich jak serwery, stacje robocze, gateway, bridge lub router. Zwykle, stacja sterowania siecią (NCS) działa jako proces zarządzający. MIB jest zwykle lokalizowany w NCS i część MIB, która jest stała dla agenta jest również umieszczana w agencie.

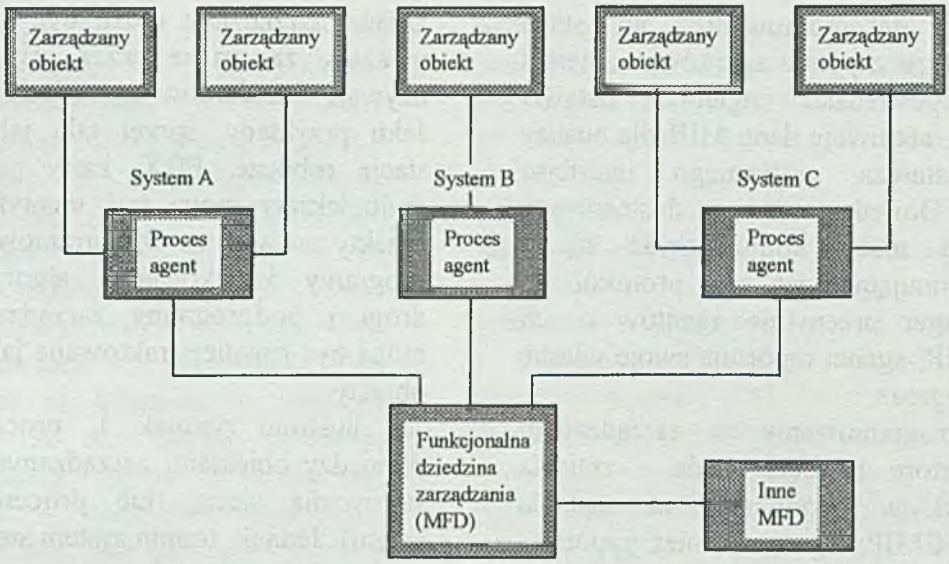
Standardy zarządzania siecią OSI używają wielu z koncepcji projektowania zorientowanego obiektowo (OOD). W mniejszym rozmiarze, model Internet i IEEE również używa koncepcji OOD.

Jak pokazano na rysunku 1, resursy, które są zarządzane lub sterowane przez zarządzanie siecią są nazywane obiektami zarządzanymi. Obiekt zarządzany może być czymś, co jest uważane za ważne przez organizacje, które używają standardów zarządzania siecią OSI. Jako przykłady, sprzęt taki jak komutatory, stacje robocze, PBX, karty portów PBX i multipleksery mogą być identyfikowane jako obiekty zarządzane. Oprogramowanie, takie jak programy kolejkowania, algorytmy wyboru drogi i podprogramy zarządzania buforami mogą być również traktowane jako zarządzane obiekty.

Jak ilustruje rysunek 1, proces agent stoi pomiędzy obiektami zarządzanymi i systemem sterowania siecią (lub procesem zarządzającym). Jednak, termin system sterowania siecią ma na tym rysunku zmienioną nazwę na dziedzina funkcjonalna zarządzania (MFD). Ten termin OSI jest używany (w sposób abstrakcyjny) do opisu, że system zarządzania może również być grupowany w procesy agentów raportujące do MFD. Zwrotnie może istnieć wiele MFD. Standardy Internet i IEEE nie używają tej koncepcji.

Kilkoma innymi ważnymi aspektami zarządzanych obiektów są:

- Dopuszczalne operacje zarządzania, które mogą być wykonywane na zarządzanym obiekcie.
- Definicja zarządzanego obiektu może również obejmować efekt jaki te operacje mają w związanych z nimi resursami systemowymi.
- Stan zarządzanego obiektu lub jego właściwości mogą określać typ operacji, która może być wykonywana na zarządzanym obiekcie.



Rysunek 1. Terminy i koncepcje zarządzania siecią

3. Definiowanie obiektów zarządzanych Internet

Z perspektywy Internet zarządzany obiekt jest opisany w sposób mniej abstrakcyjny niż obiekt OSI. Jak pokazano na rysunku 2, zarządzany obiekt jest opisywany przez:

- Składnię używaną do modelowania obiektu,
- Poziom dostępu dopuszczanego do obiektu,
- Wymaganie dla implementacji obiektu; jego stan,
- Niedwuznaczną nazwę obiektu.

Rysunek 2. Zarządzany obiekt Internet

Składnia. W modelu Internet, składnia definiuje typ danych. W przeciwieństwie do zarządzania siecią OSI, typy Internet są całkiem ograniczone. Są one zdefiniowane przy pomocy typów ASN.1 z albo INTEGER, OCTET STRING, SEQUENCE albo SEQUENCE OF. Ponadto, MIB Internet wyjaśnia użycie OCTET STRING w kontekście kiedy może być używany jako drukowalny znak. W konsekwencji, łańcuch zobrazowania może również być identyfikowany jako OCTET STRING w którym zbiory znaków ASCII są zdefiniowane.

Dostęp. Definicja dostępu dostarcza informacji o tym jak zarządzany obiekt może być dostępowany. Następujące operacje są dopuszczalne na zarządzanym obiekcie:

Tylko odczyt: Wystąpienia obiektów mogą być czytane ale nie ustawiane.

Odczyt zapis: Wystąpienia obiektów mogą być czytane i ustawiane.

Tylko zapis: Wystąpienia obiektów mogą być ustawiane ale nie czytane.

Niedostępowalny: Wystąpienia obiektów nie mogą być ani czytane ani ustawiane.

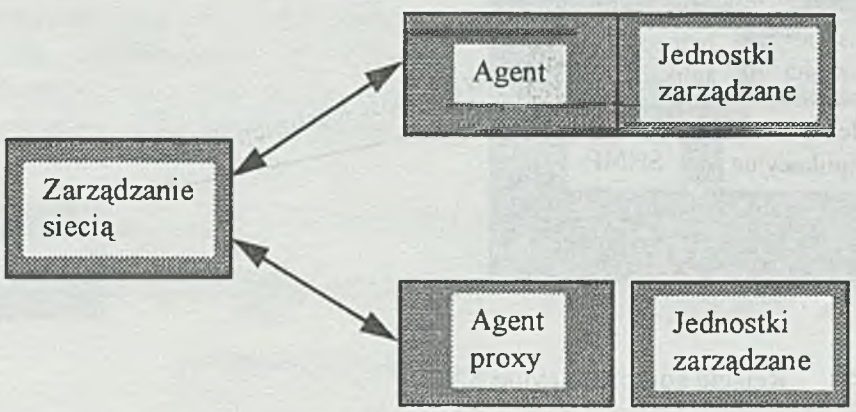
Stan. Stan obiektu jest bieżąco definiowany jako (1) obowiązujący, każdy zarządzany węzeł jest wymagany do implementowania swojego zarządzanego obiektu; (2) opcjonalny, węzły nie są wymagane do implementowania obiektu; lub (3) przestarzały, zarządzany obiekt nie jest dłużej definiowany i węzły nie potrzebują implementować go.

Nazwa. Definicja nazwy jest ANS.1 OBJECT IDENTIFIER (identyfikator obiektu), który jest używany do niedwuznacznego identyfikowania zarządzanego obiektu.

4. Model zarządzania siecią Internet

Internet używa terminu element sieciowy do opisu dowolnego obiektu, który jest zarządzany. Ten termin jest taki sam jak obiekt zarządzany OSI.

Standardy zarządzania siecią Internet są opracowane w celu dopuszczenia komunikacji informacji zarządzania pomiędzy agentami umieszczonymi w elementach sieciowych i stacją zarządzania siecią (zwykle, centrum zarządzania siecią). Z kontekstu rysunku 3, element sieciowy składa się z jednostki zarządzanej i agenta jednostki zarządzanej.



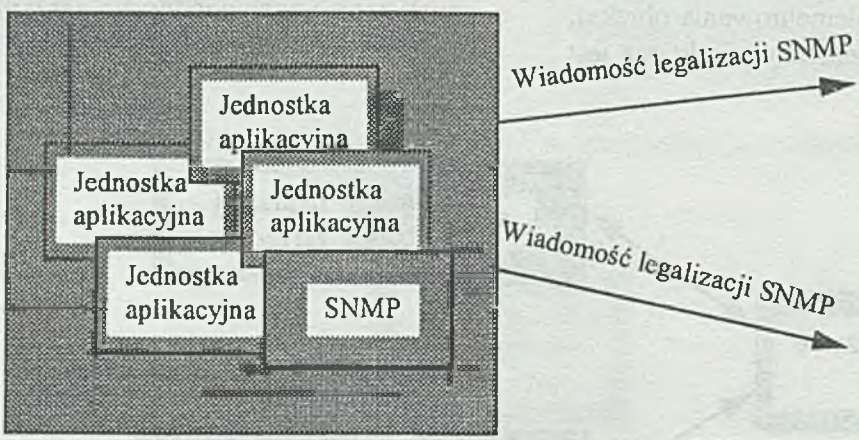
Rysunek 3. Model zarządzania siecią Internet

Rysunek 3 pokazuje również agenta proxy (pośrednika). Ta jednostka może być przywołana dla elementów, które są nieosiągalne przez używanie konwencjonalnego protokołu zarządzania. Agent proxy może dostarczać dla zbieżności funkcji takich jak konwersja protokołu i operacje filtrowania. Zarządzanie pośrednie jest wykonywane odnośnie zachowania się resursów sieciowych, które nie mogą komunikować się bezpośrednio z elementem zarządzającym. Na przykład, funkcje niskiego poziomu urządzeń takich jak bridge i modemy zwykle nie są zdolne do utrzymywania wyszukanych protokołów zarządzania. Kiedy jednostka zarządzająca chce komunikować się z urządzeniem zarządzanym, które nie może utrzymywać protokołów zarządzania, to po prostu skierowuje ruch do agenta pośrednika. Jest odpowiedzialnością agenta pośrednika zrozumienie protokołu

zarządzania i wykonanie funkcji komunikowania się z zarządzanym urządzeniem.
To podejście jest atrakcyjne ponieważ utrzymuje operacje przezroczyste w stosunku do zarządzania siecią, która tylko wie, że musi posiadać dodatkowe wystąpienie informacji do wybrania pośrednika zamiast resursu sieciowego. W konsekwencji agent pośrednik jest dopuszczony do zarządzania wieloma urządzeniami używając wystąpień informacji do dopasowania informacji.

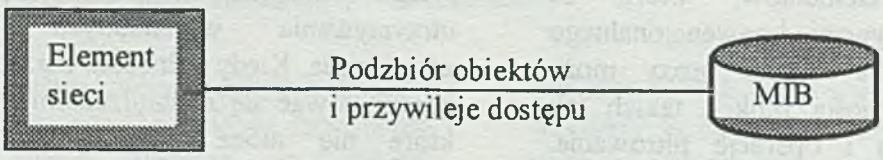
5. SNMP - prosty protokół zarządzania siecią

Zarówno platformy zarządzania jak i aplikacje zarządzania są związane z konkretnym protokołem zarządzania. Aktualnie najbardziej rozpowszechnionym na świecie protokołem zarządzania jest protokół SNMP (Simple Network Management Protocol). Protokół ten jest standardem de facto, stosowanym praktycznie przez prawie wszystkich producentów sprzętu i oprogramowania, które mogą być zarządzane. Większość urządzeń sieciowych i komunikacyjnych wyposażonych jest w moduł zarządzania (czasami opcjonalny), który realizuje oprogramowanie agenta zarządzania SNMP.
Architektura SNMP używa różnych terminów, które są wyjaśnione poniżej. Używając rysunku 4 jako punktu do dyskusji, jednostki, które rezydują w stacjach zarządzania siecią i elementy sieciowe, które komunikują się między sobą używając standardu SNMP są nazywane jednostkami aplikacyjnymi SNMP. Sparowanie jednostek aplikacyjnych z agentami SNMP jest nazywane otoczeniem SNMP. Każde otoczenie jest identyfikowane przez nazwę hierarchiczną Internet.



Rysunek 4. Relacje administracyjne SNMP

Wiadomości SNMP są zapoczątkowane przez jednostki aplikacyjne SNMP. Są one traktowane jako należące do otoczenia SNMP, które zawiera jednostkę aplikacyjną. Te wiadomości są nazywane wiadomościami legalizacji SNMP. Schematy legalizacji są używane do identyfikowania wiadomości i weryfikowania ich autentyczności. Ten proces jest nazywany usługą legalizacji.

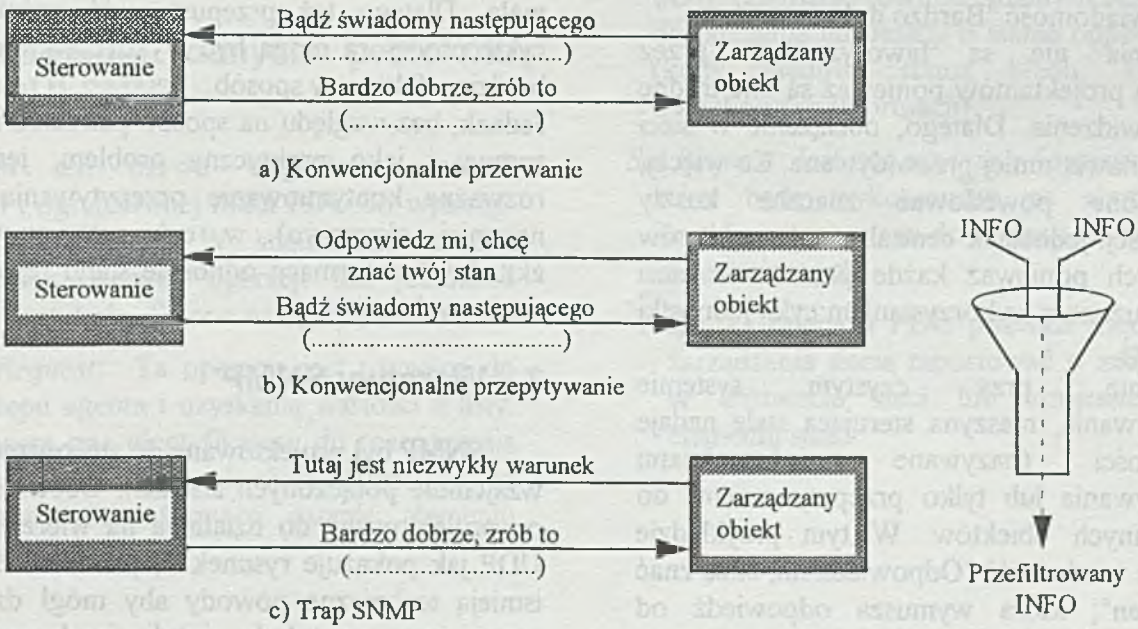


Rysunek 5. Profil środowiska SNMP

Rysunek 5 dostarcza poglądu na inne relacje administracyjne dla SNMP. Element SNMP używa obiektów z bazy informacji zarządzania Internet (IMIB). Ten podzbiór obiektów odnoszący się do tego obiektu jest nazywany SNMP MIB view. Zwrotnie, tryb dostępu SNMP reprezentuje element zbioru (na przykład, elementy tylko do odczytu, elementy tylko do zapisu). W końcu, sparowanie trybu dostępu SNMP z MIB view jest nazywane profilem otoczenia SNMP. W zasadzie, profil jest używany do określania przywilejów dostępu dla MIB view. Te relacje są zdeterminowane przez parowanie otoczenia CNMP przez rozwijanie profili nazywanych politykami dostępu SNMP. Te polityki dostępu następnie dostarczają kierunków jak agenci SNMP i elementy sieciowe mogą używać MIB.
Zwykle, informacja środowiska i inne dane są pamiętane w pliku konfiguracyjnym systemu. Plik zawiera informacje o użytkowniku i środowisku agenta. Plik musi zawierać nazwę środowiska i adres protokołu internet (IP) jednostki związanej ze środowiskiem. Przywileje dostępu takie jak tylko zapis, odczyt zapis, itp., są zdefiniowane z nazwą środowiska. Ponadto, view jest dostarczone, które opisuje podzbiór MIB jaki jest dostępny dla tej nazwy środowiska. Informacja trap jest również zwykle włączona; definiuje ona środowisko do którego trap mają być nadawane. Nazwy zmiennych są również włączone; zawierają one położenie i kontakty systemu.

6. Strategia zarządzania SNMP poprzez przepytywanie i trap

SNMP działa z dwoma funkcjami zarządzania: (1) jednostka współdziała z agentem zarządzania w celu wyszukiwania (uzyskania) zmiennych, i/lub (2) jednostka współdziała z agentem zarządzania w celu zmiany (ustawienia) zmiennych. Ideą tego prostego podejścia jest fakt, że to znacznie ogranicza funkcje, które mogą być wykonywane z SNMP, co w konsekwencji ogranicza złożoność oprogramowania.
Te funkcje mogą być implementowane poprzez operacje przepytywania. Manager SNMP może być zaprogramowany do nadawania okresowego wiadomości przepytujących do zarządzanych urządzeń w pewnych interwałach. Te interwały mogą być ustalane poprzez SNMP MIB.



Rysunek 6. Operacje przepytywania i przerwania

Koncepcja ta jest ważna w ewolucji SNMP z trzech powodów. Po pierwsze, stosowanie przepytywania utrzymuje system względnie prostym. Po drugie, ponieważ przepytywanie jest sterowane przez jednostkę zarządzającą SNMP, to może ograniczać ilość ruchu przesyłanej informacji zarządzania, która jest tworzona poprzez Internet. Po trzecie, protokół zarządzania przepytywaniem po prostu ogranicza elastyczność zarządzanych elementów do reagowania na warunki w określony sposób; dlatego ogranicza to liczbę urządzeń, które mogą być zarządzane w Internet.

SNMP nie jest kompletnym protokołem przepytywania. Dopuszcza on pewien spontaniczny ruch, nazywany trap, bazując na ograniczonych parametrach. Relacja przepytywania i trap jest pokazana na rysunku 7.

Rysunek 6 pokazuje konwencjonalny proces przerwania (Rysunek 6a). W tej sytuacji centrum sterowania siecią (nazwane sterowanie na tej ilustracji) reaguje na zarządzany obiekt, kiedy zarządzany obiekt nadaje przerwanie do niego. Przerwanie jest dobrze znaną konwencją dla zarządzania komunikacją między maszynami. Urządzenie sterujące, po odebraniu wiadomości "Bądź świadom następującego" podejmuje decyzję o podjęciu pewnych akcji i zwraca wiadomość "Bardzo dobrze, zrób to...".

Przerwania nie są faworyzowane przez pewnych projektantów ponieważ są one trudne do przewidzenia. Dlatego, obciążenie w sieci staje się nawet mniej przewidywane. Co więcej, mogą one powodować znaczne koszty eksploatacji jednostek centralnych komputerów sieciowych ponieważ każde przerwanie musi być obsłużone z wykorzystaniem cykli jednostki centralnej.

Przeciwnie, przy czystym systemie przepytywania, maszyna sterująca stale nadaje wiadomości (nazywane wiadomościami przepytywania lub tylko przepytywaniem) do zarządzanych obiektów. W tym przykładzie widzimy wiadomość "Odpowiedz mi, chcę znać twój stan", która wymusza odpowiedź od zarządzanego obiektu.

Operacje przepytywania mogą konsumować znaczną ilość kosztów przez wprowadzenie nieużytecznego ruchu danych do sieci. Te operacje są szczególnie kłopotliwe jeżeli przepytywania są wydawane często i nic produktywnego nie jest raportowane na nie. Pasmo nośnika i cykle komputera są konsumowane - bez żadnych pozytywnych efektów w efektywności sieci. Niemniej jednak, protokoły przepytywania są względnie proste do implementowania i pozwalają one projektantowi sieci opracować bardziej określony profil ruchu.

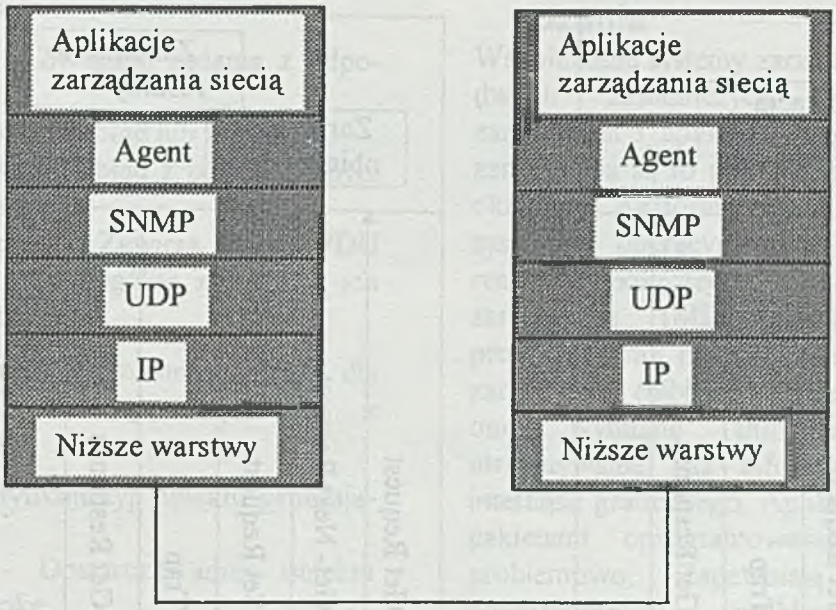
SNMP podchodzi do tej sytuacji ze zmodyfikowanym przerwaniem nazywanym trap. Ta operacja jest zilustrowana na rysunku 6c. Agent zarządzanego obiektu jest odpowiedzialny za wykonanie kontroli wartości progowych (ogólnie nazywanej filtrem) i tylko warunki raportowania które spełniają pewne kryteria progowe. Na przykład, kolejka staje się zbyt duża, itp. Po odebraniu wiadomości "Jest tu niezwykły warunek" maszyna sterująca musi zdecydować o podjęciu pewnej akcji.

No dobrze, ale trap nadal jest przerwaniem. Jednak uzyskiwane są następujące korzyści: (1) trap (przerwanie) jest wykonywane bardzo rzadko i w krytycznych warunkach, i (2) wiadomość przerwania jest bardzo prosta i mała. Dlatego też przepustowość nośnika i cykle procesora mogą być wykorzystywane w bardziej efektywny sposób.

Jednak, bez względu na sposób patrzenia na te sprawy i jako praktyczny problem, jest to rozważne kontynuowanie przepytywania (co najmniej okresowo) w celu utrzymywania aktualnych informacji odnośnie stanu resursów sieciowych.

7. Warstwy SNMP

SNMP był projektowany do stosowania we wzajemnie połączonych sieciach. Obecnie, jest on projektowany do działania na wierzchołku UDP jak pokazuje rysunek 7, pomimo, że nie istnieją techniczne powody aby mógł działać przez inne protokoły jeżeli implementator będzie rozwijał pewne moduły interfejsu do protokołu.



Rysunek 7. Warstwy SNMP

Ponieważ używa UDP, SNMP jest protokołem bezpołączeniowym. Nie ma gwarancji że ruch zarządzania jest odebrany w innej jednostce. Jak we wszystkich protokołach bezpołączeniowych, koszt przetwarzania jest redukowany i prostota jest uzyskana. Jednak, jeżeli potrzebna jest niezawodność i rozliczanie, manager sieci zbuduje operacje zorientowane na połączenia w aplikacjach wyższej warstwy.

8. Jednostka danych protokołu (PDU) SNMP

Protokół ten używa względnie prostych operacji i ograniczonej ilości PDU do wykonywania ich funkcji. W standardzie zostało zdefiniowanych pięć operacji dla jednostek danych protokołu. Są one następujące:

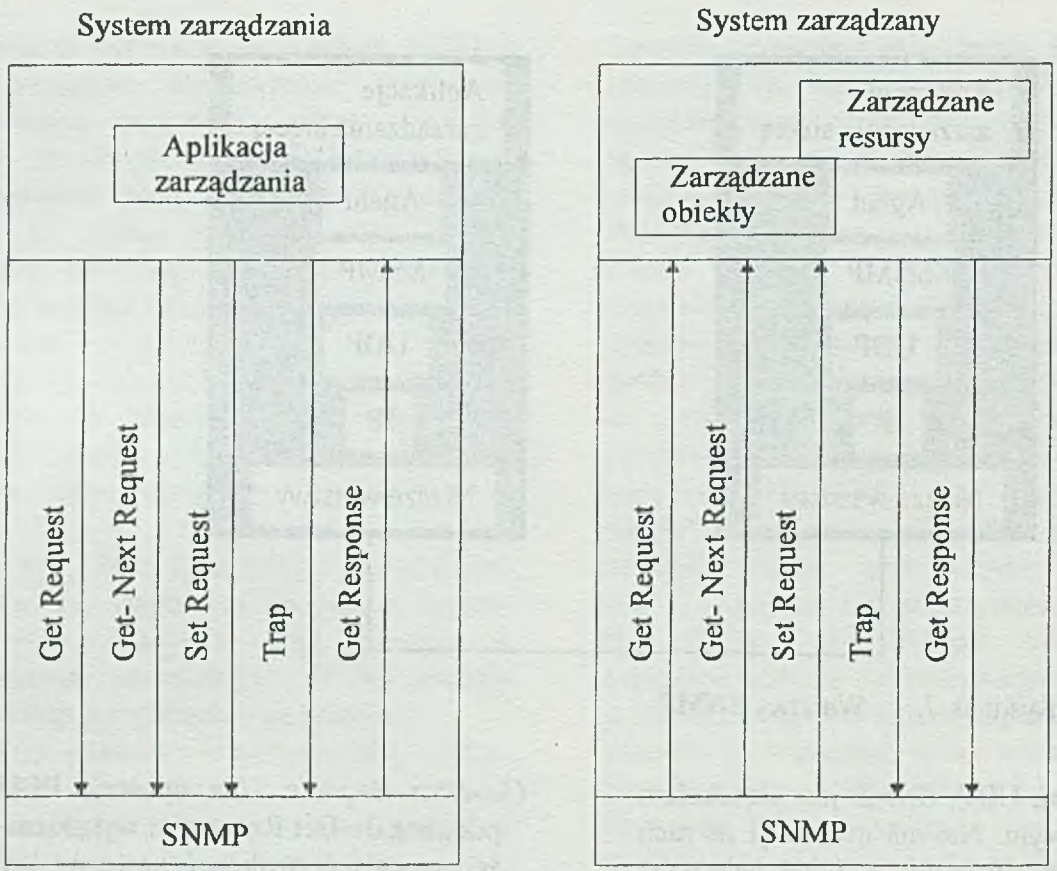
Get Request: Ta operacja jest używana do dostępu agenta i uzyskania wartości z listy. Zawiera ona identyfikatory do rozróżnienia jej od wielu żądań jak również wartości do dostarczenia informacji ostanie elementu sieciowego.

Get-Next Request: Ta operacja PDU jest podobna do Get Request, z wyjątkiem tego, że pozwala wyszukiwać następny logiczny identyfikator w drzewie MIB.

Get Response: Ta operacja PDU odpowiada na jednostki danych Get Request, Get-Next Request i Set Request. Zawiera ona identyfikator, który związuje ją z poprzednią PDU. Zawiera również identyfikatory do dostarczenia informacji o stanie odpowiedzi (kody błędów, status błędu, i listę dodatkowych informacji).

Set Request: Jest ona używana do opisu akcji, które będą wykonywane na elemencie. Zwykle jest używana do zmiany wartości w liście zmiennych.

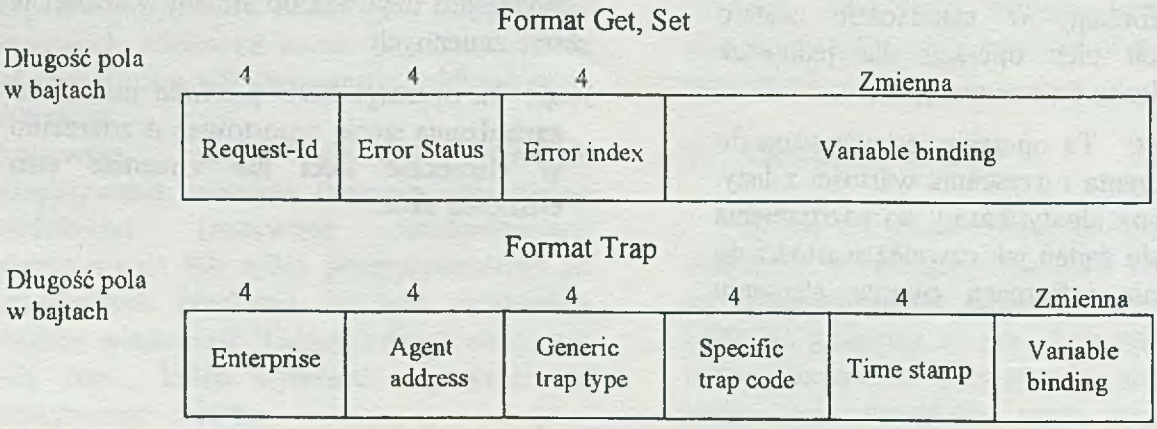
Trap: Ta operacja PDU pozwala modułowi zarządzania siecią raportować o zdarzeniu w elemencie sieci lub zmieniać stan elementu sieci.



Rysunek 8 Interfejs SNMP z wyższą warstwą

Rysunek 8 pokazuje granice pomiędzy oprogramowaniem SNMP i oprogramowaniem wyższej warstwy. Ten intefejs jest osiągnięty przez przesyłanie PDU pomiędzy dwoma jednostkami. Przepływ PDU może zachodzić w obu kierunkach, pomimo że rysunek 8 pokazuje, że PDU są przesyłane z SNMP do wyższej warstwy. Trap mogą być wydawane przez dowolną jednostkę.

Część danych wiadomości SNMP zawiera określoną operację SNMP (get, put, itd.) i związane z nią operandy. Operandy wskazują wystąpienia obiektu w transakcji SNMP.
Wiadomości SNMP są nazywane również jednostkami danych protokołu (PDU). Rysunek 9 pokazuje format pakietu SNMP.



Rysunek 9 Struktura danych PDU

PDU get i put SNMP składają się z następujących części:

Request-ID - Związuje żądania z odpowiedziami.
Error-status - Wskazuje błąd lub typ błędu.
Error-index - Związuje błąd z poszczególnym wystąpieniem obiektu.
Variable bindings - Zawiera dane PDU SNMP. Związuje poszczególne zmienne z ich bieżącymi wartościami.

PDU pułapek są nieco odmienne niż PDU dla innych operacji. Składają się one z następujących części:

Enterprise - Identyfikuje typ obiektu generującego pułapkę.
Agent address - Dostarcza adres obiektu generującego pułapkę.
Generic trap type - Dostarcza ogólny typ pułapki.
Specific trap code - Dostarcza specyficznego kodu pułapki.
Time stamp - Dostarcza ilości czasu jaki upłynął pomiędzy ostatnią ponowną inicjalizacją sieci i generacją tej pułapki.
Varible bindings - Dostarcza listy zmiennych zawierających interesujące informacje o pułapce.

9. Platformy i aplikacje zarządzania

Współczesne systemy zarządzania składają się z dwóch zasadniczych części: platformy zarządzania i aplikacji zarządzania. Platformy zarządzania są to pakiety oprogramowania dla określonej platformy sprzętowej i określonego systemu operacyjnego. Dostarczają one realizacji poszczególnych obiektów systemu zarządzania (MIB, moduł zarządzający, protokół) oraz narzędzi dla budowy aplikacji zarządzania (biblioteki procedur). Zapewniają one wymianę informacji zarządzania, utrzymywanie baz informacji zarządzania i interfejsu graficznego. Aplikacje zarządzania są pakietami oprogramowania zorientowanymi problemowo. Zapewniają one realizację poszczególnych problemowych funkcji zarządzania lub też są zorientowane na zarządzanie urządzeniami określonej firmy. Wiele aplikacji zarządzania działa nie na jednej lecz na kilku platformach zarządzania.
Zakres możliwych funkcji zarządzania, szczegółowość uzyskiwania i ustawiania poszczególnych parametrów zarządzanego obiektu zależą od konkretnego rozwiązania. Są one określane przez bazę informacji zarządzania (MIB) oraz oprogramowanie managera i agenta.
Najbardziej popularnymi platformami zarządzania protokołu SNMP dla systemu UNIX są: SunNet Manager dla komputerów firmy Sun, OpenView dla komputerów firm Hewlett Packard i Sun oraz NetView/6000 dla komputerów firmy IBM.

Zbigniew Skotniczy
Solidex (Cisco) Kraków

Sieci komputerowe i metropolitarne. Sieci bez problemów

szkolenia
doradztwo
projektowanie
realizacja projektów
autoryzowane dostawy
doradztwo techniczne
konsultacje
instalacje
serwis

Sieci bez problemów.

SOLIDEX® Ltd.

SIEĆ WAN GAZETY WYBORCZEJ
Krajowa sieć rozległa oparta o routery Cisco 4000 i Cisco 2500 łączące oddziały Agory i drukarnie na terenie całego kraju (w sieci działają połączenia ISDN).

SIEĆ BANKU PKO S.A. W WARSZAWIE
Krajowa sieć WAN składająca się z ponad 30 węzłów Cisco w oddziałach banku łącząca sieci LAN poprzez szeregowe łącza międzymiastowe (TPSA, TELBANK) z miejską siecią FDDI centrali Banku w Warszawie. FDDI o zasięgu miejskim dla Banku PKO S.A. w Warszawie.

KORPORACYJNA SIEĆ WAN TELEKOMUNIKACJI POLSKIEJ S.A.
Kilkadziesiąt routerów Cisco łączy sieci LAN Centrali, Okręgów i innych jednostek TP S.A.

SIEĆ BANKU DEPOZYTOWO-KREDYTOWEGO S.A. W LUBLINIE
Sieć WAN oparta o routery Cisco łączące oddziały i centralę Banku.

SIECI KORPORACYJNE MAN I WAN REALIZOWANE Z UDZIAŁEM SOLIDEXu

SIEĆ BANKU PRZEMYSŁOWO-HANDLOWEGO W KRAKOWIE
Krajowa sieć WAN składająca się z ponad 50 węzłów Cisco w oddziałach banku, łącząca sieci LAN poprzez szeregowe łącza międzymiastowe (TPSA, TELBANK) z miejską siecią FDDI centrali Banku w Krakowie. FDDI o zasięgu miejskim dla Banku BPH S.A. w Krakowie.

CISCO SYSTEMS
Sun microsystems
Telesyn Inc.
CHIPCOM
SYBASE
SynOptics

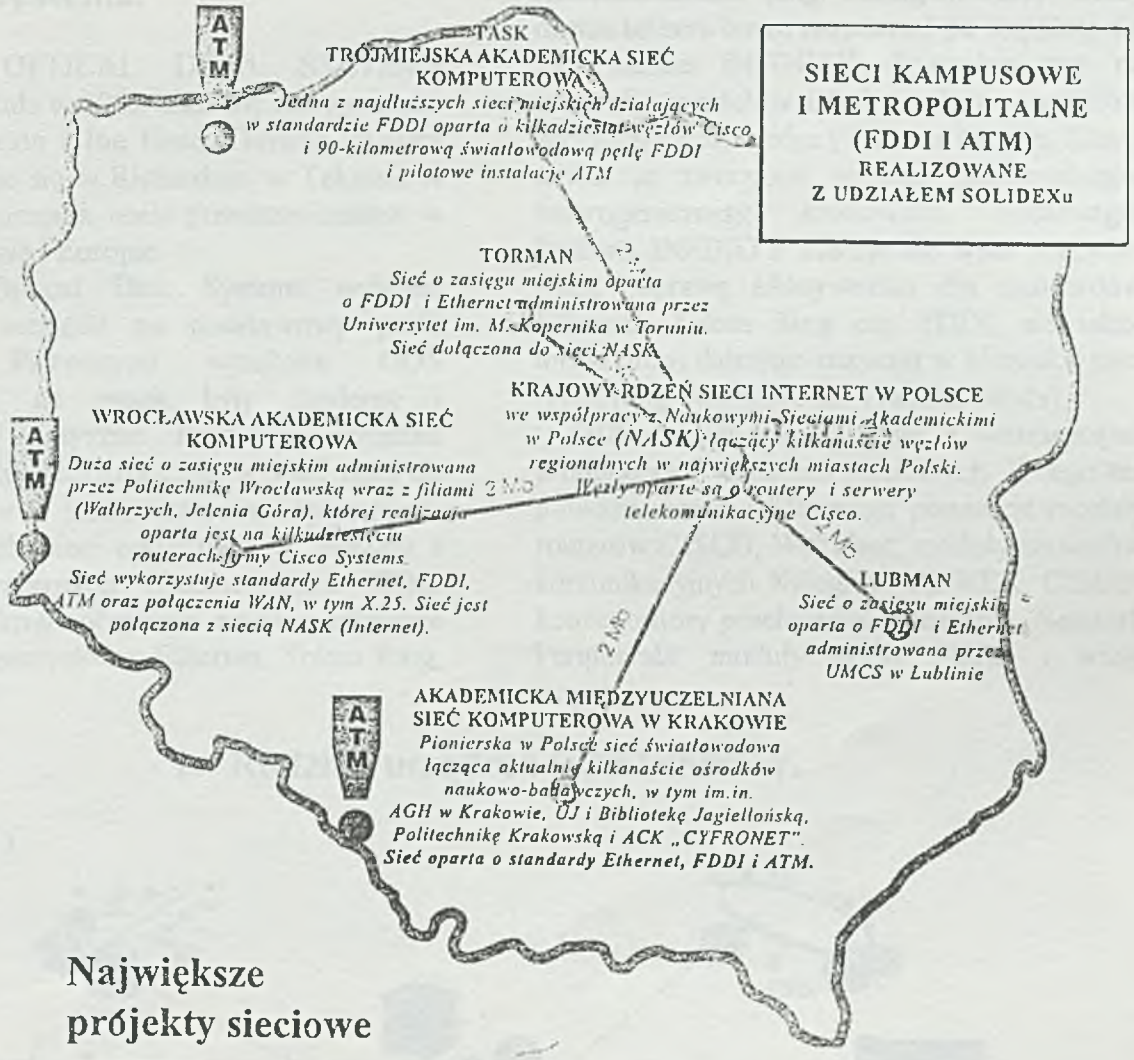
30-067 Kraków, ul. Piastowska 44
tel. (12) 36 68 29
fax. (12) 36 55 45
e-mail: solidex@solidex.krakow.pl

SOLIDEX Ltd.

Oddział w Warszawie
00-697 Warszawa, Al. Jerozolimskie 65/79
tel. (2) 630 96 11 fax (2) 630 66 13
e-mail: solidex@solidex.waw.pl

Firma SOLIDEX od pięciu lat swego istnienia notuje ciągły, dynamiczny rozwój wdrażając na polskim rynku nowoczesne technologie swoich partnerów, największych światowych producentów, dodając do produktów wiedzę i doświadczenie kadry specjalistów. Po pięciu latach działalności SOLIDEX może pochwalić się dziesiątkami instalacji sieciowych w bankach, uczelniach, instytucjach rządowych i zakładach przemysłowych. Blisko 100,000 użytkowników korzysta dzisiaj z sieci Internet opartej o zainstalowane przez SOLIDEX wieloprotokołowe routery Cisco Systems. Światłowodowe sieci miejskie w oparciu o standardy FDDI i ATM funkcjonują już w kilku ośrodkach miejskich. Kilkaset stacji i serwerów SUN ułatwia pracę Klientom SOLIDEXu.

SOLIDEX® Ltd.



Największe projekty sieciowe

SOLIDEX Ltd. uczestniczy w projektowaniu i realizacji największych sieci komputerowych LAN, MAN i WAN o różnym zasięgu terytorialnym. Wiele z tych sieci ma charakter nowatorski w skali kraju.

30-067 Kraków, ul. Piastowska 44
tel. (12) 36 68 29
fax. (12) 36 55 45
e-mail: solidex@solidex.krakow.pl

SOLIDEX Ltd.

Oddział w Warszawie
00-697 Warszawa, Al. Jerozolimskie 65/79
tel. (22) 620 66 11 fax. (2) 629 66 13
e-mail: biuro@solidex.waw.pl

Krzysztof Ślatała
Ventus Computer - Poznań

Koncentratory Wielkich Sieci

1. Przedstawienie firmy Optical Data Systems.

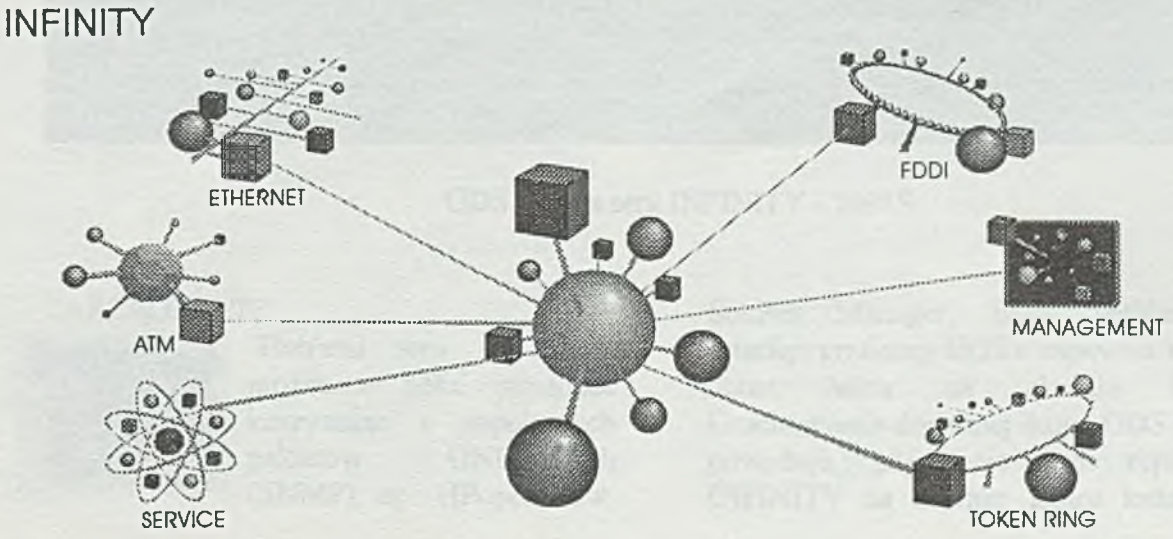
Firma OPTICAL DATA SYSTEMS (ODS) powstała w 1983 roku. Jej założycielami są Ward Paxton i Joe Head. Główna kwatera firmy znajduje się w Richardson w Teksasie w USA. ODS posiada wiele przedstawicielstw w USA, Kanadzie i Europie.

Nazwa Optical Data Systems wybrana została ze względu na podstawowy profil produkcji. Pierwszymi wyrobami ODS kierowanymi na rynek były modemy i multipleksery optyczne. Produkcję urządzeń standardu Ethernet rozpoczęto w 1987 roku od transceiver'ów i koncentratorów optycznych. Urządzenia dla sieci optycznych są jednymi z wielu wytwarzanych obecnie przez ODS. Produkcja firmy obejmuje swoim zakresem standardy przemysłowe: Ethernet, Token Ring, FDDI, ATM.

W 1992 roku ODS wprowadził na rynek nową serię produktów zwanych Inteligentnymi Koncentratorami (ang. Inteligent Hub). Urządzenia tej serii łatwo rozpoznać po wspólnej dla nich nazwie INFINITY. Pozwalają one na integrację modułów użytkowników, modułów zarządzających, bridge'y i router'ów, itp. Umożliwia to tworzenie wysoce niezawodnego, heterogenicznego środowiska sieciowego. Rodzina INFINITY oferuje nie tylko zdecydowaną poprawę efektywności dla standardów Ethernet, Token Ring czy FDDI, ale także toruje drogi dalszego rozwoju w kierunku sieci ATM (ang. Asynchronus Transfer Mode).

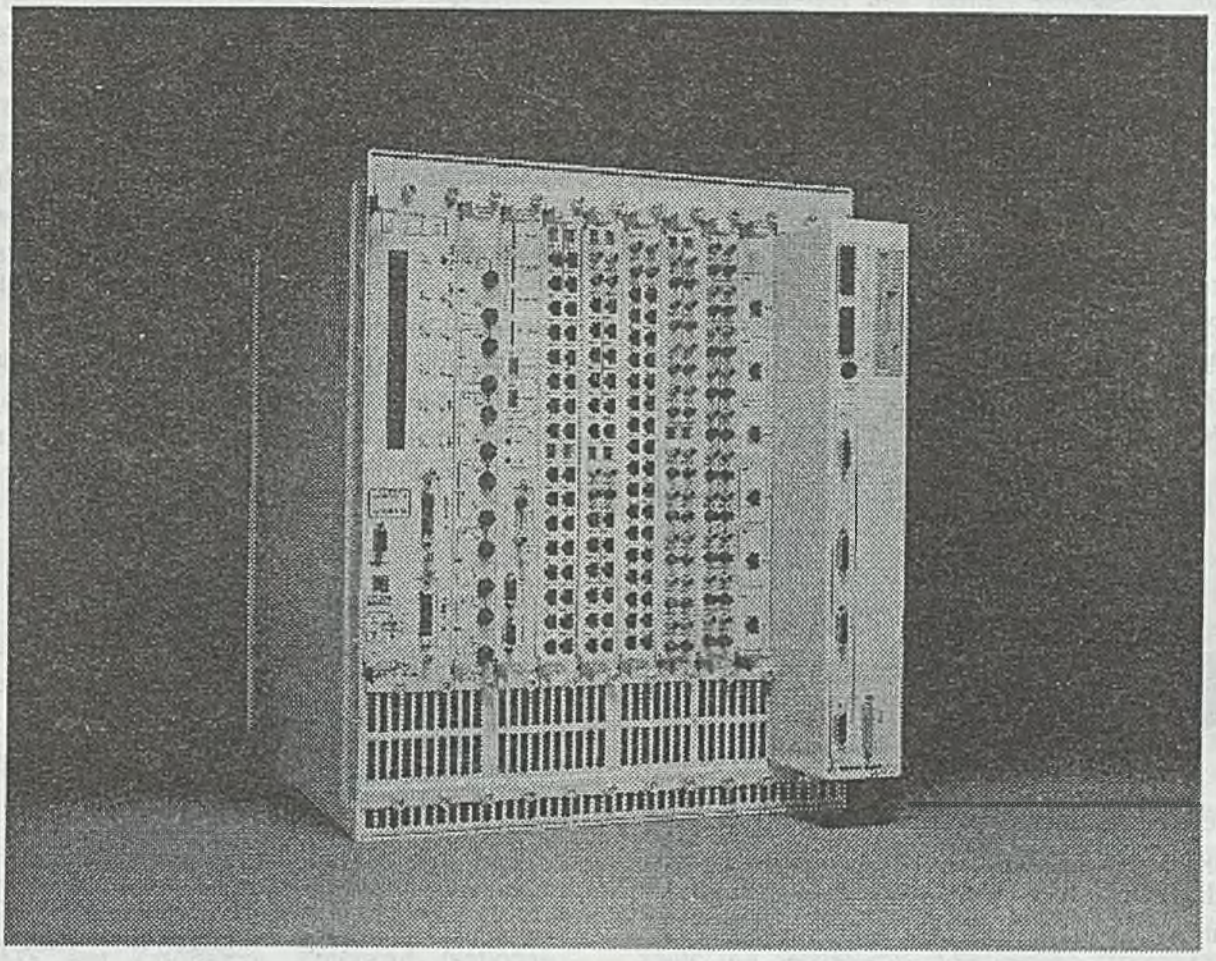
ODS od lat współpracuje z ważniejszymi producentami urządzeń sieciowych. Z tego też powodu chassis ODS mogą pomieścić moduły routerów CISCO, WellFleet; moduły serwerów komunikacyjnych Xylogics - ANNEX, CISCO; koncentratory przełączające Plaintree, Network Peripherals, moduły ATM FORE i wiele innych.

2. Rodzina urządzeń ODS INFINITY.



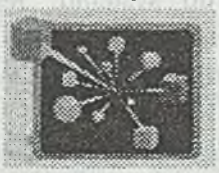
Kierunki rozwoju ODS są dyktowane zarówno wymaganiami stawianymi przez klientów jak i zmianami w technologii urządzeń sieciowych. Pod koniec 1992 roku ODS zainicjował sprzedaż hub'ów INFINITY, które pozwalają na integrację sieci Token Ring, Ethernet, FDDI i ATM w jednym chassis. Seria INFINITY zapewnia pełne bridge'owanie i route'owanie pomiędzy LAN'ami powyższych standardów jak i przyłączenie do sieci rozległych WAN (Wide Area Networks).

Pojedyńczy hub INFINITY może zapewnić do 132 portów FDDI, 352 portów Token Ring lub 528 portów Ethernet. Porty te mogą być przyłączone do dowolnego segmentu hub'a (chassis) na drodze konfiguracji programowej. Wszystkie te segmenty mogą być następnie wzajemnie route'wane lub bridge'owane (zależnie od zainstalowanego modułu). ODS dodał także możliwość pełnego odizolowania segmentu dowolnej wielkości od kręgosłupa sieci (backbone). Sieć rekonfiguruje się dynamicznie przy wykorzystaniu Simple Network Management Protocol (SNMP).



ODS chassis serii INFINITY - 1095S

ZARZĄDZANIE



Hub'ami serii INFINITY można w pełni zarządzać korzystając z popularnych pakietów UNIX'owych (SNMP), np.: HPopenView, SunNet Manager, IBM NetView /6000. Interfejs graficzny ODS'a zapewnia realistyczny obraz hub'a na ekranie komputera. Uruchomienie dowolnej ikony ODS INFINITY powoduje pojawienie się repliki hub'a INFINITY na ekranie. Kolor każdego portu

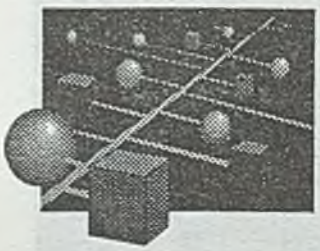
pokazuje jego aktualny stan. Dla platform PC istnieje program LanVision dla MS_Windows, LanVision dla DOS, HP_OpneView dla MS_Windows.

ODS poszerzył możliwości zarządzania sieciami poprzez zwiększenie funkcjonalności agentów pracujących w hub'ach INFINITY. Koncentratory tej serii posiadają standardowe bazy MIB (ang. Management Information Base), ODS'owe rozszerzenie bazy MIB jak i wszystkie grupy baz MIB RMON (Remote Monitoring).

Dzięki zastosowaniu hub'ów INFINITY zwiększa się znacznie bezpieczeństwo systemu, gdyż wszystkie sesje są monitorowane. Możliwa jest detekcja jakiegokolwiek dostępu z zewnątrz poprzez monitorowanie sesji ze względu na komunikację, która nie jest autoryzowana.

Jako część serii INFINITY ODS oferuje ekranowy dekoder protokołów. Pozwala on na pełne dekodowanie pakietów przechwyconych przez dowolnie odległy hub INFINITY. Uzyskane informacje mogą posłużyć do znajdowania adresu źródłowego pakietów, adresu przeznaczenia, wynegocjowanych parametrów transmisji, itp.

ETHERNET



Maksymalna liczba portów podłączonych do hub'a INFINITY wynosi 528. Każdy moduł montowany w chassis (w hub'a) daje możliwość utworzenia czterech grup WG (ang. Work Group). Każda z nich może być podłączona do dowolnego z czterech istniejących na chassis segmentów Ethernet. Każdy z tych segmentów lokalnych jest monitorowany przez agenta RMON. Grupa WG może zostać odizolowana (nie dołączona do żadnego z segmentów chassis) przez co tworzy oddzielny segment Ethernet. Dowolne z tych segmentów mogą być wzajemnie połączone poprzez router montowany jako moduł w chassis. Oczywiście na tej samej zasadzie uzyskamy połączenia do WAN.

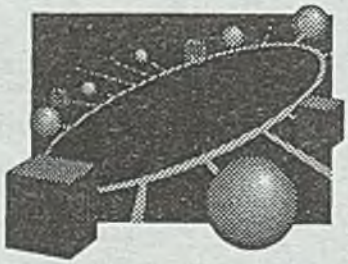
ODS zapewnia wszystkie standardowe interfejsy dla Ethernet'u: 10BaseT, 10BaseFL, 10base2 (cienki), 10Base5 (AUI); wraz z odpowiednimi przyłączami: RJ45, 50 pin'owy mass-terminated , AUI (15pin) i BNC.

Na panelu czołowym hub'a wyświetlane są wszystkie informacje o konfiguracji. Dodatkowo trzystanowe diody LED wskazują stan połączenia, ruch i kolizje dla każdego portu. Łatwo można rozpoznać, który port do której sieci fizycznej jest przyłączony. Pozwala to uniknąć problemów z adresami IP oraz ewentualnego podłączenia użytkownika do złej grupy WG.

Zapewnione jest pełne zarządzanie zgodne z protokołem SNMP dla wszystkich segmentów hub'a. Każdy instalowany moduł zarządzania ma standardowe bazy MIB plus wszystkie dziewięć grup RMON (wraz z filtrami pakietów, analizą protokołów i pełnymi detalami każdej komunikacji Ethernet'owej). Eliminuje to konieczność zakupu analizatora sieci, czy podobnych produktów programowych i sprzętowych.

ODS INFINITY Ethernet jest unikalną kombinacją technologii przełączających, analizy ruchu sieciowego i zintegrowanego route'wania.

TOKEN RING (TR)



Pojedynczy hub INFINITY zapewnia podłączenie dla 352 użytkowników z wielu oddzielnych segmentów sieci Token Ring (4 lub 16 Mb/s). Dowolny port może być przyłączony do dowolnego z czterech pierścieni logicznych. Każdy z indywidualnych segmentów jest strzeżony, dzięki czemu zwiększa się znacznie niezawodność systemu. Każdy port weryfikuje szybkość i poprawną pracę użytkownika starającego się przyłączyć do systemu. Jeśli stacja pracuje na złej szybkości transmisji lub z zepsutym adapterem to jest automatycznie odłączana. Pierścień do którego stacja ma zostać przyłączona wybierany jest programowo.

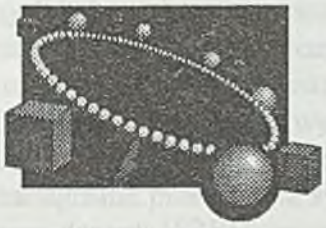
ODS INFINITY jest pierwszym chassis zapewniającym pełne przełączanie portów Token Ring'u wraz zarządzaniem SNMP i RMON MIB. Zbiera statystyki o każdym urządzeniu TR, pierścieniu czy sesji, a także zapewnia filtrowanie i przechwytywanie pakietów.

ODS zapewnia wszystkie standardowe interfejsy Token Ring'u dla: UTP, STP i Optyki; wraz z odpowiednimi przyłączami: DB9, RJ45, 50pin'owy mass terminated, ST, SMA. W przyszłości zapewnione zostaną: 802.5J i 802.5N, gdy staną się standardami.

Route'owanie i bridge'owanie pomiędzy segmentami: TR, Ethernet, FDDI, jak i sieciami rozległymi WAN jest gwarantowane.

INFINITY zapewnia dodatkowo automatyczną naprawę pierścienia po awarii co stawia go na czołowym miejscu wśród produktów standardu Token Ring. Osiągnięcia te są wynikiem wielu lat badań projektowych prowadzonych w firmie ODS.

FIBER DISTRIBUTED DATA INTERFACE (FDDI)



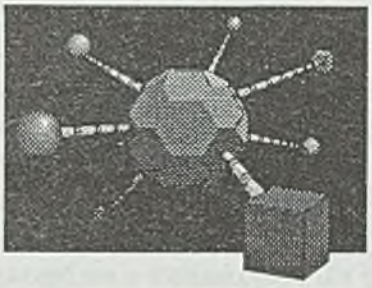
Jedno ODS INFINITY chassis zapewnia miejsce do zainstalowania maksymalnie 132 portów FDDI. Zintegrowane route'wannie pozwala na odizolowanie segmentów FDDI w celu zwiększenia ich efektywności. Router'y FDDI dokonują translacji lub kapsulacji i zapewniają połączenia z wieloma segmentami Ethernet, Token Ring i sieciami WAN (do 45Mbps - T3).

ODS jest członkiem American National Testing Center (ANTC) i University of New Hampshire (UNH) centrum testów FDDI. Urządzenia INFINITY są testowane w tych ośrodkach co zapewnia, że są w pełni zgodne z urządzeniami innych producentów. Każdy INFINITY HUB posiada pasywny segment FDDI (do czterech) i pełne zarządzanie zgodne z protokołem SNMP i SMT. Dzięki temu każdy uszkodzony element aktywnego pierścienia zostaje programowo usunięty bez zakłócania pracy pierścienia aktywnego.

Na płycie czołowej umieszczony jest wskaźnik ciekłokrystaliczny wraz z małą klawiaturą. Pozwala to na ciągłe i natychmiastowe monitorowanie parametrów (np. błędy transmisji sieciowej) jak i konfigurowanie sieci (np. wyłączanie modułów z pierścienia).

Zarządzanie siecią można realizować na wiele sposobów: SMT, SNMP poprzez Proxy Agent, Telnet jak i lokalnie SLIP. Zwiększa to znacznie swobodę wyboru sposobu zarządzania, co jest bardzo istotne w dobie tworzenia metropolitalnych sieci FDDI MAN (ang. Metropolitan Area Network). ODS jest niewątpliwym liderem w jednomodowych połączeniach FDDI na daleki dystans. Maksymalna odległość pomiędzy stacjami wynosi 60km. ODS produkuje również moduły FDDI UTP i STP - zarówno dla portów A, B i M.

ASYNCHRONUS TRANSFER MODE (ATM)



Produkty ODS'a serii ATM INFINITY zostały zaprojektowane w celu zwiększenia efektywności istniejących sieci lokalnych. Rozwiązania ODS ATM zapewniają aplikację dla przesyłania obrazu, głosu i danych czasu rzeczywistego, i współpracują ze wszystkimi istniejącymi standardami LAN. Urządzenia ODS ATM są implementacją zawansowanej technologii przełączającej bazującej na standardzie BISDN ATM (ang. matrix cell switching technlogy). Oferowane rozwiązanie może pracować z szybkościami 622Mbps. Przy matrycy ATM (switch 4x4) współpracującej z trzema obwodami OC12 (transmisja 622 Mbps) osiągnięta przepustowość wynosi 1.86 Gbps. ODS posiada moduły switch'y ATM FORE i Newbridge. Podłączenie standardów lokalnych zapewnione są poprzez instalację modułów router'a CISCO 7000 lub ExpressLAN switch FORE.

3. ATM kontra FDDI.

ODS dokończył właśnie instalację jednej z największych w świecie sieci ATM. Obsługuje ona ponad 85000 użytkowników z dwóch największych baz Armii Amerykańskiej: Fort Bragg w Północnej Kalifornii i Fort Hood w Teksasie. Kręgosłup sieci stanowi gwiazda na standardzie OC3 (155.Mb/s) podłączona do switch'a ATM. Urządzeniami peryferyjnymi sieci zapewniającymi route'owanie Ethernet/ATM są chassis ODS INFINITY z router'ami CS7000. Zainstalowano ich ponad 20. Łącznie obsługują one około 300 segmentów Ethernet. Dało to efektywne pasmo w łączności WAN dla każdego segmentu Ethernet około 7.5 Mb/s.

Przykład.

Przyjmijmy, że segmentów Ethernet jest 308, router'ów Ethernet/ATM(OC3) mamy 22. Zakładając równomierny rozkład segmentów na router'y mamy 308/22=14 segmentów przypadających na jeden router (router posiada 14 interfejsów Ethernet i 1 ATM). Każdy router łączy się bezpośrednio ze switch'em ATM linią OC3 155Mb/s. Zatem na każdy jeden segment Ethernet przypada pasmo 155/14=11,07 Mb/s. Jest to więcej niż może sam Ethernet.

Przyjmijmy natomiast, że zamiast switch'a ATM stosujemy jako kręgosłup sieci standard FDDI z 22 router'ami Ethernet/FDDI. Pasmo 100 Mb/s dzielimy wówczas pomiędzy wszystkie 308 segmentów Ethernet. W efekcie otrzymujemy 100/308 = 0.325 Mb/s na każdy segment.

Liczb tych nie trzeba komentować. Dodać tylko należy, że przy dużych instalacjach koszty budowy sieci FDDI i ATM są zbliżone.

4. Informacje o Ventus Computer.

Firma Ventus Computer jest firmą prywatną z Poznania obecną na rynku od 1989 roku. Główną naszą specjalnością są instalacje różnego rodzaju sieci komputerowych (LAN, MAN, WAN). W firmie pracuje wysoce wykwalifikowany personel techniczny rekrutujący się z absolwentów wyższych uczelni technicznych.

M. Szyprowski, Z. Królikowski, M. Januchta
Instytut Informatyki Politechniki Poznańskiej

Problemy budowy systemów multibaz danych

Streszczenie

W pracy omówiono podstawowe problemy badawcze i projektowe z dziedziny systemów multibaz danych. Przedstawiono również opracowany i wdrożony w Instytucie Informatyki Politechniki Poznańskiej prototyp systemu zarządzania multibazą danych. System ten integruje rozproszone, heterogeniczne bazy danych pracujące pod kontrolą *SZDB Oracle* i *Postgres*.

1. Wprowadzenie

W ostatnich kilkudziesięciu latach systemy plików, sieciowe, hierarchiczne i relacyjne bazy danych były wykorzystywane jako platforma do zarządzania danymi w ramach różnych aplikacji w różnych dziedzinach zastosowań. Problem tworzenia aplikacji, które wymagają dostępu do różnych heterogenicznych źródeł danych, które są zarządzane przez różne systemy zarządzania bazami danych (*SZBD*) został zdefiniowany już na początku lat 80 - tych. Jak dotąd zaproponowano kilka ogólnych metod rozwiązywania tego problemu [8].

Początkowo intensywnie badana była metoda polegająca na konwersji i migracji wszystkich wymaganych danych z jednego systemu zarządzania danymi do drugiego systemu. Genezą tutaj były prace związane z zastępowaniem systemów hierarchicznych i sieciowych baz danych systemami relacyjnymi. Niestety podejście takie nie miało charakteru uniwersalnego. Wymagało bowiem rozwiązywania szeregu problemów stricte technicznych, specyficznych dla określonych grup systemów. Przykładowo, jeśli nawet wykonanie konwersji i migracji danych z systemu A do systemu B, w celu ich przetworzenia w tym drugim systemie, jest możliwe w stosunkowo łatwy sposób, to i tak niezbędna jest konwersja w przeciwnym kierunku, danych które podczas przetwarzania w systemie B uległy modyfikacji. Co więcej, takie podejście do integracji różnych systemów charakteryzuje się niską efektywnością.

Inną metodą integracji różnych systemów baz danych, proponowaną obecnie przede wszystkim przez komercyjne firmy informatyczne, jest tworzenie bram (ang, *gateway*) pomiędzy parami różnych *SZBD*. Elementami składowymi systemów takich jak np. *Ingres*, *Oracle* czy *Sybase* są bramy do innych systemów jak np. *IMS* czy *dBase*. Rozwiązania te mają jednak istotne ograniczenie. Umożliwiają bowiem tylko translację zapytań wyrażonych w języku systemu A na równoważne zapytanie systemu B. W zdecydowanej większości przypadków bramy nie wspomagają zarządzania transakcjami dla par integrowanych systemów.

Obecnie przeważa opinia [1-3, 6, 8, 9] że najbardziej obiecującą metodą rozwiązania problemu dostępu do różnych heterogenicznych źródeł danych jest tworzenie systemów multibaz danych (ang. *Multidatabase Systems*). Zwróćmy uwagę, że w literaturze jest również używane pojęcie - systemy sfederowanych baz danych [1, 9] (ang. *Federated Database Systems*). Niewielkie różnice w interpretacji tych dwóch pojęć przedstawimy w dalszej części pracy.

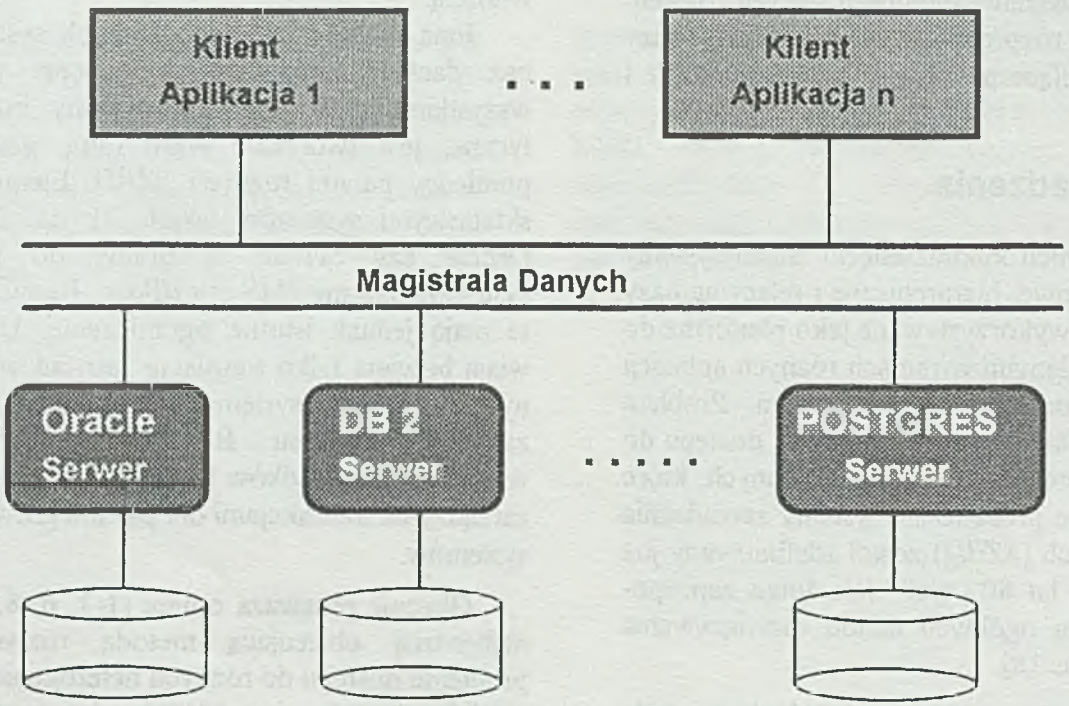
W ostatnich kilku latach obserwujemy znaczny wzrost zainteresowania i znaczenia problematyki systemów multibaz danych. Powodu ku temu są następujące [8].

Liczba różnego rodzaju aplikacji rozproszonych baz danych rośnie w ostatnich latach bardzo szybko. Do ich budowy stosowane są różne systemy zarządzania bazami danych, jak np. Oracle, Ingres, Sybase, Progress, czy Informix. Rośnie również zainteresowanie eksperymentalnymi SZBD, na przykład systemem Postgres [4, 5]. Na świecie istnieje obecnie wiele instalacji systemów baz danych znacznie różniących się między sobą. Coraz wyraźniej zarysowywuje się więc potrzeba integracji różnych, heterogenicznych systemów w ramach komputerowego systemu rozproszonego [1, 8]. Przewidywany na lata 90 - te scenariusz rozwoju systemów baz danych zakłada [1]:

- rozwój stacji roboczych (~ 100 MIPS) na których będą instalowane lokalne bazy danych,
- istnienie różnych modeli danych (np. relacyjne i obiektowe),
- istnienie różnych interfejsów z użytkownikiem.

Zakłada się również, że rozwój tych systemów będzie zmierzał w kierunku powstawania systemów, których ogólną strukturę przedstawiono na rys. 1.

W tym kontekście, coraz wyraźniej zarysowuje się potrzeba budowy heterogenicznych systemów multibaz danych. Stanowi ona obecnie ważny postulat badawczy [1, 3, 6, 8].



Rys. 1. Przewidywana struktura systemów baz danych lat 90 - tych.

Przez system multibaz danych (lub inaczej system sfederowanych baz danych) rozumiemy tutaj system składający się z co najmniej dwóch niezależnych, heterogenicznych systemów baz danych oraz mechanizmu konsolidującego wszystkie ich komponenty. Poszczególne systemy baz danych należące do systemu sfederowanego są całkowicie niezależne od siebie i mogą pracować jako samodzielne jednostki. Integrowane w ramach takiego systemu, systemy baz danych mogą różnić się w wielu aspektach, np. językiem zapytań, sposobem utrzymywania danych i konstrukcją schematu bazy danych [8].

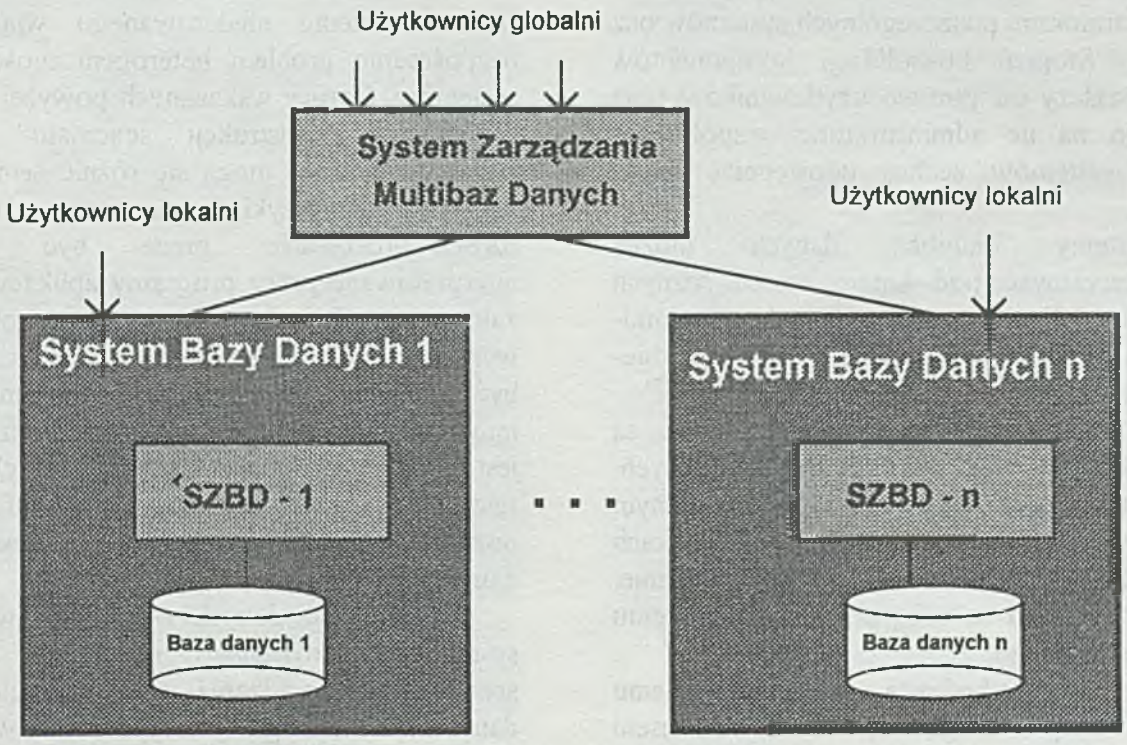
Wykorzystywanie systemów multibaz danych może mieć niemalże nieograniczony zasięg. Mogą gwarantować one dostęp do każdej informacji nie tylko jak dotychczas,

niezależnie od ich geograficznego położenia, ale również niezależnie od sposobu utrzymywania danych i od języka komunikacji z systemem. Może prowadzić to do powstania globalnie sfederowanych systemów baz danych. Każdy system składowy może być zorientowany na określoną dziedzinę rzeczywistości. Te same pojedyncze systemy baz danych mogą należeć do różnych sfederowanych systemów baz danych ukierunkowanych na inny charakter danych. Te z kolei można łączyć w kolejne systemy, tworząc coraz szersze i bogatsze (w rozumieniu tematyki informacji) systemy multibaz danych. Możliwości i zakres zastosowań systemów multibaz danych są więc potencjalnie bardzo szerokie. Jednakże, aby wykorzystać możliwości tych systemów należy rozwiązać szereg istotnych problemów związanych z ich budową. W ostatnich latach prowadzone są intensywne prace badawcze w wielu ośrodkach naukowych na świecie, mające na celu rozwiązanie tych problemów [1, 8]. Prowadzone prace bazują w dużej mierze na konstrukcjach systemów prototypowych.

W Instytucie Informatyki Politechniki Poznańskiej również podjęto próbę budowy prototypu systemu multibaz danych w oparciu o *SZDB Oracle* [7] oraz eksperymentalny *SZDB Postgres* [4, 5]. Opracowanie koncepcji i budowa takiego prototypu stanowiły główny cel pierwszego etapu tych prac.

2. Problemy budowy systemów multibaz danych

System multibaz danych (*SMBD*) jest zbiorem współpracujących ze sobą, ale niezależnych systemów baz danych *(SBD)*, przy czym składowe *SBD* mogą być integrowane w różny sposób. Oprogramowanie, które zapewnia zsynchronizowane zarządzanie komponentami *SMBD* jest nazywane systemem zarządzania multibazą danych (*SZMBD*) [8]. *SZMBD* umiejscowiony na szczycie istniejących lokalnych systemów baz danych (rys. 2.), stwarza użytkownikom iluzję jakby pracowali w jednym wielkim systemie. System zarządzania multibazą danych może utrzymywać globalny schemat bazy danych, z którego korzystają użytkownicy wydając zapytania i transakcje. Schemat globalny jest tworzony przez konsolidację schematów lokalnych baz danych.



Rys. 2. Architektura systemu multibaz danych.

Właśnie z kwestią schematu globalnego związane są różnice w interpretacji pojęć: system sfederowanych baz danych a system multibaz danych. Część autorów uważa [1], że systemy sfederowane to systemy w których dostęp do rozproszonych heterogenicznych danych jest wykonywany właśnie na podstawie schematu globalnego, natomiast w systemach multibaz danych taki schemat w formie bezpośredniej nie występuje. Nie mniej jednak, w tego typu systemach, musi istnieć taka czy inna forma integracji schematów lokalnych baz danych (niekoniecznie musi to być schemat globalny [2]). Uważamy, że z tego punktu widzenie, różnice w interpretacji tych pojęć są nieistotne i oba te pojęcia mogą być wykorzystywane zamiennie.

Wszystkie dane w systemie multibaz danych są utrzymywane i zarządzane przez lokalne systemy zarządzania bazami danych. Tak więc, jedną z podstawowych własności *SMBD* jest to, że składowe, lokalne systemy baz danych mogą jednocześnie przeprowadzać swoje "prywatne" operacje oraz uczestniczyć w pracy systemu sfederowanego. Integracją komponentów *SMBD* mogą zajmować się zarówno użytkownicy systemu jak i administrator *SMBD* we współpracy z administratorami poszczególnych systemów baz danych. Stopień konsolidacji komponentów *SMBD* zależy od potrzeb użytkowników oraz od tego na ile administratorzy współpracujących systemów zechcą udostępniać swoje zasoby.

Systemy multibaz danych można charakteryzować pod kątem trzech różnych zagadnień [8]: rozproszenia danych, autonomiczności podsystemów oraz ich heterogeniczności.

W systemie sfederowanym dane są pamiętane w wielu różnych bazach danych. Bazy te mogą znajdować się na jednym komputerze lub na wielu stanowiskach komputerowych rozproszonych geograficznie, ale połączonych ze sobą przy użyciu systemu komunikacyjnego.

Systemy wchodzące w skład systemu sfederowanego mogą się różnić zakresem autonomiczności komunikacji, wykonywania operacji i projektowania. System bazy danych z komunikacją autonomiczną jest w stanie sam decydować kiedy i jak reagować na żądania pochodzące od innych *SBD*. Autonomiczność wykonywania operacji pozwala systemowi baz danych na samodzielne decydowanie, na przykład o kolejności wykonywania operacji, bez potrzeby komunikowania się z *SZMBD*. Co więcej, *SZMBD* nie jest w stanie ingerować w operacje lokalnego *SZBD*.

Autonomiczność projektowa jest interpretowana jako możliwość wyboru własnego sposobu konstrukcji schematu bazy danych przez jej projektanta. Ilustrację tego problemu stanowi następujący przykład.

Przykład:

Rozważmy relację zawierającą dane osobowe pracowników. Relacja taka powinna zawierać między innymi atrybuty nazwisko pracownika i jego adres. Schemat takiej relacji może być zaprojektowany w różny sposób. Na przykład,

Pracownicy_1 = (nazwisko, miasto, ulica, nr_domu) lub

Pracownicy_2 = (nazwisko, adres),

gdzie atrybut *adres* relacji *Pracownicy_2* zawiera w istocie takie same dane, co atrybuty *miasto, ulica, nr_domu* relacji *Pracownicy_1*.

Z problemem autonomiczności *SBD* w ramach systemu sfederowanego wiąże się bezpośrednio problem heterogeniczności tych systemów. Oprócz wskazanych powyżej różnic dotyczących konstrukcji schematu relacji, systemy składowe mogą się różnić semantyką danych. Na przykład, znaczenie atrybutu *zarobki_pracownika* może być różnie interpretowane przez programy aplikacyjne, w zależności od waluty w której zarobki są wyrażone (w jednym systemie zarobki mogą być podawane w złotówkach, natomiast w innym w dolarach). Wykrycie tego typu różnic jest trudnym problemem, ponieważ schematy relacji z reguły nie zawierają informacji pozwalających na interpretację znaczenia danych.

Systemy baz danych wchodzące w skład systemu sfederowanego mogą się różnić sposobami zarządzania i manipulowania danymi. *SZBD* mogą być wyposażone w różne języki zapytań. Mogą to być całkiem różne języki (Quel, Postquel, SQL, Query by

Example) albo jedynie różne wersje tego samego języka (np. SQL w Oracle 6 a Oracle 7). Na przykład, *SZBD* Oracle [7] wykorzystuje język zapytań SQL (Structured Query Language), którego podstawową konstrukcją jest

SELECT
nazwa_relacji.nazwa_atrybutu
FROM *nazwa_relacji* WHERE *warunki*
.....

natomiast, w systemie Postgres jest wykorzystywany jest język Postquel [4]. Składnia zapytań języka Postquel jest kombinacją języka Quel (z systemu Ingres) oraz SQL. Polecenie języka Postquel odpowiadające poleceniu SELECT z języka SQL ma postać:

RETRIEVE
[INTO nazwa_relacji]
FROM t_1 *in* R_1, ..., t_n *in* R_n WHERE *<warunki>*
[..... [USING *operator*]]

Interpretacja struktury *retrieve* jest następująca. Jeżeli sekwencja *into nazwa_relacji* nie występuje, to domyślnie wynik selekcji jest przesyłany do aktywnego buforu (*portala*), natomiast aby zachować te dane należy podać nazwę relacji. *Warunki* tworzą tak zwaną *kwalifikację zapytania* t_i i nazywane są *surogatami* (ang. *surrogates*) lub inaczej zmiennymi krotki, tj. elementami zastępującymi wystąpienie obiektu. Są one podstawą przeszukiwania klasy (w ujęciu tradycyjnym - relacji). Za każdym razem gdy warunki są spełnione, jest formowana - na podstawie listy atrybutów - nowa krotka wstawiana do relacji wynikowej. Lista atrybutów tworzy tak zwaną *specyfikację klasy wynikowej* (ang. *target list*).

Podsumowując, system zarządzania multibazą danych powinien umożliwiać dostęp (odczyt i aktualizację) do wielu heterogenicznych, autonomicznych baz danych. Zadaniem *SZMBD* jest również translacja i zarządzanie zapytaniami i transakcjami globalnymi, ich kierowanie do odpowiednich systemów lokalnych, oraz zbieranie, scalanie i wysyłanie wyników tych operacji do użytkowników korzystających z systemu. *SZMBD* powinien zawierać następujące mechanizmy:

- uniwersalny język manipulowania danymi w różnych lokalnych bazach danych [8],
- integrację schematów lokalnych baz danych, innymi słowy, mechanizm definiowania powiązań i zależności pomiędzy danymi które mogą pochodzić z różnych lokalnych baz danych [3, 6, 8, 9, 10],
- utrzymywanie globalnej spójności w systemie multibaz danych [8, 9].

3. Prototypowy system zarządzania multibaz danych Oracle i Postgres

W ramach prac nad budową *SZMBD* prowadzonych w Instytucie Informatyki Politechniki Poznańskiej podjęto próbę rozwiązania szeregu problemów przedstawionych w rozdziale 2 tej pracy, przy czym za problem podstawowy uznano integrację schematów lokalnych baz danych. W literaturze zaproponowano do tej pory wiele rozwiązań tego problemu. Metoda wykorzystywana w przedstawionym poniżej systemie *PO-DM* stanowi rozwinięcie propozycji przedstawionej w [2]. W systemie *PO-DM* zastosowano graficzną metodę "*on the fly*" integracji schematów lokalnych baz danych.

W ramach wspomnianych powyżej prac, jako elementy składowe *SZMBD* zaprojektowano i zaimplementowano:

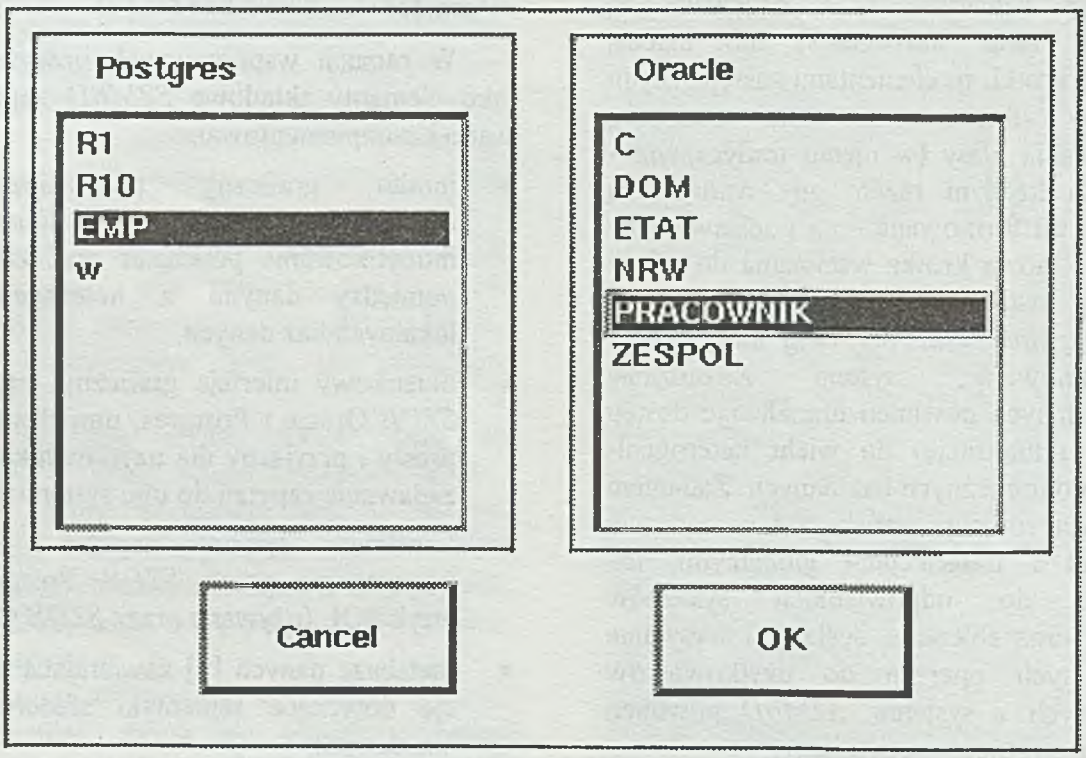
- moduł graficzny pozwalający na dynamiczne ("*on the fly*") definiowanie i modyfikowanie powiązań oraz zależności pomiędzy danymi z heterogenicznych lokalnych baz danych,
- okienkowy interfejs graficzny integrujący *SZDB* Oracle i Postgres, umożliwiający w prosty i przyjazny dla użytkownika sposób zadawanie zapytań do obu systemów,
- translator języka zapytań Postquel (używanego przez *SZDB* Postgres) na język SQL (używany przez *SZDB* Oracle),
- metabazę danych [9] zawierającą informacje dotyczące semantyki sfederowanych baz danych,

- procedury dokonujące odwzorowania struktur sfederowanych baz danych,
- procedury wykorzystujące metabazę dla konwersji pomiędzy sfederowanymi bazami lokalnymi,
- bibliotekę funkcji umożliwiających programistom tworzenie aplikacji w systemie multibaz danych.

Zbudowany prototyp systemu zarządzania multibaz danych Oracle i Postgres pracuje pod kontrolą systemu operacyjnego UNIX na komputerach Sun 4 firmy SunMicrosystems. Do budowy interfejsu z użytkownikiem systemu sfederowanego oraz modułów integrujących oba *SZDB* wykorzystano system X-Windows oraz interfejs graficzny Motif. Z punktu widzenia użytkownika prototyp jest widziany jako dwa narzędzia programowe ściśle współpracujące ze sobą. Pierwsze, to okienkowy interfejs graficzny z użytkownikiem wraz z narzędziami służącymi do definiowania różnic pomiędzy integrowanymi systemami baz danych (*PO-DM*, od ang. *Postgres/Oracle Data Manager*). Drugie narzędzie natomiast, to biblioteka funkcji umożliwiających programiście równoległą współpracę z komponentami *SMBD*.

Prototyp może współpracować z bazą danych zarządzaną przez system Postgres oraz drugą zarządzaną przez system Oracle. Zakłada się, że bazy te zostały wcześniej utworzone. Zbudowany prototyp *SZMBD* umożliwia oczywiście również lokalną, autonomiczną pracę zintegrowanych systemów. Moduł *PO-DM* zarządzający systemem multibaz danych jest systemem okienkowym, służącym administratorowi *SZMBD* do nawiązania łączności pomiędzy zintegrowanymi systemami oraz do zdefiniowania różnic pomiędzy komponentami systemu. Umożliwia on również zadawanie zapytań do obu baz danych.

Zastosowane w systemie PO-DM rozwiązanie problemu zawiązania łączności pomiędzy integrowanymi systemami baz danych rozważmy na przykładzie relacji o takiej samej strukturze. Należy wówczas jedynie powiązać te dwie relacje, tak aby system heterogeniczny traktował je jako równoważne. W dalszej części przez powiązanie relacji pochodzących z różnych komponentów systemu sfederowanego nazywać będziemy wiązaniem lub konwersją. Do tego celu służy okno dialogowe przedstawione na rys. 3.

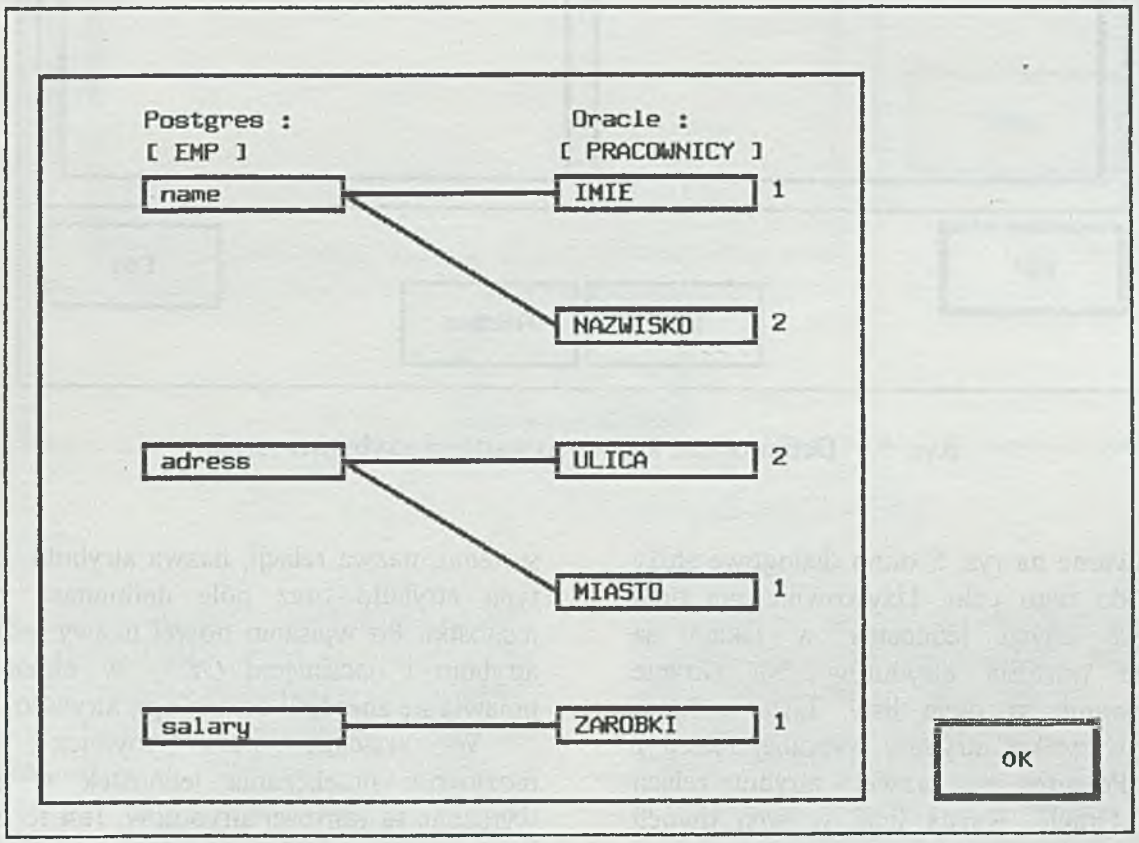


Rys. 3. Okno dialogowe wiązania relacji w systemie PO-DM.

W oknie są prezentowanie dwie listy. Pierwsza z nich - zawiera wszystkie dostępne relacje systemu Postgres, a druga - systemu Oracle. Tworzenie nowego powiązania następuje po wyselekcjonowaniu po jednej relacji z obu list.

System PO-DM pozwala również na definiowanie różnic strukturalnych relacji. Może to być różna ilość atrybutów relacji, nazwy samych atrybutów mogą być różne, jak również jednemu atrybutowi relacji *R1* może odpowiadać kilka atrybutów relacji *R2* (patrz przykład w rozdziale 2).

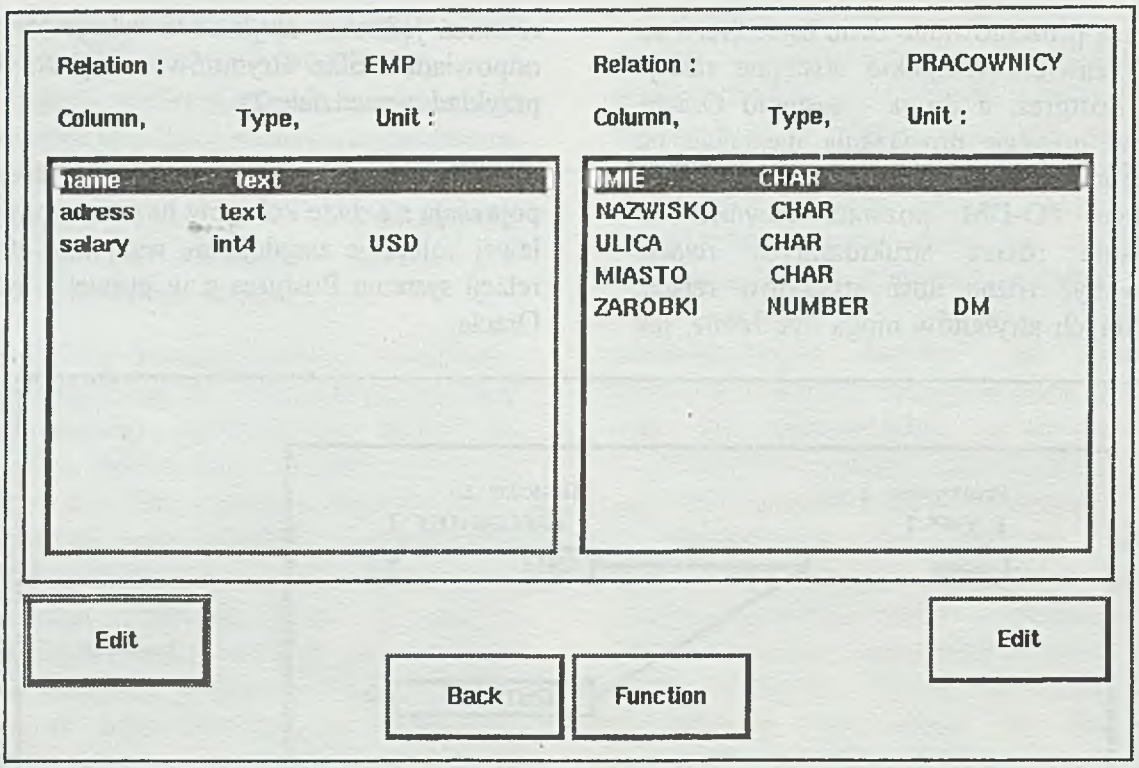
Po otwarciu okna *Alias* (rys.4) na ekranie pojawiają się dwie kolumny nazw atrybutów: w lewej kolumnie znajdują się wszystkie atrybuty relacji systemu Postgres a w prawej - systemu Oracle.



Rys. 4. Definiowanie różnic strukturalnych relacji w systemie PO-DM

Połączenie prostokątów określających atrybuty jest równoznaczne z określeniem równoważności tych atrybutów.

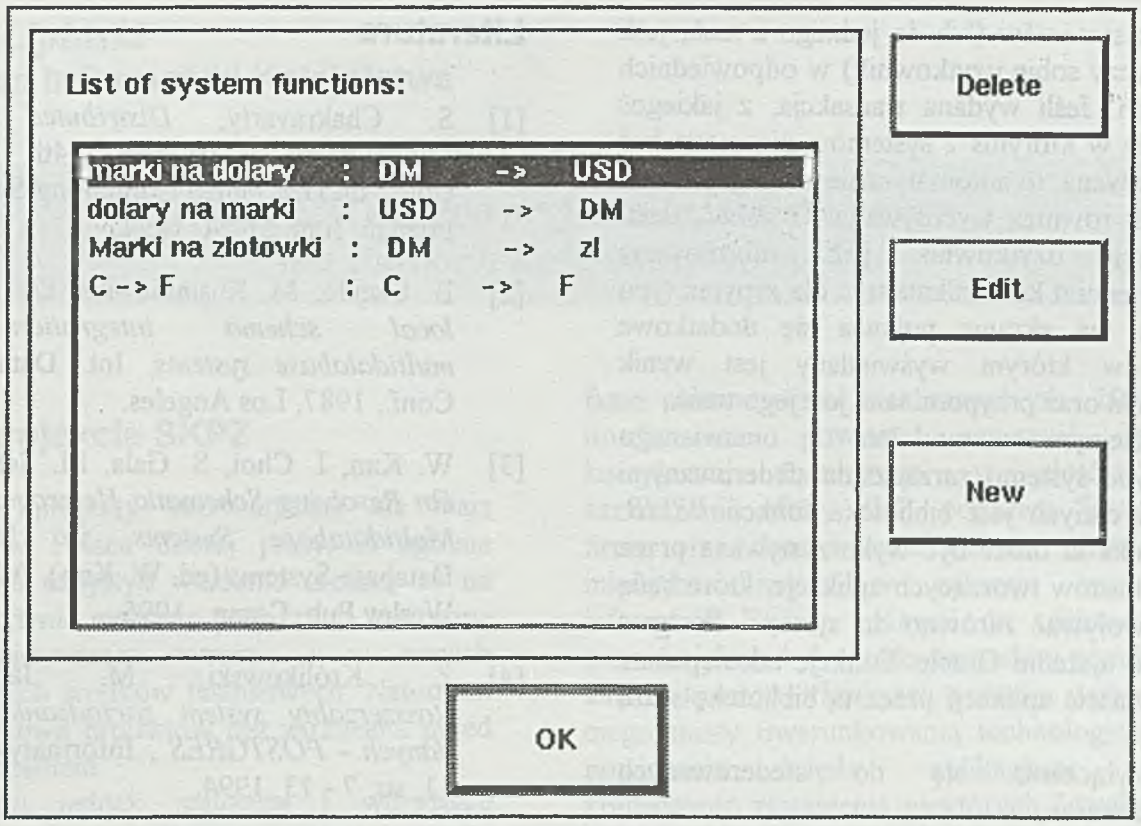
Zdefiniowane w ten sposób różnice w strukturach baz danych i sposoby konwersji tych różnic mogą być aktualizowane i modyfikowane w trakcie eksploatacji systemu. Wykorzystuje się w tym celu, tak zwany edytor konwersji. W uproszczeniu, służy on do edycji już istniejących i tworzenia nowych konwersji (powiązań) pomiędzy parami relacji, tj. do określania związków pomiędzy atrybutami i różnic w znaczeniu tych atrybutów. Użytkownik może definiować funkcje dokonujące konwersji wartości atrybutów, może również funkcje te modyfikować i usuwać.



Rys. 5. Definiowanie konwersji wartości atrybutów relacji

Przedstawione na rys. 5 okno dialogowe służy właśnie do tego celu. Użytkownik ma tutaj możliwość edycji jednostek w jakich są wyrażane wartości atrybutów. Na ekranie prezentowanie są dwie listy. Lista z lewej zawiera wszystkie atrybuty wybranej relacji z systemu Postgres, a w prawej - atrybuty relacji systemu Oracle. Każda linia w tych dwóch listach składa się z nazwy relacji, nazwy atrybutu i typu atrybutu natomiast nazwa jednostki jest początkowo pusta. Aby można było wykonać edycję nazwy jednostki, użytkownik musi wybrać określony wiersz i przycisk *EDIT*. Wtedy otwiera się okno zawierające następujące informacje: nazwa systemu, nazwa relacji, nazwa atrybutu, nazwa typu atrybutu oraz pole definiujące nazwę jednostki. Po wpisaniu nowej nazwy jednostki atrybutu i naciśnięciu *OK* - w oknie *Unit* pojawia się zmodyfikowany opis atrybutu.

W systemie musi również istnieć możliwość przeliczania jednostek w jakich wyrażone są wartości atrybutów. Jest to istotne w przypadku wydawania zapytań do kilku zintegrowanych baz danych. W systemie PO-DM problem ten rozwiązano przez wprowadzanie możliwości definiowania funkcji dokonujących odpowiednich konwersji. Konkretnie służy do tego okno dialogowe *Function Manager* przedstawione na rys. 6.



Rys. 6. Definiowanie funkcji dokonujących konwersji

Wyświetlane są wszystkie dostępne w systemie funkcje dokonujące konwersji. Pojedynczy element listy zawiera informacje o tym z jakiej i na jaką jednostkę dana funkcja konwersji dokonuje. Okno to daje również możliwość usuwania z systemu nieaktualnych funkcji, dodawania nowych oraz modyfikacji istniejących.

Dodawanie nowych i modyfikacja istniejących funkcji odbywa się w oknie *Edit Function*. W tym oknie użytkownik może określać przelicznik jaki ma być stosowany przy wyprowadzaniu danych. Okno zawiera informacje o nazwach atrybutów i ich typy. Użytkownik może wprowadzić nazwę funkcji, oraz samą funkcję. W tej wersji systemu PO-DM funkcje służą tylko do przeliczania wartości liczbowych. Format opisu funkcji jest ściśle określony:

y = x [ciało funkcji]

gdzie ciało funkcji może być ciągiem wyrażeń typu operator, operand. Operatory mogą być: +, -, *, /. Natomiast operand może być dowolną liczbą rzeczywistą lub całkowitą. Kolejność wykonywania operacji jest od lewej do prawej. Wprowadzona funkcja jest wpierw analizowana pod względem poprawności zapisu. Wykorzystywana funkcja jest ewaluowana podczas przeliczania danych, a pod symbol 'x' podstawiana jest wartość wejściowa.

Oprócz funkcji przedstawionych powyżej, system *PO-DM* umożliwia zadawanie zapytań do systemu heterogenicznych multibaz danych. Zapytania powinny być wyrażone w języku Postquel. Użytkownik może określić czy zapytanie ma być skierowane do obu systemów baz danych czy tylko do jednego z nich. W tym drugim przypadku, powinien dokonać wyboru systemu do którego zapytanie ma być wydane. Zapytania nie muszą być każdorazowo wprowadzane od nowa - można korzystać z wprowadzonych wcześniej zapytań, pamiętanych w odpowiednich tablicach systemu *PO-DM*.

Podczas translacji zapytań wykonywana jest ich analiza składniowa. W przypadku wystąpienia błędów, translacja zostaje przerwana i informowany jest o tym użytkownik. Zapytania są następnie wysyłane

do obu systemów (lub do jednego z nich, jeśli tego życzy sobie użytkownik) w odpowiednich językach. Jeśli wydana transakcja, z jakiegoś powodu w którymś z systemów nie może być zrealizowana, to automatycznie w drugim systemie jest również wycofywana. O zakończeniu transakcji użytkownik jest informowany odpowiednim komunikatem, a dla zapytań typu *retrieve* na ekranie pojawia się dodatkowe okno, w którym wyświetlany jest wynik zapytania oraz przypominana jest jego treść.

Kolejnym modułem omawianego prototypu systemu zarządzania sfederowanymi bazami danych jest biblioteka funkcji *PSLIB*. Biblioteka ta może być wykorzystywana przez programistów tworzących aplikacje, które będą się odwoływać zarówno do systemu Postgres jak i do systemu Oracle. Funkcje udostępniane programiście aplikacji przez tą bibliotekę służą do:

- przyłączania się do sfederowanych systemów,
- wyboru konwersji,
- wydawania zapytań,
- zapewnienia dostępu do zwracanych wartości krotek wynikowych,
- udostępnia struktury, informacyjne dla zapewnienia pełnej kontroli nad systemem.

4. Podsumowanie

W pracy przedstawiono podstawowe problemy badawcze i projektowe dotyczące problematyki systemów heterogenicznych multibaz danych. Przedstawiono również opis prototypu takiego systemu, zaprojektowanego i wdrożonego w Instytucie Informatyki Politechniki Poznańskiej. Ze względu na ograniczoną ilość miejsca ograniczono się tylko do zasygnalizowania niektórych problemów oraz do opisu wdrożonego systemu od strony użytkowej. Pominięto większość aspektów technicznych implementacji. System oraz jego pełna dokumentacja są dostępne dla użytkownika *anonymous ftp* na serwerze *syriusz.cs.put.poznan.pl* w kartotece *pub/PODM*.

Literatura

[1] S. Chakravarty, *Distributed Data Management - Tutorial,* 14th Intern. Conf. on Distributed Computing Systems, Poznań, June 21-24, 1994.

[2] B. Czejdo, M. Rusinkiewicz, *On the fly local schema integration in multidatabase systems*, Int. Data Eng. Conf., 1987, Los Angeles.

[3] W. Kim, I. Choi, S. Gala, M. Scheevel, *On Resolving Schematic Heterogenity in Multidatabase Systems*, in: Modern Database Systems (ed. W. Kim), Addison-Wesley Pub. Comp., 1995.

[4] Z. Królikowski, M. Januchta, *Rozszerzalny system zarządzania bazą danych - POSTGRES* , Informatyka, Nr. 11, str. 7 - 13, 1994.

[5] Z. Królikowski, M. Januchta, *POSTGRES jako narzędzie budowy aktywnych i dedukcyjnych baz danych,* Informatyka, Nr. 12, str. 4 - 11, 1994.

[6] A. Motro, *Virtual Integration of Multiple Databases,* IEEE Transaction on Software Engineering, vol. SE-13, No. 7, 1987.

[7] *Oracle Utilities User's Guide,* Oracle Corporation, Redwood Shores, 1992.

[8] M. Rusinkiewicz, *Multidatabase Systems: Technology and Applications - Tutorial,* 14th Intern. Conf. on Distributed Computing Systems, Poznań, June 21-24, 1994.

[9] A. Sheth, J. Larson, *Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases,* ACM Comp. Surveys, v. 22, No.3, 1990.

[10] M. Siegel, S. Madnick, *A Metadata Approach to Resolving Semantic Conflicts,* in: Proc. of the 17th Intern. Conference on Very Large Databases, Barcelona, 1991.

Adam Lipiński
Centrum Informatyki Kolejnictwa

Realizacja dużego projektu informatycznego dla Polskich Kolei Państwowych

1. O projekcie SKPZ

Duże projekty informatyczne nie mają ostatnio w Polsce dobrej prasy. Z ogólnie dostępnych statystyk wiadomo zresztą, że na świecie jedynie nieliczne projekty są kończone w przewidywanym terminie i w ramach planowanych środków finansowych. Natomiast ponad połowa projektów jest zarzucana przed ich ukończeniem.

Często jednak realizacja i wdrożenie dużego systemu informatycznego jest dla dużego przedsiębiorstwa jedynym sposobem zapewnienia sprawnego zarządzania, kontrolowania kosztów i wpływów oraz długoterminowego planowania działalności.

Takim systemem jest realizowany obecnie przez PKP projekt SKPZ (System Kierowania Przewozami i Zarządzania), obejmujący swoim zakresem większość obszarów działalności Polskich Kolei Państwowych. Opracowanie systemu jest ściśle powiązane z wprowadzeniem zmian w funkcjonowaniu przedsiębiorstwa i jego restrukturyzacją. Podstawowym celem całego przedsięwzięcia jest zwiększenie jakości usług, zmniejszenie kosztów eksploatacyjnych kolei oraz poprawa ogólnego wyniku finansowego.

Po etapie opracowań studialnych, w końcu 1992 r. rozpoczęto prace nad stworzeniem logicznej architektury systemu, obejmującej podział na poszczególne aplikacje oraz określenie kolejności ich realizacji, wynikającej zarówno z uwarunkowań technicznych, jak i z wykonanej analizy kosztów i korzyści. Prace te były prowadzone według metodyki Summit-S (Coopers & Lybrand). Zdefiniowanych zostało ponad 80 podsystemów, z których w pierwszej fazie postanowiono zrealizować ok. 20. Ich uruchomienie powinno zapewnić, poza korzyściami wymiernymi w złotówkach, rozszerzenie oferty PKP, poprawę "image'u" firmy oraz dostosowanie się do wymogów międzynarodowych w zakresie wymiany informacji. Wybór aplikacji do realizacji w pierwszej fazie był przede wszystkim wynikiem analizy kosztów i korzyści; ponadto wpływ na niego miały uwarunkowania technologiczne i powiązania między aplikacjami, oraz konieczność zastąpienia niektórych "wyeksploatowanych" systemów, które są niezwykle trudne i kosztowne w utrzymaniu.

2. Środowisko techniczne systemu

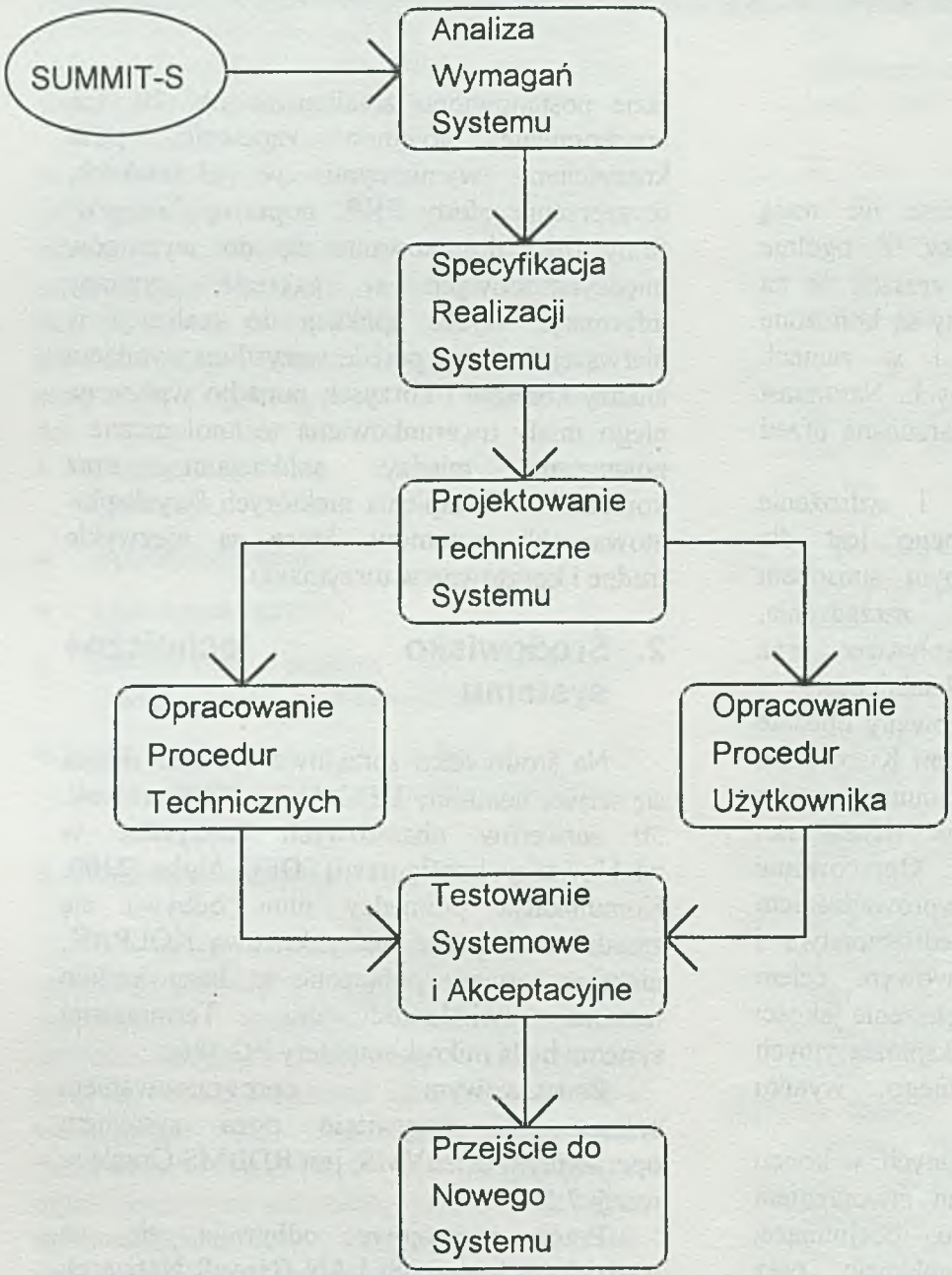
Na środowisko sprzętowe systemu składa się serwer centralny DEC-Alpha 7000 oraz ok. 30 serwerów obszarowych (wszystkie w zdublowanej konfiguracji) DEC-Alpha 2100. Komunikacja pomiędzy nimi odbywa się zasadniczo poprzez sieć pakietową KOLPAK; niektóre serwery połączone są bezpośrednio łączami światłowodowymi. Terminalami systemu będą mikrokomputery PC 486.

Podstawowym oprogramowaniem systemowym, oczywiście poza systemem operacyjnym OpenVMS, jest RDBMS Oracle w wersji 7.1.

Prace rozwojowe odbywają się na wydzielonych sieciach LAN (Novell Netware), na których zainstalowane są narzędzia Systems Engineer i Oracle. Testowanie i strojenie gotowych aplikacji będzie się odbywać już w środowisku produkcyjnym.

3. Metodyka realizacji systemu

Prace nad systemem są prowadzone zgodnie z metodyką Summit-D, zakupioną od Coopers & Lybrand (UK). Summit-D składa się z 7 podstawowych modułów, obejmujących cały cykl życia systemu, oraz dwóch dodatkowych, wykorzystywanych m.in. w procesie zakupu gotowych pakietów.



Rys. 1 Schemat modułów Summit-D

Summit-D jest metodyką uniwersalną, wyznaczającą ogólne ramy prowadzenia projektu, i nie narzucającą stosowania konkretnych metod, technik czy narzędzi. W ramach poszczególnych modułów można więc posługiwać się rozmaitymi technikami i narzędziami; ważne jest, by każdy z modułów zakończył się przewidywanym rezultatem.

W przypadku analizy wymagań (moduły AWS i SRS) posługujemy się dwiema podstawowymi technikami: diagramami przepływu danych i modelami związków encji. Przyjęta metoda ich budowania polega na:

- skonstruowaniu kontekstu systemu
- zbudowaniu fizycznego modelu przepływu danych
- wyodrębnieniu zdarzeń
- zbudowaniu pojęciowego modelu procesów poprzez analizę zdarzeń
- zbudowaniu pojęciowego modelu danych, powiązanego z pojęciowym modelem procesów.

Podstawowym narzędziem, przy pomocy którego budowane są diagramy przepływu danych i modele związków encji, jest Systems Engineer brytyjskiej firmy LBMS.

Projektowanie aplikacji oraz generowanie kodu odbywa się zasadniczo przy pomocy narzędzi Oracle (Oracle CASE*Dictionary 5.1, Oracle CASE*Generators, Oracle Forms 4 i Oracle Reports 2). Oczywiście, narzędzia są w stanie wygenerować automatycznie jedynie część kodu; pozostała część jest dopisywana ręcznie. Niemniej, dla celów dokumentacyjnych, w repozytoriach obu narzędzi (tj, Systems Engineer i CASE Dictionary) przechowywane są pełne wersje modeli.

Przejście z Systems Engineer do środowiska Oracle odbywa się w sposób częściowo zautomatyzowany. Modele danych są przenoszone automatycznie; modele procesów ręcznie, przy czym diagramy przepływu danych są przenoszone na modele hierarchii funkcji i użycia encji przez funkcje.

4. Metoda Rapid Application Development (RAD)

W 1993 r. rozpoczęto analizę wymagań dla pierwszych aplikacji. Dotyczą one obsługi zagadnień związanych m.in. z ewidencją taboru i opisem sieci kolejowej, pracą eksploatacyjną na stacjach, gospodarką wagonami należącymi do obcych zarządów kolejowych, tworzeniem rozkładów jazdy oraz handlową obsługą klienta. Podstawowe definicje aplikacji, określające ich zakres oraz cel ich realizacji, pochodzą ze studium strategicznego.

W końcu III kwartału 1994 r. zakończona została szczegółowa analiza wymagań m.in. dla dwóch niewielkich aplikacji (Ewidencja Wagonów, EWAG i Umowy Akwizycyjne, UMAK). Dalsze prace, tj. projektowanie, tworzenie oprogramowania i testowanie tych dwóch aplikacji poprowadzono metodą RAD (Rapid Application Development).

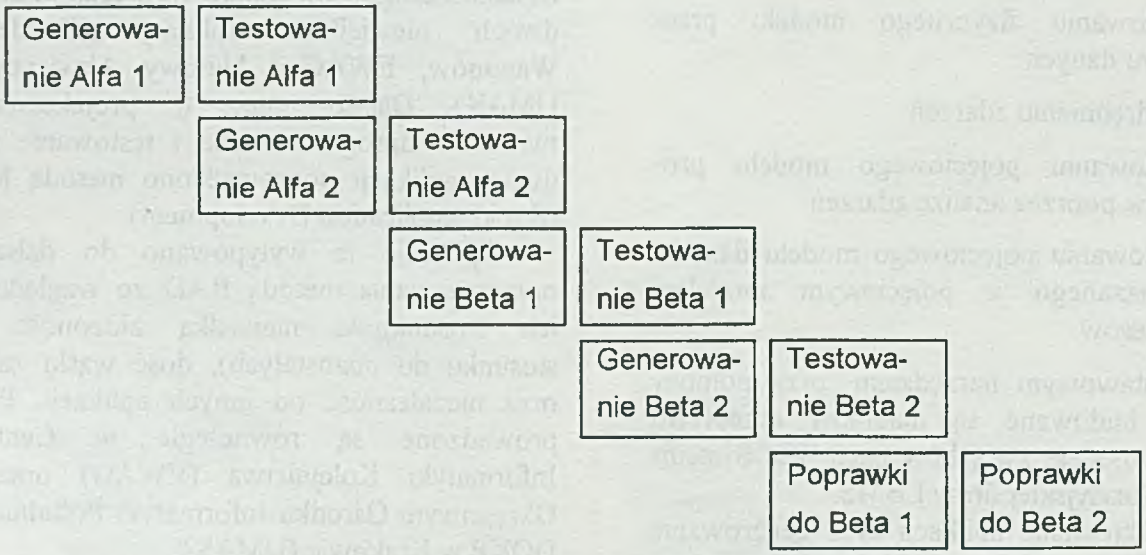
Aplikacje te wytypowano do dalszego opracowywania metodą RAD ze względu na ich stosunkowo niewielką złożoność (w stosunku do pozostałych), dość wąski zakres oraz niezależność od innych aplikacji. Prace prowadzone są równolegle w Centrum Informatyki Kolejnictwa (EWAG) oraz w Okręgowym Ośrodku Informatyki Południowej DOKP w Krakowie (UMAK).

Metoda RAD różni się nieco od tradycyjnego cyklu opracowywania systemu. Poszczególne jej kroki są następujące:

- dokonanie podziału systemu na 3 części, z których dwie są w dużym stopniu generowane przy pomocy narzędzi CASE, a trzecia jest tworzona tradycyjnie ("ręcznie"),
- wygenerowanie tzw. wersji Alfa pierwszej części systemu,
- jednoczesne testowanie wersji Alfa pierwszej części systemu i generowanie wersji Alfa drugiej części; w wyniku testowania wersji Alfa otrzymuje się wykaz rozszerzeń proponowanych przez użytkowników oraz wykaz błędów do usunięcia,

- jednoczesne testowanie wersji Alfa drugiej części systemu i opracowywanie wersji Beta części pierwszej; wersja Beta uwzględnia rozszerzenia zaproponowane podczas testowania wersji Alfa,
- jednoczesne testowanie wersji Beta części pierwszej systemu i tworzenie wersji Beta części drugiej; w wyniku testowania wersji Beta otrzymuje się wykaz błędów i usterek do usunięcia, nie rozszerza się już natomiast funkcjonalności systemu,
- testowanie wersji Beta części drugiej i nanoszenie poprawek do części pierwszej,
- dokonanie poprawek w wersji drugiej,
- ewentualne opracowanie i testowanie części trzeciej.

Poniższy rysunek ilustruje kolejność wykonywania poszczególnych kroków metody RAD.



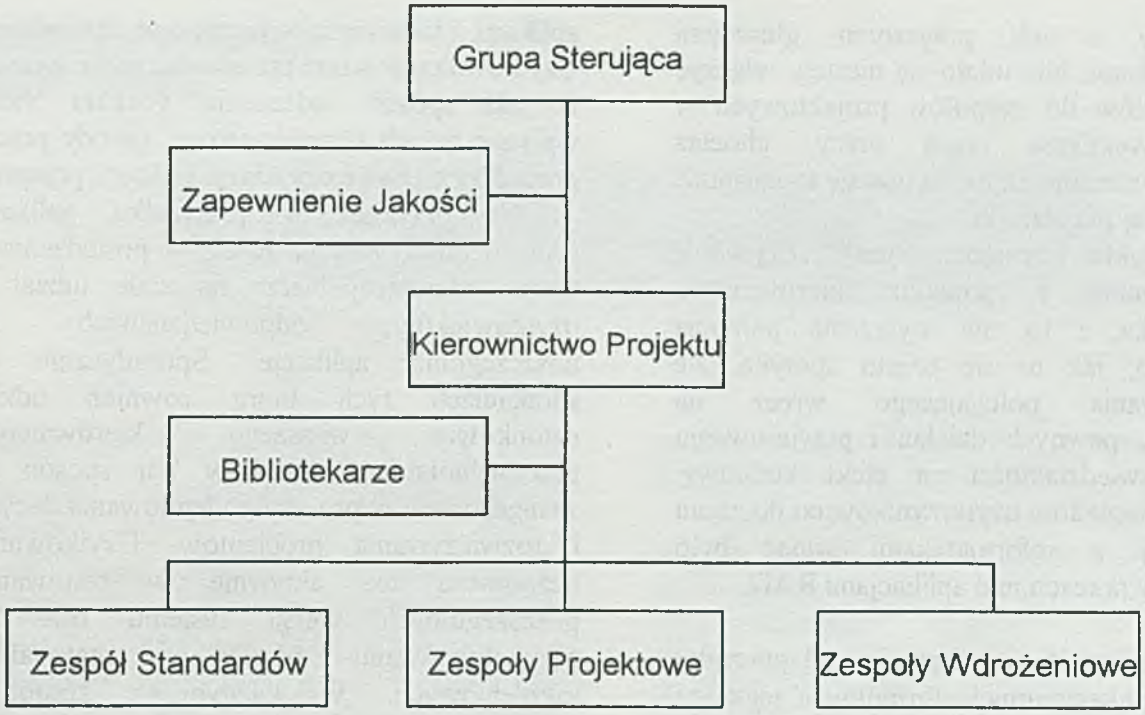
Rys. 2 Schemat metody RAD

Oczywiście, opracowywanie systemu informatycznego nie polega wyłącznie na stworzeniu oprogramowania. Inne działania niezbędne dla prawidłowego przygotowania wdrożenia systemu to m.in.:

- opracowanie dokumentacji użytkowej,
- przygotowanie i przeprowadzenie konwersji danych,
- przygotowanie szkoleń dla użytkowników końcowych,
- przygotowanie założeń eksploatacyjnych,
- przeszkolenie personelu eksploatacji i personelu użytkownika.

5. Organizacja projektu prowadzonego metodą RAD

Dla obu aplikacji opracowywanych metodą RAD utworzono wspólną strukturę organizacyjną, której schemat jest zamieszczony na następnej stronie.



Rys. 3 Organizacja projektów RAD

W skład Grupy Sterującej wchodzą przedstawiciele PKP (Centrum Informatyki Kolejnictwa i użytkownika) oraz Oracle Polska. Spotkania Grupy Sterującej odbywają się co 2 tygodnie.

Kierownictwo Projektu stanowią kierownicy projektu ze strony PKP i Oracle Polska. Kierownictwo wspomagane jest przez zespół Bibliotekarzy, gromadzących dokumentację projektu.

Zespoły Projektowe pracują w Warszawie i w Krakowie, przy czym w każdym z tych ośrodków istnieje zespół opracowujący oprogramowanie (informatycy z PKP i konsultant z Oracle Polska) oraz zespół ds. interface'ów i konwersji danych (informatycy z PKP).

Zespół Standardów pracuje na rzecz wszystkich zespołów projektowych i składa się z kilku informatyków PKP.

Zespoły wdrożeniowe (centralny w Warszawie oraz po jednym dla każdej aplikacji w Warszawie i w Krakowie) składają się z informatyków i użytkowników. Ich zadaniem jest m.in. opracowanie warunków świadczenia usług, zaprojektowanie organizacji eksploatacji, opracowanie procedur użytkownika, opracowanie dokumentacji oraz przygotowanie testów akceptacyjnych.

6. Udział i zaangażowanie użytkownika

Cechą charakterystyczną wielu projektów informatycznych prowadzonych w Polsce jest często znikome zangażowanie użytkowników. Natomiast doświadczenia z wielu projektów - udanych i nieudanych - pokazują, że niewystarczający udział użytkowników w kolejnych etapach realizacji systemu prowadzi często do odrzucenia przez nich końcowego produktu, ponieważ nie spełnia on ich wymagań.

W projekcie SKPZ udało się osiągnąć takie zaangażowanie użytkowników w prace (i to począwszy od szczebla kierownictwa przedsiębiorstwa), które znacznie zmniejsza ryzyko niepowodzenia i odrzucenia przez nich produktu końcowego.

Faza strategii (tj. definiowania aplikacji) była prowadzona wspólnie z użytkownikami odpowiedzialnymi za poszczególne obszary działalności. Analiza wymagań wszystkich aplikacji prowadzona jest przy ścisłej

współpracy z ich przyszłymi głównymi użytkownikami. Nie udało się niestety włączyć użytkowników do zespołów projektowych w pełnym wymiarze czasu pracy, chociaż wszystko wskazuje na to, że uda się to osiągnąć w najbliższej przyszłości.

Niezwykle ważne jest uzyskanie zaangażowania i poparcia kierownictwa użytkownika, i to nie wyłącznie poparcia werbalnego, jak to się często spotyka, ale zaangażowania polegającego wręcz na inicjowaniu pewnych działań i przyjmowaniu współodpowiedzialności za efekt końcowy. Ewolucję poglądów użytkowników co do zasad współpracy z informatykami widać było wyraźnie w pracach nad aplikacjami RAD.

Ważne jest, by użytkownicy od początku prac nad aplikacją umieli sformułować jej cele i wyrazić je w kategoriach mierzalnych, oraz by potrafili wskazać korzyści z wdrożenia aplikacji, i to korzyści wyrażone w złotówkach. Użytkownicy powinni też zdawać sobie sprawę, w jaki sposób wdrożenie systemu może wpłynąć na ich dotychczasowe metody pracy, procedury biurowe czy nawet niektóre przepisy.

Na przykład w przypadku aplikacji prowadzonych metodą RAD, w posiedzeniach Grupy Sterującej bierze na stałe udział 2 użytkowników, odpowiedzialnych za poszczególne aplikacje. Sporadycznie w spotkaniach tych biorą również udział członkowie wyższego kierownictwa przedsiębiorstwa, którzy w ten sposób są zaangażowani w procesy podejmowania decyzji i rozwiązywania problemów. Użytkownicy uczestniczą też aktywnie w testowaniu poszczególnych wersji systemu oraz w przygotowywaniu szkoleń i materiałów szkoleniowych. W każdym z zespołów wdrożeniowych pracuje po ok. 10 użytkowników.

Zbigniew Pojmański
Oracle Polska

Rozproszona baza danych Oracle

Oracle i sieciowe bazy danych

Ewolucja systemów przetwarzania danych

W ciągu ostatnich 30 lat systemy przetwarzania uległy zasadniczej ewolucji. Wyraźnie można tu wyróżnić 3 okresy Przed rokiem 1964 istniały tylko środowiska **jednego producenta, jednej maszyny i jednego systemu operacyjnego**. Każdy nowo powstający komputer miał odmienną architekturę sprzętową i odmienny system operacyjny. Aplikacja napisana dla jednej maszyny, w celu pracy na innej maszynie musiała być kompletnie przerabiana.

Lata 1964 - 1978 to środowiska **jednego producenta, wielu maszyn i jednego systemu operacyjnego**. Duże firmy komputerowe, takie jak IBM czy Digital Equipment Corporation, wprowadziły rodziny komputerów wyposażonych w ramach danej rodziny w ten sam system operacyjny. Gdy rodzina wzbogacała się o nowy, potężniejszy komputer, bez większych problemów można było przenieść na niego powstałe wcześniej aplikacje. Niemniej klient pozostawał związany z danym wytwórcą. Przeniesienie aplikacji na inny system operacyjny wymagało wielkich nakładów.

Od roku 1979 do dziś mamy środowiska **wielu producentów, wielu maszyn i wielu systemów operacyjnych**. Pojawienie się wielu różnorodnych komputerów z systemem operacyjnym UNIX dało wreszcie użytkownikom możliwość wyboru platformy sprzętowej. Aplikacja napisana dla komputera jednego wytwórcy mogła być stosunkowo łatwo przeniesiona na komputery innych producentów. Co więcej, pojawiła się możliwość łączenia ze sobą, w ramach sieci, komputerów różnych wytwórców.

Jednak większość dużych użytkowników posiada nie tylko systemy unixowe, lecz także zgromadzone w ciągu wielu lat najprzeróżniejsze maszyny z różnymi systemami operacyjnymi i aplikacjami. Do tego dochodzą niemal powszechne komputery osobiste. Wyzwaniem dla projektantów jest więc stworzenie środowiska, do którego można dołączać maszyny unixowe, komputery osobiste i maszyny "nietypowe" , i w którym wszystkie maszyny mogą ze sobą współpracować. Środowiska takie noszą nazwę systemów otwartych.

Korzyści z otwartości systemów

Otwartość systemu powinna umożliwiać dwie rzeczy:

- **Przenośność aplikacji**, czyli wykonywanie aplikacji na każdym z komputerów (unixowym lub "nietypowym") używanych w danej organizacji;
- **Współdzielenie danych,** czyli dostęp każdego komputera do danych przechowywanych na innym komputerze, często w odmiennym systemie zarządzania danymi.

Dzięki otwartości pojawiają się następujące korzyści:

- **Zmniejszenie kosztów** - programy napisane dla jednego komputera mogą bez zmian pracować na innych komputerach; dzięki minimalizacji kosztów produkcji i dystrybucji oszczędza się czas i pieniądze.

- **Mniejsze ryzyko inwestowania** - dzięki uniezależnieniu się od określonego wytwórcy maleje ryzyko związane ze zmienną sytuacją rynkową.
- **Większa elastyczność** - użytkownicy nie są związani z określoną architekturą sprzętową, systemem operacyjnym czy rozwiązaniem sieciowym.
- **Większa skalowalność** - dzięki wielkiej liczbie dostępnych opcji użytkownicy nie są zmuszani do inwestowania od razu w wielki system; mogą stopniowo, w miarę wzrastających potrzeb, rozbudowywać sprzęt i oprogramowanie.

Otwartość systemów uzyskuje się przez stosowanie standardów przy projektowaniu części składowych systemów. Używa się zarówno przyjętych formalnie standardów *de jure*, jak i powszechnie używanych, lecz jeszcze formalnie nie przyjętych standardów przemysłowych *de facto*.

Oracle Corporation

Firmą zajmującą się od wielu lat systemami otwartymi jest Oracle Corporation. Oprogramowanie Oracle pojawiło się na rynku oprogramowania w 1979 roku i było pierwszym na świecie relacyjnym systemem baz danych wykorzystującym SQL - język, który obecnie stał się standardem w przemyśle informatycznym. W 1983 roku Oracle był już zaimplementowany na maszynach typu mainframe, minikomputerach oraz mikrokomputerach. Oracle stał się pierwszym przenośnym systemem baz danych. Strategia Oracle Corporation to przede wszystkim praca nad nowymi technologiami oraz ciągłe udoskonalanie jakości produktów i usług. Nakłady Oracle na rozwój nowych technologii wynoszą więcej niż całkowite wpływy większości z konkurentów na rynku systemów baz danych.

Obecnie Oracle Corporation to największy na świecie producent i sprzedawca systemów relacyjnych baz danych oraz trzecia co do wielkości firma zajmująca się wyłącznie wytwarzaniem oprogramowania. W roku finansowym 1992 wpływy Oracle Corporation wyniosły ponad 1.6 miliarda dolarów amerykańskich. Oracle Corporation zatrudnia dziesięć tysięcy pracowników i jest obecna w ponad stu krajach na całym świecie. Prawie 50% światowego rynku systemów relacyjnych baz danych należy do Oracle Corporation.

Oracle Polska

Oracl Corporation działa w Polsce za pośrednictwem Oracle Polska. Przedsiębiorstwo Oracle Polska powstało w czerwcu 1992 roku i jest rozwinięciem przedstawicielstwa Oracle Corporation w Polsce, które istniało od 1991 roku. W chwili obecnej Oracle Polska jest w stanie dostarczyć polskim klientom wszystkie produkty i usługi oferowane przez Oracle Corporation na rynku światowym. Jak dotąd, Oracle jest jedynym producentem systemów baz danych, który posiada swój oddział w Polsce.

Na działalność Oracle Polska składa się sprzedaż oprogramowania, świadczenie asysty technicznej, prowadzenie szkoleń oraz dostarczanie usług konsultingowych.

Firma liczy obecnie około 25 pracowników i zamierza powiększyć skład osobowy do 35 osób do końca 1995 roku.

Serwery i klienci w sieci

Jedna logiczna baza danych

Aby umożliwić przenośność aplikacji i współdzielenie danych, znajdujących się zarówno w bazach Oracle, jak i w bazach innych wytwórców, Oracle stosuje architekturę klient/serwer. Klienci to komputery żądające dostępu do danych, a serwery to komputery przechowujące dane i udostępniające je klientom.

Zmiana konfiguracji serwera, włączając w to jego fizyczną lokalizację, platformę sprzętowo/systemową i źródło danych, nie ma wpływu na działanie istniejących na klientach lub mogących się pojawić w przyszłości aplikacji. Odseparowanie użytkownika (zarówno projektanta aplikacji, jak i użytkownika końcowego) od złożoności serwerów umożliwia mu abstrahowanie od

różnic i traktowanie zbioru baz na wszystkich serwerach jako jednej logicznej bazy danych.

Komunikacja klient - serwer

Z punktu widzenia klienta każdy serwer jest pojedynczą logiczną bazą danych. Ta logiczna baza danych jest zintegrowana z usługami zapewnianymi przez procedury i funkcje PL/SQL, które dają wiele korzyści: "wtłoczenie" do bazy reguł działania i złożonych powiązań integralnościowych, wydajne przetwarzanie dużych transakcji, dostęp do nierelacyjnych i innych niż bazy źródeł danych, integrację ze starymi systemami.

Dzięki takiemu prostemu punktowi widzenia możliwe staje się zaimplementowanie klientów na wielu różnych maszynach - od komputerów osobistych do dużych komputerów - i różnych interfejsów z użytkownikiem - obsługujących terminale znakowe, blokowe lub graficzne.

Każdy klient powinien być w stanie współbieżnie z innym klientami zmieniać i odczytywać dane w środowiskach transakcyjnych i odczytywać dane potrzebne aplikacjom wspierającym podejmowanie decyzji. Możliwe jest to dzięki obsługiwaniu przez Oracle standardów i interfejsów języka SQL. Oprócz własnych narzędzi do komunikacji klientów z bazą, takich jak SQL*Plus, Oracle*Forms czy Oracle*Reports, Oracle dostarcza prekompilatory i bibliotekę Oracle Call Interface (OCI), umożliwiające tworzenie programów 3GL komunikujących się bezpośrednio z serwerem Oracle.

Komunikacja serwer-serwer

Nieskomplikowany punkt widzenia klienta nie oddaje złożoności serwera. Aby możliwe było traktowanie zbioru oddalonych od siebie fizycznych baz danych jako jednej logicznej całości, serwer realizuje dodatkowe, skomplikowane funkcje. Obejmują one wysyłanie poleceń SQL i wywołań zdalnych procedur do innych serwerów Oracle i do gatewayów obsługujących obce źródła danych. Uwzględniają możliwość zmieniania umiejscowienia, liczby i roli serwerów odpowiednio do zmieniających się potrzeb.

Współpracujące serwery

Istotą zaimplementowanej przez Oracle technologii jest współpraca serwerów. Klient uzyskuje dostęp do wielu serwerów i gatewayów poprzez usługi dostarczane przez pojedynczy serwer, realizujący komunikację serwer-serwer. Podstawową cechą jest tu symetryczność, która oznacza, że do realizacji usług dających klientowi wrażenie pracy z jedną logiczną bazą może być użyty dowolny z serwerów.

Usługi oferowane przez server Oracle obejmują m.in.:

- Zarządzanie transakcjami
- Polecenia SQL komunikujące się z wieloma bazami
- Wywołania zdalnych procedur
- Dostęp poprzez gatewaye do odmiennych źródeł danych i usług

Oprogramowanie Oracle dla klientów i serwerów

Oracle dostarcza oprogramowanie zarówno dla klientów, jak i dla serwerów, a także oprogramowanie umożliwiające ich komunikowanie się.

Do pracy na serwerach przeznaczone jest oprogramowanie *Oracle7 Cooperative Server*, umożliwiające pracę komputera w roli serwera zarządzającego relacyjną bazą danych i komunikującego się z klientami i innymi serwerami.

Do pracy na komputerach - klientach przeznaczone są gotowe produkty, np. *Oracle Financials*, oraz narzędzia służące do tworzenia oprogramowania, wchodzące w skład *Cooperative Development Environment*, wreszcie aplikacje wytworzone w wyniku użycia tych narzędzi.

Oddzielną grupę stanowi oprogramowanie umożliwiające współpracę komputerów w sieciach, na które składają się *Protocol Adapters*, *SQL*Net* i *MultiProtocol Interchange*.

Cooperative Server Technology

Oracle7 Cooperative Server jest otwartym systemem zarządzania relacyjną bazą danych. Dzięki swej niezawodności może być środowiskiem kluczowych dla organizacji aplikacji, zaś jego skalowalność pozwala dostosowywać te aplikacje do wzrostu i zmian zachodzących w organizacji. Systemy Oracle7 charakteryzują się niskimi kosztami i dzięki swej otwartości chronią wcześniejsze inwestycje programowe.

Oracle7 pozwala transparentnie współdzielić dane i sprawia, że wszystkie stare i nowe systemy, łącznie z tymi na dużych komputerach, pracują razem w łatwym do zarządzania, ściśle zintegrowanym środowisku obejmującym całą organizację. Niezależnie od architektury sprzętowej, Oracle7 efektywnie wykorzystuje wszystkie zasoby systemu i zapewnia znakomitą skalowalność i wydajność.

Skalowalność

Oracle7 jest przenośny pomiędzy ponad 80 platformami sprzętowymi i systemowymi, od komputerów biurkowych do dużych superkomputerów. Ta przenośność daje użytkownikowi wolność wyboru platformy, uwzględniającego stosunek kosztów do wydajności oraz niezawodność teraz i w przyszłości. Skalowalność rozciągająca się od komputerów biurkowych do superkomputerów gwarantuje, że w miarę wzrostu wymagań stawianych systemowi użytkownikowi nigdy nie zabraknie mocy przetwarzania.

Wydajność

Wielowątkowa architektura serwera Oracle7 zapewnia na wszystkich architekturach sprzętowych wysoką wydajność nawet przy bardzo dużej liczbie użytkowników. Wydajność osiągana jest poprzez eliminację wąskich gardeł procesora, wejścia/wyjścia, pamięci i systemu operacyjnego, a także przez optymalizację kodu zarządzającego relacyjną bazą danych. Dodatkowa opcja o nazwie Oracle7 Parallel Server dzięki unikalnej technologii zarządzania zrównolegloną pamięcią notatnikową zapewnia skalowalną wydajność na luźno powiązanych systemach.

Wielowątkowa architektura serwera

Oracle7 stosuje samodostrajającą się, wielowątkową architekturę serwera, w której liczba procesów serwera bazy danych dynamicznie dostosowuje się do aktualnego obciążenia. Zapewnia to wszystkim użytkownikom krótki czas odpowiedzi przy minimalnym użyciu zasobów systemu. Współdzielona pamięć notatnikowa SQL pozwala wielu użytkownikom wykonującym to samo polecenie SQL współdzielić jedną kopię polecenia, minimalizując zużycie pamięci. Dynamiczny i statyczny SQL oraz pamiętane procedury współdzielone są przez użytkowników. Wszystkie te cechy minimalizują obciążenie systemu operacyjnego i zużycie pamięci, zapewniając dużą wydajność nawet dla największych aplikacji.

Optymalizacja zapytań uwzględniająca koszty

Dzięki wbudowanej inteligentnej optymalizacji zapytań uwzględniającej koszty Oracle7 gwarantuje nawet niedoświadczonym użytkownikom i projektantom błyskawiczny dostęp do danych. W celu znalezienia najbardziej wydajnej ścieżki dostępu, przy optymalizacji brane są pod uwagę takie czynniki, jak liczba wierszy przypadająca na tablicę czy selektywność indeksów.

Kontrolowanie zużycia zasobów

Administrator bazy danych posiada kontrolę nad wielkością zasobów zużywanych przez poszczególnych użytkowników. Może ustalać górne limity takich zasobów procesora, jak logiczne operacje wejścia/wyjścia i czas połączenia. Jeżeli użytkownik próbuje przekroczyć te ograniczenia, jego żądanie lub sesja ulegają skasowaniu.

Ciągła praca

Baza danych Oracle7 może pracować w sposób ciągły. Kluczowe funkcje systemowe, takie jak kopiowanie zabezpieczające, odtwarzanie po awarii, czy administrowanie bazą danych, mogą być wykonywane on-line,

bez przerywania przetwarzania transakcji, nawet w okresie intensywnej pracy. Możliwe jest uniezależnienie się od awarii sprzętu, dzięki obsłudze luźno powiązanych (*clusters*) platform sprzętowych.

Aktywna część aplikacji

Oracle7 jest aktywną częścią aplikacji w tym sensie, że realizuje pewne funkcje zazwyczaj programowane w aplikacjach. Wymusza niezawodną integralność i bezpieczeństwo danych, minimalizując jednocześnie prace przy projektowaniu, konserwowaniu i administrowaniu. Prowadzi to do zmniejszenia kosztów aplikacji i do skalowalnego środowiska projektowania aplikacji, mogącego obsłużyć setki różnych aplikacji i tysiące użytkowników. Zaimplementowana w Oracle7 w pełni zgodna z normą ANSI/ISO wersja języka SQL zapewnia całkowicie otwarte środowisko projektowania.

Deklaratywne więzy integralnościowe

Oracle7 jest w pełni zgodny z normami ANSI/ISO dotyczącymi deklaratywnych więzów integralnościowych, które pozwalają wymuszać integralność bez potrzeby programowania. Wystarczy, aby projektanci lub administrator w momencie definiowania lub zmieniania tablic określili więzy integralności, a system zarządzania bazą danych Oracle7 automatycznie będzie wymuszał ich przestrzeganie.

Więzy integralności Oracle7:

- minimalizują koszty projektowania i konserwowania aplikacji, gdyż eliminują kodowanie
- poprawiają niezawodność aplikacji, gdyż to system wymusza przestrzeganie reguł
- zmniejszają dodatkowe nakłady związane z wymuszaniem integralności, gdyż Oracle7 optymalizuje sposób wymuszania
- są w 100 procentach zgodne z normą ANSI/ISO.

Pamiętane procedury i wyzwalacze w języku PL/SQL

Pamiętane procedury i procedury wyzwalaczy napisane w języku PL/SQL umożliwiają wymuszanie na poziomie serwera złożonych reguł funkcjonowania, poprawiając niezawodność aplikacji, zmniejszając koszty projektowania i zwiększając szybkość dostępu do danych.

Procedury pamiętane to procedury składające się z jednego lub więcej zdań PL/SQL - potężnego proceduralnego języka będącego rozszerzeniem SQL - przechowywane w postaci skompilowanej i współdzielonej w bazie danych Oracle7. Procedury te mogą być wywoływane bezpośrednio z narzędzi bądź innych procedur, albo też ich wykonanie może być automatycznie wyzwalane przy wstawianiu, zmienianiu lub usuwaniu rzędów w tablicach.

Procedury wyzwalane mogą być używane do wymuszania w serwerze złożonych reguł integralności. Na przykład wyzwalacz może generować nowe zamówienie, gdy zapas danego artykułu spadnie poniżej ustalonego minimum. Wyzwalaczy można również używać do przeprowadzania kontroli danych lub synchronicznej replikacji tablic. Wyzwalacze mogą się wykonywać przed lub po wyzwalającym zdarzeniu, raz na każdy wiersz lub raz na polecenie. Niezawodna implementacja wyzwalaczy w Oracle7 jest dokładnym odwzorowaniem specyfikacji ANSI/ISO SQL3.

Procedury pamiętane i wyzwalacze zwiększają niezawodność aplikacji i redukują koszty projektowania, gdyż wystarczy tylko raz zaprojektować potrzebne procedury, po czym przechowywać i konserwować w jednym wspólnym miejscu, a nie w każdej aplikacji oddzielnie. Procedury pamiętane Oracle7 mogą być pogrupowane w pakiety procedur, ponadto mogą mieć formę zwracających wartość funkcji. Procedury i pakiety PL/SQL pozwalają zastosować techniki aktywnych baz danych w wielkich (wielu projektantów, wiele aplikacji) środowiskach projektowych.

Role - podstawa bezpieczeństwa

Oracle7 pozwala zdefiniować role, czyli ponazywane zbiory uprawnień. Zdefiniowanie ról znacznie zmniejsza wysiłek i koszty związane z kontrolowaniem bezpieczeństwa, pozwalając grupować uprawnienia dotyczące tablic i innych obiektów bazy danych i przydzielać je indywidualnym użytkownikom lub całym grupom. W ten sposób administratorzy odpowiedzialni za bezpieczeństwo za pomocą pojedynczych poleceń GRANT mogą upoważniać poszczególnych użytkowników do wykonywania całych aplikacji. Ponadto, w przeciwieństwie do tradycyjnych systemów zarządzających bazami danych, posiadających jednego wszechpotężnego administratora, role w Oracle7 umożliwiają organizacjom posiadanie wielu administratorów o ściśle kontrolowanych uprawnieniach.

Zastosowana przez Oracle7 specyfikacja ról została zaakceptowana przez komitety normalizacji SQL ISO i ANSI jako podstawa przyszłych standardów bezpieczeństwa SQL.

Obsługa języków narodowych

ORACLE zapewnia rozbudowaną obsługę niemal wszystkich języków europejskich i azjatyckich (kody 8- lub 16-bitowe). Dla każdego języka automatycznie wybierane są odpowiednie komunikaty o błędach, porządek sortowania, format daty i wiele innych opcji. Możliwe są nawet niejednorodne pod względem językowym środowiska typu klient-serwer i bazy rozproszone. Oracle7 przeprowadza wszelkie niezbędne konwersje zestawów znaków w sposób automatyczny i transparentny.

Ukrycie rozproszoności bazy danych

Dzięki Oracle7 projektanci i użytkownicy końcowi mogą traktować zbiór fizycznie rozproszonych baz danych jako logiczną pojedynczą bazę. Użytkownik może pobierać i zmieniać dane przechowywane w dowolnym miejscu rozproszonej bazy z taką samą łatwością, jak gdyby dane rezydowały na pojedynczej, lokalnej maszynie. Nawet gotowe już aplikacje, zaprojektowane w środowisku niesieciowym, bez potrzeby przeprogramowywania mogą zostać rozprzestrzenione na wiele systemów.

Rozproszone przetwarzanie zapytań

W ORACLE7 pojedyncze zdanie SQL może transparentnie zapytywać, a nawet w skomplikowany sposób łączyć dane pochodzące z wielu baz. Użytkownicy nie muszą trudzić się pamiętaniem, gdzie faktycznie przechowywane są dane, a aplikacje nie muszą być przeprogramowywane, gdy dane zostają przeniesione z jednego komputera na inny. Dzięki inteligentnej dekompozycji zapytań i globalnej, uwzględniającej koszty optymalizacji, nawet złożone, rozproszone zapytania obsługiwane są bardzo efektywnie.

Rozproszone zarządzanie transakcjami

W celu zapewnienia integralności rozproszonych transakcji zmieniających dane, Oracle7 stosuje niezawodny, transparentny, dwufazowy mechanizm potwierdzania. Transakcje obejmujące wiele baz danych zatwierdzane są za pomocą jednego standardowego polecenia SQL o nazwie COMMIT. Nie potrzebne jest żadne programowanie. Oracle7 automatycznie wykrywa i obsługuje wszelkie ewentualne niepowodzenia i zapewnia, że we wszystkich serwerach transakcje będą jednocześnie albo potwierdzone, albo wycofane. Chronione są wszelkie zmiany, także te wprowadzone przez zdalnie wywołane procedury PL/SQL. Dwufazowe potwierdzanie zapewnia najwyższy z możliwych stopień autonomii danego serwera bazy danych i zabezpieczenie w przypadku długotrwałych awarii systemów.

Dwufazowe potwierdzanie Oracle7 współpracuje także z innymi bazami danych i systemami plików - nawet z tymi starszymi, które same z siebie nie obsługują dwufazowego potwierdzania. W połączeniu z rodziną produktów Oracle Gateway możliwe jest włączenie jednej bazy innego niż ORACLE wytwórcy do systemu dwufazowego

potwierdzania Oracle7. Jeśli aplikacja wymaga rozproszonych transakcji pomiędzy wieloma obcymi systemami, to muszą one być zgodne z normą XA, i wtedy do dwufazowego potwierdzania można użyć monitora TP zgodnego z XA. System zarządzania bazą danych Oracle7 jest w pełni zgodny z normą XA.

Replikowanie tablic

W Oracle7 często używane dane można transparentnie replikować do wielu komputerów. Oracle7 automatycznie uaktualnia przeznaczone tylko do odczytu kopie tablic źródłowych (*snapshots* - zdjęcia migawkowe) z częstotliwością wybraną przez użytkownika (np. raz na godzinę, dzień czy tydzień). Uaktualniane są tylko te wiersze, które uległy zmianie po poprzednim uaktualnieniu, albo całe tablice. Replikacja tablic pozwala projektantowi na dużą elastyczność w dostarczaniu użytkownikom takiej wydajności i dostępności, jaka jest dla nich niezbędna, przy jednoczesnej minimalizacji wysiłku związanego z administrowaniem bazy danych.

Globalne nazewnictwo

Oracle7 obsługuje globalne nazwy baz danych, pozwalając w sposób jednoznaczny identyfikować i lokalizować wszystkie dane w środowisku rozproszonym. Oracle7 zaprojektowano do pracy z zewnętrznymi usługami nazewniczymi, aby wyeliminować obciążenie związane z utrzymywaniem w bazie informacji o lokalizacji danych.

Otwarta architektura Oracle7 pozwala zintegrować Oracle7 i systemy innych firm oraz najlepsze dostępne na rynku zestawy narzędzi, aplikacji i innych produktów programowych w jedno standardowe środowisko.

- Poprzez produkty z rodziny Open Gateway, Oracle7 zapewnia dostęp (proceduralny lub transparentny, przy pomocy SQL) do danych z praktycznie dowolnego źródła. Dostępne są dane z wielu źródeł, takich jak:
  - inne systemy zarządzania relacyjnymi bazami danych (DB2, SQL/DS, Rdb, NonStop SQL, itd.)
  - inne systemy zarządzania nierelacyjnymi bazami danych IMS, HP TurboImage, itd.)
  - źródła inne niż bazy danych sterowniki maszyn, liczniki, monitory procesów, itd.)

Integrowanie systemów pochodzących od wytwórców innych niż Oracle.

Oracle7 integruje wszystkie te źródła danych ze swym własnym bogatym zestawem narzędzi i aplikacji oraz aplikacjami innych wytwórców. Oracle7 współpracuje także z ogromną liczbą produktów programowych innych wytwórców. Lista obejmuje takie popularne produkty jak Lotus1-2-3, Paradox, Excel, i dziesiątki innych pakietów, od procesorów tekstu po systemy ekspertowe i narzędzia do projektowania aplikacji.

Cooperative Development Enviroment

Cooperative Development Enviroment (w skrócie: CDE) to oferowany przez Oracle zintegrowany zestaw narzędzi programowych, umożliwiający efektywne tworzenie, wdrażanie i konserwację aplikacji niezbędnych dla prawidłowego funkcjonowania organizacji. Aplikacje te wykonywane są na komputerach pełniących role klientów żądających dostępu do danych i obrazujących te dane, komunikujących się z serwerami, czyli komputerami przechowującymi i udostępniającymi bazy danych. Narzędzia CDE służą projektantowi na wszystkich etapach cyklu projektowania aplikacji: od identyfikowania procesów składających się na działalność organizacji i definiowania wymagań aplikacji, do analizy, projektowania, budowania i konserwacji aplikacji.

Narzędzia do tworzenia aplikacji dla organizacji

Organizacje stają dziś przed trudnym wyzwaniem, jakim jest szybkie i ekonomiczne tworzenie przenośnych aplikacji pracujących w

sieciach komputerów typu klient i serwer. Dotychczas przy tworzeniu aplikacji najczęściej posługiwano się niejednolitymi, niezintegrowanymi narzędziami. CDE zasadniczo poprawia sytuację. Projektanci otrzymują kompletny zestaw narzędzi, wspomagający ich na każdym etapie cyklu życiowego aplikacji, zarówno niewielkich (dla jednego użytkownika), jak i rozbudowanych, obejmujących całą organizację, używających różnorodnych źródeł danych i różnorodnych interfejsów użytkownika.

Niezależność od interfejsu graficznego użytkownika

Zarówno same narzędzia CDE, jak i utworzone przy ich pomocy aplikacje, dają się bez dodatkowych nakładów przenosić pomiędzy różnymi interfejsami graficznymi użytkownika (GUI - graphical user interface), takimi jak Windows firmy Microsoft, Macintosh firmy Apple, Open Look bądź Motif w systemie Unix).

Przenośność tę osiągnięto poprzez zastosowanie unikalnej biblioteki oprogramowania o nazwie Adaptable User Interface Toolkit. Wyeliminowała ona potrzebę oddzielnej wersji CDE dla każdego rodzaju GUI. Przy zmianie GUI klienta nie trzeba zmieniać ani jednej linii kodu programu. Ponadto narzędzia CDE i aplikacje automatycznie przyjmują wygląd i sposób pracy charakterystyczny dla GUI klienta.

Otwartość - standardowe interfejsy programowania aplikacji

CDE zbudowano przy użyciu standardowych interfejsów programowania aplikacji (API - application programming interface), takich jak ODBC i DRDA.

Dzięki temu użytkownik równocześnie z CDE może używać produktów innych wytwórców, np. Excel firmy Microsoft czy 1-2-3 firmy Lotus.

Ponadto aplikacje równie łatwo, jak z bazą danych typu Oracle7, mogą wymieniać dane z bazami innych wytwórców (DB2 czy SQL Server), a także z plikami.

Otwarta architektura pozwala na współpracę CDE z bazami danych i narzędziami różnych wytwórców zarówno dostępnymi obecnie, jak i mogącymi się pojawić w przyszłości.

Multimedia

Aplikacje CDE mogą się posługiwać multimediami - obiektami typu obraz graficzny, dźwięk lub video. Także i te obiekty są przenaszalne. Na przykład, jeśli obiekt graficzny lub dźwiękowy stworzony na Macintoshu przenosimy do Windows, CDE automatycznie dokona wszelkich niezbędnych transformacji obiektu.

Metoda CASE

Użycie CDE znacznie (nawet dziesięciokrotnie!) przyspiesza proces tworzenia aplikacji. Dzieje się tak dzięki zastosowaniu metody CASE (Computer Aided System Engineering - Inżynieria Systemowa Wspomagana Komputerowo).

Stosując tę metodę, projektant modelujacy organizację przedstawia składające się na nią obiekty i pojęcia w postaci prostokątów, zaś relacje zachodzące pomiędzy nimi - w postaci łączących te prostokąty linii. Na podstawie takich diagramów CDE automatycznie generuje kompletną, gotową do pracy aplikację.

CDE może przejąć bazę wiedzy o organizacji utworzoną przy użyciu narzędzi CASE innych wytwórców, m. in. IEF firmy Texas Instruments i IEW firmy KnowledgeWare.

Ułatwia to podjęcie decyzji o przejściu do stosowania CDE w wypadku, gdy istnieje już baza wiedzy o organizacji.

Obsługa języków narodowych

Narzędzia i aplikacje CDE obsługują wiele języków narodowych, w tym język polski.

Produkty sieciowe Oracle

SQL*Net

Połączenie wielu różnorodnych sieci w jedną "sieć aplikacji Oracle"

SQL*Net V2 pozwala przezwyciężyć trudności komunikacyjne aplikacji sieciowych. Typowe sieci ulegają scaleniu i przekształcają się w sieć wyższego poziomu - sieć aplikacji. Klienci i serwery komunikują się, mimo że komputery, na których pracują, nie mogą się bezpośrednio połączyć.

(W tym miejscu wydawca przerywa druk dalszej części artykułu ze względu na jego obszerność i szczupłe ramy niniejszego opracowania. Niżej zamieszczamy tylko spis nagłówków akapitów części nie wydrukowanej.

Połączenie wielu różnorodnych sieci w jedną "sieć aplikacji Oracle
- Poszczególne sieci o tym samym protokole stają się "społecznościami" (communities)
- Unikalność SQL*Net V2
- SQL*Net V1 - zapoczątkowanie przetwarzania klient/serwer
- SQL*Net V2 - rozszerzenie możliwości połączeniowych aplikacji
  - Większa transparentność sieci
  - Wielopoziomowe śledzenie i odnotowywanie błędów
  - Konwersja typów danych i zbiorów znaków
  - Adresowanie aplikacji
  - Operacje asynchroniczne
  - Obsługa wielu baz danych
  - Obsługa wielu protokołów
- MultiProtocol Interchange
  - Connection Manager
  - Navigator
  - Proces zestawiania połączenia
  - Modularyzacja zwiększa wydajność i niezawodność oraz umożliwia redundancję
  - Redundancja połączeń między społecznościami

Architektura sieci Oracle
- Standardy w architekturze Oracle
- Rys. 7. Standardy w architekturze Oracle
- Protokoły sieciowe
- Usługi sieciowe
- Usługi transakcyjne
- Gatewaye i API dla dostępu rozproszonego
  - Gatewaye transparentne
  - Gatewaye proceduralne

Jerzy Kruszewski
IBM Polska

DB2 - Architerktura rozproszonej, relacyjnej bazy danych

DB2 jest relacyjną bazą danych implementowaną na wszystkich platformach sprzętowych IBM i pod wszystkimi systemami operacyjnymi IBM oferuje:

- DB2/ESA, DB2/VM, DB2/VSE na serwerach serii ES/9000 odpowiednio pod kontrolą systemów operacyjnych MVS/ESA, VM/ESA,
- DB2/400 na serwerach AS/400 pod systemem operacyjnym OS/400,
- DB2/6000 na serwerach riscowych pod systemem operacyjnym AIX,
- DB2/2 na serwerach PS/2 pod systemem operacyjnym OS/2.

Oferta nie jest ograniczona do sprzętu IBM-owskiego. DB2/600 może być instalowane na komputerach firm HP i SUN w dniu dzisiejszym.
Software DB2 został zaprojektowany w taki sposób, że docelowe rozwiązanie satysfakcjonuje najbardziej wymagających użytkowników pod względem technicznym i eksploatacyjnym - wielkie korporacje banki i inne.
DB2 zapewnia:

- pełną integralność i wiarygodność danych,
- wysoką jakość przetwarzania i dostępność,
- odtwarzanie, reorganizację baz danych i stosowania innych programów posiłkowych (utilities) bez przerywania pracy systemu.

Charakterystyczne cechy DB2:

- operacyjna dostępność przez 7 dni w tygodniu 24 godziny na dobę z równoczesnym wykonywaniem programów posiłkowych (utility),
- produkcyjne wsparcie dla 2 Phase Commit - 2PC,
- doskonałe wyniki masowego przetwarzania danych,
- utrzymanie efektywnego przetwarzania wielkich tablic do wymiaru 64 GB,
- synergy czyli powiązanie procesów DB2 z architekturą hardwarową, na której baza danych jest zaimplementowana,
- kompresja danych zawartych w tablicach DB2, a pozwalająca oszczędzić do 40% przestrzeni pamięci zewnętrznej dostępna jest na razie na platformie MVS/ESA,
- równoległe, wieloprocesorowe przetwarzanie zapytań dostępne jest pakietu z DB2/6000 pod AIX na komputerze SP2.

DATABASE 2 jest systemem bazy danych:

1. Klient-serwer na wszystkich platformach softwarowych,
2. Stosuje standardy : ISO, ANSI, X/OPEN, SQL ACCES GROUP, OSF.
3. Stosuje na wszystkich platformach SQL-92.
4. Wspiera protokoły SDLC i TCP/IP.

Zabezpieczenie danych:
DB2 realizuje wielopoziomowe zabezpieczenie danych. Realizowane jest to przez:

- autoryzację nadania dla grup lub pojedynczych właścicieli obiektów na dostęp lub wykonanie funkcji zdań SQL

– poprzez zdefiniowanie zasobów DB2 w specjalistycznych pakietach odpowiedzialnych za bezpieczeństwo danych np. RACF.

DRDA - Distributed Relational Database Architecture.

DRDA stanowi integralną część pakietu RDBMS - DB2 i jest dostarczana użytkownikowi razem z pakietem DB2. DRDA jest protokołem komunikacyjnym wbudowanym w DB2, który przyjmuje zapytania klienta z sieci i przesyła je do realizacji w odpowiednim serwerze sieci. Nie gra roli, który z terminali w sieci generuje zapytanie, jest ono zawsze skierowane do właściwego serwera do wykonania. Architektura DRDA jednolicie traktuje wszystkie elementy sieci i stacja robocza może być elementem sieci pod kontrolą DB2/6000 będącą jednocześnie terminalem np. serwera ES/9000. DRDA nie jest ograniczone tylko do domeny IBM-owskiego sprzętu i produktów. Serwery HP i firmy Sun mogą pracować w sieci zgodnie z poprzednią informacją o implementacji DB2/6000 na tych komputerach. Większość liczących się na rynku systemów baz danych takich jak Oracie, Informix, Ingrase czy Sybase zrealizowała już interfacy do DRDA, co pozwala im pracować w sieci IBM-owskiej na tych samych prawach, co serwery i klienci IBM-owscy.
Wszystkie cechy DB2 dotyczące zabezpieczenia i replikacji danych, pozostają na tym samym poziomie, co przed włączeniem DRDA.
DB2 na wszystkich platformach współpracuje z Monitorem Transakcji - CICS. Dodatkowo DB2/MVS może używać, jako Transaction Monitora część komunikacyjną Information Management System-IMS TP.

VISUALGEN I VISUALGEN TEAM SUITE.

DB2 i CICS jako dwa podstawowe produkty IBM-u otrzymały wsparcie w dziedzinach: szybkiego rozwoju aplikacji - Visualgen oraz VisualGen Team Suite - kompleksowego środowiska rozwoju aplikacji, rozumianego, jako rozwój od etapu analizy do wykonywania programów w środowisku produkcyjnym. Visualgen jest bazowanym na OS/2 pakietem do rozwoju aplikacji. Aplikacje utworzone pod OS/2 mogą pracować w środowiskach: MVS/ESA, VSE/ESA, AIX, OS/2, OS/400.
Używając VGen można:

– definiować i edytować aplikacje z grafiką i bez,
– interakcyjnie testować aplikacje klient-serwer i inne,
– generować przetestowane aplikacje,
– wykonywać aplikacje w przeznaczonym środowisku.

Visualgen Team Suite - VGTs dostarcza narzędzia softwarowe do integracji środowiska rozwoju aplikacji i zapewnia:

– w pełni zintegrowane, wizualne środowisko rozwoju,
– efektywny rozwój aplikacji klient-serwer,
– wspomaganie dużego volumenu transakcji w układzie klient-serwer,
– projektowanie danych i ich administracja na systemach LANowych.

CUSTOMER INFORMATION CONTROL SYSTEM.

CICS.

Fakt posiadania i niezbędności systemu ze zdalnym dostępem do danych zwanego dawniej data communication nikogo już nie dziwi. Tylko systemy ze zdalnym dostępem są dzisiaj akceptowane ze strony użytkownika i produkowane przez twórców softwaru. CICS jest systemem, którego komponenty obsługują dane komunikacyjne w najszerszym tego słowa znaczeniu. To zezwala analitykom i programistom aplikacyjnym koncentrować się na analizie działalności przedsiębiorstwa i pisaniu aplikacji.
CICS został zaprojektowany, ulega nieustannej modyfikacji i jest stosowany od prawie trzydziestu lat jako typowy produkt OLTP - Online Transaction Processing. Ten najbardziej rozpowszechniony i stosowany w więcej niż

150 tysiącach instalacji na świecie system jest produktem, który może być zainstalowany na następujących platformach pracujących pod kontrolą systemu operacyjnego:

Rodzina CICS	System Operacyjny
CICS/ESA	MVS/ESA
CICS/VSE	VSE/ESA
CICS/400	OS/400
CICS/6000	AIX
CICS/2	OS/2

Pełna kompatybilność pomiędzy rodziną produktów CICS zapewniona jest poprzez stosowanie CICS Application Programming Interface (API), a powiązanie operacyjne poprzez cechę CICS - Intersystem Communication (ISC).
Powyższa lista odzwierciedla stan produktów na dzień dzisiejszy, historycznie zaś produktem wyjściowym był CICS pracujący pod OS/360, OS/VS1 i dalej do MVS/ESA. W międzyczasie chcąc utrzymać pełną kompatybilność międzyplatformową, firma IBM dodawała kolejne produkty o tych samych cechach wiarygodności, co CICS/OS z końca lat sześćdziesiątych. W ten sposób obecnie komputery typu RISC, AS/400, PS/2 i serwery mainframowe pracują w tej samej sieci, gdzie zainstalowane są różne systemy CICS przezroczyste dla aplikacji.

Obsługa funkcji programów aplikacyjnych

Podstawową funkcją CICS jest sterowanie wykonaniem programów aplikacyjnych.
W celu zrealizowania funkcji zapewniającej obsługę interakcji między programem aplikacyjnym a terminalem CICS zapewnia:

- funkcje niezbędne do komunikacji z lokalnymi i zdalnymi terminalami lub systemami,
- sterowanie współbieżnie przetworzonych programów obsługujących wielką ilość użytkowników online,
- wysoką dostępność wszelkiego rodzaju zasobów,
- szybkie podnoszenie systemu,
- kontrolę dostępu dla autoryzowanych użytkowników do baz danych i zbiorów,
- komunikację między różnymi systemami CICS, dostęp do różnych systemów baz danych niezależnie czy znajdują się na tym samym komputerze czy na wielu instalacjach,
- przejęcie pewnych funkcji systemu operacyjnego w niektórych jego aspektach.

Rodzina pakietów CICS została już wymieniona powyżej. Na podstawie pakietu CICS/600 pracującego pod kontrolą systemu operacyjnego AIX na serwerze RISC/6000 lub serwerach innych firm zostaną przedstawione funkcje i cechy pakietu CICS. Należy przyjąć, iż pozostałe pakiety, jak CICS/ESA, CICS/2 posiadają te same funkcje i cechy.

Dane techniczne

1. Aplikacje CICS-u i dane obsługiwane przez CICS mogą być rozproszone w następujących środowiskach:
   - TCP/IP wyłącznie,
   - SNA wyłącznie,
   - TCP/IP i SNA

   Zarówno SNA i TCP/IP są w pełni wspierane poprzez Local Area Network (LAN) i Wide Area Network (WAN). Aplikacje używające CICS mogą być rozwijane pod CICS/6000, wykonywane pod dowolnym produktem rodziny CICS. Istotne jest wsparcie dla PowerParallel SP2 oraz HACMP/6000, które zapewniają wzrost ilości procesorów oraz wiarygodność i bezpieczeństwo rozwiązania „clustera" HACMP.
2. CICS/6000 jest oparty o architekturę klient-serwer i składa się z dwu produktów instalowanych odpowiednio na serwerze i kliencie.

3. Rodzina produktów CICS wspiera API dla języków Cobol i C.
4. Większość liczących się na rynku produktów relacyjnych baz danych współpracuje z CICS-em. Do nich należą:
   - DB2,
   - Oracle,
   - Informix,
   - Sybase,
   - Ingres,
   - i inne

   Transakcje CICS realizują dostęp do danych relacyjnych baz danych poprzez włączenie odwołań SQL-a do programów CICS. Koordynacja „potwierdzenia dwufazowego" oraz przy podnoszeniu systemu po awarii jest zapewniona przez CICS dla baz danych z interfacem X/Open XA.
5. CICS zapewnia możliwość przetwarzania rozproszonego pomiędzy produktami CICS zainstalowanymi w różnych instalacjach i środowiskach. Realizowane jest to przez integralną cechę CICS-u Intersystem Communication (ISC), która umożliwia:
   - routing transakcji,
   - przesyłanie wykonania funkcji,
   - przetwarzanie rozproszone z użyciem APPC,
   - asynchroniczne przetwarzanie transakcji,
   - możliwość dołączenia programu w dowolnym miejscu środowiska rozproszonego.

   Funkcja ISC jest wspierana przez TCP/IP i SNA.
6. Integralność danych.
   CICS zapewnia integralność danych na wszystkich poziomach przetwarzania włączając w to dynamiczne wycofanie (backout) transakcji i koordynację poziomów synchronizacji dla lokalnych i zdalnych punktów synchronizacji w rozproszonym środowisku i sieciach o różnych protokołach.

Standardy implementowane przez CICS.

POSIX - CICS/6000 używa funkcji AIX, które są zgodne ze standardem POSIX,

X-OPEN - CICS/6000 używa funkcji AIX, które są zgodne ze standardem X-OPEN. Do nich należy użycie interfaców X/Open SQL, C i Cobolu jak również budowa modelu X/Open DTP.

TCP/IP - TCP/IP jest stosowane przez CICS/6000 i przez CICS/6000 Intersystem Communication.

Efektywność przetwarzania

Efektywność wykorzystania pakietu CICS zależeć będzie od następujących czynników:

- wielkości procesora,
- wielkości całkowitej pamięci dostępnej dla przetwarzania,
- wielkości pamięci dedykowanej dla CICSu,
- typu i modelu pamięci dyskowej,
- funkcji aplikacji pracujących w tym samym interwale czasowym.

Bezpieczeństwo zasobów danych

CICS/6000 używa wszystkich funkcji AIX Security Server/6000 dla zapewnienia bezpieczeństwa i kontroli:

- transakcji,
- zasobów.

Oprócz tego standardowe zabezpieczenie CICS-u dotyczy:

- dostępu do transakcji dla autoryzowanych użytkowników,

– dostępu do kontrolowanych przez CICS zasobów przez autoryzowanych użytkowników.

- CICS/6000 dostarcza interface do połączenia z zewnętrznym „security manager".

Rozwój aplikacji.

CICS dostarcza użytkownikowi wiele narzędzi do rozwoju aplikacji. Do nich należą:

- translator komend i interpretator komend,

Programy zawierające niewspierane przez CICS komendy nie zostaną przetłumaczone przez „precompiler". Na tym etapie komendy EXEC CICS konwertowane są na zadania języka C lub Cobolu wraz z weryfikacją syntaktyki:

- generacja map i ich testowanie,
- narzędzia do pomocy przy uruchamianiu programów.

Podstawowe pytanie, które należy sobie zadać: jaką funkcję w dzisiejszej technologii informacji, przy wielu systemach baz danych, przy wielu platformach sprzętu i różnych protokołach sieci spełniać ma CICS - sprawdzony i wiarygodny Monitor Transakcji? Systemy baz danych takie, jak DB2, IMS, Adabas i inne używają CICS, jako Data Communication czyli OLTP. Systemy baz danych pracujące w większości swoich zastosowań na komputerach nowej technologii riscowej posiadają OLTP wbudowane w RDBMS, dotyczy to wszystkich liczących się systemów RDBMS, jak: Sybase, Oracle, Infor mix, Ingres i innych. Przy dużej ilości terminali (np. powyżej 1000) może okazać się, że zastosowanie CICS, jako OLTP z jego wieloletnim sprawdzeniem w instalacjach bankowych i ubezpieczeniowych na całym świecie, przyczyni się do polepszenia wydajności przetwarzania.

W sytuacjach, gdzie w sieci rozproszonej mamy do czynienia z dwoma lub większą ilością systemów baz danych np.: Oracle, Informix, Ingres z wzajemnym dostępem do danych rola CICS-u jako Integratora systemów i Sieci, sterującego dostępem do zdefiniowanych w CICS-ie terminali i baz danych jest bezsporna. Jeśli do tego dodamy wydajność i wiarygodność CICS, niezależnie od platformy, niezależnie od protokołu zawsze wysoka, to w taki sposób znajdujemy odpowiedź:

CICS jest najlepszym Integratorem Systemów i Sieci jaki istnieje na rynku. Jest **NAJLEPSZYM i NAJBARDZIEJ SPRAWDZONYM** tego typu **INTEGRATOREM**.

Jochen Arms
BETA Systems

BETA Systems - product description

BETA is for :

- people with ideas, people with energy, people with needs, people who want change, people that understand the needs of today's global marketplace, people that understand business

BETA's products Streamline operations in answer to the increase in data and the growth of user demands. BETA's mulit-platform, client/server solutions - bridge the last remaining cleft between 'glass house' and end user.

BETA's partners are 1500 of the largest information processing centers in fifty countries around the globe. And our world-wide distribution net, provides seamless support and service .

Products

BETA's product range provides a well-rounded base for strategic data center automation. Each product itself offers a suite of additional options including client/server applications enhancing each product's effectiveness in enterprise-wide usage.

BETA 92 Output Manager*

All of your system output, online, at your disposal - automatically checked for errors, automatically archived, ready for retrieval at a moment's notice.

BETA 92 Output Manager / Personal Computer Facility

All job information collected by BETA 92 Output Manager is displayed in clear, concise color-coded graphics on a PC workstation. Historical data such as when jobs ran, which ended in error and deviance from the planned schedule can be used for statistical analysis and planning.

BETA 93 Distribution Manager*

Enterprise-wide report distribution, from host to end user, printed or online, local or remote, over any MVS, AS/400 or PC environment. Using XEROX's UNIX-based XPSM print server, documents can be printed anywhere in the enterprise. Powerful options for storage and retrieval from tape or optical disk render provide unsurpassed document availability.

BETA 93 Distribution Manager / Personal Computer Facility

Makes reports available to PC users on individual workstations or on a LAN. BETA 93 PCF offers full viewing support of AFP and XEROX data streams. Users can export their lists into other PC applications for further processing.

BETA 93 Distribution Manager / AS/400 Facility

Makes reports available to AS/400 users on a medium familiar to AS/400 users. BETA 93 ASF also serves PC workstations in a LAN environment.

BETA 91 Balancing Manager*

Production balancing ensures detection of production errors through verification of control totals, volser numbers, record counts and all other production data. Data errors, program errors, system errors and handling errors are all caught as they occur and not after consequential damage has been done. BETA 91

offers interfaces to restart and recovery systems to avoid costly reruns.

BETA 44 Presentation Manager

Precise, comprehensive documentation of your production schedule in PC flowcharts. BETA 44 eliminates the need of lengthy reports from your ZEKE, OPC/ESA, OPC/A, CA-7 and CA-Scheduler* scheduling system.

BETA 55 Media Optimization Manager

Dataset stacking optimizes tape utilization, increases your scratch pool, maximizes the capacity of your automatic tape library and reduces resource utilization to a minimum. BETA 55 positions your media management for the high capacity tapes of the future.

BETA 77 Setup Manager

Well-structured, standardized construction of JCL modules for a more organized JCL library. Automatically generated JCL decreases manual effort and errors. Interfaces to scheduling systems provide automatic setup of the JCL for the daily production run.

BETA 88 Security Manager*

Easy administration of your RACF database with a structured user interface and command generation facility.

BETA 88 Security Manager / Personal Computer Facility

RACF access and scope structures are displayed in color-coded PC graphics. BETA 88 PCF is easy to use and its displays easy to understand making it the right toolkit for DP auditors.

VTAM Access Facility, CICS Access Facility, IMS Access Facility

BETA offers product options which allow the BETA products (those marked with '*') to be used in the MVS environments outside of TSO: IMS, VTAM and CICS. Besides being able to offer users the same interface as if they were working under ISPF, these facilities help the data center save on required address space.

ZEKE - Enterprise Scheduling

A production scheduling system for planning and executing your DP center's daily workload on one or more CPU's. ZEKE provides step-level condition code validation, workload balancing and parameter substitution, as well as schedule forecasting and simulation, and SAF security. ZEKE / GWS - Enterprise Scheduling Workstation, a feature of ZEKE, is a sophisticated workstation providing real-time graphical representation of ZEKE's processing tasks. ZEKE/GWS' Schedule Vision lets users watch their event flow as it occurs.

ZACK - Automated Operations

An automated systems operations tool for continually monitoring all console activity, instantly reacting to all messages and commands.

Z/TEAM 400 - Distributed Systems Management

Complete automated operations and scheduling functions for your integrated mainframe and AS/400 environments.

Z/TEAM UNIX - Distributed Systems Management

Complete automated operations and scheduling functions for your integrated mainframe and distributed environments.

ZARA - Removable Media Management

A revolutionary tape management system monitoring, managing and recording all tape processing while providing online, real-time access to all tape management data.

ZEBB - Rerun/Restart Management

A comprehensive job rerun/restart management system improving data center productivity by reducing the manual tasks required to restart or rerun a job.

BETA SYSTEMS SOFTWARE AG Alt-Moabit 97a D-10559 Berlin tel. +4930-39926290

Mr. Jochen Arms fax +4930-39926292

Borys Czerniejewski
MM Telecables Ltd

Moduł automatycznego tłumaczenia jako uniwersalny interfejs do bazy danych

1. Wstęp

Jednym z pierwszych niematematycznych zastosowań komputerów było wykorzystanie ich do tak zwanego przetwarzania języka naturalnego[1] (**Natural Language Processing — NLP**). Wiąże się to ściśle z odwiecznym marzeniem człowieka o stworzeniu maszyn myślących, jako że język jest systemem, w którym myśli wyrażamy. Dlatego też przetwarzanie języka zalicza się do dziedzin tzw. sztucznej inteligencji (**Artificial Intelligence — AI**). Ponadto komunikacja językowa znakomicie ułatwiłaby człowiekowi obsługę programów komputerowych, gdyż jest ona najczęstszym i najbardziej intuicyjnym, choć nie najprostszym, sposobem porozumiewania się używanym w życiu codziennym między ludźmi. Ideałem byłoby skonstruowanie takiego sprzętu i oprogramowania, żeby różnica pomiędzy komunikacją z maszyną, a komunikacją z drugim człowiekiem była niezauważalna.

Problem automatycznego przetwarzania języka posiada obszerną bibliografię. Niestety postępy w tej dziedzinie są nadal niewielkie, choć pojawia się coraz więcej komercyjnych programów wykorzystujących metody NLP. Wiąże się to przede wszystkim z dużym stopniem skomplikowania takich programów i koniecznością dogłębnej znajomości, czasami nie do końca sprawdzonych, teorii lingwistycznych i psychologicznych. Dlatego też przetwarzanie języka rzadko wychodzi poza obręb laboratoriów.

Jednymi z głównych działów przetwarzania języka naturalnego są automatyczne (maszynowe) tłumaczenie (**Machine Translation — MT**) i konstrukcja interfejsów w języku naturalnym (**Natural Language Interfaces — NLI**). Funkcjonalnie programy tłumaczące diametralnie różnią się od interfejsów naturalnojęzykowych, jednakże ze względu na konieczne do zastosowania metody przetwarzania języka są im bardzo bliskie. W niniejszym artykule omówiono program ***Tłumacz Pytań*** — interfejs do bazy danych tłumaczący pytania w języku polskim na kwerendy strukturalnego języka zapytań SQL. Podjęto też próbę pokazania podobieństwa pomiędzy wymienionymi dziedzinami maszynowego przetwarzania języka.

2. Problem rozumienia wypowiedzi

Pierwotnie systemy automatycznego tłuma-czenia, powstające w latach sześćdziesiątych, dokonywały przekładu metodą „słowo za słowo", tzn. kolejne słowa tekstu zastępowane były ich słownikowymi odpowiednikami w języku docelowym tłumaczenia. Szybko jednak odkryto wady tej metody. Przy weryfikacji jakości przekładu poprzez zwrotne tłumaczenie na język oryginału problem wieloznaczności słów uwidocznił się szczególnie jaskrawo w słynnym już tłumaczeniu sloganu „w zdrowym ciele, zdrowy duch," który po przetłumaczeniu z angielskiego na rosyjski i z powrotem brzmiał

[1] „język naturalny — język utworzony w procesie ewolucji biologicznej lub społecznej (...)" — za „*Słownikiem informatyki polsko — angielsko — rosyjskim*" pod red. Hanny Jezierskiej, WNT, Warszawa (wyd. 3.) 1989

„spirytus był świeży, ale mięso cuchnęło."[2] Rozwiązaniem tego problemu było wyposażenie programów tłumaczących w narzędzia do dokonywania wstępnych lub końcowych poprawek w przekładanym tekście (**pre–** i **post–editing**) i skoncentrowanie się raczej na programach wspomagających tłumaczenie (**Computer Aided/Assisted Translation — CAT**) niż na „niemożliwych do stworzenia" programach w pełni zautomatyzowanego przekładu o wysokiej jakości (**Fully Automated High Quality Translation — FAHQT**). Stanowisko to znalazło wyraz w artykule pioniera komputerowego przetwarzania języka, profesora Bar–Hillela z uniwersytetu w Jerozolimie, zatytułowanym: „Automatyczne tłumaczenie — koniec złudzeń."[3] Dopiero późniejsze prace, prowadzone na początku lat osiemdziesiątych, pokazały że automatyczne tłumaczenie jest możliwe, jednakże tylko przy założeniu że komputer będzie **rozumiał** tłumaczony tekst. Okazuje się bowiem, że język ma strukturę fraktalną i nawet najprostszy problem przenoszenia wyrazów (**hyphenation**) i podziału ich na sylaby wymaga pełnej analizy semantycznej. Na dowód tego można przytoczyć dwa różne podziały (i dwie wymowy!) homoformu „obmierzać," zależnie od jego znaczenia wynikającego z szerszego kontekstu językowego i sytuacyjnego.[4]

Zastanówmy się więc, co oznacza **rozumienie** jakiejkolwiek wypowiedzi: *„(...) proces rozumienia definiujemy jako tworzenie się reprezentacji tekstu w umyśle (...)"*[5] A zatem podstawowym problemem przetwarzania języka jest stworzenie systemu reprezentacji znaczeń i algorytmu przejścia od tekstu wypowiedzi do struktur znaczeniowych. Barierą na jaką tu napotykamy jest nasza nieznajomość reprezentacji znaczenia w mózgu ludzkim i niemożność bezpośredniego zbadania procesu rozumienia u człowieka. Posługujemy się tu metodami pośrednimi: *„W dziedzinie sztucznej inteligencji przyjmuje się, że rozumienie przez system wypowiedzi użytkownika jest określone przez reakcję systemu na tę wypowiedź."*[6] Reakcją na wypowiedź może być na przykład wykonanie zleconej czynności lub inna wypowiedź, powiedzmy: odpowiedź na pytanie, czy skierowana do trzeciego rozmówcy wypowiedź będąca tłumaczeniem wypowiedzi pierwotnej na inny język. Rodzaj reakcji zależy oczywiście od szerszego kontekstu sytuacyjnego w którym pojawiła się analizowana wypowiedź, na który składają się: język (kod), scena, wspólna wiedza rozmówców i temat komunikacji. Tutaj zajmiemy się jedynie sceną, a dokładniej wymuszanymi przez nią schematami komunikacji.[7]

3. Schemat komunikacyjny

W najprostszym przypadku dialogu dwóch osób na scenie znajdują się tylko rozmówcy, którzy na przemian reagują wzajemnie na swoje wypowiedzi. Jeśli pominiemy w tym miejscu występujące w teorii informacji etapy kodowania i dekodowania, jako czynności czysto techniczne, nasz schemat komunikacji wygląda następująco:

[2] Przypadek ten jest tak często cytowany bez podania źródła, że autorowi nie udało się dotrzeć do oryginału. Zauważmy, że w niniejszej pracy posługujemy się polskimi przekładami obu fraz.

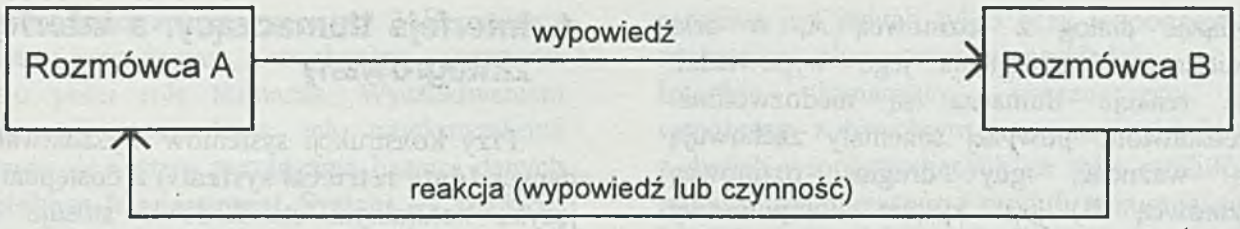
[3] Yeoshua Bar-Hillel, *„Machine Translation — The end of illusion"*

[4] Zauważmy, że „zaczynać budzić niechęć" może samo „pobieranie wymiarów," czyli „obmierzać może obmierzanie," ale nie odwrotnie.

[5] Ida Kurcz, Anna Polkowska, *„Interakcyjne i automatyczne przetwarzanie informacji językowych, na przykładzie procesu rozumienia tekstu czytanego na głos,"* Ossolineum, Wrocław 1990, str. 8

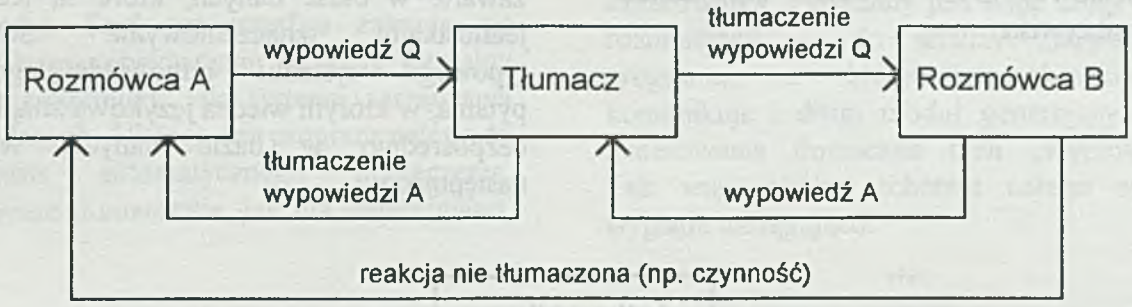
[6] Leonard Bolc, Małgorzata Cichy, Ludmiła Różańska, *„Przetwarzanie języka naturalnego,"* WNT, W-wa 1982

[7] Po pojęciem „komunikacja" rozumiemy tu naprzemienny ciąg wypowiedzi i reakcji.



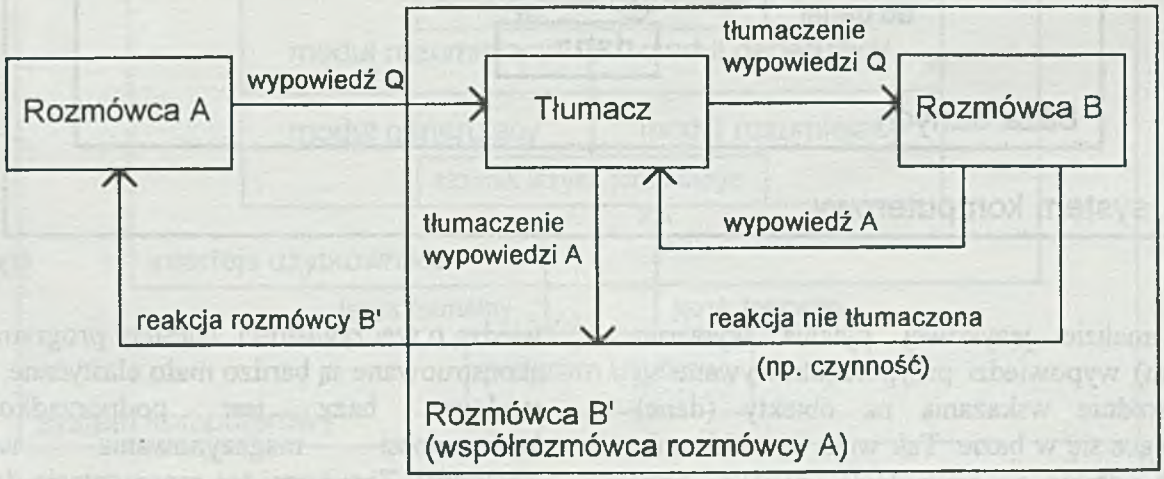
(rys. 1.)

Reakcją rozmówcy B może być wypowiedź (np. odpowiedź na pytanie rozmówcy A), czynność lub obie jednocześnie. W powyższym schemacie zakładamy, że dialog odbywa się pomiędzy rozmówcami używającymi tego samego języka (kodu). W przeciwnym przypadku w schemacie musi pojawić się tłumacz (będący de facto trzecim rozmówcą). Oczywiście, jeśli reakcją rozmówcy B jest jakaś wypowiedź, to skierowana ona zostaje również do tłumacza, który tłumaczy ją na język zrozumiały dla rozmówcy A:



(rys. 2.)

Z punktu widzenia rozmówcy A, najwygodniej jednak jest, by tłumacz był „przezroczysty", tzn. by jego obecność była niezauważalna tak, by dialog z rozmówcą B był w jak największym stopniu podobny do sytuacji w której obaj rozmówcy porozumiewają się w tym samym języku. Rozmówca A skłonny jest więc uznać za swego współrozmówcę system złożony faktycznie z tłumacza i rozmówcy B:



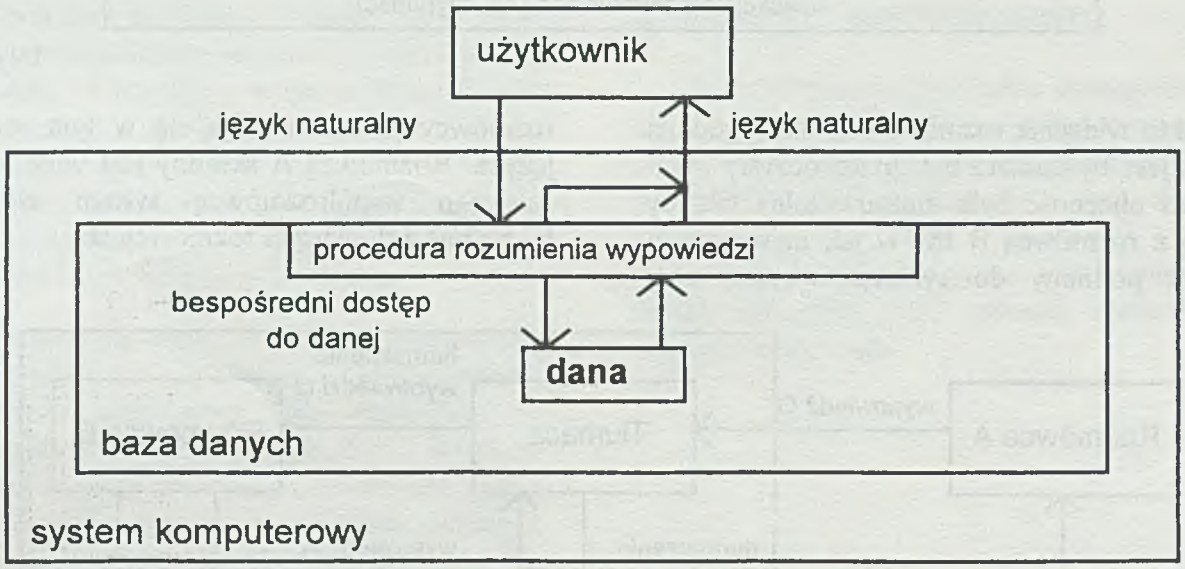
(rys. 3.)

Zauważmy, że w schemacie tym reakcje pochodzące bezpośrednio od rozmówcy B i reakcje tłumacza są nierozróżnialne. Zauważmy również, że tłumacz może w tym przypadku tylko przetłumaczyć wypowiedź rozmówcy A i przekazać ją (jej tłumaczenie) rozmówcy B, lub

nawiązać dialog z rozmówcą A, w celu pełniejszego zrozumienia jego wypowiedzi. Inne reakcje tłumacza są niedozwolone. Przedstawione powyżej schematy zachowują swą ważność, gdy drugim rozmówcą (rozmówcą B) jest system komputerowy (komputer wraz z oprogramowaniem). Rozmówca A zwany jest wtedy użytkownikiem (systemu). Tłumaczem pomiędzy użytkownikiem a systemem może być program automatycznego tłumaczenia. Program ten może być integralną częścią systemu, z którym użytkownik się porozumiewa (interfejsem użytkownika). Wtedy postrzeganie przez użytkownika całego systemu jako swego współrozmówcy (rys. 3.) staje się jeszcze bardziej oczywiste.

4. Interfejs tłumaczący, a interfejs zintegrowany

Przy konstrukcji systemów wyszukiwania danych (**data retrieval systems**) z dostępem w języku naturalnym standardowo stosuje się schemat komunikacji przedstawiony na rys. 1. Dlatego też systemy te nazywane są często systemami odpowiadającymi na pytania sformułowane w języku naturalnym (**natural language question answering systems**). Moduł analizy języka jest w nich jedynie mocno zintegrowanym interfejsem odwzorowującym wyekstrahowane z tekstu obiekty i relacje bezpośrednio na odpowiednie obiekty i relacje zawarte w bazie danych, które są jedynymi jednostkami znaczeniowymi. Struktura typowego systemu odpowiadającego na pytania, w którym wiedza językowa znajduje się bezpośrednio w bazie danych, wygląda następująco:



(rys. 4.)

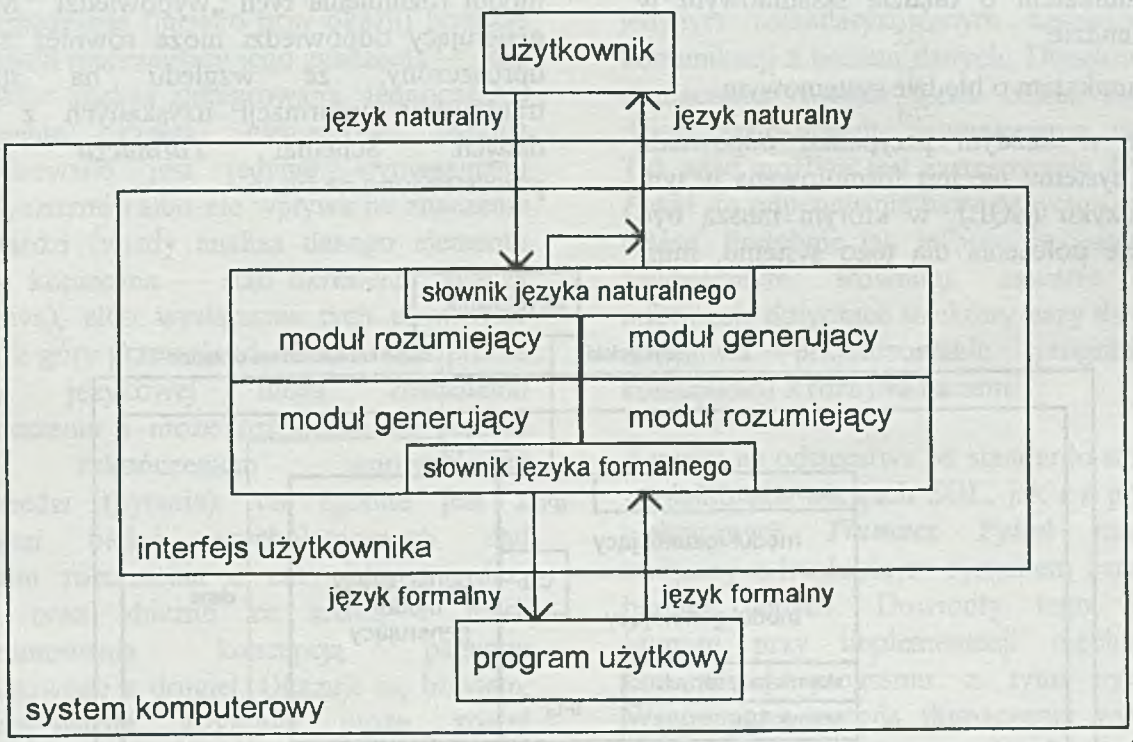
Przy analizie językowej pytania wyrazom (frazom) wypowiedzi przyporządkowywane są bezpośrednie wskazania na obiekty (dane) znajdujące się w bazie. Tak więc wyszukiwanie danych odbywa się równolegle i niejako „przy okazji" językowego rozpoznawania elementów wypowiedzi (wyszukiwania ich w słowniku). Podejście takie jest zgodne z teoriami psychologicznymi z których wiadomo, że wiedza leksykalna ściśle wiąże się z naszą wiedzą o rzeczywistości. Niestety programy tak skonstruowane są bardzo mało elastyczne, gdyż struktura bazy jest podporządkowana konieczności magazynowania wiedzy językowej. Zazwyczaj też reprezentacja danych (wiedzy) i mechanizmy operacji na nich są w takich programach tworzone od podstaw.

Przy konstrukcji ***Tłumacza Pytań*** zastosowano natomiast podejście

odpowiadające schematowi z rys. 3. Rozmówcą B jest tu baza danych, a mechanizm rozumienia tekstu pełni rolę tłumacza. Wyszukiwaniem informacji i prezentacją ich użytkownikowi zajmuje się system zarządzania bazami danych (**Database Management System — DBMS**). To jest właśnie reakcja systemu obserwowana przez użytkownika. Oczywiście z punktu widzenia użytkownika jego współrozmówcą jest system komputerowy traktowany jako całość (rozmówca B'). Przy konstrukcji programu przyjęto, że dane będące odpowiedziami bazy na zadane przez użytkownika pytanie, są zrozumiałe same przez się i nie wymagają komentarza. W praktyce można więc było zrezygnować z tłumaczenia odpowiedzi. Stąd też interfejs zajmuje się jedynie tłumaczeniem pytań na język formalny (SQL), „zrozumiały" dla systemu zarządzania bazami danych. Dlatego też program należy do programów automatycznego tłumaczenia. Rzeczywiste tłumaczenie, jak już wspomniano, możliwe jest jednak tylko przy jednoczesnym zrozumieniu tłumaczonej wypowiedzi.

Interfejs tłumaczący, przeznaczony do współpracy z dowolnym programem składa się z dwóch współpracujących ze sobą modułów: modułu rozumiejącego i modułu generującego. Moduł rozumiejący buduje reprezentację wewnętrzną znaczenia wypowiedzi użytkownika. Moduł generujący zajmuje się zaś generacją odpowiedniej wypowiedzi w języku formalnym.

W zwykłym przypadku tłumaczenia dialogu, wypowiedzi drugiego (komputerowego) rozmówcy są również analizowane przez interfejs, a następnie tłumaczone na język użytkownika. Potrzebny jest więc drugi moduł rozumiejący — do analizy „wypowiedzi" programu, z którym użytkownik się komunikuje i drugi moduł generujący — do generowania tłumaczeń tych „wypowiedzi". Tak więc ogólny schemat całego systemu wygląda następująco:



(rys. 5.)

Na rysunku zaznaczono, że moduł rozumiejący wypowiedzi użytkownika może dopytywać się o pewne fragmenty wypowiedzi, których nie potrafi zrozumieć (lub są niejednoznaczne). Zakłada się, że z programem użytkowym można komunikować się przy pomocy jakiegoś

języka formalnego, lub innego ograniczonego zestawu poleceń. Reakcja programu jest również ściśle określona (nawet jeśli wystąpił błąd w jego działaniu). Dlatego nie jest konieczne, by moduł rozumiejący odpowiedzi programu użytkowego miał możliwość dopytywania się o znaczenie odebranej od tego programu informacji.

5. Budowa *Tłumacza Pytań*

Powyżej przedstawiono ogólną strukturę interfejsu tłumaczącego do dowolnego programu użytkowego. W naszym przypadku programem, z którym użytkownik „rozmawia" jest system zarządzania bazami danych, umożliwiający odpytywanie bazy przy pomocy języka zapytań do relacyjnych baz danych — SQL. Systemy tego typu posiadają taką właściwość, że na komendę języka SQL reagują wypowiedzią, która jest (alternatywnie):

a) informacją (zestawem danych), której żądał użytkownik,

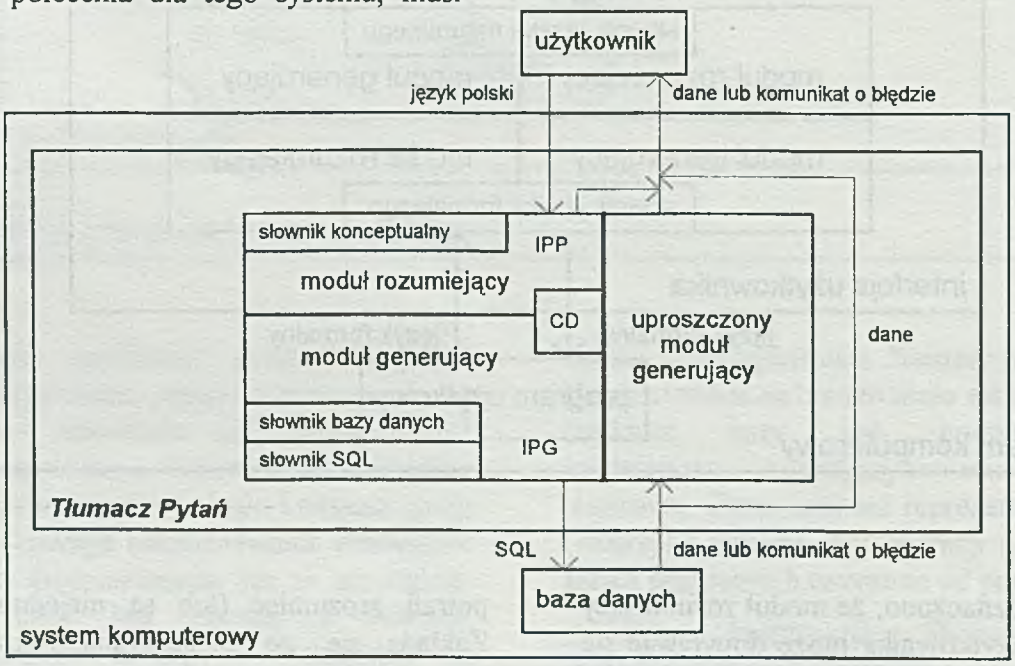
b) komunikatem o błędzie składniowym w komendzie,

c) komunikatem o błędzie systemowym.

Ponieważ w każdym przypadku odpowiedź (reakcja) systemu nie jest formułowana w tym samym języku (SQL), w którym muszą być zapisywane polecenia dla tego systemu, musi być ona traktowana jako czynność. Symetria przedstawionego powyżej schematu zostaje więc zachwiana. Przy konstrukcji ***Tłumacza Pytań*** przyjęto następujące założenia:

1. dane zwracane przez system obsługi baz danych nie podlegają interpretacji i jako wystarczająco czytelne są bezpośrednią odpowiedzią dla użytkownika,
2. w generowanych automatycznie przez generator komendach SQL nie ma błędów składniowych,
3. komunikaty o błędach systemowych (zazwyczaj kod błędu i komentarz w języku angielskim) są zastępowane odpowiednimi, przygotowanymi wcześniej komunikatami w języku polskim, z pominięciem analizy znaczeniowej komunikatów systemu.

Jak z tego widać „wypowiedzi" (reakcje) systemu zarządzania bazami danych nie są właściwie interpretowane (dokonuje się tylko rozróżnienia ich typów). Zbędny jest więc moduł rozumienia tych „wypowiedzi". Moduł generujący odpowiedzi może również zostać uproszczony, ze względu na sposób traktowania informacji uzyskanych z bazy danych. Schemat ***Tłumacza Pytań*** przedstawiono na rys. 6.:



(rys. 6.)

Tłumacz Pytań posiada własny system pojęć, które można opisać zarówno w języku wejściowym (polskim) jak i docelowym (SQL). Pojęcia te konstruowane są z użyciem struktur zaczerpniętych z teorii zależności pojęciowych (**Conceptual Dependency — CD**) opracowanej przez Rogera C. Schanka na uniwersytecie w Yale (Stany Zjednoczone), w latach 1969-1981 ([Sch-75], [Sch-81]). Można więc powiedzieć, że program istotnie rozumie tłumaczone pytania. Zastosowanie tej teorii w zasadniczy sposób odróżnia ***Tłumacza Pytań*** od innych, powszechnie znanych programów przetwarzających języki naturalne. Otóż nie ma tu tradycyjnego rozbicia procesu rozumienia na etapy analizy gramatycznej i semantycznej. Do analizy językowej (parsingu) wypowiedzi wykorzystano metodę dedykowaną teorii zależności pojęciowych zwaną zintegrowaną, częściową analizą językową (**Integrated Partial Parsing — IPP**), opracowaną również przez zespół R. C. Schanka ([Sch-75], [Sch-80], [Sch-81]). Związki gramatyczne występujące w pytaniu są tu uwzględniane (niejako przy okazji) podczas konstrukcji reprezentacji jego znaczenia — stąd określenie: analiza zintegrowana. Jednocześnie wystąpienie części elementów pytania uwarunkowane jest jedynie wymaganiami gramatycznymi i albo nie wpływa na znaczenie wypowiedzi (wtedy analiza danego elementu nie jest konieczna — stąd określenie: analiza częściowa), albo wystąpienie tych elementów można z góry przewidzieć. Dzięki temu proces analizy językowej ulega znacznemu przyspieszeniu i może rozpocząć się jeszcze przed zakończeniem wprowadzania wypowiedzi (pytania), co zgodne jest z wynikami badań psychologicznych nad procesem rozumienia u człowieka z jednej strony oraz zbieżne ze znaną z teorii oprogramowania koncepcją parsingu przyrostowego z drugiej. Okazuje się bowiem, że wypowiedź językowa może zostać zrozumiana jeszcze przed odebraniem jej zakończenia lub mimo zakłóceń w odbiorze (np. w trakcie rozmowy telefonicznej). Zastosowanie tradycyjnej analizy gramatycznej w takich przypadkach nie daje pożądanego rezultatu. Dokładniejsze omówienie tego problemu można znaleźć w pracach R. C. Schanka oraz powstałych w Polsce pracach bazujących na jego teorii ([Lub-90], [Cze-94]).

6. Dążenie do uniwersalności — ograniczenia

Jednym z podstawowych założeń pracy dyplomowej autora było stworzenie możliwie uniwersalnego interfejsu, który będzie współpracował z komercyjnymi systemami obsługi baz danych. Jak już powiedziano ***Tłumacz Pytań*** jest interfejsem umożliwiającym użytkownikowi odpytywanie bazy danych z użyciem języka naturalnego. W obecnej wersji do odpytywania stosowany jest język polski. Z uwagi na uniwersalność mechanizmu (wszystkie informacje językowe zawarte są w zewnętrznym słowniku) możliwe jest przystosowanie programu do komunikacji w innym języku. ***Tłumacz Pytań*** przekłada pytania w języku polskim na kwerendy strukturalnego języka zapytań SQL, który jest jedynym ustandaryzowanym narzędziem do komunikacji z bazami danych. Docelowy język tłumaczenia również może zostać zmieniony dzięki zastosowaniu zewnętrznego słownika. Tak więc możliwe jest zastosowanie ***Tłumacza Pytań*** do odpytywania bazy danych w systemie dBase. Podobnie jak informacje językowe w zewnętrznym słowniku zawarte zostały informacje dotyczące struktury bazy danych, co umożliwia przystosowanie programu do komunikacji z różnymi bazami.

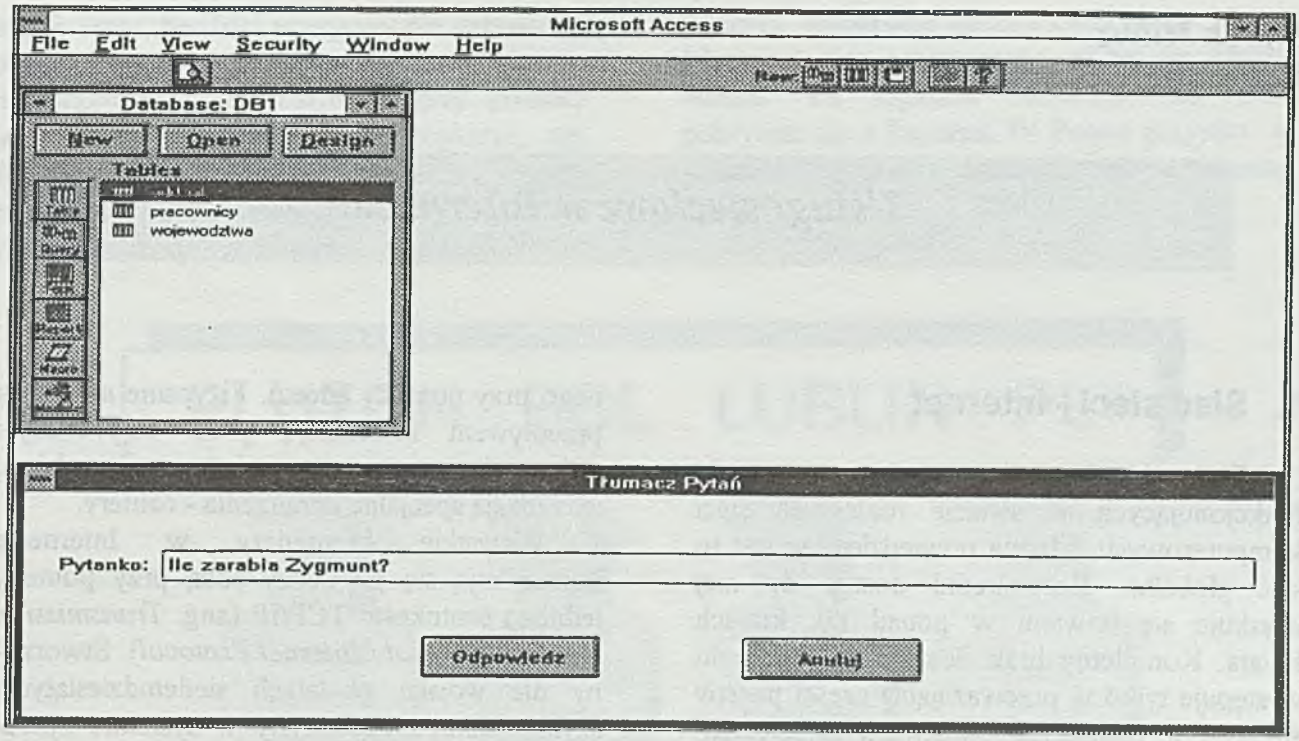
Z uwagi na odstępstwa od standardu stosowane w niektórych wersjach SQL, jak i z powodów technicznych ***Tłumacz Pytań*** musi być związany z konkretnym systemem zarządzania bazami danych. Dowiodły tego niezbicie kłopoty przy implementacji mechanizmów komunikacji programu z tymi systemami. Wspomniana metoda tłumaczenia wydaje się jednak dostatecznie uniwersalna, by można było ją stosować z dowolną bazą danych, obsługiwaną przez dowolny system zarządzania bazami danych, zawierającą informacje z dowolnej dziedziny. Przy adaptacji do nowej bazy danych wymaga to tylko wymiany

słownika bazy i (jeśli baza dotyczy innej dziedziny) słownika języka naturalnego. Przy zmianie języka kwerend, lub języka naturalnego może się okazać konieczne poczynienie pewnych zmian w kodzie programu. Przewiduje się jednak, że zmiany te dotyczyć będą jedynie dodania nowych akcji metajęzyków analizatora lub generatora. Operacja taka nie jest jednak trudna i nie wymaga śledzenia całości kodu programu. Uzupełnianie bazy danych nowymi informacjami nie powoduje konieczności jakichkolwiek zmian w słowniku. Umożliwia to mechanizm rozpoznawania nazw. Niestety użycie tego mechanizmu powoduje, że generowane kwerendy stają się bardziej skomplikowane i wydłuża się czas reakcji systemu. Możliwe jest też, że system nie będzie potrafił znaleźć poprawnej formy podstawowej nazwy. Konieczne jest wtedy uzupełnienie słownika nazw. Mechanizm ten jest również ściśle zależny od języka polskiego. Oczywiście wzorce końcówek zawarte w słowniku nazw mogą być wymienione, lecz w innych językach zasada odmiany nazw może być zupełnie inna. Trzeba jednak stwierdzić, że z omówionymi poniżej ograniczeniami, program jest przenośny ze względu na bazę danych, język manipulacji na danych i język naturalny.

8. Zakończenie

Obecnie ***Tłumacz Pytań*** umożliwia komunikację z bazą zawierającą informacje na temat oddziałów i pracowników hipotetycznego przedsiębiorstwa. Baza została utworzona przy pomocy oprogramowania *Microsoft Access*. Interfejs umożliwia zadawanie pytań o rozmaitej konstrukcji. Przy budowie programu zwrócono uwagę na: dowolny szyk wyrazów w pytaniu, pytania o osoby z użyciem nazwiska lub imienia i nazwiska w dowolnej kolejności, użycie nazwy (np. nazwiska) w dowolnym przypadku gramatycznym, pytania o dwa i więcej obiektów (np. osób), pytania o dwa i więcej predykatów, pytania o informacje spoza bazy (np. o czas systemowy). Możliwe jest wystąpienie kilku z wymienionych problemów w jednym pytaniu. Problem rozpoznawania nazw i ich odmiany został rozwiązany heurystycznie, lecz w ostatecznej wersji należałoby wprowadzić mechanizm uczenia się. Nie zostały rozwiązane problemy rozpoznawania liczb i zasięgu spójników oraz interpunkcji. Analiza wypowiedzi jest dokonywana dopiero po jej całkowitym wprowadzeniu. Jednak mimo to czas potrzebny na dokonanie tłumaczenia pytania na kwerendę SQL jest o dwa rzędy wielkości krótszy od czasu wyszukiwania danych w bazie ok. 3000 pracowników przez system *MS Access* i jest krótszy od jednej sekundy. Użycie programu ułatwia graficzny interfejs. Konstrukcja tego interfejsu, duża szybkość działania programu i możliwość współpracy z komercyjnym systemem obsługi baz danych możliwe były dzięki zastosowaniu języka programowania C^{++}.

Poniższy rysunek przedstawia ***Tłumacza Pytań*** podczas pracy z systemem *MS Access*:



(rys. 7.)

9. Literatura

[Cze-94] Borys Czerniejewski, Mechanizm rozumienia tekstu jako inteligentny interfejs, Praca dyplomowa, Katedra Informatyki AGH, Kraków 1994

[Lub-90] Wiesław Lubaszewki, Rozumienie tekstu przez komputer, Wprowadzenie do problematyki i inteligentny program komputerowy, Instytut Języka Polskiego PAN, Kraków 1990

[Sch-75] Roger C. Schank, Conceptual Information Processing, seria: Fundamental Studies in Computer Science, tom 3., North-Holland, Amsterdam, etc. 1975 (wyd. 2., 1984)

[Sch-80] Roger C. Schank, Michael Lebowitz, Lawrence Birnbaum, An Integrated Understander, American Journal of Computational Linguistics, Vol. 6, No. 1, January - March 1980

[Sch-81] Roger C. Schank, Christopher K. Riesbeck (red.), Inside Computer Understanding: Five Programs Plus Miniatures, Lawrence Erlbaum Associates, Hillsdale 1981

Marek Miłosz
Politechnika Lubelska

Usługi sieciowe w Internecie

1. Sieć sieci - Internet

Sieć Internet jest jedną z wielu funkcjonujących w świecie rozległych sieci komputerowych. Można powiedzieć, że jest to sieć globalna. Bezpośredni dostęp do niej uzyskuje się bowiem w ponad 80. krajach świata. Kompletny brak dostępu do Internetu występuje tylko w przeważającej części państw Afryki i w niektórych państwach azjatyckich. W Europie białą plamą są jedynie Bałkany - niektóre państwa byłej Jugosławii i Albania. Sieć Internet nie jest największą siecią komputerową. Nie obsługuje także największej liczby użytkowników - prześcigają ją w tym sieci komercyjne. Jednak od paru lat corocznie podwaja się liczba włączonych do sieci Internet komputerów, a co za tym idzie i liczba użytkowników. Skąd więc taki burzliwy rozwój?

Internet jest siecią łączącą wiele podsieci, zorganizowanych w hierarchiczną strukturę. Łączone są przy tym sieci różnych typów. Internet ukrywa bowiem przed użytkownikiem szczegóły budowy sprzętu i umożliwia transfer informacji pomiędzy komputerami niezależnie od ich typu i umiejscowienia. Duża popularność Internetu wynika z minimalnych wprost wymagań dotyczących instalacji sieciowej i dołączanych do niej komputerów. Większość istniejących lokalnych sieci komputerowych (LAN, ang. *Local Area Network*), zarządzanych przez takie systemy jak VMS, Unix czy Novell, może być bezpośrednio włączone do Internetu. Podobnie zresztą jak i najprostszy, pojedynczy PC.

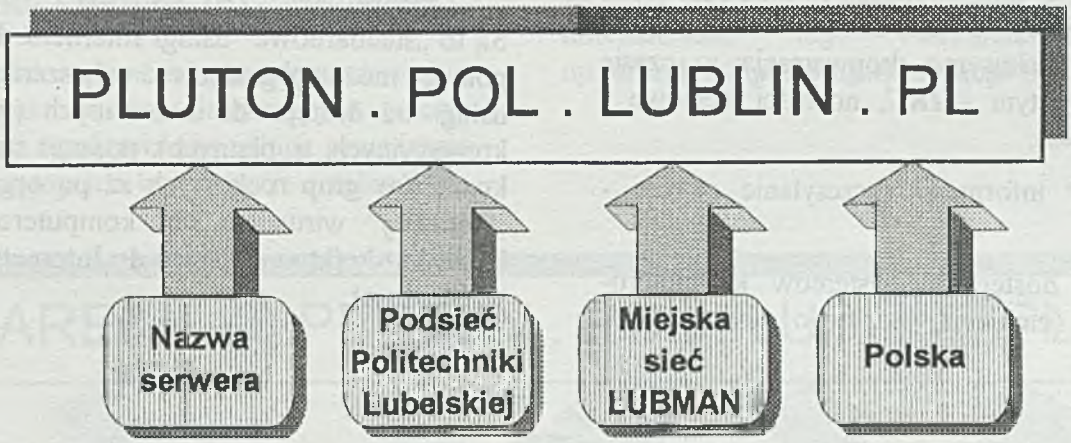
Komunikacja pomiędzy komputerami w sieci Internet z logicznego punktu widzenia jest bardzo prosta. Komputer-nadawca wysyła komunikat do komputera-odbiorcy, określonego przy pomocy adresu. Fizycznie natomiast przepływem informacji, poza najprostszym połączeniem wewnątrz pojedynczej podsieci, zarządzają specjalne urządzenia - routery.

Wszystkie komputery w Internecie komunikują się pomiędzy sobą przy pomocy jednego protokołu: TCP/IP (ang. *Transmission Control Protocol / Internet Protocol*). Stworzony dla wojska w latach siedemdziesiątych, szybko zaimplementowany w systemie operacyjnym Unix i dostępny obecnie na praktycznie wszystkich typach komputerów (w postaci odpowiedniego oprogramowania) protokół TCP/IP łączy w sobie dwa standardy:

- IP - podstawowy mechanizm transmisji danych, realizujący bezpołączeniowy (tzw. datagramowy) model komunikacji; dane podzielone na fragmenty, są odpowiednio oznaczane i wysyłane w wirtualną sieć; IP kontroluje poprawność transmisji pojedynczego fragmentu (tzn. datagramu, ramki);
- TCP - jest protokołem odpowiadającym za podział danych u nadawcy i ich późniejsze łączenie w odpowiedniej kolejności w komputerze-odbiorcy; umożliwia także pełną kontrolę nad poprawnością transmisji całości danych i wymusza ewentualną powtórną transmisję „zagubionego" fragmentu.

Każdy komputer-serwer (np. komputer z systemem Unix lub serwer sieci Novell) w sieci Internet posiada swój unikalny identyfikator - adres logiczny (IP). Adres ten składa się z numeru sieci, podsieci i konkretnego komputera-serwera. Adresy są liczbami 32 bitowymi podzielonymi kropkami na 4 grupy 8. bitowe, np. ***148.81.16.51***. W sieci Internet

istnieje drugi, bardziej przyjazny dla człowieka, sposób adresowania. Komputer-serwer jest w tym przypadku identyfikowany przy pomocy ciągu znaków oddzielonych kropkami, np. ***pluton.pol.lublin.pl***. Poszczególne grupy znaków wyznaczają miejsce danego komputera w hierarchicznej strukturze logicznej sieci Internet. Najbardziej ogólna klasa hierarchiczna (domena) jest umieszczana z prawej strony adresu. Ta logiczna struktura nie musi pokrywać się z fizyczną. W Polsce przyjęto w zasadzie terytorialny podział sieci na domeny (rys. 1).



Rys. 1. Struktura przykładowego adresu komputera-serwera w polskiej domenie sieci Internet

„Tłumaczeniem" adresów (w sposób niewidoczny dla użytkownika, choć trochę trwający) zajmują się specjalne komputery-serwery nazw domenowych (ang. *Domain Name Server*).

Począwszy od 1992 roku nadawanie adresów internetowych zostało zdecentralizowane z zachowaniem hierarchicznej struktury. Nadrzędną dla Europy jest organizacja RIPE z siedzibą w Amsterdamie, a w Polsce: NASK (Naukowa i Akademicka Sieć Komputerowa, ul. Bartycka 18, 00-716 Warszawa). W poszczególnych miejskich sieciach (MAN) funkcje te spełniają ich operatorzy.

Sieć Internet posiada połączenia z innymi globalnymi i krajowymi sieciami komputerowymi, w tym z jedną z najstarszych akademickich sieci - BITNET. Sieć ta znana w Polsce również pod nazwą EARN (ang. *European Academic and Research Network*) została podarowana międzynarodowej społeczności akademickiej przez koncern IBM. W chwili obecnej w Polsce pracuje w sieci EARN 16 komputerów (przeważnie dużych maszyn typu mainframe, głównie produkcji firmy IBM).

Z poziomu Internetu łączność z innymi (w tym również i czysto komercyjnymi) sieciami można uzyskiwać przy pomocy różnorodnych komputerów pośredniczących (ang. *gateway*). Postać adresu komputerów w różnych sieciach jest różny. Przykładowo dostęp do komputera węzłowego w sieci BITNET z poziomu sieci Internet uzyskuje się przy pomocy adresu, składającego się z nazwy komputera w sieci BITNET i po kropce słowo: bitnet, np. ***plszus11.bitnet***.

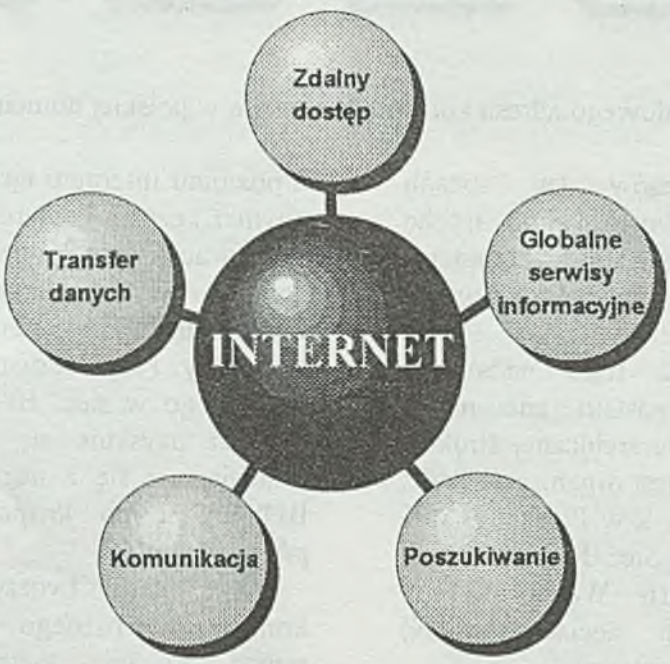
Sieć Internet tworzy obecnie ponad 10 mln komputerów różnego typu. Korzysta z niej ponad 50 mln ludzi na całym świecie. Użytkownicy ci mają do wyboru całą gamę różnorodnych usług sieciowych, olbrzymie zasoby informacji przechowywane na różnorodnych serwerach (a udostępniane w przeważającej części bezpłatnie), a także szybką, dość tanią metodę nieskrępowanego niczym transferu informacji.

2. Typy usług sieciowych

Podstawową usługą każdej sieci komputerowej jest transfer informacji.
W Internecie, w zależności od przeznaczenia i obudowy informacyjno-programowej, można wyróżnić pewne grupy dostępnych usług (rys. 2):

- komunikacja (poczta elektroniczna - ***e-mail***, grupy dyskusyjne, konwersacja w czasie rzeczywistym - ***IRC***, nowości sieciowe - ***NetNews***);
- transfer informacji (przesyłanie plików - ***ftp***);
- zdalny dostęp do systemów komputerowych (emulacja zdalnego terminalu - ***telnet***);
- globalne serwisy informacyjne (rozproszone systemy informacyjne: znakowe - ***Gopher*** i hypertekstowe - ***WWW***);
- poszukiwanie informacji o zasobach sieci (wyszukiwanie zasobów sieciowych - ***Archie***, przeszukiwanie baz danych - ***WAIS***) i użytkownikach sieci (lokalizacja użytkownika - ***Whois***, ***Netfind***).

Są to „standardowe" usługi Internetu. Przy ich pomocy można organizować cały szereg innych usług: od dostępu do baz danych (w tym i komercyjnych, tj. płatnych), poprzez transmisję koncertów grup rockowych aż po organizację „globalnej" wirtualnej gry komputerowej, w której każdy (kto ma dostęp do Internetu) może wziąć udział.



Rys. 2. Grupy usług dostępnych w Internecie

W Internecie można organizować obliczenia rozproszone, w których równolegle weźmie udział 100, 1 000 lub nawet 10 000 komputerów, można uzyskać za darmo oprogramowanie (w tym i wersje źródłowe), można prowadzić nie kończące się dyskusje na tematy naukowe czy też damsko-męskie, przeprowadzić akcję protestacyjną, zapoznać się z zabytkami miast w celu zaplanowania włóczęgi wakacyjnej lub dowiedzieć się jaki film grają dzisiaj w kinie uniwersytetu w Oxfordzie. Nie sposób wyliczyć wszystkie usługi oferowane przez sieć Internet.

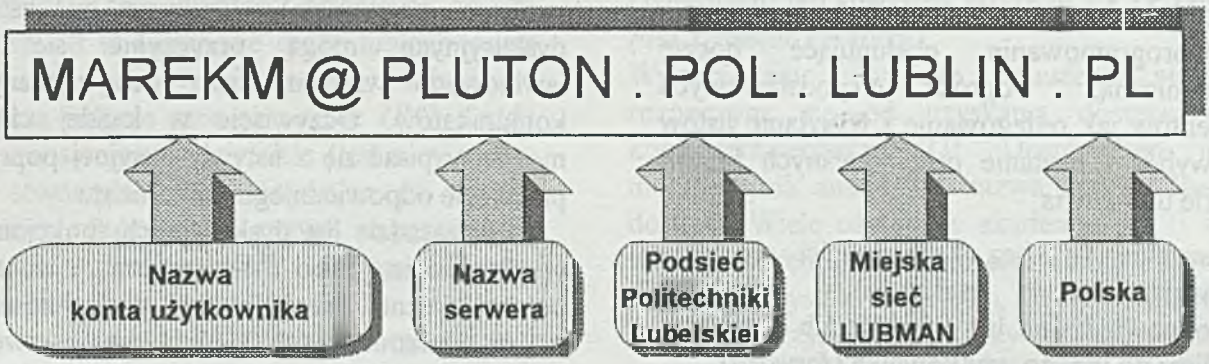
Należy podkreślić, że informacja w sieci Internet jest na ogół jawna. Użytkownik sieci sam powinien zadbać o bezpieczeństwo swoich danych poprzez np. odpowiednie kodowanie informacji. Ponadto włączenie systemu

komputerowego do sieci globalnej naraża ten system na nieustanne ataki różnego typu włamywaczy. Problemy ochrony bezpieczeństwa sieciowych systemów komputerowych ze szczególnym uwzględnieniem sieci Internet zostały szerzej przedstawione w pracy [6].

3. Komunikacja przy pomocy sieci Internet - e-mail, IRC, NetNews

Jedną z najbardziej popularnych usług realizowanych w sieci Internet jest poczta elektroniczna i pokrewne jej usługi: organizacja grup dyskusyjnych i konwersacja (realizowana na razie głównie przy pomocy klawiatury) w czasie rzeczywistym.

Poczta elektroniczna (*e-mail*) umożliwia przesyłanie informacji od jednego użytkownika sieci Internet do drugiego. Użytkownik w tym przypadku jest identyfikowany przy pomocy tzw. adresu pocztowego. Adres ten składa się z nazwy konta użytkownika na komputerze-serwerze, po którym następuje znak @ i adres internetowski tego komputera-serwera, np. **marekm@pluton.pol.lublin.pl** (rys. 3)



Rys. 3. Struktura przykładowego adresu elektronicznego w sieci Internet

Przy wysyłaniu listów w ramach jednej podsieci możliwe jest na ogół skracanie adresów poprzez pomijanie końcowych nazw poszczególnych podsieci. W skrajnych przypadkach przesyłania poczty do użytkownika, który posiada konto na danym komputerze-serwerze, zbędne jest podawanie adresu komputera. Listy można wysyłać (i otrzymywać) do adresatów mających konta (skrytki pocztowe) w innych niż Internet sieciach, np. wspomnianej już sieci BITNET, USENET, Compuserve czy też America Online. Problemem jest w tym przypadku odpowiednie adresowanie listów. Pomimo dużego podobieństwa każdorazowo jednak należy dowiedzieć się o sposobie ich adresowania. Adres poczty elektronicznej w akademickiej sieci BITNET tworzy się w identyczny sposób jak w sieci Internet, np. **ZS@PLSZUS1.BITNET**. Dla innych sieci może okazać się niezbędna znajomość konkretnej drogi listu i adres może być bardziej skomplikowany, np. **pw%sun-rc.cp.co.uk@ukc.ac.uk**.

Nadchodzące listy są gromadzone w plikach dyskowych na komputerze-serwerze użytkownika-odbiorcy i czekają na odczytanie. Można zatem przejrzeć je w wolnej chwili. Ta cecha, jak i szereg innych, przyczyniła się do dużej popularności poczty elektronicznej.

Pocztę elektroniczną obsługują różne programy, w tym i cały szereg ogólnie dostępnych (w sieci Internet oczywiście) i bezpłatnych (typu: *Public Domain* lub *Shareware*). Programy te mają z reguły podobne funkcje i sposoby obsługi. Wbudowane funkcje obsługi poczty posiada Unix, OS/2 czy też Windows'95.

Listy, wysyłane przy pomocy poczty elektronicznej, mają zwykle postać znakową i

są poprzedzone nagłówkiem, zawierającym adresy nadawcy i odbiorcy, czas nadania listu i krótki komentarz o zawartości listu. Do listu można dołączać pliki w różnych formatach (tj. zawierające teksty w popularnych edytorach tekstów, pliki binarne czy też postscriptowe), tworząc tym samym cały pakiet. Zbiory binarne są kodowane (*uuencode*) do postaci ASCII i tak przesyłane po sieci. Odbiorca musi je oczywiście odkodować. W ten sposób można przesyłać pomiędzy kontynentami teksty z rysunkami, dźwięki, obrazy, sekwencje filmów itp. Należy podkreślić, że do przesyłania dużych plików binarnych służy specjalna usługa w sieci Internet - ***ftp***, która w tym zakresie jest bardziej efektywna.

Oprogramowanie obsługujące pocztę elektroniczną, oprócz podstawowych elementów jak redagowanie i wysyłanie listów lub wybór i czytanie otrzymywanych listów, zwykle umożliwia:

- szybkie (tj. bez konieczności ręcznego wpisywania nagłówka) wysyłanie odpowiedzi na list (*replay*) lub przesłanie listu do innego użytkownika (*forward*);
- automatyczne informowanie użytkownika o posiadaniu nieprzeczytanych listów (np. w trakcie dołączania się do systemu), niedostarczeniu listu lub też o przeczytaniu (lub nie) listu przez adresata;
- tworzenie i zarządzanie książkami adresowymi (tj. spisami naszych korespondentów) i stałymi listami rozsypki (tj. zbiorami adresów odbiorców korespondencji grupowej);
- zarządzanie listami, od podstawowych jak usuwanie listów, ich drukowanie czy też kopiowanie na dyskietki, aż do bardziej zaawansowanych: tworzenie szuflad tematycznych (ang. *folder*) i automatyczne sortowanie do nich listów wg. różnych warunków;
- wyszukiwanie adresów pocztowych użytkowników danego komputera-serwera.

Programy obsługi poczty elektronicznej są wykorzystywane także do uczestnictwa w listach dyskusyjnych. W Internecie dostępna jest organizacja list dyskusyjnych w postaci tzw. list adresowych. Właściciel listy dyskusyjnej (tj. osoba nią zarządzająca), a właściwie jego serwer pocztowy, rozsyła trafiające do niego listy do wszystkich zarejestrowanych członków. Listy dyskusyjne mogą dotyczyć różnorodnych tematów. Mogą mieć one charakter otwarty lub zamknięty. Członkiem otwartej listy dyskusyjnej można się stać w sposób automatyczny, wysyłając zwykłe na określony adres pocztowy (zwykle jest to ***listserv@<adres serwera>*** lub ***listproc@<adres serwera>***) wiadomość typu: ***sub <nazwa_listy> <imię-nazwisko>***. Do list zamkniętych przyjmują (zapisują) ich właściciele - trzeba pisać do nich listy (e-mailowe oczywiście). Dyskusje w listach dyskusyjnych mogą przyczynić się do lawinowego wzrostu liczby otrzymywanych komunikatów. Oczywiście w każdej chwili można wypisać się z listy dyskusyjnej poprzez przesłanie odpowiedniego komunikatu.

Na zasadzie list dyskusyjnych funkcjonuje w Internecie prasa elektroniczna, w tym polskojęzyczna. Przez Internet dystrybuowane są nieodpłatnie DONOSY (adres serwera: listproc@fuw.edu.pl), DYRDYMALKI (adres właściciela: zbigniew@engin.umich.edu), GAZETA (serwer: listproc@poniecki.berkley.edu), krakowski GROCH Z KAPUSTA (właściciel: bielewcz@uwpg02.uwinnipeg.ca), anglojęzyczne PIGULKI (właściciel: zielinski@acfcluster.nyu.edu) czy też poważny PRZEGLĄD AKADEMICKI (serwer: listserv@uci.agh.edu.pl), a nawet BILETYN GIEŁDOWY (serwer: listserv@appli.mimuw.edu.pl).

Istnieje w Internecie lista dyskusyjna Polskiego Towarzystwa Informatycznego, zrzeszająca członków i sympatyków naszego towarzystwa. Właścicielem listy, a właściwie jej administratorem, jest kol. Leszek Bogusławski z Instytutu Ochrony Środowiska w Warszawie (jerome@frodo.nask.org.pl). Adres pod którym można zapisać się na listę **PTI-L** (bo taką ma nazwę lista dyskusyjna PTI): ***listserv@plearn.bitnet***. Lista dyskusyjna PTI udostępnia także archiwalną zawartość. By ją otrzymać na powyższy adres należy wysłać wiadomość: ***get pti-l logYYMM***. Przy czym ***YY***

oznacza rok, a *MM* numer miesiąca, z którego chciałoby się otrzymać korespondencję.

Spis list dyskusyjnych sieci Internet można znaleźć np. przy pomocy ogólnie dostępnego (przy pomocy ***telnet***) Gophera serwera: ***hum.amu.edu.pl***.

Inna usługa sieci Internet - **IRC** (skrót od ang. *Internet Relay Chat*) służy do organizowania rozmów na żywo, tj. w czasie „prawie" rzeczywistym. Najważniejszą zaletą tej usługi jest organizacja wirtualnych grup dyskusyjnych (ang. *virtual groups*), tworzonych w sposób spontaniczny, na pewien czas, przy barku ograniczeń co do liczby osób, należących do takiej grupy. Powstająca grupa wybiera sobie kanał (ang. *channel*) nazywając go. W danej chwili w Internecie może funkcjonować bardzo dużo aktywnych kanałów. Ich liczbę ogranicza jedynie moc serwerów *IRC*. Każdy kanał ma swojego właściciela (jest nim osoba, która stworzyła kanał nazywając go po raz pierwszy), który ma prawo zapraszać (i wypraszać, w żargonie *IRC* - wykopywać, ang. *kick*) rozmówców.

Przy pomocy serwisu IRC można organizować także prywatne grupowe rozmowy, do których nie są dopuszczane osoby postronne, lub też rozmowy pomiędzy dokładnie dwoma uczestnikami (ang. *query session*).

Nowości sieciowe *NetNews* są zbiorem informacji dostarczanych przez użytkowników sieci Internet. Są one gromadzone na serwerach, przechowywane na nich przez kilka dni i udostępniane wszystkim zainteresowanym. Ze względu na olbrzymią ilość informacji, nowości są podzielone na kategorie (np. nowości: **news**, komputery: **comp**, nauka - **sci**, rozrywka - **rec** itp.) i grupy zainteresowań, w ramach każdej kategorii (np. bezpieczeństwo w systemach komputerowych: **comp.risk**). Podział jest przeprowadzany dalej. Możliwe są zatem nazwy grup takie jak: **comp.databases.oracle** czy też **alt.binaries.pictures.fractals**. Użytkownik sieci, posiadający odpowiednie oprogramowanie, może przeglądać nowości w dowolnej grupie. Może dołączać do niej własne wiadomości, które automatycznie stają się częścią strumienia informacji (nowości) rozpowszechnianych w sieci. Strumień ten jest olbrzymi i osiąga obecnie ok. 60MB tygodniowo [1].

Nowości przesyłane są za pomocą protokołów: UUCP (*Unix-Unix Copy Protocol*) i NNTP (*Network News Transport Protocol*). Do ich odczytu wymagane jest specjalne oprogramowanie - czytnik NNTP (np. program *tin*) i dostęp do serwera NNTP.

4. Przesyłanie plików - ftp

Przy pomocy poczty elektronicznej można przesyłać pliki nie tylko tekstowe. Jednak, jak już było wspomniane, bardziej efektywną metodą transferu dużych zbiorów (tekstowych i binarnych) jest specjalny protokół - **FTP** (ang. *File Transfer Protocol*).

Wykorzystanie ***ftp*** do transferu plików rozpoczyna się od uzyskania dostępu do komputera-serwera FTP. Dostęp ten jest możliwy jeśli znana jest nazwa konta i hasło dostępu. Wiele ośrodków akademickich (i nie tylko) udostępnia w Internecie swoje serwery FTP dla wszystkich chętnych. Dostęp do takich serwerów uzyskuje się przy pomocy konta anonimowego (nazwa konta: ***anonymous*** lub ***ftp***) bez hasła lub z własnym adresem poczty elektronicznej, podanym jako hasło (wyłącznie do celów statystycznych).

Usługa ***ftp*** umożliwia głównie przeglądanie zawartości katalogów serwera i przesłanie wybranych zbiorów do i od serwera.

Zbiór poleceń ***ftp*** jest niewielki i bardzo prosty w użyciu. Wykonanie niektórych z nich może wymagać posiadania określonych praw na serwerze.

5. Zdalny dostęp do systemów komputerowych - telnet

Zdalny dostęp jest jedną z najstarszych usług sieci Internet. Pozwala ona na przyłączenie się do konkretnego odległego komputera i zdalną, interakcyjną pracę na nim. Pod pojęciem zdalnej pracy rozumie się przede wszystkim uruchomienie programów na komputerze zdalnym i uzyskiwanie rezultatów ich pracy na komputerze lokalnym, który w tym przypadku staje się zwykłym, nieinteligentnym terminalem komputera zdalnego. Usługa ta

dostępna jest praktycznie dla wszystkich systemów operacyjnych komputerów-serwerów (począwszy od Unixa, VMS aż do mainfraimów czy też superkomputerów).

Przy pomocy usługi ***telnet*** można wykorzystywać oprogramowanie, które jest możliwe (z różnych względów) do uruchomienia na konkretnym komputerze. W ten sposób można dokonywać obliczeń na superkomputerze, wyszukiwać informacje w bazach danych (np. katalogach bibliotek lub też zbiorach archiwalnych) lub też korzystać z innych usług Internetu.

Praca na zdalnym komputerze możliwa jest oczywiście tylko w zakresie przyznanych uprawnień. Tylko nieliczne centra komputerowe dopuszczają bezpłatne wykorzystanie swoich zasobów przez nieznanych użytkowników (identyfikowanych jako goście: *guest* lub nieznani: *anonymous*). Prawa takich użytkowników są zwykle bardzo ograniczone. Zawsze jednak dostanie się informację na temat sposobu kontaktu z administratorem serwera, który to wskaże sposób uzyskania (i zwykle zapłaty) szerszych uprawnień.

Polecenia usługi ***telnet*** umożliwiają dołączenie się do zdalnego komputera (**open**), zawieszenie połączenia (**z**), odłączenie (**close**), zakończenie pracy (**quit**) i zmianę parametrów emulowanego terminala (**set, mode**).

By uzyskać dostęp do danego odległego komputera niezbędna jest znajomość jego adresu (i bardzo często sposobu dołączenia, tj. identyfikatora i hasła anonimowego użytkownika).

6. Iobalne serwisy informacyjne - Gopher, WWW

W Internecie powstało w sposób spontaniczny szereg rozproszonych baz danych (o różnej strukturze) świadczących usługi informacyjne. Serwisy te są dostępne nie poprzez ***telnet***, ale przy pomocy specjalnego oprogramowania. Oprogramowanie to w większości przypadków udostępniane jest bezpłatnie z zasobów sieci Internet. Zresztą, większość oprogramowania obsługującego serwery i klientów takich rozproszonych serwisów informacyjnych jak ***Gopher***, ***WWW*** czy też ***WAIS*** została wytworzona przez entuzjastów. Dopiero później powstawały ich wersje komercyjne. Serwisy informacyjne w Internecie działają z wykorzystaniem architektury klient-serwer i stanowią przykład globalnych, „silnie" rozproszonych baz danych.

System informacyjny ***Gopher*** (pol. *Świstak*) oferuje dostęp do różnorodnych dokumentów (głównie w postaci tekstowej, ale także i binarnej). Informacje dostępne przez ***Gophera*** są przygotowywane przez właścicieli serwerów systemu i ich skala, zakres i postać zależą od osób, które je przygotowują. Brak jest w tym względzie jakiś standardów, ale zakres informacyjny serwisów ***Gophera*** można podzielić na informacje merytoryczne (typu: prace naukowe, komunikaty, przepisy, spisy publikacji itp.), towarzyskie (listy pracowników, zawiadomienia o konferencjach, seminariach itp.), publiczne (turystyczne, książki telefoniczne, repertuary kin itp.) i rozrywkowe (gry i zabawy, różnorodne zdjęcia, piosenki, plebiscyty itp.).

Poprzez ***Gophera*** dostępne są w coraz szerszym zakresie archiwa serwerów ***ftp***, a także usługi typu ***telnet***. Skupia on zatem trzy funkcje: serwis informacyjny, transmisja plików i zdalny dostęp.

Struktura informacji zawartych na serwerach Gophera jest typowo hierarchiczna. Praca z systemem polega na dołączeniu się do konkretnego serwera i przeszukiwaniu jego zasobów przy pomocy szeregu kolejnych menu. W trakcie pracy system może automatycznie (po wybraniu przez użytkownika kolejnej opcji z menu) zmieniać serwer w sposób przezroczysty dla użytkownika. Taka „wędrówka" po systemie nawiązuje do poszukiwania plików w systemie katalogów komputerów i jest bardzo łatwa do intuicyjnego opanowania. Minimalny jest także zakres funkcji programu klienta Gophera.

W zależności od konkretnej konfiguracji dostęp do programu klienta Gophera uzyskuje się z poziomu systemu operacyjnego (polecenie: **Gopher**) bądź też dołączając się do serwera Gopher przy pomocy usługi ***telnet*** (podając jako nazwę użytkownika: *gopher* lub *guest*, zwykle bez hasła). Tak uzyskuje się dostęp do np. do polskich ogólnie dostępnych serwerów

gopher.torun.edu.pl (trudno się dostać ze względu na ograniczoną do 4 liczbę jednoczesnych użytkowników anonimowych) lub **hum.amu.edu.pl.** Jak przystało na hierarchiczną strukturę istnieje Gopher-matka wszystkich innych serwerów systemu: **gopher.micro.umn.edu**. Znajduje się on w uniwersytecie w Minnesocie, w którym powstały pierwsze programy i serwery Gophera.

Światowa pajęczyna informatyczna - *WWW* (ang. *World-Wide Web*) spełnia analogiczne funkcje jak Gopher. Jest obecnie jedną z najmodniejszych i najbardziej „zasobożernych", ogólnie dostępnych usług w sieci Internet. *WWW* jest rozproszoną pomiędzy tysiące serwerów hipertekstową bazą danych, która może być przeglądana zarówno w trybie tekstowym, jak i graficznym. Ten ostatni tryb, w połączeniu z szybkim, odpowiednio wyposażonym w urządzenia zewnętrzne komputerem i bardzo szybką transmisją danych poprzez sieć umożliwia korzystanie ze wszelkich efektów multimedialnych oferowanych przez serwery systemu *WWW*.
Struktura hipertekstowa bazy danych systemu *WWW* sprawia, że poruszanie się po zasobach informacyjnych systemu różni się znacznie od hierarchicznego dostępu do danych w Gopherze. Charakteryzuje się ona brakiem uporządkowania, ale też dużą dowolnością i elastycznością w dostępie do informacji. Nowoczesna multimedialna szata danych zawartych w bazach *WWW* zachęca wręcz do ich przeglądania, a intuicyjna obsługa (kliknięcie na wyróżnionym polu, wyrazie lub obrazku) nie nastręcza trudności nawet słabo przygotowanemu użytkownikowi. Serwery WWW powstają praktycznie w każdym liczącym się ośrodku naukowym - zawierają one bardzo różnorodne informacje od lokalnych i ogólnych po monotematyczne przeglądy literatury. W systemie WWW coraz częściej dostępne są różnego typu multimedialne aplikacje: przewodniki po miastach (np. po Londynie czy Krakowie), parkach rozrywki (np. po EuroDisneylandzie) i muzeach (np. Luwr), książki i encyklopedie, wideoklipy itp.

By mieć możliwość skorzystania z usług systemu *WWW* należy posiadać przeglądarkę (ang. *browser*), która jest oprogramowaniem typu klient dla serwerów *WWW*. Przeglądarki te można uzyskać z zasobów sieci Internet (jak programy typu *Public Domain*). Jedną z najbardziej popularnych i dostępną dla systemów MS Windows i X Windows przeglądarkę *Mosaic* można uzyskać z zasobów serwera ***ncsa.uiuc.edu*** (gdzie w podkatalogach katalogu ***pub/mosaic*** znajdują się różne wersje oprogramowania, a także pokaźna dawka dokumentów opisujących system *WWW*).

7. Wyszukiwanie informacji w sieci Internet - Archie, WAIS, Whois, Netfind

Poruszanie się po dużych i rozproszonych zasobach Internetu jest utrudnione. Pomocą w tym skomplikowanym komputerowym świecie służą użyteczne narzędzia do wyszukiwania potrzebnej informacji, niezależnie od miejsca w którym się ona znajduje. Do nich zaliczają się serwisy informacyjne ***Archie***, ***Veronica*** i ***WAIS*** (*Wide-Area Information Server*).

Archie pozwala odszukiwać pliki przechowywane na ogólnie dostępnych serwerach ***ftp***. System wykorzystuje do tego celu rozproszoną bazę danych o zasobach ***ftp*** indeksowanych przy pomocy słów kluczowych i/lub tytułów. Bazy te są regularnie aktualizowane i stanowią niezastąpione źródło informacji o zawartości serwerów. Oprogramowanie systemu ***Archie*** ma charakter klienta (w stosunku do serwerów Archie). Jest ono, podobnie jak i analogicznie działająca w odniesieniu do stron Gophera ***Veronica***, zwykle dołączane do oprogramowania ***Gophera*** (choć nie tylko). Oprócz wyszukiwania umiejscowienia (adres + katalog + nazwy) potrzebnych plików umożliwia przeprowadzenie szeregu dodatkowych operacji. Najciekawszą jest dostęp do bazy danych z obszerniejszymi opisami programów i jej przeszukiwanie: ***whatis***.

WAIS (*Wide-Area Information Server*) umożliwia korzystanie w jednolity sposób z całego szeregu baz danych (przeważnie

tekstowych) o różnorodnej tematyce. Oprogramowanie klienta systemu ***WAIS*** umożliwia przeszukiwanie dużych plików tekstowych pod kątem występowania w nich słów podanych przez użytkownika. Przeszukuje się przy tym wybrane (wskazane) bazy danych. Rezultatem poszukiwania jest lista dokumentów uporządkowana według znormalizowanych (do zakresu: 0-1000) liczb wystąpień wewnątrz nich podanych słów. W systemie WAIS nie wyróżnia się specjalnych słów kluczowych. Przeszukiwane są całe teksty, a nie specjalne indeksy jak to miało miejsce w systemie ***Archie***. Znalezione dokumenty mogą być w prosty sposób (poprzez wskazanie) przeglądane.

Przy pomocy ***WWW*** można uzyskać dostęp do całego szeregu baz danych, w których podjęto próbę skatagolizowania zasobów Internetu. Bardzo często dostępna jest w nich cenna możliwość przeszukiwania tych katalogów. Niektóre z nich można znaleźć pod wskazanymi w artykule B.Biernatowskiego „Porządek w Internecie" adresami (w nawiasie nazwy serwisów):

- http://www.einet.net/ (EINet Galaxy Directory Service);
- http://info.cern.ch/hypertext/DataSources/bySubject/ (The WWW Virtual Library);
- http://teal.nosc.mil/info.html (PLANET EARTH);
- http://www.gnn.com/wic/newrescat.toc.html (GNN-The Whole Internet Catalog).

Wspomniany artykuł był zamieszczony w Monitorze Sieciowym (nr 8/95 z dnia 10.04.1995) - piśmie (oczywiście elektronicznym, rozprowadzanym w sieci Internet) Wirtualnej Akademii (oczywiście elektronicznej, dostępnej poprzez Internet).
W Internecie istnieją także serwisy, które automatycznie przeszukują zasoby sieci w celu wyszukania potrzebnej informacji.

W Internecie dostępnych jest cały szereg usług, związanych z lokalizacją użytkowników sieci.

Whois udostępnia informacje z rozproszonej bazy danych o „parametrach" konkretnych użytkowników (w tym: adres e-mailowy, nazwa i opis instytucji, numer telefonu i faksu), komputerów/podsieci (nazwa instytucji - właściciela hosta/podsieci, dane administratorów itp.) i domenach (nazwa instytucji, która obsługuje domenę, parametry administratorów, adresy serwerów obsługujących domenę itp.). Specjalny program ***Whois*** pracujący w architekturze klient-serwer umożliwia odszukiwanie osób (***PErson***), nazw domen (***DOmain***), hostów (***HOst***), podsieci (***NEtwork***) i organizacji (***Organisation***) na podstawie podanego wzorca (adresu, nazwy, nazwiska itp.). Serwery z danymi o użytkownikach Internetu prowadzą organizacje zarządzające nadawaniem adresów (InterNIC w USA, RIPE w Europie, NASK w Polsce, operatorzy MANów w poszczególnych miejscowościach).

Tych samych serwerów co ***Whois*** używa inna usługa sieci Internet - ***Netfind***. Przy jej pomocy można ustalić na podstawie nazwiska podstawowe parametry osoby: nazwa instytucji, numer telefonu, nazwę konta i datę jego ostatniego użycia.

8. Podsumowanie

Większość usług w Internecie wymaga specjalnego oprogramowania. Jest ono dostępne w sieci bezpłatnie bądź też możliwe do zakupu w postaci specjalnych pakietów, które zwykle łączą w sobie dostęp do wielu usług jednocześnie. Ich opis i ocenę można znaleźć np. w [1]. Posiadając tylko możliwość zdalnego dołączania się do komputerów (***telnet***) można korzystać z takich usług jak ***Gopher***, ***WWW***, ***Archie*** czy też ***WAIS*** dzięki temu, że są one udostępniane za darmo użytkownikom anonimowym na niektórych serwerach, np.: **gopher.torun.edu.pl** (system Gopher); **www.icm.edu.pl** (WWW); **quake.think.com** (WAIS, login: wais); **whois.internic.net** (Whois); **archie.icm.edu.pl** (Archie).

W ciągu dnia w Internecie pojawia się olbrzymia ilość informacji. Dostęp do niej jest bezpłatny (jeśli nie liczyć dużych kosztów transmisji informacji w sieci i traconego czasu) i niemalże natychmiastowy. Internet otwiera więc nieograniczone wprost możliwości

w wymianie informacji. Niektórzy użytkownicy spędzają przy monitorach cały swój wolny (i nie tylko) czas, przeszukując knieje Internetu.

Problem: *jak z tego oceanu informacji wyszukać potrzebne?* pomagają rozwiązać różne serwisy informacyjne. Wystarczy o nich wiedzieć i nauczyć się korzystać.

Oczywiście wszystko ma swoje negatywne strony. By się o nich przekonać wystarczy np. należeć do paru (nawet średnio aktywnych) list dyskusyjnych. Pomimo ciekawych dyskusji zwykle nie ma się czasu na czytanie wszystkich wypowiedzi, a traci się go na ustawiczne uporządkowywanie (czytaj: usuwanie) korespondencji.

Internet (a właściwie swoboda wymiany informacji i pewien brak uwarunkowań prawnych) jest także nadużywany. Można poczytać o nadużywaniu sieci w celach komercyjnych, propagandowych (np. wykorzystywanie przez neonazistów luk prawnych w rozprowadzaniu wiadomości przy pomocy sieci komputerowych), pornograficznych (np. istnieją pornograficzne grupy dyskusyjne, kanały IRC i serwery WWW) czy wręcz przestępczych (bardziej lub mniej dowcipni i bezkarni włamywacze, złodzieje oprogramowania i usług obliczeniowych). Można tego doświadczyć niekiedy na własnej skórze.

Literatura

1. Janyszek J.: Usługi sieci Internet. Materiały konferencji Miejskie Sieci Komputerowe w Nauce i Gospodarce POLMAN'95. Poznań, 10-12.04.1995, str. 162 - 171.
2. Motoszko J., Neska I.: Sieć Internet w Polsce. Materiały seminarium Naukowa i Akademicka Sieć Komputerowa. Miedzeszyn, wrzesień 1994, str. 35 - 40.
3. Kozłowski M.: NASK w sieciach komputerowych Europy i świata. Materiały seminarium Naukowa i Akademicka Sieć Komputerowa. Miedzeszyn, wrzesień 1994, str. 41 - 48.
4. Zaperty B.: Serwisy baz danych i katalogi biblioteczne dostępne poprzez NASK. Materiały seminarium Naukowa i Akademicka Sieć Komputerowa. Miedzeszyn, 1994, str. 55 - 65.
5. Neska I.: Serwisy informacyjne dostępne w sieci Internet. Materiały seminarium Naukowa i Akademicka Sieć Komputerowa. Miedzeszyn, wrzesień 1994, str. 71 - 85.
6. Silicki K.: Bezpieczeństwo w sieciach komputerowych. Materiały seminarium Naukowa i Akademicka Sieć Komputerowa. Miedzeszyn, wrzesień 1994, str. 96 - 113.
7. Neska I., Motoszko J.: Przewodnik po sieci Internet. NASK, Warszawa, luty 1994, 56 str.
8. PC Magazine po polsku. styczeń 1995, nr 1/23.
9. LaQuey T., Ryer J.C.: Internet i okolice. Przewodnik po światowych sieciach komputerowych. Biznet, Warszawa, 1994, 237 str.
10. Levine J.R., Baroudi C.: Internet dla opornych. README, Warszawa, 1994

Jarosław Kępkowicz
Marian Suskiewicz
Telekomunikacja Polska S.A

Kierunki rozwoju transmisji danych w sieciach TPSA

1. Wprowadzenie

Wraz z rozwojem i rozszerzaniem zastosowań komputerów powstała nowa dziedzina telekomunikacji - transmisja danych. Budowa sieci transmisji danych w krajach zaawansowanych w dziedzinie zastosowań techniki komputerowej jest problemem nie mniej ważnym niż rozwój innych usług w dziedzinie łączności (szeroko rozumianego systemu przesyłania informacji). Problem ten jest rozważany zarówno przez administracje łączności (ministerstwa) w tych krajach, jak i przez operatorów odpowiedzialnych za rozwój systemów teleinformatycznych. Również w Polsce stajemy coraz częściej przed koniecznością wyboru kierunków rozwoju systemów transmisji danych, doboru środków technicznych do ich realizacji, jak również przed problemem efektywnego wykorzystania już istniejących zasobów telekomunikacyjnych.
TPSA jako główny operator sieci telekomunikacyjnej w Polsce, opiera rozwój sieci transmisji danych na bazie istniejących systemów telekomunikacyjnych. Do celów budowy sieci transmisji danych wykorzystuje z powodzeniem posiadaną sieć telefoniczną (łącza analogowe i cyfrowe oraz centrale telefoniczne różnego typu). Systemem podkładowym dla przesyłania sygnałów transmisji danych jest stale unowocześniana sieć telefoniczna (kablowa, radiowa) i systemy satelitarne. Na początku bieżącej dekady nastąpił przełom w rozwoju telekomunikacji w Polsce. Przyczyniły się do tego m.in. następujące czynniki:

- jasno sprecyzowane cele i plany rozwoju usług i sieci telekomunikacyjnych w TPSA;
- możliwość uzyskania kredytów zagranicznych na rozwój telekomunikacji;
- zniesienie wielu ograniczeń na eksport nowoczesnych technologii;
- zakup i wdrożenie do eksploatacji cyfrowych central telefonicznych i światłowodowych systemów przesyłania informacji.

Powyższe czynniki umożliwiły reorganizację systemów telekomunikacyjnych będących w posiadaniu TPSA i podjęcie śmiałej rozbudowy sieci transmisji danych stwarzając jednocześnie bazę transportową dla systemów przesyłania informacji również innych operatorów.

2. Cyfryzacja systemów telekomunikacyjnych

Do końca lat 80-tych dominującym systemem w transmisji danych były systemy analogowe. Wprowadzenie cyfrowych central telefonicznych spowodowało, że sieć oparta wyłącznie o linie analogowe stała się niewydolna i trudna do wykorzystania przy budowie nowoczesnych systemów transmisji danych. W związku z powyższym w TPSA podjęto decyzje o budowie linii cyfrowych optotelekomunikacyjnych (światłowodowych) i radiowych.
Podstawowymi systemami cyfrowymi zbudowanymi w sieci magistralnej TPSA, o przepływności 140 Mbps, są:

- podmorski kabel optotelekomunikacyjny z Koszalina do Danii;
- cyfrowa linia radiowa pomiędzy Koszalinem i Warszawą;
- cyfrowy system łączności satelitarnej na region Oceanu Atlantyckiego;

- linia optotelekomunikacyjna od granicy zachodniej do Olsztyna ;
- linia optotelekomunikacyjna łącząca Wybrzeże z Cieszynem;
- szesnaście linii radiowych, prod. NEC.

Ponadto TPSA zainstalowała i rozpoczęła eksploatację wielu elektronicznych central cyfrowych w głównych miastach Polski [3].
W najbliższych latach przewiduje się dalszą rozbudowę cyfrowej sieci dalekosiężnej, co zapewni doprowadzenie łączy cyfrowych do wszystkich central międzymiastowych w Polsce oraz przejście z cyfrowych systemów plezjochronicznych (PDH) na systemy synchronicznej transmisji (SDH). Porównanie własności sieci PDH i SDH wskazuje na celowość i konieczność budowy sieci SDH z uwagi na:

- większą elastyczność i mniejszy koszt budowy sieci SDH;
- większą niezawodność;
- łatwość wdrażania systemów zarządzania i utrzymania.

Równolegle z budową sieci dalekosiężnych prowadzone są prace przy konstrukcji sieci cyfrowych na niższych poziomach. Utworzona podstawowa sieć cyfrowa będzie bazą do tworzenia sieci wtórnych powszechnego użytku (telefonia) jak i sieci teleinformatycznych - organizowanych zarówno przez TPSA, jak też przez innych operatorów.

3. Analiza stanu zapotrzebowań na transmisję danych

Z analizy obecnej sytuacji na krajowym rynku transmisji danych wynika potrzeba zbudowania przez TPSA uniwersalnej sieci, umożliwiającej dostarczenie różnym klientom szerokiego zakresu usług. Jednoznacznie trudno jest określić potencjalną ilość użytkowników poszczególnych usług, będą to ilości zróżnicowane w zależności od typu usługi, lokalizacji systemów komutacyjnych, istniejącej sieci łączy dostępowych i aplikacji posiadanych przez użytkowników na systemach komputerowych.
W Polsce wielu użytkowników sieci lokalnych posiada obecnie różnorodne routery i sieci LAN, stąd też są oni zainteresowani dzierżawą łączy typu punkt-punkt. Dla operatora oznacza to konieczność zapewnienia możliwości dołączenia użytkowników do dzierżawionych kanałów cyfrowych o określonej przepustowości lub dołączania użytkowników do sieci typu Frame Relay (retransmisja ramek). Również dotychczasowi użytkownicy istniejących sieci X.25, ograniczeni stosunkowo niską prędkością transmisji, mogą wkrótce chcieć uzyskiwać dostęp do szybkich sieci Frame Relay. Z punktu widzenia operatora publicznego, jakim jest TPSA, w pierwszej kolejności należy dostarczyć do siedziby klienta łącza o gwarantowanych parametrach transmisyjnych. Najwłaściwszym rozwiązaniem byłoby dostarczenie poprzez jedno łącze różnych usług (np. ISDN), dopasowanych do zmieniających się potrzeb użytkowników. Kolejnym problemem jest łączenie wieloprotokołowych sieci lokalnych poprzez wielowejściowe routery lub poprzez sieci typu Frame Relay. Problemem jest również doprowadzenie interfejsów bezpośrednio do siedziby użytkownika, co wiąże się z koniecznością doprowadzenia urządzeń transmisyjnych na niewielkie odległości od użytkownika, zwykle poprzez kabel miedziany (tzw. problem "ostatniej mili"). Wymaga to odpowiedniego sposobu kontroli i zarządzania siecią transmisji danych. Siłą TPSA jest to, iż jest właścicielem sieci kablowej i posiada odpowiednie urządzenia teletransmisyjne, może dostarczyć kompletny serwis, biorąc na siebie odpowiedzialność za całe łącze "od końca do końca" tj. łącznie z kablem. Problemem jednak pozostaje efektywne wykorzystanie łącza.

Drugą grupę zwiększonych zapotrzebowań na transmisję danych generują nowoczesne usługi teleinformatyczne typu: poczta elektroniczna, zdalne i rozproszone przetwarzanie, dostęp do baz informacyjnych, w szczególności poprzez ostatnio burzliwie rozwijającą się sieć Internet.

4. Trendy rozwoju sieci transmisji danych

W ostatnich latach obserwujemy podstawowe zmiany w transmisji danych na duże odległości. W chwili obecnej nowoczesne systemy transmisyjne muszą być przystosowane

do przekazywania różnorodnych informacji (danych, obrazu, głosu, ...). Szczególnego znaczenia nabierają połączenia pomiędzy sieciami lokalnymi, o różnorodnych protokołach komunikacyjnych, ale wymagających szybkich łączy, o małych opóźnieniach. Po zastosowaniu światłowodów i wysoce niezawodnego sprzętu transmisyjnego osiąga się bardzo dobre połączenia na obszarach miejskich. Obserwuje się natomiast zmniejszające się zainteresowanie usługami pakietowych sieci X.

Protokół X jest doskonałym mechanizmem do uzyskania bezbłędnych transmisji, ale jest obarczony dużymi narzutami systemowymi, które w rezultacie powodują obniżenie szybkości transmisji całego systemu. Dodatkowo obsługa kolejek i kontrola powodują powstawanie znaczących opóźnień. Wymienione powody wymusiły powstanie nowych systemów transmisyjnych o wysokiej wydajności, szybkości transmisji i małych opóźnieniach. Frame Relay (retransmisja ramek) i ATM (asynchroniczna transmisja komórek) to dwie technologie, które stworzono do realizacji tych nowych potrzeb. Charakteryzują się one wysokim wykorzystaniem pasma, przy prędkościach od 2 Mbps do 622 Mbps, bez wysokich narzutów związanych z korekcją błędów. Idealne rozwiązanie powinno połączyć zalety X.25 z wydajnością Frame Relay czy ATM.

Frame Relay jest technologią przewidzianą do stosowania na łączach cyfrowych, jest to protokół komutacji pakietów, opracowany dla efektywnej i szybkiej transmisji pakietów o zmiennej długości, z minimalnym opóźnieniem. Sam protokół jest podobny do X.25, tyle że nie posiada zbędnych narzutów kontrolnych, umożliwia transmisję z dużymi prędkościami, potencjalnie aż do 34 Mbps. W ostatnich latach obserwuje się wręcz lawinowy przyrost dochodów ze sprzedaży usług i urządzeń pracujących w standardzie Frame Relay.

Od szeregu lat nie obserwowano tak dużego zainteresowania żadną inną technologią w zakresie transmisji danych, jak usługami Frame Relay.

Protokół ATM został w zasadzie opracowany jako protokół warstwy fizycznej dla użycia w szerokopasmowym ISDN, przeznaczony do niezwykle szybkiej, jednoczesnej transmisji głosu, wideo, obrazów i danych. ATM opiera się na transmisji 53 bajtowych komórek (48 danych i 5 kontrolnych) poprzez bardzo szybkie łącza. Standardowe prędkości wg. IEEE 802.6 wynoszą 155 Mbps i 622 Mbps. Wszystkie źródła przewidują, że ATM wkrótce będzie dominującą na rynku technologią, zarówno na rynku WAN jak i LAN. ATM jest bardzo ciekawą technologią, która jak się wydaje, jest skalowalną technologią (nadaje się dla sieci lokalnych, rozległych, dla małych i dużych prędkości), dająca możliwość przesyłania zarówno danych, głosu jak i obrazu. ATM jest dobrym rozwiązaniem dla budowy sieci miejskich, oczywiście przy zachowaniu odpowiednich mechanizmów ochrony (security). Dla praktycznego zastosowania technologii ATM potrzebna jest odpowiednia infrastruktura teletransmisyjna w hierarchii SDH (którą właśnie buduje TPSA), o bardzo dużych prędkościach międzywęzłowych.

Prognozy rozwoju rynku komputerowego wskazują, że wielu użytkowników sieci planuje przeprowadzenie gruntownych zmian w swoich sieciach lokalnych i rozległych [4]. W zakresie sieci WAN utrzyma się trend polegający na przechodzeniu od dedykowanych, prywatnych sieci ku bardziej elastycznym usługom typu Frame Relay, jak również planuje się niebawem wprowadzać produkty i usługi ATM. Prawie połowa ankietowanych zamierza się pozbyć posiadanych protokołów komunikacyjnych na rzecz protokołu TCP/IP. Natomiast łącza internetowe planuje się używać jak strategicznego nośnika w wymianie informacji ekonomicznej. Siła, jaką niosą ze sobą otwarte protokoły sieciowe, nowe technologie LAN i WAN o dużej szybkości przesyłania oraz publiczne usługi, takie jak Internet - pozwoli przedsiębiorstwom znacznie rozszerzyć zasięg swoich sieci.

5. Kierunki zmian w zakresie transmisji danych w TPSA

TPSA na bazowej sieci telekomunikacyjnej (w znacznej części już cyfrowej) nadbudowuje warstwy usługowe w zakresie transmisji głosu

(usługi serii 800- Infolinia 800, audioteks, itd.) jak również w zakresie transmisji danych (poczta elektroniczna, bazy danych, Internet, itd.).
W przyszłości planuje się budować inteligentne sieci telefoniczne i wówczas będzie zacierać się granica między usługami telefonicznymi i transmisji danych, powstaną szeroko rozumiane usługi telekomunikacyjne.
Od samego początku usługi teleinformatyczne powstawały w Polsce w warunkach pełnej konkurencji. W obecnej chwili poza TPSA na rynku tym działa kilku innych operatorów oferujących zbliżoną usługę. Tym co wyróżnia TPSA jest integracja sieci transmisji danych o nazwie POLPAK z innymi sieciami telekomunikacyjnymi takimi jak: łączność satelitarna VSAT, system przywoławczy Polpager, sieć teleksowa a wkrótce połączenie z siecią ISDN oraz z siecią radiokomunikacyjną [2].
Rozbudowa sieci POLPAK odbywa się w kilku kierunkach m.in. poprzez:

a/ instalację kolejnych węzłów sieci i koncentratorów terminali. Zakup nowych dużych węzłów i 18 koncentratorów zwiększy pojemność sieci dwu i pół krotnie, a także rozszerzy zasięg sieci;
b/ zamianę łączy analogowych na łącza cyfrowe w połączeniach międzywęzłowych i dostępowych, tam gdzie będzie to możliwe. Cyfryzacja łączy międzywęzłowych i zapewnienie prędkości min. 64 Kbps oraz wprowadzenie protokołu Frame Relay i prędkości do 2 Mbps jako międzywęzłowej dla węzłów zlokalizowanych w Warszawie, Katowicach, Wrocławiu, Poznaniu, Gdańsku, Krakowie;
c/ wymianę oprogramowania węzłów komutacji pakietów na najnowowszą generację w pełni realizującą zalecenia serii X.25 i umożliwiającego również transmisję z protokołem Frame Relay;
d/ wymianę starych modemów na nowoczesne modemy o większych prędkościach transmisji;

Wykorzystując podstawową zaletę transmisji pakietowej X.25 (dopuszcza pracę na łączach o niższej jakości), można oczekiwać w niedalekiej przyszłości, że usługa ta będzie również w większym zakresie dostępna na poziomie mniejszych miast (np. gmin) przy czym jako sieć dostępowa może być wykorzystany system VSAT. Przełomu w rozwoju telekomunikacji dokonało wprowadzenie cyfrowych sieci z tzw. integracją usług. Przesyłanie głosu, obrazu i danych komputerowych - równocześnie i tą samą linią - od niedawna jest możliwe także w sieci TPSA. Usługę tę wprowadzono na razie w ograniczonym zakresie, w wydzielonej sieci KOMERTEL przeznaczonej dla klientów biznesowych. Obecnie komertelowska sieć ISDN jest dostępna dla abonentów w Warszawie, Gdańsku i Kielcach. Ten dostęp ma być rozszerzony dla innych krajowych abonentów. Wkrótce dostęp do sieci ISDN mają uzyskać abonenci w Gdyni, Poznaniu, Szczecinie, Krakowie. Na powszechny dostęp do sieci ISDN w skali ogólnokrajowej trzeba jeszcze trochę poczekać ze względu na złożoność przedsięwzięcia pod względem technicznym (wymagana pełna cyfryzacja sieci i implementacja systemu sygnalizacyjnego nr.7), organizacyjnym i ekonomicznym, ponieważ terminale ISDN-owskie są stosunkowo drogie.
W 1995r. rozpoczęta zostanie budowa przez TPSA 12 miejskich sieci w głównych miastach Polski oraz sieci szkieletowej.
Sieć szkieletowa POLPAK -T będzie spełniać kilka podstawowych zadań, m.in.:

- łączyć budowane sieci metropolitalne (MAN) w dwunastu miastach Polski. Protokołem międzywęzłowym będzie protokół Frame Relay z możliwością migracji w kierunku ATM. Początkowe szybkości międzywęzłowe będą wynosiły n x 2 Mbps z możliwością migracji w kierunku 34 Mbps, w zależności od potrzeb;
- będzie stanowić płaszczyznę transmisyjną dla obecnej sieci transmsji danych POLPAK, traktowanej później jako sieć dostępowa;
- będzie stanowić podstawę do wydzielania sieci wirtualnych na potrzeby dużych klientów, takich jak banki, duże przedsiębiorstwa handlowe, urzędy i instytucje państwowe mające szczególne wymagania dotyczące niezawodności transmsji, jej jakości i ochrony danych.

Ogólna koncepcja budowy sieci MAN polega na zainstalowaniu węzłów sieci (urządzeń

komutacyjno - transmisyjnych) w obiektach central miejskich i połączeniu ich cyfrowymi łączami 2 Mbps. W obiektach central międzymiastowych będą zainstalowane urządzenia do połączeń międzysieciowych:

- z innymi sieciami MAN;
- z innymi sieciami transmisji danych np. POLPAK, ISDN;
- z sieciami innych operatorów;

oraz siecią szkieletową.

Sieci MAN będą miały strukturę dwupoziomową:

- poziom komunikacyjny realizowany przez węzły połączone kanałami cyfrowymi o przepływności 2 Mbps utworzonymi na bazie międzycentralowych łączy światłowodowych;
- poziom dostępowy realizowany poprzez przyłączenie systemów komputerowych użytkowników do najbliższych węzłów sieci MAN.

Topologia sieci MAN będzie dostosowana do istniejącej infrastruktury telekomunikacyjnej TPSA w poszczególnych miastach.

Urządzenia sieci MAN zapewnią wymianę danych pomiędzy sieciami LAN pracującymi zgodnie z protokołami: TCP/IP, Novell IPX, DECnet, HP Advancenet i innymi, a także współpracować będą z sieciami WAN realizującymi protokoły X.25, Frame Relay, PPP, SNA a w przyszłości ATM [1].

Niezależnie od budowanych sieci metropolitalnych, sieć POLPAK będzie w dalszym ciągu eksploatowana, stanowiąc naturalne uzupełnienie wszędzie tam, gdzie jeszcze nie istnieje struktura łączy cyfrowych (światłowodowych) wysokiej jakości.

6. Usługi teleinformatyczne

O atrakcyjności działania poszczególnych sieci poza ich szybkością transmisji decydują usługi teleinformatyczne, jakie są dostępne dla potencjalnych abonentów. Oczywiście w chwili obecnej trudno jest jeszcze mówić o powszechnych usługach multimedialnych, choć na pewno jest to niedaleka przyszłość, ale usługi takie jak poczta elektroniczna będą wkrótce dostępne dla klientów TPSA. Aktualnie jest na ukończeniu postępowanie przetargowe, prowadzące do instalacji poczty elektronicznej działającej zgodnie ze standardem X.400 wraz z książką teleadresową wg. X.500 oraz modułem EDI. Szerszy opis tych usług znajduje się w innym referacie.

Wychodząc naprzeciw potrzebom naszych klientów w zakresie transmisji danych, sieci będące w posiadaniu TPSA poza pocztą elektroniczną będą świadczyć usługi typu:

- dostęp do krajowych i zagranicznych baz informacyjnych;
- dostęp do usług sieci INTERNET;
- szybką obsługę połączeń i przydzielanie pasma przenoszenia na żądanie;
- ochronę przesyłanych i przechowywanych danych.

Ponadto rozwój sieci ISDN umożliwi realizację usług takich jak: widetelefonia, wideokonferencje, dostęp do baz grafiki i danych, jak również pozwoli na świadczenie usług multimedialnych.

TPSA stwarza możliwości dołączania prywatnych baz danych do swoich sieci. Jeśli właściciel bazy danych chce pobierać opłaty za korzystanie z tej bazy, to musi stworzyć własny system taryfikacji, np. przydzielać każdemu użytkownikowi uprawnienia (hasło) do korzystania z bazy. Ponadto powinien ubiegać się o status service providera TPSA.

Telekomunikacja Polska S.A. planując rozwój usług teleinformatycznych, współpracuje i korzysta z doświadczeń z uczelni, instytutów, ośrodków naukowo - badawczych oraz innych operatorów (np.NASK).

7. Literatura

1. Karpeta M., Suskiewicz M.: Ogólnodostępne sieci metropolitalne TPSA, Materiały z konferencji POLMAN`95, OWN Poznań 1995r.
2. Kępkowicz J.: Tygodnik Gospodarczy, Informator Targów, Infosystem-Multimedia-Poligrafia, Poznań 1995r.
3. Lisiecki B.: SDH moda, czy konieczność, Świat Telekomunikacji, nr.2., Warszawa, październik 1994r.
4. Technology Planning Survey - Network World, 4 NetWorld, luty 1995r.

Sławomir Michalski
Marian Suskiewicz
Telekomunikacja Polska S.A

Koncepcja realizacji poczty elektronicznej X.400 W TPS.A.

1. Wstęp

Poczta elektroniczna (ang. e-mail) jest popularną nazwą usług zaliczanych do usług dodanych. Usługa ta oferuje użytkownikom szybką wymianę informacji zwanych wiadomościami z wykorzystaniem komputerów i mediów teletransmisyjnych.
Poczta elektroniczna oferuje dużo większe możliwości niż tradycyjne formy międzyludzkiej komunikacji:

- jest usługą asynchroniczną , tzn. nie jest wymagane, aby odbiorca wiadomości był aktywny (online) wtedy, gdy wiadomość przychodzi;
- jest usługą wieloadresową, pozwala zaadresować wiadomość do wielu odbiorców i przekazać tę wiadomość jednocześnie do wskazanych adresatów;
- jest usługą otwartą, pozwala na wymianę informacji z różnymi innymi systemami przesyłania np. teleksem, telefaksem, itp.

Telekomunikacja Polska S.A. - największy publiczny operator telekomunikacyjny w Polsce - postanowiła rozszerzyć swoje usługi właśnie o elektroniczną wymianę informacji opartą na standardzie X.400.
Po przeprowadzeniu działań marketingowych stwierdzono coraz większe zainteresowanie taką usługą w Polsce i postanowiono zakupić odpowiedni system komputerowy realizujący tę usługę.
Publiczny System Wymiany Wiadomości MHS (*Message Handling System*) otrzymał nazwę POLKOM/400, historycznie nawiązując do eksploatowanego wcześniej w telekomunikacji Systemu Retransmisji Faksów, Teleksów i Telegramów - POLKOM. W stosunku do swego poprzednika, obecny system posiada całkowicie nową jakość realizacji usług publicznych i poza nazwą niewiele ma z dawnym POLKOM'em wspólnego.
W wyniku przeprowadzonego przetargu na Publiczny System Wymiany Wiadomości MHS, oparty na standardzie X.400, TP S.A. wybrała jako integratora i dostawcę, francuską firmę SYSECA. System POLKOM/400 zaproponowany przez tę firmę bazuje na produktach znanego na świecie dostawcy systemów wymiany informacji - firmy ISOCOR. System ten stanowi kompletne rozwiązanie zarówno dla operatora publicznego jak i dla użytkownika końcowego.

2. Ogólne informacje o systemie POLKOM/400

System POLKOM/400 oparty jest generalnie o standard X.400 (84/88/92), który definiuje Publiczny System Wymiany Wiadomości MHS i protokoły wymiany przesyłek:

- Protokół P1 między węzłami pocztowymi MTA (*Message Transfer Agent*)
- Protokół P2 między modułami obsługi użytkowników UA (*User Agent*)
- Protokół P3 między użytkownikami, a węzłami pocztowymi MTA
- Protokół P7 między użytkownikami, a pamięcią wiadomości MS (*Message Store*), w której przechowywane są skierowane do nich przesyłki

Określa on także sposób dostępu najbardziej rozpowszechnionej usługi realizowanej w sieciach komputerowych - poczty elektronicznej, do innych systemów przesyłania wiadomości. Ponadto opisuje elementy koperty elektronicznej, na której umieszczony jest znormalizowany adres nadawcy i odbiorcy oraz takie cechy jak np. pilność, stopień poufności, termin ważności, zawiadomienie o dostarczeniu bądź niedostarczeniu wiadomości itp.
W kolejnej wersji standard X.400 został wzbogacony o elementy bezpieczeństwa zapewniające między innymi: uwiarygodnienie nadawcy i odbiorcy oraz bezbłędne przesyłanie. Ponadto zdefiniowano w nim sposób dołączenia do systemu X.400 takich usług, jak usługa teleksowa, faxowa czy fizyczne doręczenie, polegające na wydrukowaniu i dostarczeniu przesyłki adresatowi, do którego dotarcie inną drogą nie jest możliwe.

3. Architektura systemu POLKOM/400

System POLKOM/400 zostanie dostarczony w przedstawionej na rys. 1 następującej konfiguracji sprzętowej:

- Serwer główny:
  - Stratus R5 Risc Fault Tolerant Computer - z dwoma procesorami
  - 2 x 64 MB RAM
  - 4 x 1.4 GB HDD (disk duplexing)
  - 2 Gb DAT (backup cartridge)
  - 4 x K118-D12 (64 Kb/s)
  - 2 x K124 (2 Mb/s)
  - 32 porty asynchroniczne (19.2 Kb/s)
- Serwer komunikacyjny do innych systemów wymiany informacji (PC UNIX/SCO)
- Stanowisko zarządzania systemem (PC/Windows)
- Podsystemy współpracy z abonentami sieci teleksowej i faksowej (DISTEL)
- Moduły obsługi użytkowników (RUA/P7)
- Interfejsy dostępowe (X.25, ASYNC, PAD, ISDN)

4. Podstawowe moduły systemu POLKOM/400

System POLKOM/400 oparty na oprogramowaniu firmy ISOCOR ma budowę modularną. W jego skład wchodzą następujące moduły podstawowe:

- MTA ISOPLEX ADMD
- MS ISOPLEX MS
- X.500 ISOPLEX DS
- ISOGATE (cc:Mail, Ms:Mail, Novell-MHS, Lotus Notes, SMTP)
- ISOTRADE (moduł EDI - *Electronic Data Interchange*)
- ISOMAN (moduł administracji i zarządzania)
- AU (ISOTELEX, ISOFAX, ISOROUTE)
- UA (LUA, RUA, Sec RUA)
- API (MAPI, SSAPI, XOPEN, XDS, FPI)
- ACCOUNTING, BILLING, STATISTICS, ALARMS

Struktura funkcjonalna systemu POLKOM/400 została przedstawiona na rys.2.

5. Opis modułu MTA Isoplex 800

Moduł przekazywania wiadomości MTA odpowiada za transport komunikatów pomiędzy węzłami systemu X.400. Jest on odpowiednikiem urzędu pocztowego sortującego korespondencję. Korespondencja wychodząca jest umieszczana w odpowiedniej skrytce pocztowej jeśli odbiorca jest obsługiwany przez dany moduł, w przeciwnym wypadku, jeżeli adresat nie jest lokalny, wiadomość jest transportowana do innych MTA. Głównym środkiem transportu jest sieć

transportowa, a punktem docelowym - zdalny MTA, właściwy dla adresata.
W systemie POLKOM/400 wiadomości mogą być przesyłane po sieci za pomocą następujących protokołów komunikacyjnych:

- RFC 1006 on TCP/IP
- ISO CLNS on 802.3
- ISO CONS on X.25
- Asynchronous with calling mode on PAD X.3/X.28/X.29
- APS on asynchrone connections
- Frame relay

Moduł MTA Isoplex 800 jest całkowicie obsługiwany przez administratora systemu.

6. Opis modułu obsługi użytkownika - UA

Moduł obsługi użytkownika działa jak skrzynka pocztowa rozumiana w/g zasad amerykańskich, tzn. podobnie jak w Europie poczta przychodząca jest przechowywana do momentu jej odebrania przez adresata i odmiennie jak w Europie, w tej samej skrzynce można umieszczać korespondencję do wysłania.
Użytkownik systemu POLKOM/400 postrzega moduł UA poprzez jego oprogramowanie obsługujące dialog. Moduł użytkownika UA wykonuje wszystkie funkcje związane z przygotowaniem, wysłaniem i odbiorem wiadomości. Obejmuje to takie funkcje jak przypomnienie wypełnienia pól adresu, tematu i treści wiadomości oraz np. funkcje redakcji tekstu. Do zadań UA należy również ułatwienie użytkownikowi manipulacji wiadomościami odebranymi z MTA.
W systemie POLKOM/400, poza dowolnymi UA komunikującymi się z MTA za pomocą protokołu P7, posiadamy dwa dodatkowe UA firmy ISOCOR:

- graficzny ISOPRO (GUI for Windows 3.1 and NT)
- znakowy ISOMAIL 800 (for DOS)

7. Opis modułu pamięć wiadomości -MS

Pamięć wiadomości MS jest dodatkową uniwersalną funkcją MHS, która działa jako pośrednik pomiędzy UA i MTA. MS może uzupełniać zaimplementowany na komputerze osobistym moduł użytkownika UA, zapewniając bezpieczny, stale dostępny mechanizm pamięciowy, działający na rzecz UA. UA jest informowany o napłynięciu wiadomości do MS i może ją stamtąd pobrać. Wysyłanie wiadomości przez UA współpracujące z MS polega na przekazaniu wiadomości do MS, która bezzwłocznie przekazuje wiadomość do MTA. Na żądanie UA możliwe jest wysyłanie wiadomości przechowywanej w MS.

- W systemie POLKOM/400 zaimplementowano następujące sposoby dostępu do MS:
- P7 on X.25
- P7 on X.28 and X..29 (PAD)
- P7 on TCP/IP
- P7 on APS asynchronous connections

W systemie POLKOM/400 usługę X.400 MS realizuje moduł ISOPLEX MS.

8. Opis modułu dostępowego ISOGATE

W systemie POLKOM/400 zaimplementowano współpracę z wieloma popularnymi systemami pocztowymi różnych dostawców, takimi jak:

- MS-Mail
- cc:MAil
- Lotus Notes
- Novell MHS Server
- Internet based on SMTP product
- QuickMail
- DEC ALL-IN-1
- IBM PROFS/Office Vision

- HP Desk Manager
- Banyan Mail
- Wang Office

Komunikacja ze światem internetowym zapewniona jest za pomocą protokołu SMTP (*Simple Mail Transfer Protocol*). Protokół ten określa format wiadomości wysyłanych przez proces klienta w komputerze źródłowym do procesu serwera w komputerze docelowym. Komunikacja pomiędzy nadawcą i odbiorcą odbywa się w czytelnej formie, za pomocą rozkazów i odpowiedzi reprezentowanych tekstowo. Protokół SMTP jest więc protokołem typu klient-serwer. Pełny opis SMTP można znaleźć w dokumencie RFC 822. Natomiast w dokumencie RFC 1090 opisany jest standard stosowania protokołu SMTP w sieciach X.25, a w dokumencie RFC 1006 w sieciach TCP/IP. Wszystkie te dokumenty zostały między innymi zaimplementowane w module ISOGATE systemu POLKOM/400.

9. Opis modułów ISOTELEX i ISOFAX

Od wielu lat używane są dwa tradycyjne systemy elektronicznego przesyłania informacji: telex i fax. Telex działa jako autonomiczna sieć z własnymi centralami umożliwiająca przekazywanie informacji reprezentowanej w bardzo ograniczonym zbiorze znaków i z małą szybkością (standardowo 50 bodów). Fax natomiast wykorzystuje do przesyłania dokumentów normalną publiczną komutowaną sieć telefoniczną. Usługa ta ma także bardzo istotne ograniczenia.
W systemie POLKOM/400 zaimplementowano dostęp zarówno do sieci telexowej jak i faxowej.

10. Interfejsy programowe - APIs

W systemie POLKOM/400 dostępne są następujące interfejsy programowe APIs (*Application Programming Interface*):

- API ISOPLEX 800 - interfejs programowy MTA
- API XDS++ - interfejs programowy X.500
- ISOTRADE - interfejs programowy EDI
- API X/Open - interfejs programowy UA
- MAPI - interfejs programowy firmy Microsoft
- SSAPI - Security profile provider, Key store provider, Algorithm provider

11. Elektroniczny spis abonentów - X.500

Aby MHS X.400 mógł się stać systemem o zasięgu globalnym konieczne było rozwiązanie problemu implementacji i utrzymania elektronicznego spisu użytkowników. Zagadnienie to jest przedmiotem zaleceń serii X.500, definiujących model funkcjonalny spisu abonentów.
Spis jest zbiorem otwartych systemów współpracujących przy utrzymywaniu logicznej bazy danych o zbiorze obiektów realnego świata. Użytkownicy spisu, którymi mogą być osoby lub programy, mają możliwość odczytu i modyfikacji informacji, pod warunkiem posiadania uprawnień. Podczas dostępu do spisu każdy użytkownik jest reprezentowany przez moduł dostępu do spisu DUA (*Directory User Agent*). Moduł ten jest procesem aplikacyjnym w sensie modelu OSI.
W systemie POLKOM/400 usługę X.500 realizuje moduł ISOPLEX DS.

12. Elektroniczna wymiana danych - ISOTRADE

W oparciu o standard X.400 powstały systemy EDI (Electronic Data Interchange), które pozwalają na przesyłanie gotowych dokumentów wprost między programami przetwarzającymi te dokumenty u różnych użytkowników. Termin EDI oznacza systemy elektronicznego przesyłania dokumentów handlowych takich jak zamówienia, faktury,

płatności. Głównymi standardami w dziedzinie EDI są: ANSI X.12 w Stanach Zjednoczonych i EDIFACT (*Electronic Data Interchange for Administration, Commerce and Trade*) w Europie. Oba są zgodne z nowym zaleceniem CCITT X.435 definiującym między innymi sposób wykorzystania systemu X.400 do transportu komunikatów EDIFACT'owych i X.12.

13. Standardy i profile systemu POLKOM/400

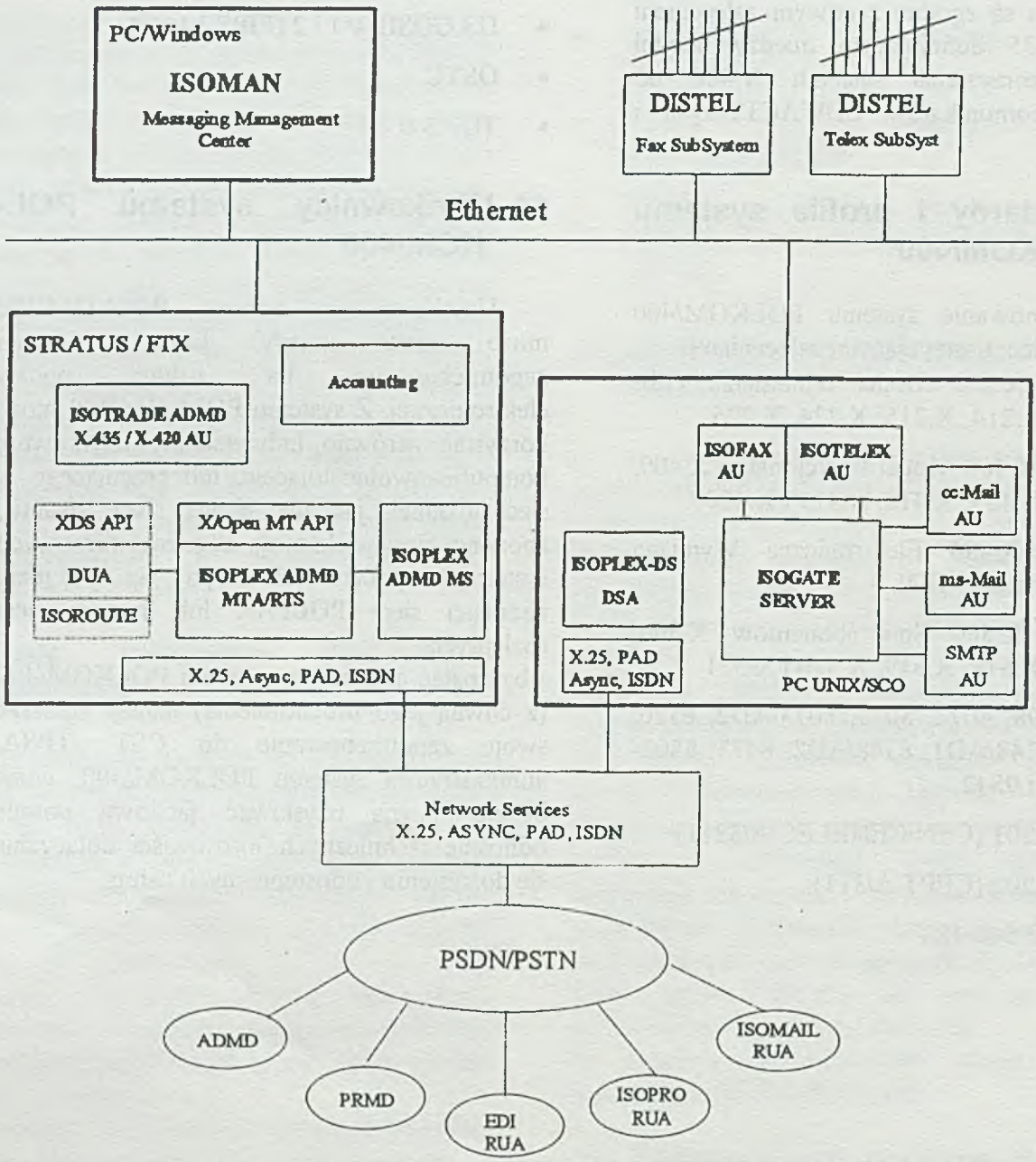
Funkcjonowanie systemu POLKOM/400 będzie zgodne z następującymi zaleceniami:

- CCITT X.200 Model odniesienia OSI, X.213, X.214, X.215, X.224, X.225
- CCITT X.400 Model funkcjonalny X.400, X.401, X.409, X.410, X.411 i X.420
- CCITT X.435 Elektroniczna Wymiana Dokumentów - EDI
- CCITT X.500 Spis abonentów X.500, X.509, X.518, X.519, X.520 i X.521
- ISO 7498, 8072, 8073, 8073/AD2, 8326, 8327, 8348/AD1, 8348/AD2, 8473, 8802-2, 8880 i 9542
- ENV 41201 (CEN/CENELEC A/3211)
- ENV 41202 (CEPT A/311)
- NIST SP 500-183
- Australian GOSIP v 2.0
- NORDIC GOSIP
- UK GOSIP v 4.0 i 4.1
- US GOSIP v 1 i 2 (FIPS 146-1)
- OSTC
- TOP 3.0

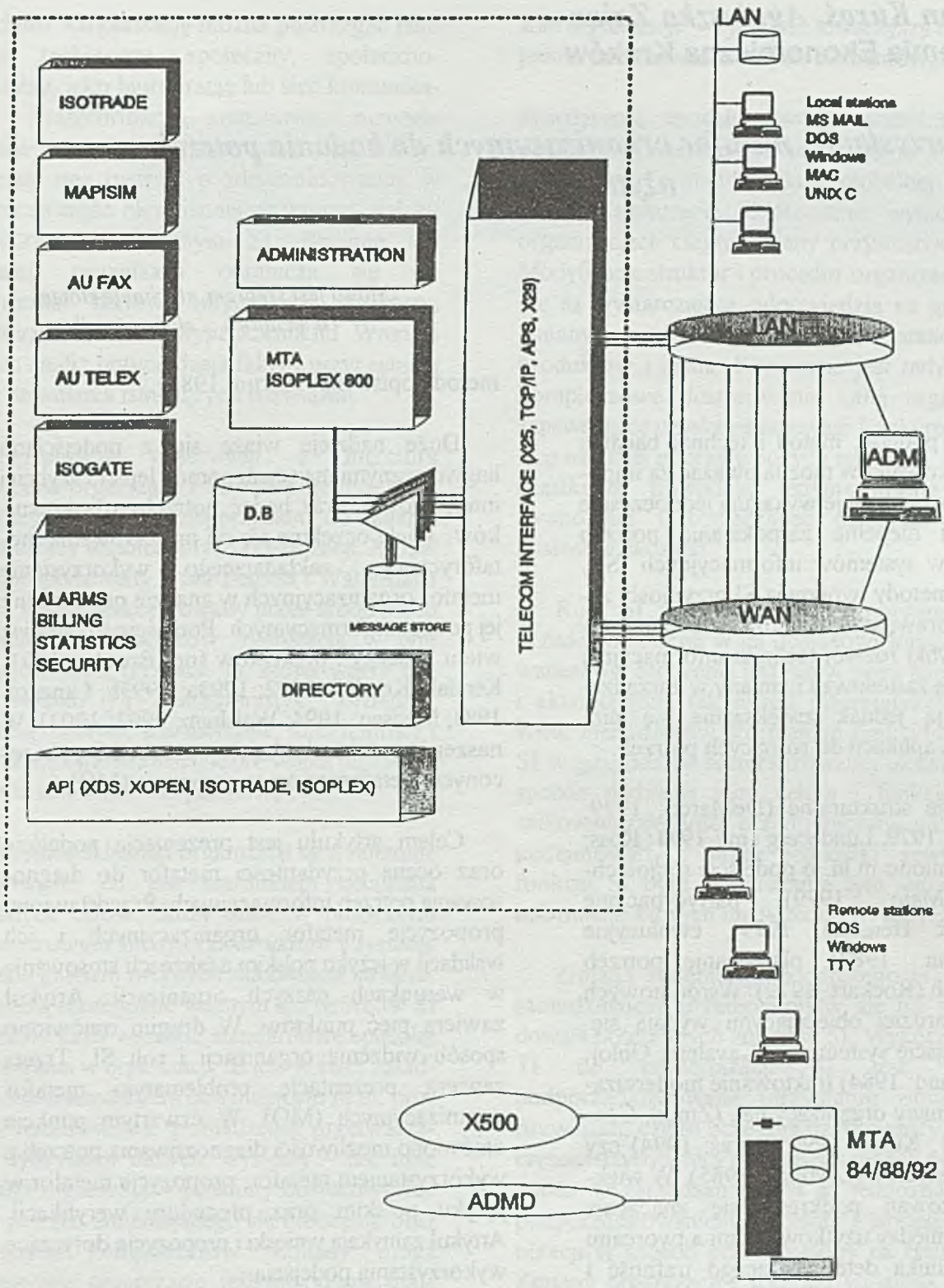
14. Użytkownicy systemu POLKOM/400

Użytkownikiem systemu POLKOM/400 może zostać każdy, kto posiada zapotrzebowanie na usługi poczty elektronicznej. Z systemu POLKOM/400 może korzystać zarówno indywidualny użytkownik komputera wolnostojącego lub pracującego w sieci lokalnej, jak też serwer sieci lokalnej, abonenci sieci teleksowej, abonenci firmowych systemów pocztowych typu np. cc:mail, abonenci sieci POLPAK lub innych sieci rozległych.

Aby zostać abonentem systemu POLKOM/400 (z chwilą jego uruchomienia) należy zgłaszać swoje zapotrzebowanie do CST -TPSA, administratora systemu POLKOM/400, gdzie będzie można uzyskiwać fachową poradę odnośnie technicznych możliwości dołączania się do systemu i udostępnianych usług.



Rys. 1. : Konfiguracja sprzętowa systemu POLKOM/400



Rys. 2. : Struktura funkcjonalna systemu POLKOM/400

Marian Kuraś, Agnieszka Zając
Akademia Ekonomiczna Kraków

Wykorzystanie metafor organizacyjnych do badania potrzeb użytkownika

Mowa jest srebrem, słuchanie złotem
a metafora ... złotym środkiem.

1. Wstęp

Rozwój podejść, metod i technik badania potrzeb użytkowników można uważać za imponujący. Liczne publikacje wykazują jednocześnie niedostatki i niepełne zaspokajanie potrzeb użytkownikow systemów informacyjnych (SI). Strukturalne metody tworzenia SI przyniosły zauważalną poprawę zgodności zastosowań z potrzebami. Szybki rozwój techniki informacyjnej (TI), wzrost jej zastosowań i zmiany w zarządzaniu powodują jednak zwiększanie się niedostosowania aplikacji do rosnących potrzeb.

Koncepcje strukturalne (DeMarco: 1979; Gane, Sarson: 1979; Lundeberg i in.: 1981; Ross: 1977) uzupełniono m.in. o podejście socjotechniczne (DiMaio: 1979), partycypacyjne (Mumford & Henshall: 1979), ewolucyjne (Rzevski i in.: 1982), planowanie potrzeb informacyjnych (Rockart: 1979). Wśród nowych podejść najbardziej obiecującymi wydają się: miękkie podejście systemowe (Cavalere, Obloj: 1993; Checkland: 1984) traktowanie modernizacji SI jako zmiany organizacyjnej, (Zmud, Cox: 1979; Hayder, Kuraś: 1994; Kuraś: 1994) czy analiza lingwistyczna (Lyytinen: 1985). W większości opracowań podkreśla się znaczenie komunikacji między użytkownikami a twórcami SI jako czynnika determinującego trafność i poprawność definicji potrzeb. Eliminowaniu bądź ograniczaniu zakłóceń w komunikacji między uczestnikami prac służą języki graficznego opisu SI (np. DeMarco: 1979; Kuraś i in.: 1992; Lundeberg i in. 1981; Ross: 1977) czy formalne metody opisu (np. Martin: 1985).

Duże nadzieje wiąże się z podejściami lingwistycznymi mającymi pomóc lepiej i szybciej interpretować oraz badać potrzeby użytkowników. Wiele oczekuje się od projektowania metaforycznego zakładającego wykorzystanie metafor organizacyjnych w analizie organizacji i jej potrzeb informacyjnych. Podejście to omawia wielu badaczy i praktyków (np. Brooke: 1991; Kendall, Kendall: 1992; 1993a; 1993b; Lanzara: 1983; Madsen: 1994; Walsham: 1991; 1993). W naszej literaturze brak jest opracowań poświęconych metaforom organizacyjnym (MO).

Celem artykułu jest prezentacja podejścia oraz ocena przydatności metafor do diagnozowania potrzeb informacyjnych. Przedstawiono propozycje metafor organizacyjnych i ich walidacji w języku polskim a także ich stosowanie w warunkach naszych organizacji. Artykuł zawiera pięć punktów. W drugim omówiono sposób widzenia organizacji i roli SI. Trzeci zawiera prezentację problematyki metafor organizacyjnych (MO). W czwartym punkcie omówiono możliwości diagnozowania potrzeb z wykorzystaniem metafor, propozycje metafor w języku polskim oraz procedurę weryfikacji. Artykuł zamykają wnioski i propozycje dotyczące wykorzystania podejścia.

2. Pojęcie organizacji i jej SI

Rozmaitość punktów spojrzenia na organizację jest przyczyną zróżnicowania metod

jej badania. Organizację można postrzegać jako system techniczny, społeczny, społeczno-techniczny, jako biurokrację lub sieć komunikacyjną. Transformacja systemowa, niewiele zmieniła w praktyce traktowania organizacji. Pozostaje ona tworem o zdeterminowanej, w zasadniczej części niezmiennej strukturze, stałych procedurach i sztywnym SI. Badanie tak widzianej organizacji ogranicza się do stwierdzania faktów dotyczących struktury, procedur, zatrudnienia, wyposażenia itd. Wnioski z takich analiz potwierdzają fakty i przyczyniają się do utrwalania istniejących rozwiązań.

Konfrontacja obserwacji z literaturą poświęconą organizacji i SI wskazuje model 7's McKinseya jako najbardziej przydatny do badania i modernizacji współczesnych organizacji. Model ten, zaproponowany przez Petersa i Watermana (1984), został zaakceptowany przez środowisko akademickie i praktyków. rganizacja w tym ujęciu jest siecią (rysunek 1), której węzły stanowi siedem jej charakterystyk: *strategia, struktura, systemy, zatrudnianie, umiejętności* i *styl* oraz *wspólne wartości*, które wiążą pozostałe składniki stanowiąc ich podstawę[1].

Wszystkie składniki organizacji są wzajemnie dostosowane, co jest warunkiem osiągania założonych celów. Służy temu w pierwszym rzędzie *strategia* firmy oraz *struktura* i *systemy* określane mianem *twardych składników modelu*. Elementem szczególnie ważnym dla twórców SI są systemy czyli wszelkie standardowe sposoby postępowania w organizacji. SI jest w swej zasadniczej części standardem obejmującym m.in. procedury rejestrowania, gromadzenia, przetwarzania i dystrybucji danych. Systemy - jak inne składniki organizacji - powinny dostosowywać się do potrzeb zmieniającego się otoczenia oraz do innych komponentów. Zmiany mogą dezintegrować organizację, jeśli nie odpowiadają zidentyfikowanym warunkom - *miękkim składnikom organizacji*[2] - i jeśli nie towarzyszy im wzajemne dostosowanie innych elementów.

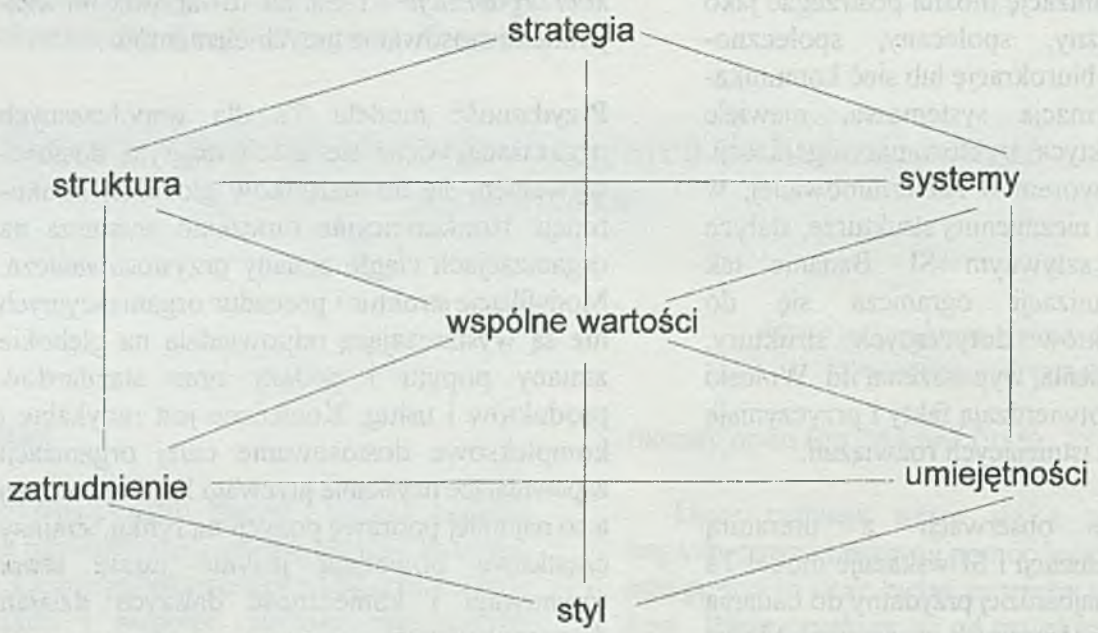
Przydatność modelu 7's dla współczesnych organizacji wiąże się z ich ciągłym dostosowywaniem się do warunków globalnej konkurencji. Konkurencyjne otoczenie wymusza na organizacjach ciągłe zmiany przystosowawcze. Modyfikacje struktur i procedur organizacyjnych nie są wystarczającą odpowiedzią na głębokie zmiany popytu i podaży oraz standardów produktów i usług. Konieczne jest radykalne i kompleksowe dostosowanie całej organizacji zapewniające uzyskanie przewagi konkurencyjnej a co najmniej poprawę pozycji na rynku. Zmiany cząstkowe powodują jedynie utratę stanu równowagi i konieczność dalszych działań dostosowawczych.

Rola SI w funkcjonowaniu organizacji a przede wszystkim w jej dostosowywaniu się do warunków otoczenia jest nie w pełni rozumiana i akceptowana tak wśród informatyków jak i wielu menadżerów. Formalizm funkcjonowania SI w gospodarce scentralizowanej ukształtował sposób widzenia jego celów i funkcji. Przy całkowitej zależności 'przedsiebiorstw' od decyzji podejmowanych przez jednostki zewnętrzne funkcją SI było dostarczanie tym jednostkom informacji, których nie wykorzystywały.

Zmiana systemu gospodarczego przyniosła samodzielność przedsiębiorstw ale nie spowodowała potrzebnych zmian w SI. Wykorzystanie TI do modernizacji SI jest zwykle podporządkowywane utrzymaniu istniejących rozwiązań, czego oczekują użytkownicy. Coraz częściej użytkownicy dostrzegają jednak potrzebę zmian w zarządzaniu. Jest to jednoznaczne z podporządkowaniem SI celom i strategii organizacji w sposób przedstawiony na rysunku 2. Zmiany organizacji i sposobu widzenia roli TI wymagają wykorzystania odpowiednich metod i

[1] Więcej na temat modelu 7's i jego przydatności w analizie SI w pracy (Kuraś: 1994).

[2] Pozostałe składniki modelu 7's: wspólne wartości, zatrudnianie, umiejętności i styl stanowią tło społeczne albo podstawę i warunki działania twar' dych składników organizacji.



Rys. 1. Model 7-s McKinseya.

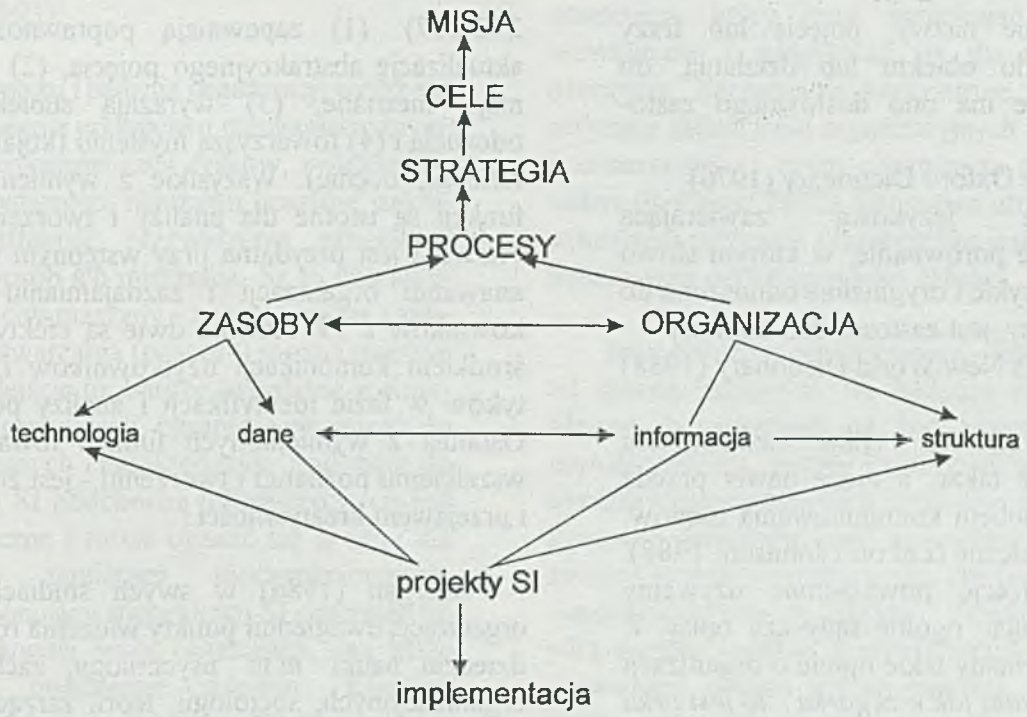
Źródło: [Peters, Waterman: 1984]

technik tworzenia SI. Metody tworzenia SI powinny umożliwiać badanie otoczenia przyszłego SI: strategii, struktury, innych systemów a także miękkich składników organizacji. SI ma służyć organizacji i musi z nią zintegrować się, zatem jego założenia muszą wynikać ze strategii a nowe rozwiązania muszą odpowiadać ograniczeniom narzucanym przez inne składniki.

Wprowadzanie TI do organizacji jest zmianą organizacyjną, nawet jeśli nie jest to zamierzone albo inicjatorzy sobie tego nie uświadamiają. Jest nią również wówczas, gdy kierownictwo organizacji postanawia wprowadzić TI miast dokonywać zmiany. Jest przy tym zmianą, która nie skoordynowana z innymi komponentami organizacji, powoduje zakłócenia i konieczność dostosowania. Modernizacja SI wymaga zbadania organizacji, jeśli oczekuje się efektów w postaci co najmniej zwrotu nakładów poniesionych na stworzenie i wdrożenie SI.

Metody, jakimi posługuje się informatyka są mało przydatne do badania organizacji i definiowana potrzeb użytkowników SI. Krytyka dotyczy przede wszystkim filozofii i podejścia a w konsekwencji metod wykorzystywanych w tworzeniu SI. Klein i Lyytinen (1985) dostrzegają źródła nieprzydatności większości podejść, metod i technik w tym, że wywodzą się one z metody naukowej. Powoduje to, że sąone nieskuteczne i nieefektywne w badaniu zachowań jednostek i grup.

Odmienne założenia przyjęto w *miękkiej metodzie systemowej* (MPS; *SSM - Soft System Methodology*; Checkland: 1984; Cavalere, Obloj: 1993; Waring: 1989). MPS stawia sobie za cel rozpoznanie problemu - strukturyzację problemów źle określonych i słabo ustrukturyzowanych. Ma pomóc znaleźć przyczyny zakłóceń i wskazać zakres zmian. MPS zapewnia przeprowadzenie analizy pojęciowej a następnie dobór odpowiednich *twardych metod systemowych* (TMS). MPS pomagają uzyskać odpowiedź na pytanie: *co i po co należy zrobić?*, podczas gdy TMS odpowiadają na pytanie: *jak to zrobić?*



Źródło: opracowanie własne

Rys. 2. Strategia postępowania przy modernizacji SI.

Można uznać za paradoks to, że wprowadzanie rozwiązania technicznego, po którym wiele obiecuje sobie kierownictwo, nie przynosi spodziewanych efektów, gdy korzysta się z twardych metod systemowych. Rozwiązanie techniczne jest jedynie środkiem a nie istotą zmiany nieodzownej do przystosowania się organizacji do wymagań otoczenia (Kuraś: 1994). Dostosowanie dotyczy zachowań organizacji i jej członków a TI ma uczynić je skutecznymi i efektywnymi. W pierwszym zatem rzędzie jest konieczne zidentyfikowanie zachowań ludzi i wynikających z nich potrzeb. Zachowania ludzi w organizacji często interpretuje się jako nieracjonalne. Zastosowania TI - wprowadzane jako czynnik zamierzonej ich racjonalizacji - przynoszą negatywne skutki, gdyż racjonalizacja polega na podporządkowywaniu ludzi rozwiązaniom technicznym.

Wykorzystanie metafor do analizy należy zaliczyć do miękkich metod systemowych. Ich stosowanie jest możliwe we wszystkich podejściach. Są jednak szczególnie przydatne w diagnozowaniu problemów organizacji, niezbędnemu w celu dobrania odpowiednich metod badania. Zauważmy, że metafory pozwalają na określenie sposobu postrzegania organizacji akceptowalnego dla wszystkich uczestników procesu. Metafory pozwalają uprościć i skrócić analizę zwiększając zarazem trafność diagnozy.

3. Metafory organizacyjne - definicje i przegląd podejść

Metafory - wszechobecne w działalności wymagającej komunikowania - wykorzystuje się w sposób nieuświadomiony jako środek komunikacji w analizie organizacji. Badają ją specjaliści z różnych dziedzin posługujący się różnymi językami, co utrudnia przekazanie obrazów, ocen, osądów i wrażeń. Metafory usuwają bariery i dają analitykom możliwość wyrażenia treści trudnych do przedstawienia w językach opierających się na odmiennych symbolach. Można zatem twierdzić, że są naturalnym narzędziem analizy systemowej w szczególności, definiowania systemów. Metafora jest znana z literatury, ale używa się jej w języku potocznym. Definiuje się ją w następująco:

- zastosowanie nazwy, pojęcia lub frazy opisowej do obiektu lub działania, do którego nie ma ono dosłownego zastosowania.

  Concise Oxford Dictionary (1976)
- konstrukcję językową zawierającą sugerowane porównanie, w którym słowo lub fraza zwykle i oryginalnie odnoszona do jednej rzeczy, jest zastosowana do innej.

  Webster's New World Dictionary (1988)

Metafory są nie tylko ozdobnikami wypowiedzi, ale także, a może nawet przede wszystkim, sposobem komunikowania osądów, ocen, wrażeń i odczuć (Lakoff i Johnson: 1988). Badając organizację powszechnie używamy metafor formułując ogólne sądy czy opisy. Z doświadczenia znamy takie opinie o organizacji jak: *'wszystko działa jak w zegarku'*, *'to jest orka na ugorze'*, *'to jeden wielki cyrk'* itp.

Metafora w sposób charakterystyczny dla pewnej kultury syntetycznie wyraża coś, co jest niewyrażalne, niemierzalne a postrzegane, doświadczane i odczuwane. Wiążąc fragmenty w znaczącą całość, będącą pewnym idealnym obrazem, komunikuje odczucia, wrażenia i doświadczenia w warunkach, gdy brak jest wspólnego języka opisu a trzeba wskazać (podkreślić) istotne cechy obiektu. Metafora odwołuje się zatem do intuicji, doświadczenia i wiedzy odbiorcy czyli jego zdolności kojarzenia obrazów.

Metafory są przydatne w procesie uczenia się jako środek abstrakcji pojęciowej. Pomagają w budowie modelu pojęciowego problemu, co jest wstępnym krokiem tworzenia modelu empirycznego a następnie rozwiązania. W analizie systemów metafory są szczególnie przydatne do budowy modelu SI. Przykładów wykorzystania metafor dostarcza literatura poświęconej problematyce tworzenia i rozwoju SI (np. Cash i in. 1992; Nolan: 1979, Yourdon: 1989). Autorzy nazywają zjawiska i procesy używając pojęć metaforycznych nie związanych z SI ale mających jednoznaczne i pomocne konotacje. MO spełniają cztery funkcje (Weaver: 1967; ss.: 252-257): (1) zapewniają poprawność lub aktualizację abstrakcyjnego pojęcia, (2) wyjaśniają nieznane, (3) wyrażają subiektywne odczucia i (4) towarzyszą myśleniu (kojarzeniu, refleksji, ocenie). Wszystkie z wymienionych funkcji są istotne dla analizy i tworzenia SI. Pierwsza jest przydatna przy wstępnym rozpoznawaniu organizacji i zaznajamianiu użytkowników z TI. Kolejne dwie są efektywnym środkiem komunikacji użytkowników i analityków w fazie identyfikacji i analizy potrzeb. Ostatnia z wymienionych funkcji towarzyszy wszelkiemu poznaniu i tworzeniu - jest źródłem i przejawem kreatywności.

Morgan (1986) w swych studiach nad organizacją uwzględnił punkty widzenia różnych dziedzin nauki: m.in. psychologii, zachowań organizacyjnych, socjologii, teorii zarządzania. Rezultatem studiów jest dziewięć metafor organizacyjnych czyli różnych sposobów widzenia organizacji: (1) maszyna, (2) organizm, (3) mózg, (4) kultura, (5) system polityczny, (6) dom wariatów, (7) lawina, potok, (8) transformacja i (9) instrument dominacji (Morgan: 1986). Walsham (1993) omawia cztery spośród wymienionych metafor (maszynę, organizm, kulturę i system polityczny) szczególnie przydatnych do rozpoznania i analizy organizacji. Wyznaczają one podejście do badania organizacji, której rozumienie jako systemu o cechach wybranej metafory determinuje dalszy sposób postępowania i wyboru metod.

Maszyna - podejście mechanistyczne. Widzenie organizacji jako **maszyny** jest głęboko zakorzenione w społeczeństwach przemysłowych, gdzie człowiek czuje sie jedynie "trybikiem" w wielkiej maszynie. Naukowe zarządzanie zapoczątkowane przez Taylora opiera się na takim właśnie widzeniu organizacji. Taylor w swych pracach w logiczny sposób dowodził możliwości stworzenia perfekcyjnej maszyny produkcyjnej, będącej konstrukcją złożoną z ludzi i środków wytwórczych. Prace te, szeroko krytykowane przez następne 50 lat, miały jednak duży wpływ na organizację i zarządzanie oraz na

sposób, w jaki mówimy o życiu organizacji (Walsham 1993).

Sposób opisu i badania organizacji wciąż ma wiele wspólnego z podejściem mechanistycznym. Mówiąc o maksymalizacji zysków, zwiększeniu produkcji, wydajności, obniżaniu kosztów, zakłada się możliwość przewidzenia przyszłych efektów i sposób ich mierzenia. Są to charakterystyczne cechy maszyny - znając wejścia i sposób ich przetwarzania (procesy) można mierzyć wyjścia. Podejście to, bardzo przydatne w funkcjonowaniu organizacji, ogranicza się jednak do mechanistycznych charakterystyk. Podobnie do organizacji i SI podchodzą informatycy, co jest krótkowzroczne i może okazać się groźne dla powodzenia zamierzeń modernizacyjnych. Analityk dokonujący specyfikacji SI opierając się na takiej naiwnej wizji organizacji, ma małe szanse na osiągnięcie sukcesu. Efektem mechanistycznego podejścia są sztywne i niedostosowane do potrzeb SI, których nie można adaptować do zmieniających się warunków otoczenia. Nadto, pominięcie subtelności kontekstu społecznego SI ma negatywny wpływ na zaspokojenie potrzeb i aspiracji ludzi a z perspektywy kierownictwa powoduje brak motywacji, niską wydajność i opór wobec zmiany (Walsham 1993).

Organizm - podejście nowoczesne. Bardziej przystaje do praktyki porównanie organizacji do **organizmu**, czyli struktury rodzącej się i rozwijającej i umierającej. W przeciwieństwie do maszyny (systemu zamkniętego), to podejście akcentuje otoczenie, w którym organizm funkcjonuje. Walsham (1993) zauważa, że genezy tego podejścia można szukać już w latach dwudziestych, gdy Hawthorne zwrócił uwagę na czynnik ludzki jako środek zwiększania efektywności organizacji (Robbins: 1992). W 1943 roku Maslow zaproponował model hierarchii potrzeb człowieka, uświadamiając ważność potrzeb wyższego rzędu w tym potrzeby przynależności do grupy, jaką jest organizacja. Prace Maslowa zapoczątkowały socjo-techniczne podejście do organizacji. Organistyczne widzenie organizacji daje podstawy dla podejścia sytuacyjnego. Jego istotą jest postrzeganie organizacji jako systemu otwartego, który musi zaspokajać potrzeby wewnętrzne i adaptować się do warunków otoczenia. Zarządzanie koncentruje się na koordynacji zadań, form organizacyjnych i łączności z otoczeniem, z czym organizacja musi sobie radzić (Robbins: 1992). Głównym utrudnieniem stosowania podejścia sytuacyjnego jest założenie traktowania organizacji jako całości.

Inną metaforą organizacyjną - **kulturą** - już od dawna zajmowali się badacze organizacji. Można ją rozumieć na trzy sposoby: *jako wspólną wiedzę, system wspólnych znaczeń* lub *wyrażanie podświadomych procesów myślowych*. Pierwsze znaczenie jest najogólniejsze, gdyż zawiera system wspólnych znaczeń (czyli pewną wiedzę) oraz wiedzę "ukrytą" i dziedziczoną, za jaką można uważać podświadomość. Kultura organizacji zwykle jest niejednolita, zwłaszcza kiedy mamy do czynienia z dużymi organizacjami. Często niezbędne jest wyodrębnienie i badanie podkultur (subkultur).

Kulturę organizacyjną omawia wiele publikacji z zakresu zachowań organizacyjnych i zarządzania (np. Daft: 1992; Robbins: 1991). Należy ją rozumieć jako *system wyznawanych wartości*[3] *i ich wspólne pojmowanie przez członków organizacji*. Takie rozumienie kultury organizacji pozwala kształtować i egzekwować pożądane postawy pracowników nieodzowne do podejmowania zbiorowego wysiłku służącego wypełnianiu misji i osiąganiu założonych celów. Kultura jest zatem ważnym narzędziem menadżerów w kierowaniu firmą. Morgan (1986) zauważa, że mogą oni wpływać na rozwój kultury, poprzez uświadomienie sobie symbolicznego znaczenia ich poczynań i przez próby wprowadzania pożądanych wartości oraz innych symboli.

[3] Robbins (1991) wymienia dziesięć grup charakterystyk kultury organizacyjnej: *inicjatywę indywidualną, tolerowanie ryzyka, ukierunkowanie, integrację, wspomaganie ze strony zarządzania, sterowanie i kontrolę, stopień identyfikacji, system motywowania, torlerowanie konfliktu* oraz *wzorce komunikowania się.*

Metafora **systemu politycznego** zachęca do widzenia organizacji jako luźnej sieci ludzi mających sprzeczne interesy ale współdziałających ze sobą w celu osiągnięcia korzyści (Morgan: 1986). Kluczową charakterystyką tej metafory jest władza, pojmowana jako narzędzie rozwiązywania konfliktów lub koordynacji różnych interesów (Walsham 1993). Władza jest umiejętnością osiągania efektów; bez względu na jej powiązanie z czysto grupowymi interesami (Giddens 1984). Władza jako taka nie jest wrogiem wolności, jest jej wręcz pokrewna, choć znane są jej niszczące własności.

Problemy polityczne pojawiają się w tworzeniu SI we wszystkich fazach. Wynikają z możliwych (antycypowanych) przesunięć władzy w wyniku zmiany dostępu do danych (zwiększenia/zmniejszenia zakresu) co jest ograniczeniem lub rozszerzeniem władzy. W praktyce tworzenia SI często pomija się te aspekty, co rodzi głębokie konflikty, których skutkiem może być później bojktowanie lub odrzucenie nowego narzędzia.

J. i K. Kendall (1992: 1993a: 1993b) zaproponowali zestaw metafor przydatnych w analizie organizacji. Są one zrozumiałe dla wielu użytkowników końcowych, choć nie są typowedla zarządzających. Wyróżnili oni dziewięć metafor oraz główne ich atrybuty istotne z punktu widzenia wykorzystania w tworzeniu SI (tablica 1).

Metafora "**gry**" (dotyczy raczej drużyny niż indywidualnego gracza) jest związana z pojęciami celu, trudności, wiadomego ryzyka, zabawy, pracy zespołowej i przywództwa. W organizacji opisywanej przez tę metaforę każdy gracz ma specyficzną rolę do spełnienia (pozycja na boisku). Sukces drużyny zależy od kompetencji poszczególnych członków współpracujących by osiągnąć cel. Celem jest **wygrać** co oznacza, że w danym starciu są zarówno przegrani jak i zwycięzcy. Taka organizacja stale dąży do uzyskiwania coraz lepszych możliwości i wyników. Najlepszych zawodników ceni tak trener jak i drużyna, gdyż to oni mają największy wpływ na rezultaty.

Głównym wymaganiem zastosowania metafory "**maszyny**" jest przewidywalność zachowań w konkretnych warunkach. Maszyna służy określonemu celowi, a jej zachowanie jest przewidywalne. Maszynę zaprojektował/stworzył projektant/konstruktor (osoba lub zespół). Projektant może być przekonany o swej wszechwiedzy, gdyż zna sposób zachowania się organizacji i zachodzące w niej procesy oraz jej możliwości. Taka "wszechwiedza" może jednak być niepożądana a nawet niebezpieczna. Potrzeby użytkowników będą zaspokajane w części lub w małym stopniu, gdy o rozwiązaniach będzie decy

Tablica 1
Metafory użytkowników bezpośrednich i ich kluczowe charakterystyki.

Metafora	Orientacja na	Otoczenie	Lider
Gra	cel	porządek	trener
Maszyna	cel	porządek	projektant
Podróż	cel	chaos	kapitan
Dżungla	alternatywy	chaos	przewodnik
Rodzina	alternatywy	porządek	głowa domu
Zoo	alternatywy	chaos	opiekun

Społeczność	alternatywy	porządek	przywódca
Wojna	cel	chaos	generał
Organizm	różnie	zróżnicowane	innowator i żywiciel

Źródło: Kendall, Kendall 1993

dował wyłącznie twórca systemu.

Metaforę "**podróży**" (morskiej) charakteryzuje lider, załoga i sytuacja: nieprzewidywalność, ryzyko ale także przygoda. Największy nacisk należy położyć na lidera, który powinien mieć cechy przywódcze, bardzo dobrze znać swoją załogę i potrafić zjednoczyć ją w dążeniu do wspólnego celu. Powinien też posiadać wiedzę o otoczeniu i prawach jakimi się rządzi. Organizacja opisywana przez tę metaforę ma określony cel do którego dąży, chociaż cel ten może być bardzo odległy. Jego osiągnięcie wszelako jest w pełni uzależnione od współpracy członków organizacji.

"**Dżungla**" charakteryzuje się przede wszystkim nieprzewidywalnością i małą znajomością otoczenia oraz ukierunkowaniem na alternatywy. Jest dobrą ilustracją zasady "każdy sobie" tzn. każdy odpowiada za siebie i na własną rękę stara się rozwiązywać napotykane problemy. Charakterystyczną cechą tej metafory jest agresja - nie do przyjęcia w innych metaforach. Należy ograniczyć ufność, bo niebezpieczeństwo może czaić się wszędzie. Ceni się spostrzegawczość, niezbędną by je rozpoznać i próbować zaradzić, przy czym nie ma ściśle określonego sposobu postępowania. Należy kierować sie doświadczeniem, wiedzą oraz intuicją. W przypadku "dżungli" nie ma przywódcy - można co najwyżej liczyć na przewodnika, którego zadaniem jest wyprowadzenie organizacji z "dżungli" i zażegnanie niebezpieczeństw.

Metafora "**rodziny**" wiąże się ze zróżnicowaniem członków organizacji. Określenie: *"jesteśmy jedną wielką rodziną"* odnosi się do tego typu organizacji. Kontakty opierają się na wzajemnym zrozumieniu, przywiązaniu i przyjaźni zapewniającej atmosferę komfortu psychicznego oraz poczucia przynależności. Każdy ma odpowiednie miejsce w hierarchii i rolę do spełnienia naturalną dla zajmowanej pozycji. Ważną cechą jest zróżnicowanie potrzeb członkow rodziny i świadomość odpowiedzialności wobec innych. Metafora ta jest związana z negocjacyjnym rozwiązywaniem konfliktów. Ostateczne decyzje podejmuje głowa rodziny po uprzednim wysłuchaniu opinii wszystkich zainteresowanych. Ważną charakterystyką jest zapotrzebowanie na nowych członków - powiększanie się rodziny związane jest z uczuciem szczęścia i satysfakcji.

Nieprzewidywalność i chaos to cechy "**zoo**". Zwierzęta zamknięte w klatkach nie odczuwają zagrożenia ze strony naturalnych wrogów, co powoduje zatracanie instynktu. Uwaga jest skierowana na własne potrzeby, sympatie i antypatie. "Zoo" jest zgrupowaniem jednostek nie kontaktujących się ze sobą w sposób naturalny i nie mających jednoznacznego celu. Zwierzęta nie mogą przetrwać bez zapewnienia odpowiednich warunków i nie mogą też dbać same o siebie, co wyzwala potrzebę opieki ze strony opiekuna. Jest on liderem z całkowitą władzą nad członkami takiej organizacji.

Metaforyczną "**społeczność**" cechuje orientacja na alternatywne cele w zależności od potrzeb i możliwości ich zaspokojenia. Zauważa się pokrewieństwo z metaforą "rodziny", z tą różnicą, że w społeczności większe jest zróżnicowanie jej członków i ich zainteresowań. Często też brakuje zrozumienia i zaufania w stosunku do innych osób. Każdy chce osiągnąć własny cel. Społeczność rządzi się własnymi prawami, kieruje się zasadami, dlatego jej zachowanie może być przewidywalne. Liderem jest głowa społeczności (np. państwa, regionu, towarzystwa), Jego zadaniem jest egzekwowanie prawa, wprowadzanie nowych zasad,

rozstrzyganie sporów itd. Ważną cechą jest negocjacyjna droga osiągania kompromisu pomiędzy sprzecznymi interesami grup.

Główną cechą metafory "**wojny**" jest cel: zwycięstwo. Wojna wymaga najlepszego lidera i strategii działania - elementów niezbędnych do wygrania. Związana jest z niebezpieczeństwem, ryzykiem, stratami i poświęceniem. Ważne znaczenie w tym typie organizacji ma informacja: właściwa we właściwym czasie. Zbieranie informacji o wrogu, jego możliwościach, silnych i słabych stronach decyduje o zwycięstwie. Przepływ informacji w samej organizacji odbywa się z góry na dół - od generała do żołnierzy coraz niższych stopni. Każdy z członków organizacji otrzymuje tyle informacji, ile jest niezbędne do podjęcia właściwych decyzji i działań. Mimo że otoczenie "wojny" określono jako chaos, kieruje się ona określonymi prawami i zasadami.

Podobnie jak w ujęciu klasycznym **"organizm"** jest metaforą kontrastującą z metaforą "maszyny". Organizacja nie jest całkowicie chaotyczna - jej części tworzą nierozłączną całość - ale nie jest w pełni ustrukturyzowana. Bardzo duży wpływ na rozwój organizmu mają warunki zewnętrzne, których pewnej części nie może kontrolować, choć na niektóre można wpływać. Inną cechą organizmu jest jego umiejętność adaptowania się do otoczenia, co determinuje jego zdolność przeżycia.

Przedstawione podejścia i metafory nie wyczerpują możliwości wiążących się z ich stosowaniem. Wykorzystanie metafor zdefiniowanych w innym kręgu kulturowym wymaga nadto ich zbadania i weryfikacji oraz uzupełnienia o metafory specyficzne dla kultury, języka polskiego i rzeczywistości gospodarczej.

4. Diagnozowanie potrzeb informacyjnych z wykorzystaniem metafor

Oczekiwania, jakie można wiązać z wykorzystaniem metafor organizacyjnych, dotyczą przyspieszenia i poprawy trafności formułowania potrzeb użytkowników. Omówione w punkcie 3 MO wykorzystuje się do diagnozowania potrzeb w kulturach, w których zdefiniowano je. Ich przydatność w naszych organizacjach trzeba zweryfikować. Zajdzie również potrzeba zdefiniowania metafor swoistych dla naszej kultury i języka.

4.1. Możliwości i warunki diagnozowania potrzeb

Proces asymilacji TI przez organizację prowadzi do zwiększenia wpływu kierownictwa i bezpośrednich użytkowników na tworzone zastosowania. Mimo pojawiania się nowych metod i technik tworzenia SI nierozwiązanym problemem pozostaje komunikacja. Jej mała skuteczność powoduje, że użytkownicy - mimo uczestniczenia w tworzeniu SI - nie wykorzystują w pełni możliwości oddziaływania na jego rozwiązania. Główną przyczyną jest techniczna orientacja informatyków i bezradność użytkowników wobec TI. Ze względu na specjalizację uczestników procesu tworzenia SI, konieczne jest posługiwanie się wspólnym językiem. Jest nim mieszanka języka naturalnego zawierającego metafory oraz języków sztucznych (graficznych i formalnych). Ewolucję komunikacji i jej wpływu na akceptację zmodernizowanego SI przedstawia tablica 2.

Tablica 2.

Wpływ komunikacji na akceptację SI

	sytuacja	komunikacja	język	akceptacja
1	o wszystkim decyduje programista	brak	techniczny	brak
2	o wszystkim decyduje analityk	brak	techniczny i zarządzania	brak
3	o wszystkim decyduje kierownik	brak	zarządzania	brak

4	o rozwiązaniach decydują uczestnicy ale niezależnie i niespójnie	zakłócona	różne języki specjalistyczne	ograniczon a
5	o rozwiązaniach współdecydują wszyscy uczestnicy	zakłócana ... skuteczna	język potoczny z wykorzystaniem metafor	ograniczon a ... pełna

Źródło: opracowanie własne

Wieloletnie prace nad wykorzystaniem podejścia metaforycznego do potrzeb użytkowników pozwoliły na sformułowanie pewnych prawidłowości. W tablicy 3 przedstawiono prawdopodobieństwo powodzenia prac nad stworzeniem odpowiedniego SI w organizacjach opisywanych przez określone metafory organizacyjne. Przypisanie odpowiednich metafor organizacyjnych może również służyć doborowi odpowiednich podejść i metod tworzenia systemu (Kendall, Kendall: 1992). Prawidłowości takie stwierdzono w wyniku badań powiązania między metaforami a podejściami i metodami.

Wykorzystanie metafor w tworzeniu SI jest tak powszechne (choć nieuświadomione), że można mówić o 'projektowniu metaforycznym.' Madsen (1994) określa tak *proces świadomego wyszukiwania, tworzenia i wykorzystywania metafor w procesie analizy*. Proces ten ma doprowadzić do zgromadzenia zestawu metafor zrozumiałych przez wielu użytkowników. W ten sposób analityk rozwija narzędzia służące zwiększeniu skuteczności komunikacji. W efekcie skraca się czas definicji potrzeb oraz specyfikacji SI.

Projektowanie metaforyczne przebiega w trzech fazach, w których (1) zbiera się, (2) ocenia i (3) rozwija zestaw metafor charakterystycznych dla określonych sytuacji i warunków działania. Madsen (1994) zaleca **generowanie** metafor na podstawie wypowiedzi użytkowników, nawiązujących do wytworów (artefaktów) a jeśli to możliwe, do metafor użytych w określeniu problemu i wykorzystywanych w praktyce. Wygenerowane metafory należy poddać **ocenie** i wybrać takie, które mają szeroki zakres znaczeniowy także w dziedzinie zastosowań oraz są dostosowane do zdolności percepcji użytkowników. Powinny one mieć znaczenie zbliżone do dosłownego oraz być uzupełnione o pojęcia wiążące. W ostatnim kroku **wyjaśnia** się pojęcie (metaforę), szuka jego nowych znaczeń, uzupełnia o opis sytuacyjny, zmienia sposób kojarzenia rzeczywistości, identyfikuje niewykorzystane znaczenia i rozpoznaje sprzeczne konotacje.

Metafory tworzy się i wykorzystuje w każdym zespole projektowym, jednak brakuje metod i technik ocen jakościowych wykorzystania metafor w projektowaniu. Opracowanie metod tworzenia i stosowania metafor w projektowaniu pozwoli na ocenę przydatności tego podejścia w tworzeniu SI.

4.2. Metafory organizacyjne w języku polskim

Literatura anglosaska podkreśla wyjątkową metaforyczność języka angielskiego. Można twierdzić, że każdy język jest pełen metafor z powodu przystępności i konieczności posługiwania się nimi. Nie budzi wątpliwości odmienność niektórych metafor, gdyż są one elementem kultury. Wpływ tradycji, wierzeń, obyczaju i języka na kształtowanie metafor i ich używanie jest oczywisty. W przypadku MO na ich kształtowanie wpływają takie czynniki jak: (1) otoczenie organizacji, (2) praktyka funkcjonowa

Tablica 3

Prawdopodobieństwo sukcesu tworzenia różnych typów SI w organizacjach

Typy systemów informacyjnych	najmniejsze prawdopodobieństwo sukcesu	największe prawdopodobieństwo sukcesu
SI wspomagający współpracę np. wspomaganie biura	społeczność zoo	podróż, gra organizm
SI zorientowany na zwiększenie konkurencyjności	zoo, rodzina społeczność	wojna, gra organizm
Tradycyjny system informowania kierownictwa	wojna dżungla	społeczność maszyna, rodzina
System doradczy	dżungla zoo	maszyna, gra organizm
System wspomagania decyzji	wojna podróż	rodzina, społeczność organizm
SI wyższego kierownictwa	podróż zoo	organizm gra

Źródło: Kendall, Kendall: 1993b

nia, (3) profil i poziom wykształcenia pracowników oraz (2) kultura i tradycja. Ewentualne wykorzystanie metafor używanych w innych językach wymaga wcześniejszej ich weryfikacji. Ponadto ponieważ nie są znane badania nad MO i ich wykorzystaniem do diagnozowania naszych organizacji trzeba znaleźć i zweryfikować metafory opisujące polskie firmy. Badania nad metaforami wymagają wstępnych studiów prowadzących do stworzenia metody gromadzenia, opisu, analizy, walidacji i wykorzystywania do diagnozowania[4].

Metafory: **folwarku, kotła i mafii** są przykładami metafor występujących w polskich organizacjach. **Folwark** opisuje organizację, w której lider (ekonom) ma nieograniczoną władzę i nie w pełni dba o interesy właściciela ani pracowników. **Kocioł** charakteryzuje instytucję o niskim poziomie zorganizowania, w której pod przywództwem lidera (kucharza) przeprowadza się akcje mające zapobiegać katastrofie. Mianem **mafii** określa się organizację spójną, hermetyczną, dbającą jedynie o interesy jej członków, której lider ('capo di tutti capi') egzekwuje bezwzględne posłuszeństwo. Łatwo zauważyć, że przyjęte metafory mają negatywne znaczenia. Są to metafory charakterystyczne dla organizacji funkcjonujących w gospodarce centralnie sterowanej. Nowe ukształtują się i upowszechnią w miarę rozwoju małych i średnich prężnych firm zdolnych do konkurowania na rynku dzięki silnym kulturom organizacyjnym.

4.3. Próba walidacji metafor organizacyjnych - opis gry

Metafory jako narzędzia diagnozowania organizacji należy poddać weryfikacji poprzez odniesienie do modelu organizacji. Modelem powszechnie uznanym jako najbardziej od-

[4] Artykuł jest sprawozdaniem z rospoczętych prac nad metodą gromadzenia i walidacji metafor wykorzystywanych w organizacjach (Kuraś, Zając: 1995). Na podstawie analizy wypowiedzi pracowników firm, w których wykonywano projekty, sformułowano szereg metafor. Będą one opisane, analizowane i ocenione zanim podejmie się dalsze prace nad stworzeniem metodyki.

powiadający wyzwaniom, jakie stoją obecnie przed organizacjami, jest model 7's (omówiony w punkcie 2). Wstępną propozycja badania przydatności metafor zaproponowanych w pacy (Kendall, Kendall: 1992) i trzech metafor używanych w języku polskim w odniesieniu do sytuacji organizacyjnych opiera się na siedmiu zmiennych definiujących organizację w modelu 7's oraz charakterystyce lidera.

Oceny trafności i przydatności metafor dokonuje się poprzez zgromadzenie i porównanie cech opisujących metafory oraz rzeczywiste obiekty i sytuacje, którym je przypisano. Weryfikacja ich przydatności musi być oparta na danych uzyskanych w toku badań na wielkiej i zróżnicowanej co do zawodu, wykształcenia i doświadczenia próbie pracowników różnych organizacji. Gra, jaka będzie przeprowadzona podczas Szkoły ma doprowadzić do wskazania metafor odpowiednich dla podanych sytuacji organizacyjnych. Zadaniem uczestników będzie rozpoznanie różnych typów organizacji, wskazanie metafor jak i ocena trafności ich dopasowania. Celem tej gry będzie z jednej strony zaznajomienie uczestników ze sposobem funkcjonowania metafor organizacyjnych a z drugiej, sprawdzenie hipotez przedstawionych w tekście artykułu.

5. Zakończenie

Metafory organizacyjne są skutecznym narzędziem usprawniającym komunikację między uczestnikami procesu tworzenia SI. Są one wykorzystywane w praktyce badania organizacji i tworzenia SI w sposób spontaniczny a nie metodyczny. Duże zainteresowanie ich wykorzystaniem obserwuje się w krajach zachodnich, gdzie metody tworzenia SI są bardziej zaawansowane. Również w naszej praktyce tworzenia zastosowań powszechnie stosuje się metafory np. do prezentacji możliwości TI. Wykorzystanie metafor jest w pełni naturalne ale zwykle pozbawione refleksji metodologicznej. Brak jest również opracowań na temat wykorzystania metafor organizacyjnych zdefiniowanych w krajach zachodnich. Wstępna ocena tych metafor oraz trzech używanych w polskich organizacjach w oparciu o model 7's daje zachęcające wyniki.

Przeprowadzona próba oceny przydatności znanych metafor powinna pozwolić na dopracowanie metody oraz podjęcie dalszych prac nad wykorzystaniem metafor w badaniu organizacji i potrzeb użytkowników. W dalszej perspektywie wyniki badań będą wykorzystane do stworzenia metodyki badań organizacji oraz potrzeb wprowadzania TI jako narzędzia zmiany organizacyjnej. Kontynuowane będą badania metafor z wykorzystaniem modelu 7's oraz modelu Hofstede (różnic kulturowych).

Bibliografia:

Brooke, C. (1991). Journeys Through the Quality Gap: Information Technology in Two Organizations. PhD. Thesis. London: City University Business School.

Cash, J.I.Jr., McFarlan, F.W., McKenney, J.L. (1992). Corporate Information Systems Management. The Issues Facing Senior Executives. 3rd Ed. Homewood. Ill. - Boston, Ma.: Irwin.

Cavalere, S., Obloj, K. (1993). Management Systems: A Global Perspective. Wadsworth Publ.

Checkland, P.B. (1984). Systems Thinking, Systems Practice. Chichester-N.York: Wiley.

Concise Oxford Dictionary (1976). Oxford: Oxford University Press

Daft, R.L. (1992). Organization Theory and Design. St.Paul: West Publishing Company.

DeMarco, T. (1979). Structured Analysis and System Specification. New York: Yourdon Press.

DiMaio, A., Bartezzaghi, E, Zanarini, G. (1979). A New Systems Analysis Method Based on the STS (Socio-technical Systems) [in:] H.J.Schneider (ed.). Formal Models and Practical Tools for Information Systems Design. Amsterdam: North-Holland.

Gane, C., Sarson, T. (1979). Structured System Analysis. Englewood Cliffs: Prentice-Hall.

Giddens, A. (1984). The Constitution of Society. Cambridge: Policy Press.

Hayder, G., Kuraś, M. (1994). Modernizacja systemu informacyjnego jako zmiana organizacyjna. Zeszyty Naukowe AE w Krakowie. Nr 421.

Kendall, K.E. & Kendall, J.E. (1992). Existence of Predominant Organizational Metaphors and their Implications for Information Systems Development. Kraków: AE.

Kendall, J.E., Kendall, K.E. (1993a). Metaphores and Methodologies: Living Beyond the Systems Machine. MIS Quarterly. Vol.17, No.2 (June 1993).

Kendall, J.E., Kendall, K.E. (1993b). Metaphores and Their Meaning for Information Systems Development. European Journal of Information Systems.

Klein, H.K., Lyytinen, K.J. (1985). The Poverty of Scientism in Information Systems [in:] Mumford et al. (eds). Research Methods in Information Systems. Amsterdam: Elsevier.

Kuraś, M. (1994). Zmiana organizacyjna jako cel modernizacji systemu informacyjnego. IV Konferencja Naukowo-Techniczna Kierunki Rozwoju Systemów Informatycznych i ich Bazy Sprzętowej w Hutnictwie. Kraków - Krynica: SIiTPH - AGH.

Kuraś, M., Hayder, G. (1994). Organizational Change in Transitional Poland: Obstacles and Method. Journal of East - West Business. Vol.1, No.2.

Kuraś, M., Sarga, D., Zaliwski, (1992). Metoda planowania i tworzenia systemów informacyjnych. Raport z prac badawczych 29/ISEI/7/91/5. Kraków: AE.

Kuraś, M., Zając, A. (1995). Metafory organizacyjne - narzędzie

badania i diagnozowania potrzeb użytkowników. Zeszyty Naukowe Akademii Ekonomicznej w Krakowie (w druku).
Lakoff, G., Johnson, M. (1988). Metafory w naszym życiu. Warszawa: PIW.
Lanzara, G.F. (1983). The Design Process: Frames, Metaphores and Games [in:] U.Briefs et al. (eds.). Systems Design For, With and By the User. Amsterdam: North-Holland.
Lundeberg, M., Goldkuhl, G., Nilsson, A. (1981). Information Systems Development. A Systematic Approach. Englewood Cliffs, N.J.: Prentice-Hall.
Lyytinen, K.J. (1985). Implications of Theories of Language for Information Systems. MIS Quarterly. Vol.12 No.1.
Lyytinen, K.J., Klein, H.K. (1985). The Critical Theory of Jurgen Habermas as a Basis for Theory of Information Systems [in:] E.Mumford (ed.). Research Methods in Information Systems. Amsterdam: Elsevier.
Madsen, K.H. (1994). A Guide to Metaphorical Design. CACM. Vol.37. No.12 (December 1994).
Martin, J. (1985). System Design from Provably Correct Constructs. Englewood Cliffs: Prentice-Hall.
Morgan, G. (1986). Images of Organization. Beverly Hills: Sage Publications.
Mumford, E., Henshall, D. (1979). A Participative Approach to Computer Systems Design. London: Associated Business Press.
Nolan, R.. (1979). Managing the Crisis in Data Processing. Harvard Busines Review. (March-April).
Peters, T., Waterman, R.H. (1984). In Serach of Excellence. Lessons from America's Best Run Companies. New York: Warner Books.
R
obbins, S.P. (1991). Organizational Behavior. Concents, Controversies, Applications. 5th Ed. Englewood Cliffs, N.J.: Prentice-Hall.
Rockart, J.F. (1979). Chief Executives Define Their Own Data Needs. Harvard Business Review. March-April 1979.
Ross, D.T. (1977). Structured Analysis (SA): A Language for Communicating Ideas. IEEE Transactions on Software Engineering. (January 1977).
Rzevski, G., Trafford, D.B., Wells, N. (1982). The Evolutionary Design Methodology Applied to Information Systems [in:] T.W. Olle, H.G. Sol, A.A. Verijn-Stuart (eds.). Information Systems Design Methodologies: A Comparative Review. Amsterdam: North-Holland.
Schon, D. (1987). Educating the Reflective Practitioner: Toward a New Design for Teaching and Learning Organization. San Francisco: Jossey-Bass.
Walsham, G. (1991). Organizational Metaphores and Information Systems Research. European Journal of Information Systems. Vol.1. No.2.
Walsham, G. (1993). Interpreting Information Systems in Organizations. Chichester: Wiley & Sons.
Weaver, R.M. (1989). A Rhetoric and Handbook.. New York: Holt, Reinhart & Winston.
Webster's New World Dictionary (1988). New York: Miriam Webster.
Yourdon, E. (1989). Modern Structured Analysis. Englewood Cliffs, N.J.: Prentice-Hall.
Zmud, R.W., Cox, J.F. (1979). The Implementation Process: A Change Approach. MIS Quarterly. Vol.3, No.2 (June 1979).

Elżbieta Miłosz i Marek Miłosz
Politechnika Lubelska

Organizacja sieciowych serwisów informacyjnych dla małych i średnich przedsiębiorstw

1. Wstęp

Proces transformacji gospodarczej w Polsce związany jest z dużą dynamiką zmian liczby i struktury podmiotów gospodarczych. W ostatnich 3 latach tylko w województwie lubelskim powstało prawie 40 000 nowych firm. Wśród nich przeważającą większość stanowią małe i średnie przedsiębiorstwa (MŚP). Firmy te początkowo mające charakter handlowy często zmieniają zakres i profil swojej działalności. Coraz więcej funkcjonuje firm usługowych i produkcyjnych. Taka tendencja wzrostu i rozwoju MŚP powoduje wzrost zainteresowania tym niedocenianym do tej pory sektorem rynku, stanowiącym około 40% wartości produkcji w Polsce.

W Raporcie Bangemanna [1] opracowanym w 1994 r. dla Rady Europejskiej wytyczono kierunki rozwoju Europy w dziedzinie informatyzacji. Wskazano na drogi, możliwości i zagrożenia budowy europejskiego społeczeństwa globalnej informacji. W społeczeństwie tym większą konkurencyjność w biznesie zapewniają sprawne połączenia informacyjne między klientem a dostawcą. Małym i średniej wielkości przedsiębiorstwom "potrzebne są połączenia o łatwym dostępie do efektywnych, pod względem kosztów, sieci dostarczających informacji na temat produkcji i możliwości rynkowych. Konkurencyjność całego sektora przemysłowego zaostrzy się, jeżeli ich stosunki z dużymi przedsiębiorstwami będą oparte na nowych technologiach. ... Połączenia sieciowe zmniejszają również izolację małych i średniej wielkości przedsiębiorstw w mniej uprzywilejowanych regionach Europy pomagając im podwyższyć jakość ich produktów i znaleźć większe rynki zbytu." [1]

2. Przesłanki tworzenia serwisów informacyjnych dla MŚP

W sektorze MŚP, jednostkowo mało interesującym, wytworzyła się potrzeba dostępu do ogólnie przyjętej informacji gospodarczej. Dostęp ten powinien być jednostkowo tani i dostarczać aktualnych informacji. W obecnych czasach informacja staje się coraz cenniejszym towarem, towarem, który można sprzedać i towarem, który można kupić. Ta firma, która ma dostęp do większej ilości dobrze uporządkowanych informacji, może szybciej podjąć właściwą decyzję taktyczną czy strategiczną i w rezultacie uzyskać większe zyski. Dla MŚP właściwa informacja to często "być albo nie być" dla firmy, dlatego problem dostępu do odpowiednich serwisów informacyjnych jest tak ważny w tym sektorze.

W Polsce istnieje wiele instytucji zorientowanych na poprawę sytuacji społeczno - gospodarczej poszczególnych regionów kraju i dysponujących zasobami informacji z różnych dziedzin. Są to np. izby handlowe, różnorodne fundacje i stowarzyszenia przedsiębiorców, urzędy administracji państwowej. Okresowo wydawane są różnego rodzaju katalogi - panoramy firm, informatory, przewodniki gospodarcze itp. Powstają także wyspecjalizowane w tej dziedzinie przedsiębiorstwa konsultingowe. Niestety, często te zasoby informacyjne są niepełne, nieaktualne, nieupowszechniane i trudno dostępne. Uzyskanie odpowiedniej informacji jest czasochłonne i nie przynosi oczekiwanego efektu. Wyłania się więc konieczność

stworzenia systemów informacyjnych zawierających w miarę pełną, aktualną, łatwo dostępną i tanią informację społeczno-gospodarczą.
Dlaczego problem ten tak ważny jest dla MŚP?

Producentom i dostawcom różnego rodzaju produktów i usług systemy informacyjne są wprost niezbędne dla ich działalności. Takie systemy powstają w dużych, międzynarodowych korporacjach i gromadzą dane o ofercie, danych technicznych, ekonomicznych, dane dotyczące strategii firmy, ... - słowem dane niezbędne pracownikowi do jego efektywnej pracy. Część danych z tych systemów jest również dostępna dla partnerów tych firm. Duże przedsiębiorstwa krajowe są w stanie pozyskać odpowiednie informacje za pomocą własnych służb marketingowych, posiadają dostęp do zasobów komercyjnych sieci komputerowych, są w stanie nawet utrzymywać własne gospodarcze, techniczne, kadrowe, itp. bazy danych.

Pozyskanie odpowiednich informacji przez MŚP nie dysponujące dużymi zasobami finansowymi i ludzkimi wiąże się z przezwyciężeniem następujących problemów:

- **rozproszoność informacji** (brak scentralizowanego serwisu informacyjnego powoduje konieczność szukania informacji w różnych źródłach, zwiększa się prawdopodobieństwo ominięcia cennych informacji);
- **aktualność informacji** (informacje zawarte w wydanych drukiem katalogach z długim cyklem wydawniczym (zbieranie, przygotowanie, druk, dystrybucja) mogą być częściowo nieaktualne ze względu na szybką dynamikę zmian dotyczącą założenia, likwidacji, zmiany zakresu działalności firm - szczególnie MŚP);
- **cena informacji** (w cenę żądanej informacji należy wliczyć koszty jej wyszukania, pozyskania, dostępu, gromadzenia, przetworzenia do określonej postaci - dla odbiorcy informacji lub cenę reklamy dla nadawcy informacji);
- **czas uzyskania informacji** (rozproszoność danych i sposób ich przetwarzania (ręczny, komputerowy) warunkują czas otrzymania informacji w określonej postaci, a zbyt długi czas zmniejsza prawdopodobieństwo, że informacja jest jeszcze aktualna);
- **zasięg informacji** (z pozycji nadawcy i odbiorcy informacji ważny jest zasięg docierania z ofertą gospodarczą w obszar regionu, kraju czy za granicę).

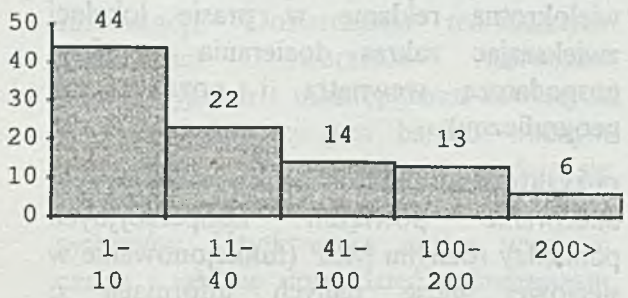
Małe i średnie firmy z powodu wysokich kosztów pozyskiwania informacji często nie mają możliwości dostępu do pełnej informacji, zadowalając się informacjami cząstkowymi zdobytymi z różnych źródeł, a na utrzymanie własnych systemów informacyjnych ich nie stać. Dla tak masowego użytkownika, jakim mogłyby być MŚP, społecznie i finansowo opłacalnym przedsięwzięciem wydaje się więc udostępnianie odpowiednich tanich informacji z bazy danych zlokalizowanej w sieci komputerowej, np. Internet.

3. Sieć komputerowa jako narzędzie pozyskiwania, przesyłania i udostępniania danych

W Polsce sieć komputerową Internet jako narzędzie pozyskiwania, przesyłania i udostępniania danych najszybciej zaczęli wykorzystywać pracownicy i studenci uczelni wyższych. Mając poprzez sieć Internet dostęp do różnych informacyjnych baz danych, niestety w większości tylko zagranicznych, wykorzystywano ich zasoby głównie w celach poznawczych lub naukowych. Duże przedsiębiorstwa przemysłowe i instytucje państwowe podłączone do Internetu mają również możliwość korzystania z potężnych zasobów informacyjnych zgromadzonych w sieciowych bazach danych, o ile one istnieją (Polska) i zawierają potrzebną danemu użytkownikowi informację.

Powstała struktura techniczna sieci komputerowych w Polsce (NASK, Telbank, Kolpak i inne) umożliwia jej użytkownikom dostęp do wielu usług, w tym i do różnorodnych baz danych. W chwili obecnej

rozwój lokalnych, miejskich, rozległych sieci komputerowych w Polsce stwarza szansę dostępu MŚP do zasobów sieciowych. Wśród użytkowników sieci dużą grupę stanowią właśnie MŚP - rys.



Rys. Struktura wielkości przedsiębiorstw podłączonych do sieci Internet w Polsce [2]

Dla MŚP istotne wydają się informacje dotyczące sytuacji społeczno - gospodarczej regionu, w którym funkcjonują oraz regionów, z którymi zamierzają podjąć różnego rodzaju współpracę. Dlatego oprócz możliwości dostępu do sieci o wiele bardziej istotnym problemem wydaje się istnienie odpowiednich regionalnych zasobów danych zawierających aktualne społeczno - gospodarcze informacje.

Niestety, obecnie polski użytkownik sieci, w tym i biznesmen, najczęściej wykorzystuje informacje zawarte w bazach danych krajów Europy Zachodniej i USA. Prościej jest niekiedy uzyskać notowania i prognozy giełdy w Londynie czy Frankfurcie, niż w Warszawie. W wyspecjalizowanych bazach danych szybko można (za odpowiednią opłatą znaleźć dostawców określonego towaru lub usług, partnera ekonomicznego, czy też firmę zainteresowaną odbiorem określonego produktu. Informacje te można zdobyć (nie ruszając się ze swojego biura) lecz tylko dla firm niemieckich, angielskich, francuskich, itp. a nie polskich. Brak jest bowiem sieciowych baz danych o polskich firmach.

Pewnym wyjątkiem jest system informacyjny „KOLIBER" dostępny jest w sieci X.25. Opracowany przez poznańską spółkę EUROSTAR [3] bezpłatnie udostępnia bazy danych z: ofertami firm polskich i zagranicznych, katalogiem polskiego oprogramowania i danymi o polskich firmach. Komercyjne bazy zawierają między innymi informacje o wystawcach i ofertach w Międzynarodowych Targach Poznańskich i o danych adresowych firm (baza: TELEFAX Poland).

4. Problemy budowy i funkcjonowania sieciowych serwisów informacyjnych dla MŚP

Ekonomiczną opłacalność inwestycji budowy sieciowych serwisów informacyjnych można ocenić szacując wielkość rynku ewentualnych przyszłych jego użytkowników. Liczbę firm zainteresowanych reklamą własnej działalności w różnych regionach można oszacować na podstawie zawartości wydawnictw reklamowych typu np. Panorama firm; podobne dane można uzyskać w Regionalnych Izbach Gospodarczych, Izbach Przemysłowo - Handlowych.

Biorąc więc pod uwagę jako przyszłego użytkownika MŚP na podstawie nawet pobieżnej oceny widać, że budowa regionalnych baz danych jest opłacalną ekonomicznie i społecznie inwestycją, wspomagającą rozwój firm, a więc jednocześnie i regionu.

Jakie założenia od strony przyszłego użytkownika powinna spełniać baza danych o regionie przeznaczona głównie dla MŚP?

Dotyczą one następujących kwestii:

1. Informacje zawarte w bazie danych powinny obejmować swoim zakresem całokształt danych o sytuacji społeczno - gospodarczej regionu, być w miarę aktualne i doprowadzone do postaci jaką potrzebuje użytkownik (zestawienie, wykres,...).
2. Informacje powinny być łatwo dostępne w różnych fizycznie miejscach - instytucjach, miastach,... co pozwoli na rozszerzenie zasięgu użytkowników bazy danych, skrócenie czasu uzyskania informacji oraz zwiększenie jakości usług związanych z pozyskaniem lub rozpowszechnianiem informacji.

3. Dostęp do zasobów bazy nie powinien być uwarunkowany przyłączeniem firmy do lokalnej sieci komputerowej gromadzącej zasoby; informacja powinna być możliwa do uzyskania zarówno za pomocą mechanizmów sieciowych jak i poza nimi z wykorzystaniem telefonu, faxu czy drogą listowną.

Te wymagania może spełnić jedynie serwis informacyjny funkcjonujący w sieciowej bazie danych wykorzystujący istniejącą infrastrukturę informatyczną regionu i powiązania jej z otoczeniem.

Istniejące w wielu miastach i regionach miejskie sieci komputerowe mogą być wykorzystane jako narzędzia przesyłania informacji między użytkownikami, partnerami gospodarczymi, natomiast dostęp do sieci Internet pozwoli jeszcze bardziej rozszerzyć zakres serwisu informacyjnego umożliwiając docieranie z ofertą gospodarczą także poza granice kraju.

Koncentracja informacji gospodarczej, społecznej w takim serwisie pozwoli na łatwiejsze wyszukanie informacji i skrócenie czasu jej pozyskania. Automatyzacja prac związanych z wyszukiwaniem żądanych informacji i doprowadzeniem jej do odpowiedniej postaci (np. prognoza, wykres) pozwoli na szybsze i pełniejsze zaspokojenie potrzeb użytkownika.

Na inny jakościowo poziom podniesione zostaną operacje związane z dostępem i przepływem informacji do użytkownika (siedząc przy biurku można otrzymać niezbędne informacje).

Istnienie podobnych serwisów informacyjnych będzie miało nie tylko duże znaczenie gospodarcze aktywizujące sektor MŚP ale i znaczenie społeczne. Informacje gospodarcze rozszerzone o dane o zatrudnieniu, oferowanych miejscach pracy pozwoli realizować politykę społeczną, promować kadry, przeciwdziałać bezrobociu, ustalać potrzeby kształcenia kadr na przyszłe potrzeby regionu (możliwość przewidywania pożądanych kierunków kształcenia na podstawie gromadzonych danych).

Korzyści wynikające z dostępu do sieciowych serwisów informacyjnych dla MŚP są następujące [4]:

- **szeroka reklama własnej działalności** (informacja o firmie zawarta w sieciowej bazie danych znacznie przewyższa wielokrotną reklamę w prasie lokalnej zwiększając zakres docierania z ofertą gospodarczą wewnątrz i poza region geograficzny);
- **pozyskiwanie partnerów do współpracy,** budowanie powiązań kooperacyjnych pomiędzy różnymi MŚP (funkcjonowanie w sieciowej bazie danych informacji o aktualnych ofertach współpracy między firmami, potrzebach w zakresie kooperacji zwiększa szansę znalezienia odpowiedniego partnera, a tym samym prowadzi do rozwoju przedsiębiorczości, rozszerzania zakresu działalności i zwiększenia potencjału ekonomicznego MŚP);
- **pozyskiwanie kadry** (za pośrednictwem sieci MŚP mogłyby pozyskiwać odpowiednich pracowników zgodnie z założonymi wymaganiami);
- **opracowywanie analiz ekonomicznych, prognoz rynkowych, kształtowanie strategii rozwoju firmy.**

Organizacja sieciowych serwisów informacyjnych dla MŚP wymaga rozwiązania następujących problemów:

Wymagania techniczne. Dla realizacji systemów informacyjnych konieczny jest odpowiedni sprzęt komputerowy o dużych pojemnościach pamięci masowych i szybkim czasie dostępu. Sprzęt powinien umożliwiać przekaz danych, dźwięku i obrazu, a informacje te powinny docierać bezpiecznie do wskazanych adresatów. Potrzebne są też odpowiednie aplikacje pozwalające gromadzić i wyszukiwać informacje, jak również dobra sieć teleinformatyczna pozwalająca na dostęp do informacji z wielu miejsc. Sposób świadczenia usług powinien być rozszerzony również dla potencjalnych użytkowników nie podłączonych do sieci

(w okresie przejściowym). Stanowiska udostępniania informacji powinny być wyposażone w modemy i faxy, które mogą być wykorzystane jako narzędzia zapytań i transmisji danych do użytkownika.

Upowszechnienie tej metody zdobywania informacji. Dostarczenie technicznych możliwości tworzenia serwisów informacyjnych i udostępniania zawartych w nich informacji jest bardzo ważnym problemem , ale ważniejszym wydaje się powszechne nauczenie korzystania z takich systemów. Użytkownicy muszą wiedzieć czego i jak się dowiedzieć najmniejszym kosztem. Jednocześnie usługi te powinny być tanie, a jest to możliwe tylko wtedy, gdy są masowe. W różnych sektorach gospodarki odpowiednie instytucje muszą utworzyć centra gromadzenia i udostępniania informacji.

Aktualność danych. Użytkownicy powinni mieć zagwarantowane zbieranie i aktualizowanie danych. Problem aktualizacji danych powinna rozwiązać instytucja obsługująca bazę danych.

Ochrona i bezpieczeństwo danych. W systemach informacyjnych należy zapewnić programowe środki ochrony określonych grup danych przed niepowołanym dostępem, co nie zawsze jest wystarczające. Istotnym problemem staje się wdrożenie zasad postępowania dla różnych grup uprawnionych użytkowników, tak by świadomie lub nieświadomie nikt z nich nie udostępnił innym swoich uprawnień.

Taryfikacja usług. Finansowanie funkcjonowania sieciowych serwisów informacyjnych możliwe jest kilkoma sposobami: przez firmy umieszczające swoje dane w bazie (wówczas proste informacje dostępne byłyby bezpłatnie) lub przez firmy korzystające z informacji (wówczas wpis do bazy mógłby być bezpłatny) albo też koszty funkcjonowania rozłożone byłyby na obie grupy użytkowników.

5. GOSLUB - Komputerowy System Wspomagania Rozwoju Gospodarczego Regionu Lubelskiego

Przykładem sieciowego serwisu informacyjnego dla regionu lubelskiego może być opracowywany przez Zakład Systemów Informatycznych Katedry Zarządzania Politechniki Lubelskiej oraz przez Polską Fundację Ośrodków Wspomagania Rozwoju Gospodarczego "OIC POLAND" Komputerowy System Wspomagania Rozwoju Gospodarczego Regionu Lubelskiego - GOSLUB [5].

Celem systemu jest działanie na rzecz rozwoju regionu poprzez analizowanie, przetwarzanie i przesyłanie informacji dotyczących sytuacji gospodarczej, ekonomicznej i społecznej regionu. System wykorzystywał będzie sieć LUBMAN do przesyłania informacji między użytkownikami w Lublinie a także sieć Internet w celu nawiązywania kontaktów gospodarczych w kraju i współpracy gospodarczej Wschód - Zachód.

GOSLUB gromadził i udostępniał będzie aktualne dane o gospodarce regionu lubelskiego, przeprowadzał będzie analizy danych społeczno - ekonomicznych i wspomagał proces podejmowania decyzji dotyczących rozwoju regionu lubelskiego. Przewidywany krąg użytkowników systemu to przede wszystkim: instytucje administracji państwowej, MŚP, uczelnie, fundacje i stowarzyszenia przedsiębiorców.

Istnienie systemu GOSLUB w regionie wymusi powstanie podobnych systemów współpracujących w innych regionach co przyczyni się do ogólnego rozwoju społeczno - ekonomicznego i integracji regionalnej.

Literatura

1. Europa i społeczeństwo globalnej informacji. Zalecenia dla Rady Europejskiej. Raport Bangemanna. Bruksela, 26.05.1994 (w Materiały 1 Kongresu Informatyki Polskiej, Poznań, 1994).

2. Raport z badania administratorów i użytkowników sieci szkieletowej. RUN (Research & University Network), Warszawa, luty-marzec 1995.
3. Zaperty B.: Serwisy baz danych i katalogi biblioteczne dostępne poprzez NASK. Materiały seminarium Naukowa i Akademicka Sieć Komputerowa. Miedzeszyn, 1994, str. 55-65.
4. Boguciński T., Bojar E., Miłosz E., Smołka J., Miłosz M., Gońka L.: Sieciowa baza danych o regionie GOSLUB - nowa jakość w nawiązywaniu kontaktów gospodarczych. Materiały konferencji: Współpraca Gospodarcza Wschód-Zachód. Politechnika Lubelska, Lublin, 1994, str. 89-93.
5. Wstępna koncepcja Komputerowego Systemu Wspomagania Rozwoju Gospodarczego Regionu Lubelskiego. GOSLUB. OIC Poland, Politechnika Lubelska, Lublin, 1994.

Andrzej Rogowski
Dariusz Koncewicz

Praktyczne aspekty budowy rozproszonej bazy danych

1. Wstęp

Przedstawione doświadczenia powstały w Przedsiębiorstwie Techniki Bankowej **elba** w trakcie prac nad Systemem Kompleksowej Automatyzacji Rachunkowości Bankowej **skarb**. PTB elba specjalizuje się w produkcji oprogramowania dla banków i instytucji finansowych. W połowie 1992 roku rozpoczęto prace nad koncepcją Systemu **skarb**. Aktualnie pracuje pilotowa instalacja w dwóch oddziałach jednego z banków komercyjnych stanowiąc równocześnie poligon weryfikacji przyjętych założeń i zebrania doświadczeń eksploatacyjnych. System jest przewidziany dla dużych banków o znacznym zróżnicowaniu wielkości banku (liczba klientów, strumień transakcji), zróżnicowanej strukturze organizacyjnej banku, zróżnicowanym profilu prowadzonych produktów bankowych i usług. Tak postawione założenia wymagały między innymi rozpoczęcia prac nad rozproszoną bazą danych. W tego typu systemie, w zależności od wymagań banku i doboru środowiska informatycznego, należy rozpatrywać możliwość zmiany skali bazy: od architektury całkowicie scentralizowanej (serwer podstawowy i ewentualny dubler, dla bezpieczeństwa) do architektury rozproszonej, w której znaczna część bazy jest "przybliżona" do oddziałów operacyjnych banku i umieszczona na wielu serwerach, np w ośrodkach regionalnych, a z pozycji centrali jest widziana jako całość.
Budowa systemu o takich założeniach i skali wymagała wypracowania technologii projektowania i kierowania przedsięwzięciem gwarantującej osiągnięcie sukcesu, a w przyszłości możliwość sterowania jakością według wymagań normy ISO-9000 w systemie TQS (Total Quality Systems). Zastosowano narzędzia CASE (Computer Assisted System Engineering). W projektowaniu systemu użyto narzędzia klasy "Upper case", a w prototypowaniu i realizacji systemu "Lower case" generujący oprogramowanie aplikacyjne.
W pierwszej części opracowania przedstawiono ogólne założenia stosowanej metodologii projektowej, a w drugiej części szczegółowe rozważania poświęcone problemom powstającym w budowie bazy rozproszonej.

2. Jak projektujemy model danych

W projektowaniu systemów MIS (ang. Management Information Systems) kluczową rolę odgrywa odpowiednio zaprojektowana i skonstruowana baza danych. We współczesnych systemach jest to najczęściej baza relacyjna. Punktem wyjścia do projektu bazy danych jest analiza zasad funkcjonowania danej organizacji, identyfikacja wymagań klienta, a następnie sporządzenie modelu danych. W technikach strukturalnych podstawowym narzędziem wykorzystywanym do tego celu jest diagram ER - Entity Relationship.

2.1. Model logiczny

Punktem wyjścia są informacje zabrane od przyszłych użytkowników. Na podstawie tych informacji i wiedzy teoretycznej tworzone są kolejne wersje modelu danych, które po konsultacjach z grupami roboczymi przyszłego użytkownika są uzupełniane lub modyfikowane.

Po przetworzeniu zebranych danych powinien powstać kompletny i poprawny model danych na poziomie logicznym, tj. zawierający wszystkie dane jakie zostały zidentyfikowane w analizowanym obszarze funkcjonowania danej organizacji oddający związki zgodne z rzeczywistością.
Model danych na poziomie logicznym posiada następujące cechy

- jest w postaci 3 Normalnej,
- jest niezależny od przyszłej techniki realizacji,
- nie zawiera szczegółów związanych z archiwizacją danych, kontrolą dostępu, itd.,
- nie zawiera żadnych rozwiązań związanych z przepustowością systemu, wielkością przetwarzanych danych, wymaganą szybkością przetwarzania, itd.,
- nie zawiera żadnych uwarunkowań związanych z lokalizacją systemu i jego ewentualnym rozproszeniem.

Jak z tego wynika logiczny model danych ujmuje samą istotę problemu i abstrahuje od jakichkolwiek szczegółów związanych z faktycznym rozwiązaniem zastosowanym w przyszłym systemie.

2.2. Model fizyczny i optymalizacja

Model fizyczny powstaje z modelu logicznego po uwzględnieniu wszystkich aspektów związanych z realizacją systemu. W przypadku bazy relacyjnej nie potrzeba wykonywać prawie żadnych dodatkowych czynności gdyż encje z modelu logicznego odpowiadają niemal wprost tablicom w relacyjnej bazie danych. W przypadku modelu sieciowego lub hierarchicznego niektórych związków nie da się bezpośrednio przenieść z modelu logicznego; model danych trzeba dodatkowo przekształcić.
W następnym kroku uwzględnia się wymagania związane z czasem przetwarzania, czasem odpowiedzi systemu, wielkością przetwarzanych i gromadzonych danych. Zmiany w modelu danych najczęściej polegają na przemieszczaniu danych pomiędzy tablicami, łączeniu tablic lub wydzielaniu części danych do osobnej tablicy, dodaniu zbiorów pośrednich, dublowaniu tablic lub ich części, itp.
W kolejnym kroku należy uwzględnić problem archiwizacji danych, śladu kontrolnego, transferu danych pomiędzy systemami oraz kontroli dostępu do systemu.
Wprowadzenie powyższych zmian do modelu logicznego przekształca ten model i w efekcie otrzymuje się model fizyczny, tj. taki który odpowiada faktycznej realizacji danych w systemie. Model fizyczny z reguły nie zachowuje postaci 3 Normalnej. Jak pokazuje praktyka w niektórych przypadkach nie jest zachowana nawet pierwsza postać Normalna !!.

2.3. Przesłanki rozproszenia bazy

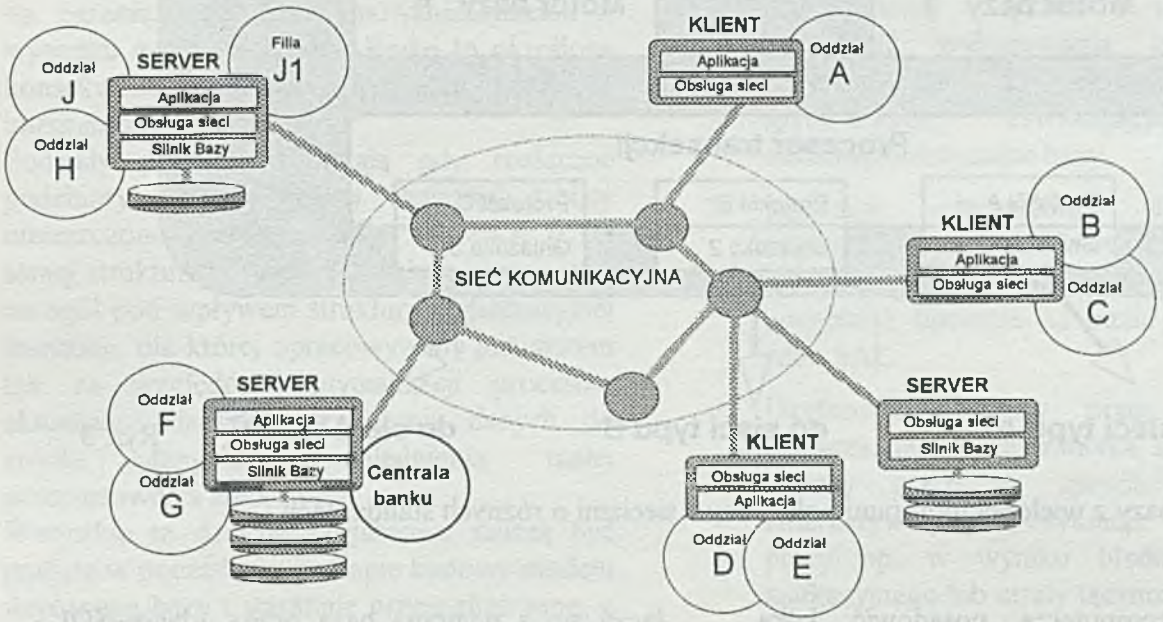
Dotychczasowe rozważania zakładały, że system jest scentralizowany, tzn. baza danych znajduje się w jednym miejscu. Przykładem takich systemów są instalacje jednostanowiskowe na komputerach osobistych, instalacje w sieciach lokalnych lub systemy oparte o duży komputer typu mainframe i terminale.
W przypadku systemów rozległych, rozwiązanie scentralizowane nie zawsze jest możliwe z uwagi na wymagane czasy reakcji systemu, przepustowość kanałów transmisyjnych, wielkość przetwarzanych danych, itd. Często dodatkowym utrudnieniem jest konieczność integracji kilku istniejących podsystemów, zazwyczaj na różnych platformach sprzętowo-programowych.
W takich przypadkach jedynym rozwiązaniem jest zaakceptowanie rozproszenia danych w systemie oraz zastosowanie architektury klient-serwer.

3. Jakie możliwości daje INFORMIX

Rozważany poniżej INFORMIX jest systemem zarządzania relacyjną bazą danych tej samej klasy co SYBASE lub ORACLE. Wszystkie one korzystają z SQL do wykonywania operacji na bazie relacyjnej.
Poniżej zostaną krótko scharakteryzowane środki programowe, oferowane do budowy rozproszonej bazy danych, przez systemy tej klasy na przykładzie INFORMIX'a.

3.1. Koncepcja relacyjnej bazy na wielu serwerach

Rozproszona baza danych w swej idei powinna zapewniać całkowitą niezależność fizycznego rozmieszczenia tablic w różnych komputerach połączonych siecią komunikacyjną od logicznego widzenia bazy danych przez aplikacje oraz od komputerów, na których przetwarzane są programy aplikacyjne.
Model nic nie zakłada o wymaganej przepustowości sieci komputerowej, komputerach udostępniających fizyczne części bazy (serwery o niezbędnej mocy obliczeniowej) jak i o komputerach, na których pracują aplikacje przetwarzające bazę
Na rysunku 1 pokazano przykładowy system dla banku złożonego z wielu oddziałów i centrali. Niektóre oddziały banku pracują z aplikacjami osadzonymi w komputerach "pozbawionych danych" (data less), inne z aplikacjami pracującymi na tych samych komputerach w których znajdują się dane.

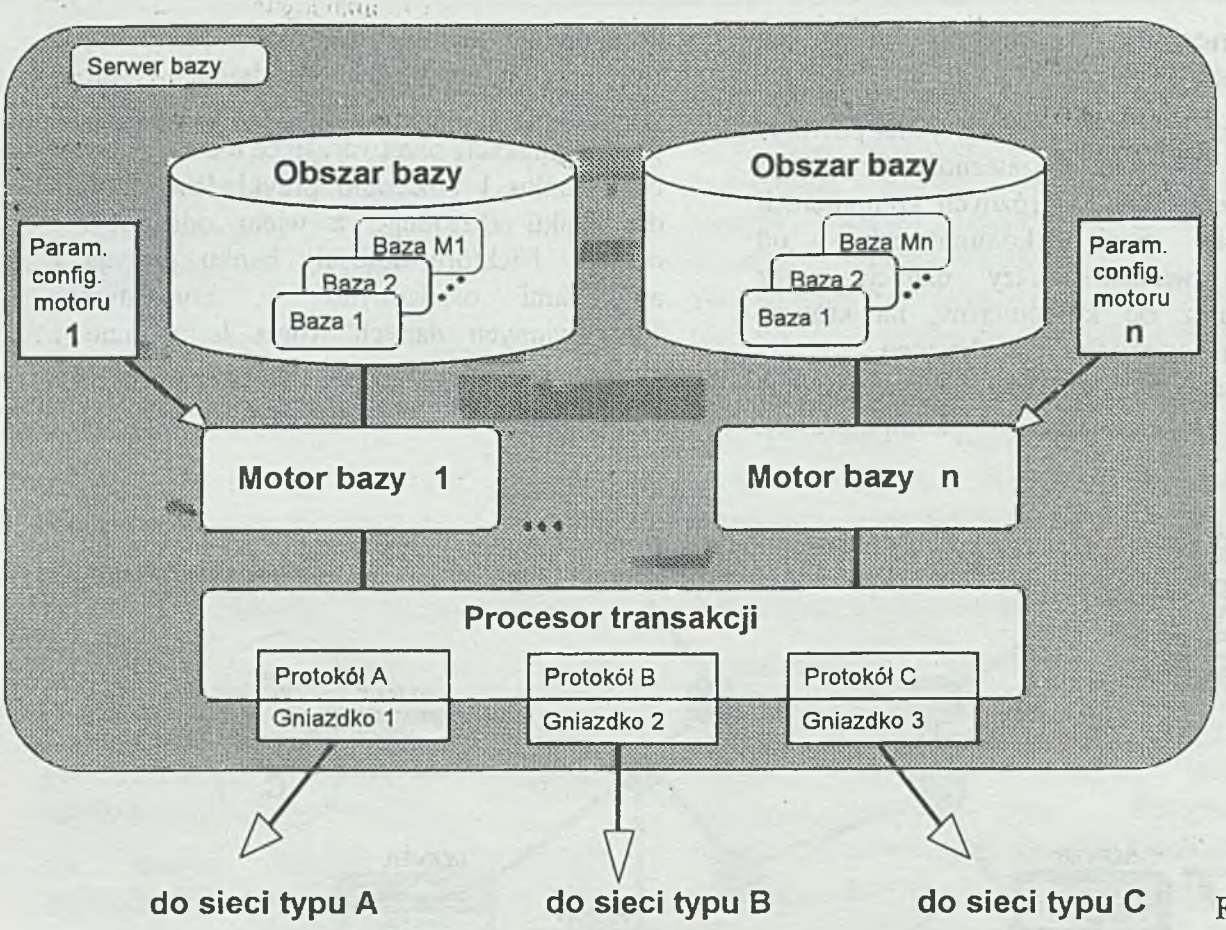


Rys. 1 Przykładowy konfiguracja systemu bankowego z rozproszoną bazą danych

Zasygnalizowana na rysunku 1 architektura pod kątem struktury oprogramowania zapewnia konfiguracje klient-serwer w następujących wariantach:

- aplikacja (napisana np. w 4GL lub C z emedded SQL) lub wiele aplikacji, pracuje na serwerze bazy danych,
- aplikacja (napisana np. w 4GL) lub wiele aplikacji pracuje na komputerze nie będącym serwerem bazy, posiadającym dyski wykorzystywane lokalnie (data less computer) lub pozbawionym dysków (disk less computer); rozwiązanie typu serwer aplikacji
- aplikacja pracuje na stacji roboczej; jest napisana w języku odpowiednim dla danej stacji z zanurzonym SQL; typowa konfiguracja klient serwer.

Oprogramowanie systemu zarządzania relacyjna bazą danych INFORMIX zapewnia konfiguracje klient serwer niezależnie od umieszczenia aplikacji. Motor bazy danych spełniający te wymogi to Informix OnLine, a procesor transakcji (zespół demonów UNIX'owych utrzymujących obsługę transakcji rozproszonych) to Informix Star.
Motor bazy obejmujący procesor zdań SQL jest jednostką logiczną, na ogół częścią oprogramowania (i/lub sprzętu) obsługującą jeden obszar przechowujący tablice bazy danych.



Rys. 3

Serwer bazy z wieloma motorami połączony z sieciami o różnych standardach

Na jednym komputerze posadowić kilka motorów bazy, z których każdy obsługuje niezależnie odrębny obszar bazy, analogicznie jak gdyby znajdowały się one na różnych fizycznie komputerach (rysunek 3). Każdy z motorów bazy pracuje w oparciu o własny zestaw parametrów instalacyjnych dostrajający jego własności do specyfiki przetwarzanego obszaru bazy oraz do własności komputera. Motory przetwarzające transakcję rozproszoną komunikują się pomiędzy sobą za pośrednictwem procesora transakcji (warstwa obsługująca transakcje rozproszone), tj. przez Unix'owe demony Informix Star.

Od strony aplikacji, operacje na rozproszonej bazie danych (RBD), wykonywane są przy pomocy zdań języka SQL. Zdania uformowane przez aplikację, wysyłane są poprzez warstwę komunikacyjną i trafiają do motoru macierzystego wyznaczonego pomocy parametrów środowiska aplikacji. Aplikacja łączy się z wybraną bazą przez zdanie SQL (DATABASE NazwaBazy), deklarując tym samym lokalną bazę, do której będzie przyłączona dana aplikacja. Dalej poprzez definicje synonimów związanych z tą bazą, aplikacja "widzi" wszystkie składające się na nią tablice rozproszonej bazy.

3.2. Podział poziomy i pionowy

Zabiegi projektowe na modelu logicznym zmierzają w kierunku zmian powodujących pojawienie się zagregowanych i powielonych danych, czyli utratę normalizacji danych. Celem jest otrzymanie możliwie małej liczby tablic lepiej dopasowanych do procesów aktualizacji i wymagań konkretnego systemu. Zapadają decyzje o ewentualnych kierunkach podziału bazy na jednostki związane z fizyczną implementacją. Podyktowane są one klasą struktur organizacyjnych instytucji, którym system ma służyć i strukturą samego systemu czyli między innymi wymaganiami administracji

systemem i bazą danych. Istnieją dwie możliwości dokonywania podziałów w bazie danych: tzw. podziały pionowe i podziały poziome.
Podziały pionowe prowadzą do wskazania tablic lub grup tablic, które w całości mogą być umieszczane w jednym obszarze bazy (pod kontrolą jednego motoru bazy). Wybór obszaru jest kwestią optymalizacji. Podklasą takiego podziału jest świadome powielenie pewnych tablic (np. w okrojonej postaci) dla zapewnienia np. ograniczonego funkcjonowania systemu w wypadku awarii łączności. Rodzi to określone konsekwencje związane z utrzymaniem integralności bazy.
Podziały poziome powstają gdy rozłączne podzbiory wierszy jednej logicznej tablicy umieszczone zostają w kilku tablicach o tej samej strukturze. Takie podziały powstają one na ogół pod wpływem struktury organizacyjnej instytucji, dla której opracowywany jest system lub ze względów optymalizacji procesów aktualizacji danych, przybliżenia danych do źródła danych, uwzględnienia małej przepustowości sieci, itp.
Wszystkie te decyzje projektowe muszą być podjęte w początkowym etapie budowy modelu fizycznego bazy i starannie przeanalizowane, a nawet przetestowane w prototypach aplikacji na testowej bazie.

3.3. Dostępny zestaw środków programowych

3.3.1. Jak aplikacja "widzi" bazę danych

Aby zilustrować sposób "widzenia" przez aplikację bazy danych oraz wykonywania na niej operacji, wymieniono poniżej ważniejsze etapy pracy użytkownika z systemem.

1. Użytkownik łączy się z systemem operacyjnym wybranego komputera (login, sign on, etc.). Zostaje zidentyfikowany podając swój identyfikator (identification), następnie zostaje skontrolowana jego wiarogodność (authentication).
2. System operacyjny dostarcza określonego środowiska operacyjnego zdefiniowanego przez administratora systemu (między innymi interfejs sql) i wówczas użytkownik uruchamia wymaganą przez siebie aplikację.
3. Aplikacja, korzystając z przygotowanego środowiska, odwołuje się do określonego środowiskiem motoru bazy kierując do niego zdanie SQL otwierające/określające bazę danych, z którą aplikacja będzie pracować.
4. Po otwarciu bazy staje się dostępna jej definicja. Otwarcie bazy daje aplikacji uruchomionej przez użytkownika możliwość wykonywania operacji na tablicach bazy i na niektórych tablicach systemowych zawierających dane definicyjne dotyczące bazy.
5. Aplikacja odpowiednio do działań użytkownika za pośrednictwem klawiatury lub innych urządzeń wejściowych, podejmuje operacje na bazie za pomocą zdań SAL.
6. Użytkownik kończy pracę. Aplikacja rozłącza się z bazą, zamyka środowisko i kończy pracę w sposób poprawny. Alternatywną jest sytuacja przerwania pracy np. w wyniku błędu programu aplikacyjnego lub utraty łączności.

Wymienione powyżej etapy pracy użytkownika z systemem są niezależne od tego czy aplikacja została uruchomiona na komputerze będącym serwerem, czy komputerem aplikacyjnym (data less computer), czy na stacji roboczej. Różne jedynie są odpowiednie do komputera wykonania modułów warstwy iterfejsu sql, procesora transakcji, itd. związanych ze środowiskiem.

Dla projektanta systemu, kluczową sprawą jest ustalenie możliwości uniezależnienie aplikacji od środowiska w którym aplikacja będzie pracować. Podobnie ważną jest możliwość niezależności od narzędzia w którym określony fragment struktury funkcjonalnej projektu stanowiący daną aplikację zostanie zrealizowany.

3.3.2. Synonimy nazw tablic

Konstrukcja językowa podana poniżej pozwala na wykonanie odwzorowania bazy danych na

faktyczne rozmieszczenie w sieci komputerowej.

PRZYKŁAD

W centrali banku zdefiniowano synonimy w sposób pozwalający zobaczyć tablice zawierające opisy klientów banku. Tablice klienci, na serwerach regionalnych, to klienci zarejestrowani i prowadzeni przez jeden oddział (synonimy, o nazwach klienci1, klienci2, ..., klienciM). Ten zbiór tablic składowych może być widzianych równorzędnie np. na poziomie systemu centrali jako całość, a zdania SQL odwołujące się do nich muszą być łączone operatorem UNION (suma).

CREATE SYNONIM klienci1 FOR baza_oddz@serwer_oddz1: oddz1.klienci_oddzialu

CREATE SYNONIM klienci2 FOR baza_oddz2@serwer2 : oddz1.klienci_oddzialu

CREATE SYNONIM klienci3 FOR baza_oddz3@serwer2 : oddz3.klienci_oddzialu

...

CREATE SYNONIM klienciM FOR baza_oddzN@serwerK: oddzN.klienci_oddzialu

Informacja o zdefiniowanych synonimach w każdym serwerze bazy jest pamiętana w tablicach systemowych. Zachowują one "trwałość" podobną jak inne obiekty (np. tablice, widoki, indeksy itd.) Są one dostępne do momentu wykonania komendy likwidującej dany obiekt (komenda DROP...).

3.3.3. Transakcja w rozproszonej bazie

Istotą transakcji na bazie danych jest takie wykonywanie zmian danych w bazie, aby wszystkie zmiany objęte transakcją zostały wprowadzone do bazy albo, w przypadku niepowodzenia, żadna zmiana bazy nie została wykonana.

Transakcja obejmuje inicjację transakcji (sygnał od aplikacji określający rozpoczęcie sekwencji czynności) treść transakcji (czynności wykonywane na danych w bazie) i zakończenie lub wycofanie transakcji (sygnał od aplikacji określający koniec sekwencji czynności).

W przypadku rozproszonej bazy danych od strony aplikacji postępowanie ma identyczną postać: inicjacja, treść, zakończenie transakcji. Procesy zachodzące w warstwie komunikacyjnej pomiędzy Motorami Bazy są złożone i obejmują dwufazowy protokół zatwierdzenia transakcji (two phase commit). Stanowi on sposób uzgodnienia poprawnego zakończenia czynności z tablicami obsługiwanymi przez różne Motory Bazy, a dopiero po zebraniu tych sygnałów formuje odpowiedź dla aplikacji.

W rozproszonej bazie kluczową sprawą jest uwzględnienie dopuszczalnych opóźnień pojawienia się sygnałów potwierdzenia poprawnego wykonania czynności przez Motory Bazy uczestniczące w częściach pracy na poszczególnych komputerach oraz ewentualne tworzenie procedur wycofania niezakończonej, rozproszonej transakcji.

3.3.4. Operacje w rozproszonej bazie

Decydując się na wybór relacyjnej bazy danych dla realizacji systemu dokonuje się automatycznie niejako wyboru SQL jako języka, w którym się operuje na tej bazie. Obejmuje on następujące klasy operacji: Selekcję (select), Projekcję (Projection), Złączenie (join) oraz operacje mnogościowe: Sumę (Union), Przecięcie (Intersection), Różnicę (Difference). W wyniku dodatkowych kwalifikatorów i własności SQL faktyczna liczba dostępnych odcieni operacji jest znacznie większa. Poniższe rozważania dotyczą operacji w postaci takiej jaką oferuje język SQL.

Upraszczając rozważania można stwierdzić, że operacje na bazie danych wiążą się bądź z:

- szeroko rozumianą aktualizacją, wstawianie nowych zapisów (Create, Insert), zmiana wartości w już istniejących zapisach (Update), kasowanie istniejących zapisów (Delete, Cancel),
- albo z wyszukiwaniem danych z bazy (Inquiring); wiąże się to z formułowaniem zapytań do bazy (Inquiry) celem otrzymania określonych danych sprawozdawczych (operacje kasy czytaj (Read)).

Wymieniony .podział operacji ma istotne znaczenia z punktu widzenia projektowego zaklasyfikowania poszczególnych funkcji systemu. Jest to realizowane z użyciem macierzy CRUD (Create, Read, Update,

Delete) na poziomie projektu funkcji i przekłada się je na odpowiednie konstrukcje SQL realizujące poszczególne operacje na bazie.
Ogólną wytyczną projektową jest taka budowa bazy danych aby operacje szeroko rozumianej aktualizacji CUD (Create, Update, Delete) były wykonywane z udziałem możliwie małej liczby tablic "widzianych" z jednej formatki ekranowej (jedna, dwie tablice), najlepiej znajdujące się w jednym komputerze.
W odniesieniu do zapytań należy rozważać następujące klasy zapytań:

- obejmującą zapytania "sprofilowane", dopasowane do użytkownika, wynikające z zasad funkcjonowania systemu, posiadające minimalną liczbę parametrów określanych ad oc. przez użytkownika,
- stanowiącą sparametryzowane predefiniowane raporty lub formatki zapytaniowe o dużej liczbie parametrów definiowalnych przez użytkownika systemu,
- nie zawierającą żadnych predefiniowalnych własności, które dałoby się wykorzystać w procesie projektowym, np. pytania formułowane przez użytkownika znającego SQL, za pośrednictwem ogólnych narzędzi.

Wszystkie operacje typu "zapytania" są klasy "czytaj" R (Read) co upraszcza rozważania dotyczące integralności bazy i jedynie ma swoje konsekwencje w czasie wykonania i w możliwości ich optymalizacji (odpowiednio do klasy zapytań).
Operacje wiążące dwie lub więcej tablic (w szczególności w różnych komputerach) w przypadku operacji typu złączenie, tworzenia kursora lub tablicy roboczej wymagają starannej optymalizacji projektowej z wykorzystaniem narzędzi oferowanych przez producenta systemu zarządzania bazą. Zasadniczym rozstrzygnięciem projektowym są decyzje, w których sytuacjach polegać na wbudowanym optymalizatorze wybierającym automatycznie komputer, na którym umieszcza wyniki pośrednie i końcowe operacji, a w jakich wypadkach wymuszać "ręcznie" lokalizację tablicy będącej rezultatem operacji.

3.3.5. Constraints, Trigers, SPL

Aby zrealizować system o architekturze rozproszonej należy dysponować oprogramowaniem serwera które umożliwia nie tylko organizację i obsługę bazy danych, ale także pozwala zdefiniować procedury do przetwarzania tych danych. Informix posiada dwa tego rodzaju mechanizmy. Pierwszy, SPL (Stored Procedure Language) jest językiem, który służy do definiowania tzw. **procedur wbudowanych** (procedur składowanych, ang. Stored Procedures). Procedury wbudowane są fizycznie przechowywane na serwerze w postaci półskompilowanej. W momencie ich uruchomienia są ładowane do pamięci operacyjnej serwera i tam wykonywane. W ten sposób przepływ danych można zredukować do minimum. Dodatkowo, redukcja czasu przetwarzania jest związana z tym, że serwer zwykle dysponuje dużo większą mocą obliczeniową od stacji klientowskiej.
Mechanizmem wspomagającym korzystanie z procedur wbudowanych jest mechanizm **wyzwalaczy** (ang. trigers). Są to konstrukcje automatycznie uruchamiające procedury SPL zawsze, gdy na wskazanej tablicy lub kolumnie jest przeprowadzana określona operacja (np. Update). Za pomocą wyzwalaczy można w łatwy sposób zrealizować utrzymywanie w kilku tablicach powielonych danych.
Mechanizmem do kontroli i utrzymania integralności bazy danych jest mechanizm **więzów** (ang. constraints). Więzy są to zadeklarowane warunki nałożone na dane przechowywane w bazie, które muszą być zawsze spełnione po zakończeniu każdej transakcji.
Opisane mechanizmy ułatwiają utrzymanie integralności w bazie danych i w dużym stopniu czynią system mało wrażliwym na błędy oprogramowania.

4. Wnioski

Nie ma ogólnej odpowiedzi czy centralizować, czy wymuszać jej rozproszenie lub, czy tylko dopuszczać rozproszenie. Najwyższy priorytet

w decyzjach projektowych winny mieć względy konserwacji i administrowania systemem w trakcie eksploatacji. W drugiej kolejności złożoność implementacyjna, a w trzeciej ewentualne komplikacje i pracochłonność prac nad projektem.

Rozproszona baza zapewnia:

1. dopasowanie bazy do struktury organizacyjnej i funkcjonalnej, której służyć ma projektowany system informatyczny;
2. "przybliżenie" operacyjnej działalności instytucji do przetwarzanych danych;
3. przybliżenie przetwarzania do użytkownika (serwer aplikacji "bliżej" użytkownika i architektura klient-serwer), co zmniejsza obciążenie sieci rozległej;
4. dopasowanie bazy danych i systemu do struktury zasobów informatycznych bez modyfikacji oprogramowania aplikacyjnego i odwrotnie: restrukturyzację zasobów bez konsekwencji dla aplikacji, zależnie od postawionego celu;
5. lepszą absorbcję skutków awarii, jeśli system posiada pewną nadmiarowość przepustowości sieci, mocy obliczeniowej i zasobów pamięci masowej;
6. niezależność aplikacji od rozwiązań i konfiguracji sprzętowych, a stąd możliwości skalowania systemu;
7. możliwość budowy bazy danych w heterogenicznym sprzętowo środowisku, korzystającym z tego samego systemu zarządzania bazą, a co najmniej operującym przy pomocy SQL.

Cena, którą się płaci za korzyści (często potencjalne, nie wykorzystywane przez projektowany system) jest:

1. rozbudowany system administrowania bazą danych i strukturą komunikacyjną serwisów,
2. kosztowne oprogramowanie systemów zarządzania bazą danych, bardziej złożone, wymagające "mocniejszego" procesora/ procesorów,
3. duże wymagania, a tym samym koszty sieci komunikacyjnej, zarówno inwestycyjne jak i eksploatacyjne w porównaniu z klasycznym OLTP,
4. skomplikowanie konstrucji aplikacji i pracy administratora w zakresie obsługi sytuacji anomalnych i awaryjnych,
5. skomplikowane, wymagające uwzględnienia większej liczby czynników i mechanizmów projektowanie systemu, praktycznie niemożliwe bez narzędzi CASE,

Analizując konsekwencje wprowadzenia rozproszenia bazy, oprócz kwestii zasygnalizowanych w opracowaniu, należy:

1. rozpatrzyć kwestie szyfrowania i reglamentacji dostępu do danych w rozproszonej bazie oraz reglamentację dostępu do funkcji systemu i ich skutki dla bazy danych gdyż kontrola praw dostępu i wykonywanych operacji są znacznie bardziej złożone niż w przypadku bazy centralnej,
2. opracować odpowiednie techniki administrowania bazą, a w szczególności procedury naprawcze po wystąpieniu awarii oraz dostrajanie bazy do przepustowości sieci i mocy obliczeniowych. Właśnie one będą ostatecznie decydowały o efektywności wdrożonego systemu,
3. przeanalizować bardzo starannie wszystkie zagadnienia zasygnalizowane w tej pracy w zestawieniu z modelem logicznym systemu (modelem danych i funkcjonalnym) w zestawieniu z wymaganiami projektowymi.

Wszelkie wątpliwe kwestie nie dające się ocenić metodami analitycznymi najbezpieczniej weryfikować przy użyciu rozwiązań prototypowych, zarówno bazy bazy danych jak i funkcji systemu.

Tomek Kuźmierz
SAS Institute, Warszawa

Implementacja Data Warehouse w rozproszonym środowisku informatycznym - metodologia SAS Institute

Współczesne środowiska przetwarzania danych charakteryzują się dużą złożonością i różnorodnością. Co więcej, ich dalszy rozwój nie powoduje uproszczenia drogi przepływu informacji w rozbudowanych systemach. Taki stan rzeczy sprawia, że w wielu firmach na świecie pojawiają się głosy o koniecznej realizacji rozwiązania dla takiego status quo. Jedną z odpowiedzi na wyraźnie zaznaczające się tendencje jest: *Data Warehouse*.

Skąd pochodzi pomysł budowania Data Warehouse?

U podstaw idei Data Warehouse leży istniejący w wielu instytucjach i firmach podział na warstwy: decyzyjną i wykonawczą. Informatyczna infrastruktura rzadko jednakże odzwierciedla taki naturalny podział i koncentruje się zwykle wokół operacyjnych baz danych zawierających stan i przebieg codziennej pracy służb wykonawczych w instytucji. Usprawnienie polega więc na wydzieleniu (bądź dobudowaniu w przypadku braku takowego) składnika analitycznego w informatycznej infrastrukturze instytucji.

Data Warehouse - co to takiego?

Data Warehouse jest reakcją środowiska informatyków na coraz powszechniej spotykane w przetwarzaniu danych problemy takie jak:

- czasowo i przestrzennie rozproszone źródła danych;
- dostęp do danych z różnorodnych aplikacji, pracujących w heterogenicznych środowiskach operacyjnych i baz danych;
- różnorodne formaty tematycznie jednolitych danych ograniczające dostęp do nich dla nie-informatyków;
- ograniczenia standardowych systemów transakcyjnych baz danych w dostarczaniu zagregowanej informacji dla celów decyzyjnych;
- powszechne stosowanie różnych platform sprzętowych w różnych działach instytucji.

Proponowane rozwiązanie umożliwia hierarchiczno-tematyczną organizację informacji, przechowywanej na różnorodnych poziomach agregacji.

Data Warehouse - podstawowe cechy

Rozwiązania typu Data Warehouse wyróżniają się następującymi czterema cechami:

- **orientacja tematyczna**: umiejscowienie danych (poszczególne komputery w sieci, rozmieszczenie w odpowiednio dostępnych tablicach, etc.) jest generowane przez ich tematykę, przez co ułatwiony jest dostęp do danych dla osób nie będących profesjonalnymi informatykami i zyskują one na czytelności;
- **integralność**: osiągana przez standaryzację, system zarządzający (oraz jego zarządca-człowiek) dba o spójność i zgodność danych oraz ich formatów zgodnie z przyjętymi w obrębie całego Data Warehouse. Dane napływające z różnorodnych źródeł są w miarę potrzeby ujednolicane i przekształcane na etapie zasilania bazy informacyjnej Data Warehouse;

- **wielowersyjność czasowa**: dane w bazie przechowywane są w wielu wersjach czasowych a kolejne zestawy danych nie są nadpisywane nad poprzednimi ale dopisywane w kolejnej warstwie czasowej. Umożliwia to śledzenie zachodzących zjawisk w funkcji czasu na podstawie analizy odpowiednich szeregów czasowych;
- **nieulotność**: raz zapisane dane do Data Warehouse nie podlegają dalszym zmianom. Uznaje się ich poprawność w momencie zasilania informacyjnej bazy danych a jej użytkownicy mogą jedynie czytać (a nie dopisywać, wykreślać czy też modyfikować) zawarte w niej informacje. Nie jest to transakcyjna baza do rejestracji bieżących wydarzeń, ale zbiór wybranych informacji zagregowanych w zdefiniowany sposób na podstawie wszystkich operacji w instytucji.

Operacyjne systemy baz danych a systemy informacyjne

Tradycyjne, operacyjne systemy baz danych (takie jak np.: system rezerwacji biletów w firmie turystycznej, księga główna działu księgowości w praktycznie każdej instytucji, system rozliczeń z klientami) zbierają dane i wykonują ogromne ilości prostych transakcji. Chociaż są one niezwykle istotne dla prowadzenia firmy, aplikacje tego typu są systemami statycznymi, organizują i realizują jedynie rutynowe czynności a przepływ danych - nie informacji - jest rezultatem ich działania. Dane te, mają często niewielką przydatność dla decydentów, którzy potrzebują raczej zsyntetyzowanych informacji pochodzących z ogromnych ilości danych, analizowanych następnie w celu wykrywania trendów i możliwości rynkowych. Umożliwia to podejmowanie celnych i skutecznych decyzji mających podstawowe znaczenie dla dobra firmy.

Systemy informacyjne, z drugiej strony, dostarczają informacji tym, którzy potrzebują jej najbardziej, poprzez umożliwienie wygenerowania mniejszej ilości, za to bardzo skomplikowanych zapytań do bazy danych. Aplikacje tego typu są systemami dynamicznymi, które w przeciwieństwie do systemów statycznych automatyzujących czynności rutynowe, wspomagają kreatywność ich użytkownika. Przykładem zapytań tego typu mogą być następujące: „Które z naszych usług przynoszą nam największe zyski?", „Kim są nasi najlepsi klienci?", „Na których dostawcach możemy polegać najbardziej?". Aby dostarczyć odpowiednie rozwiązania dla - coraz częściej pojawiających się ze strony dyrektorów i ich doradców czy współpracowników - wymagań tego typu w instytucji powinna zostać zaimplementowana nowa technologia informacyjna, która - zarządzana i realizowana za pomocą właściwych narzędzi i w odpowiedni sposób - pozwoli dostosowywać się firmom i instytucjom do ciągle zmieniających się warunków rynkowych, dając jednocześnie wymierną przewagę nad istniejącą konkurencją. W takich warunkach, odpowiednie inwestycje w informacyjne bazy danych z całą pewnością z zyskiem zwrócą poniesione na ich wybudowanie koszty w każdej instytucji zmagającej się z realiami dzisiejszych, wymagających, rynków.

Rozproszona architektura informacyjnej bazy danych Data Warehouse

Lokalizacja bazy operacyjnej na komputerze o dużej mocy obliczeniowej jest niewątpliwie uzasadniona. Jednak zapewnienie szybkiego wielodostępu do niej wymaga z reguły oparcia się dodatkowo na sieci lokalnej i większej ilości terminali lub stacji roboczych. Taka rozproszona architektura nie tylko ułatwia dostęp do danych, ale i ma, jak się okazuje, także inne zalety. Rozbudowa aplikacji analitycznych nie wpływa ujemnie na wydajność głównego komputera, gdyż to nie on obciążony jest kosztem ich wykonania, lecz stacje robocze. Poza tym pozwala na większą elastyczność i wprowadzanie zmian w wewnętrznej strategii informacyjnej i metodach syntezy danych w zależności od zmieniających się potrzeb.

Pełne wykorzystanie zalet rozproszonego środowiska możliwe jest tylko wtedy, gdy narzędzia użyte do budowy zbiorczej bazy danych przewidują pracę w trybie *klient-serwer*. Baza zbiorcza organizowana w środowisku rozproszonym pozwala określić struktury informacji potrzebnej na różnych szczeblach, w różnych departamentach tematycznych i oddziałach terytorialnych.

W dużych instytucjach stosuje się centralną bazę zbiorczą na głównym serwerze, odpowiednie podzbiory informacji na serwerach departamentalnych, a najbardziej zagregowane informacje udostępniane są kadrze kierowniczej na ich komputerach personalnych. Odpowiednia struktura przepływu informacji jest gwarancją szybkości dostępu do właściwej informacji.

Zbiorcza baza danych Data Warehouse w Systemie SAS

Zbiorczą bazę danych można stworzyć całkowicie w Systemie SAS, począwszy od procedur niskiego poziomu realizujących dostęp do baz operacyjnych, a skończywszy na kształtowaniu logiki systemu informacyjnego i jego graficznego interfejsu. Unika się przy tym problemu łączenia i dopasowywania różnych narzędzi różnych producentów.

Namacalnym argumentem za użyciem Systemu SAS mogą być wyniki analizy porównawczej przeprowadzonej w firmie Dun & Bradstreet. Kluczowe znaczenie dla Dun & Bradstreet miała szybkość zasilenia bazy zbiorczej wybranymi danymi z baz operacyjnych. Testy przeprowadzone na bazach Sybase, Informix i SAS wykazały, że System SAS wykonał to zadanie w czasie 17-krotnie krótszym niż konkurenci. Przy dużej bazie zbiorczej oznaczało to kilkudniowe obciążenie systemów zasilaniem bazy zbiorczej w transakcyjnym systemie bazy danych lub kilkugodzinne przy wykorzystaniu nietransakcyjnej bazy SAS. Tak drastyczna różnica efektywności spowodowała wybór bazy SAS.

System SAS oferuje swoim użytkownikom skuteczną metodologię przekształcania surowych danych operacyjnych w informacje służące zarządzaniu oraz mechanizmy uznawane obecnie za kluczowe w syntezie danych:

- **Łatwość dostępu do różnorodnych, rozproszonych źródeł danych** (DB/2, Oracle, SQL/DS, Adabas, Informix, Ingres, AS/400) także przechowywanych w dawniejszych częściach informacyjnego środowiska firmy, eliminuje problem komunikacji miedzy bazą zbiorczą a bazami operacyjnymi oraz pozwala zachować bez zmian działające i sprawdzone w nich rozwiązania. SAS świetnie radzi sobie z indeksowaniem i kompresją zbiorów. Potrafi lokalnie lub zdalnie ewaluować zapytania w języku SQL. Dzięki wzbogaceniu o ODBC (*Open Database Connectivity*), jego zbiory stają się bezpośrednio i szybko dostępne zewnętrznym aplikacjom. Cechy te kwalifikują System SAS do grona systemów otwartych.
- **Bogate możliwości funkcjonalne** obejmują m.in. duży wybór funkcji do różnorodnego raportowania, analiz statystycznych i ekonometrycznych oraz przyjaznej graficznej komunikacji z użytkownikiem.
- **Praca w rozproszonym środowisku i trybie klient-serwer** daje pełną swobodę wyboru optymalnych rozwiązań sprzętowych od PC po różne warianty połączeń sieciowych.
- **Całkowita przenoszalność w obrębie platform sprzętowych i systemów operacyjnych** pozwala uniknąć kosztów związanych z adaptacją oprogramowania do nowego środowiska pracy.

Kompletne rozwiązanie zbiorczej bazy danych obejmuje organizację, zarządzanie i eksploatację. Przedstawimy w skrócie odpowiadające im cechy Systemu SAS, dzięki którym jest on właściwym rozwiązaniem zarówno dla informatyków w fazach organizacji i zarządzania, jak i dla użytkowników w fazie eksploatacji:

- **Organizacja**:
  - ◊ czerpanie odpowiednich danych z systemów baz operacyjnych oraz meta baza w module SAS/EIS;
  - ◊ różnorodne struktury reprezentacji informacji;
  - ◊ relacyjne, hierarchiczne i wielowymiarowe, wbudowana baza SAS;
  - ◊ różne poziomy agregacji dostosowane do różnych potrzeb;
  - ◊ pełna automatyzacja zasilania i agregacji;
  - ◊ prototypowanie, obiektowa budowa aplikacji.
- **Zarządzanie**:
  - ◊ dostęp do wszystkich systemów baz operacyjnych z ich różnorodnymi danymi;
  - ◊ szerokie możliwości przekształcania danych jak ujednolicanie formatów, usuwanie części danych, sprawdzanie poprawności, a więc wykluczenie zarządzania przez kopiowanie;
  - ◊ dane operacyjne na różnych poziomach;
  - ◊ łączenie różnorodnych struktur;
  - ◊ dostęp do danych zewnętrznych jak struktury hierarchiczne, niestandardowe źródła elektroniczne itp.
- **Eksploatacja**:
  - ◊ optymalizacja metod dostępu do danych - 4GL, SQL, interfejsy okienkowe i w języku naturalnym;
  - ◊ narzędzia raportowania i prezentacji danych;
  - ◊ stymulacja kreatywności użytkowników końcowych: dla analityków i ekspertów: narzędzia analityczne, wspomagania podejmowania decyzji; dla decydentów: systemy informowania kierownictwa (EIS).

Firma SAS Institute jest postrzegana jako lider na polu analizy i przetwarzania informacji, a szereg już zrealizowanych i aktualnie realizowanych przy pomocy Systemu SAS zbiorczych baz danych potwierdza tę pozycję.

Werdykt niezależnych odbiorców i konsultantów

Poniżej zamieszczone są jedne z wielu opinii wieloletnich użytkowników Systemu SAS oraz firm zajmujących się niezależną od jakiegokolwiek producenta (w szczególności SAS Institute) analizą rynku oprogramowania i komputerów:

- „System SAS posiada bardzo naturalne dla Data Warehouse rozwiązania a klienci mają pełne zaufanie do implementacji rozwiązań informacyjnych za pomocą ich narzędzi. Widzę, że produkty SAS Institute wypełniają pustkę pomiędzy dyrektorami a Data Warehouse poprzez wyciąganie informacji z baz danych, ich analizę i prezentację osobom podejmującym decyzje" - Bill Inmon, konsultant w dziedzinie informatyki, „ojciec" koncepcji Data Warehouse.
- „System SAS uzyskał reputację w dziedzinie dostarczania informacji, której nie da się porównać z innymi dostawcami (oryginał: *„which is second to none"*) - audyt technologiczny Gartner Group, 1994.
- „(O systemie SAS) Kompletne rozwiązanie dla wszelkich wymagań w dziedzinie dostarczania informacji" - komentarz użytkownika zamieszczony w biuletynie Ibex firmy Xephon, 1995.
- „Wraz ze swą obsługą danych, usługami komputerowymi oraz narzędziami typu *front-end* i *back-end* System SAS w efektywny sposób wspomaga budową systemów Data Warehouse dla celów wspomagania podejmowania decyzji oraz budowania systemów informowania kierownictwa w technologii klient/serwer" - opracowanie pt. „The SAS System", Datapro International, 1995.

SAS Institute, producent Systemu SAS w Polsce

Producentem Systemu SAS jest amerykańska firma SAS Institute, która w zestawieniach światowych plasuje się na ósmym miejscu pośród niezależnych producentów oprogramowania w świecie (wg. *Software Magazine*). SAS Institute powstał w 1976 roku i od początku rozwijał jednorodne produkty, wszystkie podporządkowane idei, celom i regułom Systemu SAS, służące przekształcaniu danych w informacje.

Warszawski oddział SAS Institute udziela wszelkich informacji o funkcjonalności i możliwościach wykorzystania Systemu SAS. Dysponuje kadrą polskich fachowców z wieloletnim doświadczeniem praktycznym w używaniu oprogramowania SAS, którzy prowadzą szkolenia, wspomagają wdrażanie Systemu SAS i tworzenie pierwszych aplikacji w SAS'ie. Prowadzone w Polsce i po polsku szkolenia obejmują pełen zakres funkcjonalności Systemu SAS i zostały przygotowane lokalnie dla potrzeb użytkowników polskich.

Aktualnie w Polsce System SAS ma kilka tysięcy użytkowników. Są to głównie banki (NBP, BRE, PKO S.A., ...), instytucje sektora publicznego (Ministerstwo Rolnictwa, Ośrodki Doradztwa Rolniczego, ...), instytuty naukowe (Medycyny Pracy, Metali Nieżelaznych, ...), uczelnie wyższe (wszystkie Akademie Medyczne i wszystkie Akademie Rolnicze w całym kraju, uniwersytety i politechniki). W coraz większym stopniu produkty SAS Institute są akceptowane również w polskim przemyśle, czego dowodem jest m. in. instalacja systemu w Centralnym Ośrodku Informatyki Górnictwa w Katowicach.

Oprogramowanie Systemu SAS jest licencjonowane przez SAS Institute wraz z pełną obsługą serwisową. Obsługa oprogramowania obejmuje dostarczanie wszystkich nowych wersji, rozszerzeń i uaktualnień Systemu SAS wraz z dokumentacją oraz pomoc techniczną w zakresie instalacji i eksploatacji oprogramowania przez cały okres licencji bez dodatkowych opłat. Ze względu na specyfikę i zaawansowanie produktu, SAS Institute nie oferuje możliwości zakupu oprogramowania bez jego obsługi. Poza tym, SAS Institute oferuje pełen zakres szkoleń w swoim Centrum Szkoleniowym w Warszawie.

Użytkownikom zainteresowanym uzyskaniem wsparcia w zakresie analizy potrzeb i budowy systemów wspomagania podejmowania decyzji oraz tworzenia różnorodnych aplikacji zapewniamy możliwość współpracy z partnerami SAS Institute. Status Quality Partner SAS Institute otrzymują firmy, które specjalizują się w świadczeniu usług programistycznych i analitycznych w Systemie SAS, stale udoskonalają swoje umiejętności oraz mają za sobą konkretne wdrożenia w Systemie SAS. SAS Institute rekomenduje wybranych partnerów, a użytkownik sam decyduje z którym będzie współpracować i w jakim zakresie.

Włodzimierz Adamski
WSK PZL MIELEC Zakład Lotniczy

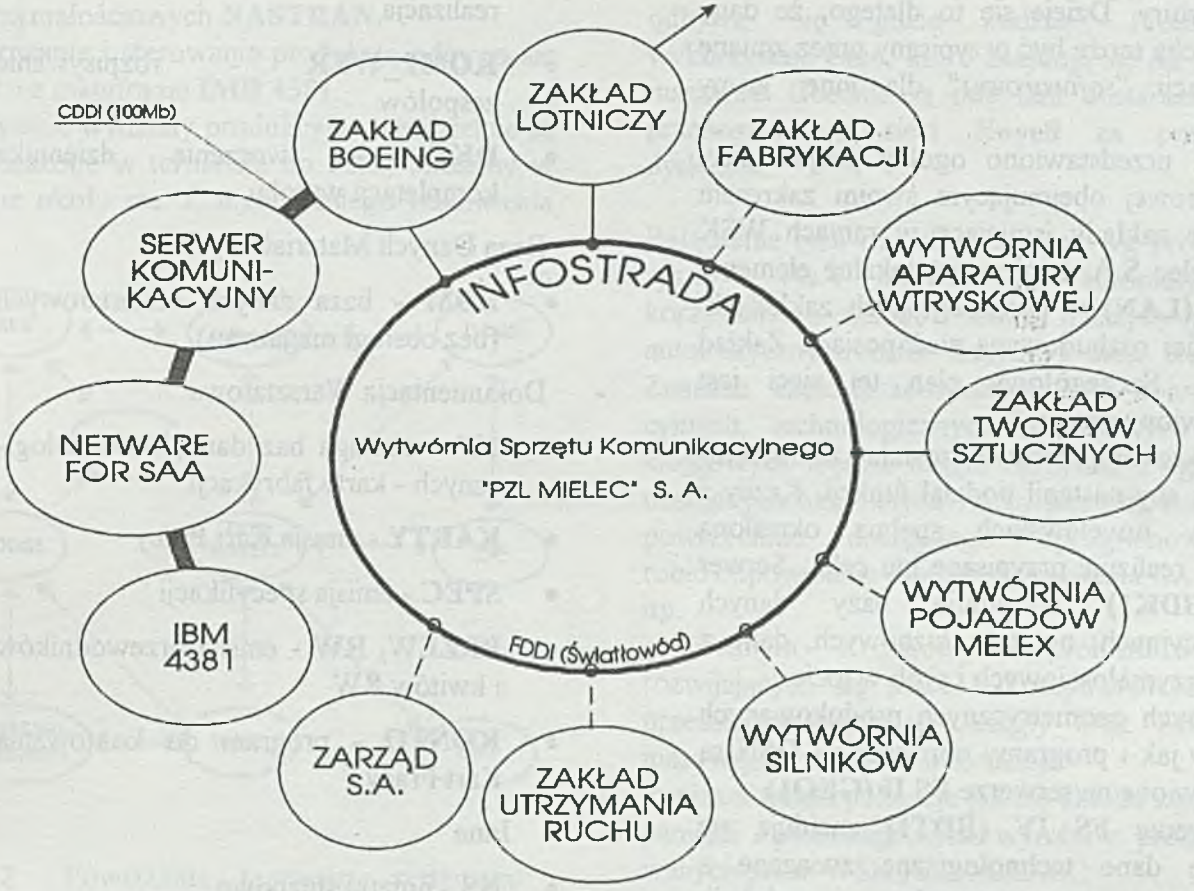
System informatyczny w rzeczywistych warunkach istniejącego przedsiębiorstwa

Upowszechnienie się systemu rynkowego na wszystkich kontynentach oraz globalna rewolucja informatyczna, to dwa procesy cywilizacyjne, które utworzą obraz i oblicze kapitalizmu XXI wieku. Zarządzanie przedsiębiorcze, bezwzględny dyktat konsumenta, nowa zamożna klasa średnia oraz wyzwania edukacyjne, to główne cechy gospodarki przyszłości.

Prosta próba nadążania za rosnącymi wymaganiami rynku prowadzi do złożonej, nieefektywnej struktury przedsiębiorstwa. Główną przyczyną wzrostu złożoności struktury jest coraz ostrzejsza konkurencja wynikająca z umocnienia się globalnego charakteru rynku. Spełnienie wymagań klientów powoduje wzrost i uruchomienie nowych coraz bardziej złożonych modeli wyrobów.

Obecnie na świecie obserwuje się zwiększenie udziału produkcji zróżnicowanej asortymentowo, co jest spowodowane indywidualizacją wymagań użytkowników wyrobów do warunków ich eksploatacji. Ponad 70 % czasu potrzebnego do uruchomienia nowego modelu zajmuje jego **zaprojektowanie oraz przygotowanie procesu produkcji.** Obecnie w przedsiębiorstwach znacznie wzrosła rola działalności koordynacyjnej - 3/4 pracowników nie przetwarza materiałów, lecz informację a 4/5 wszystkich czynności jest tylko pośrednio związanych z produkcją. Na podstawie tych prostych danych widać jak ważną sprawą dla pracowników jest swobodny dostęp do potrzebnych informacji. Zapewniają to systemy informatyczne. Dzięki technologiom informatycznym możliwe jest zupełnie nowe rozwiązanie przedsiębiorstwa, jako zintegrowanego systemu produkcyjnego. Już dzisiaj istnieją na świecie przedsiębiorstwa, których wartość oprogramowania , sprzętu komputerowego i bazy danych przekracza 50% wartości całego majątku trwałego firmy.
Jedną z najważniejszych cech informatyzacji przedsiębiorstwa jest to, że w stosunku do tradycyjnych struktur organizacyjnych gdzie rozmywa się odpowiedzialność udostępnienie pracownikowi odpowiedniej informacji w sieci komputerowej powoduje, że pracownik zamiast oczekiwać na decyzje zwierzchnika, staje się i czuje upoważniony (i jednocześnie odpowiedzialny) do szybkiego podejmowania decyzji.

Czynnikiem decydującym o efektywności i niezawodności pracy sieci jest komunikacja pomiędzy sieciami lokalnymi. Istnieje kilka metod realizacji takich połączeń. Najprostszy jest *bridging* wykorzystywany głównie do łączenia segmentów małych sieci lokalnych. Obecnie najczęściej stosowaną metodą komunikacji międzysieciowej jest *routing*, chociaż ostatnio coraz większe znaczenie zyskuje również *switching*, w szczególności do budowy szybkich połączeń. Mankamentem sieci budowanych z wykorzystaniem *routerów* jest fakt iż grupy robocze użytkowników są przywiązane do fizycznej infrastruktury sieci. Kiedy następują zmiany w organizacji instytucji powoduje to modyfikację sprzętu, inne połączenia kablowe i zmiany adresacji urządzeń w sieci. Znacznie tańszą i bardziej efektywną niż *routing* dając jednocześnie możliwość realizacji wirtualnych połączeń, uniezależniając w ten sposób logiczną i fizyczną strukturę sieci jest metoda *switching*.



PROJEKT SIECI KOMPUTEROWEJ WSK "PZL MIELEC" S. A.

Aby sprostować coraz większemu zapotrzebowaniu na przepustowość w sieciach komputerowych wielu projektantów rozważa wykorzystanie nowych technologii. Podstawowe nadzieje wiąże się z technologią **ATM** (Asynchronous Transfer Mode), która obiecuje bardzo duże szybkości przesyłania danych, obrazu i dźwięku. Drugą podstawową, coraz częściej wykorzystywaną technologią, jest technologia **FDDI** lub **CDDI**. Trzecia to tzw. **LAN** *switching*, który umożliwia komunikację między dwoma urządzeniami sieci komputerowej z przepustowością standardu (np. **Ethernet**). Urządzenia komunikujące się ze sobą nie dzielą przepustowości medium z innymi, zestawiając między sobą połączenia na potrzeby własnej komunikacji.

Wykorzystując technologię **ATM** do realizacji rdzenia sieci, technikę **LAN** *switching* do organizacji sieci lokalnych oraz technologię *routingu*, mamy możliwość budowy sieci komputerowych o nowych własnościach dotyczących przepustowości, realizacji grup roboczych i technologii *klient-serwer* oraz zarządzania i ochrony sieci. W takim środowisku pojawia się możliwość realizacji tzw. ***"sieci wirtualnych"*** (**Virtual LAN**).

Przypomnijmy, że obecnie użytkownik sieci należącej do określonej grupy roboczej podłączony jest do tego samego koncentratora co inni przedstawiciele grupy (lub tego samego portu *routera*). Każdy koncentrator lub pojedynczy **LAN** umożliwiają komunikację w obrębie grupy roboczej. Wszelkie zmiany w takiej strukturze wymagają zmian okablowania (np. krosowanie połączeń dla okablowania strukturalnego) oraz zmian w adresacji sieci. Dla sieci wirtualnych zmiany logiczne sieci nie

pociągają za sobą konieczności zmian fizycznej infrastruktury. Dzieje się to dlatego, że dany port switcha może być przypisany przez zmianę konfiguracji *"softwarowej"* dla innej grupy roboczej.

Na rys.1 przedstawiono ogólny projekt sieci komputerowej obejmującym swoim zakresem wszystkie zakłady istniejące w ramach WSK PZL Mielec S.A. Istnieją już lokalne elementy tej sieci **(LAN)** w poszczególnych zakładach. Najbardziej rozbudowaną sieć posiada Zakład Lotniczy. Szczegółowy plan tej sieci jest przedstawiony na rys. 3.

Sieć mikrokomputerowa została zaprojektowana tak aby nastąpił podział funkcji. Każdy z serwerów novelowskich spełnia określoną funkcję i realizuje przypisane mu cele . Serwer **FSI (BDK1)** utrzymuje bazy danych konstrukcyjnych, notatek służbowych, dane z prób wytrzymałościowych i prób w locie.

Bazę danych geometrycznych produkowanych wyrobów jak i programy obróbcze na **OSN** są umiejscowione na serwerze **FS II (GEO1)**.

Na serwerze **FS IV (BDT1)** znajdują się wszystkie dane technologiczne związane z technicznym przygotowaniem produkcji, dziennikami kompletacji rozdziału robót, emisji dokumentacji warsztatowej, bazami danych materiałowych itp. Są one obsługiwane przez odpowiedni system.

System ten umożliwia prowadzenie rozdziału robót dla dowolnego produktu firmy. Każdy wyrób może posiadać dowolną ilość kompletacji i wersji i nie ma to wpływu na zachowanie się programów, wymaga jednak uzupełnienia zbiorów tzw. słowników z kompletacjami i wersjami, które są w każdym katalogu dla dowolnego wyrobu.

Przykładowy spis programów wchodzących w skład systemu

1. Programy podstawowe.

- Baza Danych Konstrukcyjnych i Technologicznych
  - **RR** - obsługa bazy danych i rozdział robót
  - **KZ** - tworzenie kart zmian i ich realizacja
  - **KOMP_WER** - rozpisywanie zespołów
  - **DKW** - tworzenie dziennika kompletacji wyrobu
- Baza Danych Materiałowych
  - **BDM** - baza danych materiałowych (bez obsługi magazynu)
- Dokumentacja Warsztatowa
  - **KF** - obsługa baz danych technologicznych - karty fabrykacji
  - **KARTY** - emisja Kart Pracy
  - **SPEC** - emisja specyfikacji
  - **PRZEW, RW** - emisja przewodników i kwitów RW
  - **KONTO** - program do kontowania Kart Pracy
- Inne
  - **NS** - notatki służbowe
  - **PRZYRZ** - program do ewidencji przyrządów
  - **TECH** - technologie jednorazowe

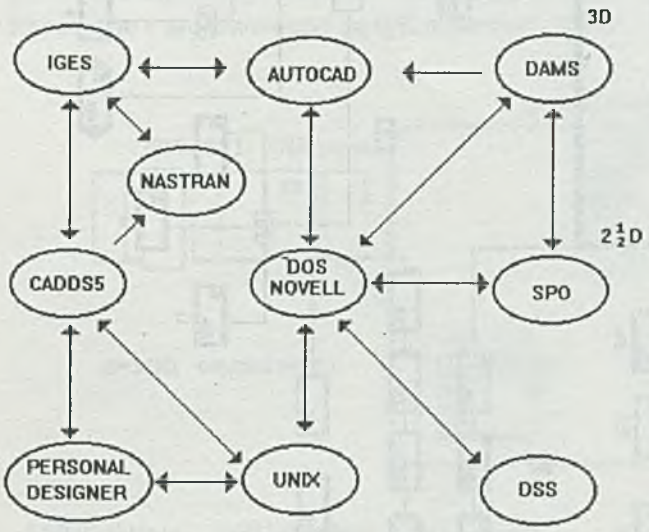
2. Programy pomocnicze.
   - **DS** - drukowanie specyfikacji konstrukc.
   - **ELE_WERS** - wykaz elementów
   - **IL_WYR** - oblicz. ilości części na wyrób
   - **SEEK** - program użytkowy do porządkowania baz
   - **PS** - poprawianie struktury baz dla wszystkich wyrobów

   i inne.

Dane administracyjno-kadrowe, majątkowe, środki trwałe znajdują się na serwerze **FSIII (FK1).** Z kolei sytem finansowo-księgowy pracuje na niezależnej sieci UNIX wykorzystując oraclowską bazę danych.

Niezależnie od już połączonych czterech serwerów Novellowskich w tej samej sieci są podłączone dwie stacje graficzne **SUN SPARC 10**.

a na drugiej system do obliczeń wytrzymałościowych **NASTRAN.**
Planowanie i sterowanie produkcją odbywa się poprzez mainframe **IMB 4381**.
Wszystkie wydziały produkcyjne, rozdzielnie są wyposażone w terminale do dużej maszyny w liczbie około stu. Z tego prostego zestawienia



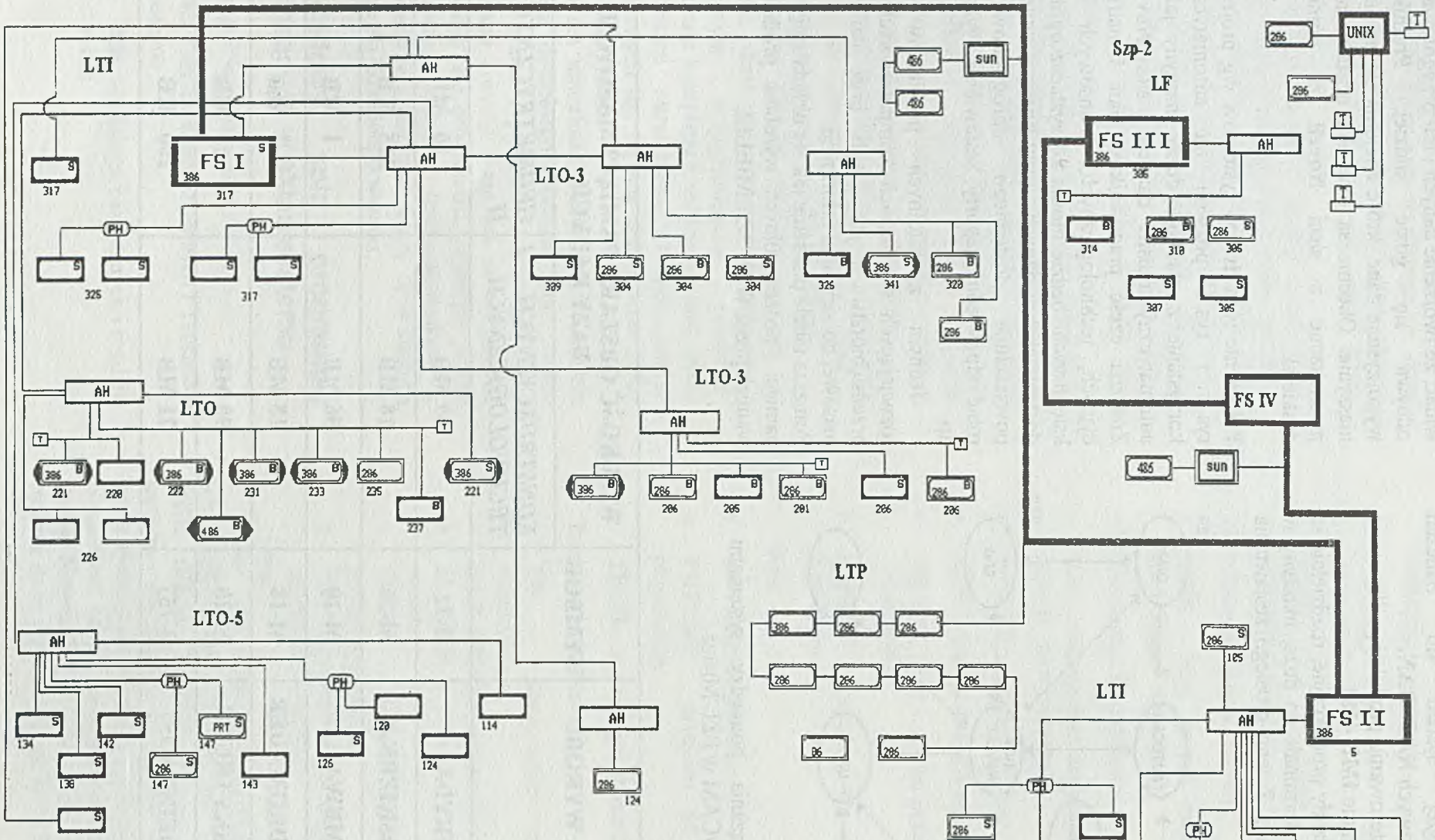
Rys.2 Powiązania pomiędzy systemami CAD/CAM w PZL Mielec

widać, że tworzenie danych i ich przygotowanie odbywa się gdzie indziej. Produkcja wykorzystuje dane, które znajdują się na dużej maszynie. Obecnie są one tam dostarczane i przenoszone z sieci **Novell** za pomocą dyskietek.

Połączenie tych dwóch systemów wg projektu rys.1 i rys.4 pozwoli na automatyczne korzystanie z zasobów dużej maszyny jak i automatyczny transfer danych z sieci **Novell.** Znaczna część pracowników biur konstrukcyjnych, technologicznych, finansowych czy księgowych będzie mogła korzystać z ogólnie dostępnych baz danych i przy pomocy taniego powszechnie dostępnego oprogramowania robić odpowiednie raporty, zestawienia, wykazy itp.

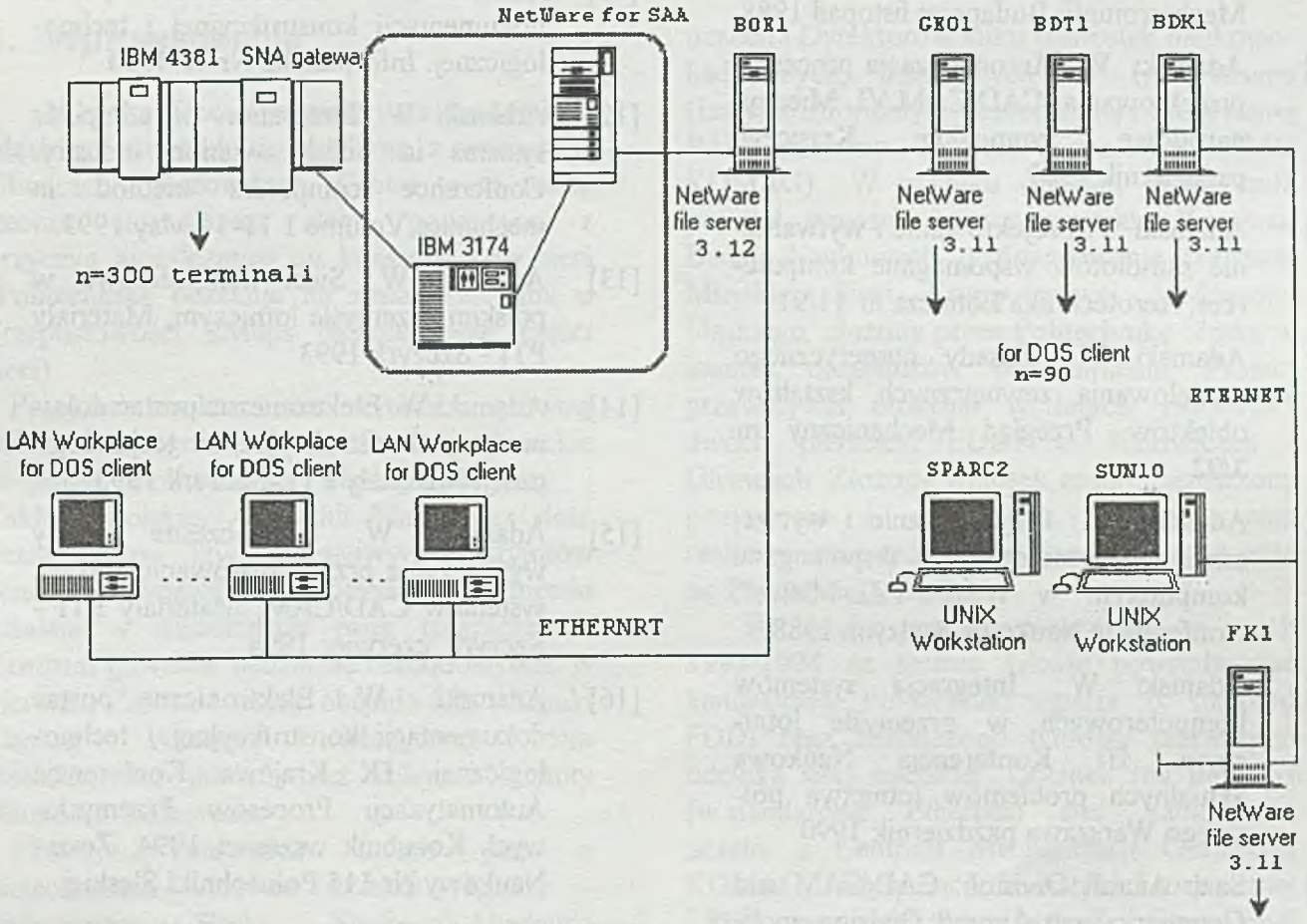
Jednym z istotnych problemów w rozwijających się sieciach komputerowych w przedsiębiorstwach jest ciągły brak pamięci masowej do zapisania informacji.
Poniższa tabela pokazuje jak się układa zajętość pamięci poszczególnych wyrobów produkowanych przez WSK PZL MIELEC

LP	NAZWA WYROBU	SYMBOL	WIELKOŚĆ OBSZARU PAMIĘCI MASOWEJ BAZY DANYCH	
			KONSTRUKCYJNYCH I TECHNOLOGICZNYCH	*GEOMETRYCZNYCH*
1	Samolot **IRYDA**	**I-22**	**59 MB**	**10 MB**
2	Samolot **ISKIERKA**	**M-26**	**18 MB**	**325 KB**
3	Samolot **MEWA**	**M-20**	**36 MB**	**1 MB**
4	Samolot **DROMADER**	**M-18**	**15 MB**	**1 MB**
5	Samolot **SKYTRUCK**	**M-28**	**98 MB**	**13 MB**
6	Zepół **BOEINGA 757**	**B-757**	**21 MB**	**100 KB**



RYS.3 SIEĆ MIKROKOMPUTEROWA NOVELL
WSK PZL MIELEC

Widać wyraźnie, że bazy danych dotyczące specyfikacji konstrukcyjnej i bazy danych technologicznych zajmują znacznie więcej miejsca niż bazy związane z numerycznym kształtem danego wyrobu. Wynika to szczególnie ze specjalnego podejścia przy gromadzeniu bazy danych geometrycznych. Pamiętane i archiwowane są tylko powierzchnie źródłowe, pozostałe dane wynikowe z tych powierzchni źródłowych są danymi "*chwilowymi*" czasowo przebywającymi w komputerze. Można powiedzieć, że im szybciej rozwijają się urządzenia związane z zapisywaniem danych tym szybciej użytkownicy je zapełniają.



Rys.4 Połączenie IBM 4381 z siecią Novell

Rozwój technologii informatycznej jest przyczyną powstania coraz to nowych teorii i koncepcji w bazach danych. Przedsiębiorstwa nieustannie dążą do zwiększenia funkcjonalności baz danych stąd ich naturalna tendencja do tworzenia baz danych tam gdzie one powstają. W zakładzie lotniczym PZL Mielec każdy z serwerów ma swoje określone zadania i pracownicy realizujące te zadania są do tych serwerów jakby przypisani na *stałe*. Nie znaczy to oczywiście, że nie mogą skorzystać z innych zasobów sieci (pod warunkiem, że posiadają odpowiednie uprawnienia)

Zastosowanie nowoczesnych technologii informatycznych powoduje wzrost produkcji przynajmniej o 25 %, dlatego popyt na te technologie nieustannie rośnie mimo wysokich kosztów ich wprowadzenia i przekształcenia przedsiębiorstw. Komputer dzisiaj to już nie zabawka a potężne narzędzie, które uwielokrotnia efekty pracy, a przez to zyski, czyli w efekcie dobrobyt pracowników.

Literatura

[1] Ralph J.Mayer: One answer to the problems of CAD database exchange. Byte no. VI/1987

[2] Adamski W., Rybak Cz., Gancarz T.: "PZL-Mielec" Vallalatnal folyo szerkesztesies technologiai munkak automati zalasa. Konferencja Naukowa Mechatroninfo Budapeszt listopad 1988.

[3] Adamski W.: Automatyzacja procesów projektowania CAD/CAM.VI Miedzynarodowe Sympozjum Krasiczyn październik 1989.

[4] Adamski W. Projektowanie i wytwarzanie samolotów wspomagane komputerem Aerotechnika Lotnicza nr 11/91

[5] Adamski W.: Zasady numerycznego modelowania zewnętrznych kształtów obiektów. Przegląd Mechaniczny nr 3/92

[6] Adamski W.: Projektowanie i wytwarzanie samolotów wspomagane komputerem w WSK "PZL-Mielec". Konferencja Naukowa Rydzyna 1988

[7] Adamski W.: Integracja systemów komputerowych w przemyśle lotniczym. III Konferencja Naukowa aktualnych problemów lotnictwa polskiego Warszawa październik 1990

[8] Saab Aicraft Division: CAD/CAM and Geometry Saab Aircraft Division application of advanced technology II 1991.

[9] Adamski W.: Integracja systemów komputerowych w polskim przemyśle lotniczym.VII Krajowa Konferencja Automatyzacji Procesów Przemysłowych Kozubnik wrzesień 1992, Zeszyt Naukowy Nr 110 Politechniki Śląskiej.

[10] Adamski W.: Stosowane systemy CAD/CAM w polskim przemyśle lotniczym. XV Miedzynarodowe Sympozjum Naukowe Zielona Góra IV 1993

[11] Adamski W. Elektroniczna postać dokumentacji konstrukcyjnej i technologicznej. Informatyka Nr 10/1993

[12] Adamski W. Integration of computer systems in polish aviation industry. Conference computers method in mechanics Volume 1 11-14 May 1993

[13] Adamski W. Sieci komputerowe w polskim przemyśle lotniczym, Materiały PTI - Szczyrk 1993

[14] Adamski W. Elektroniczna postać dokumentacji konstrukcyjnej i technologicznej, Materiały PTI -Szczyrk 1993

[15] Adamski W. Nowoczesne formy wytwarzania przy zastosowaniu dużych systemów CAD/CAM, Materiały PTI - Szczyrk, czerwiec 1994

[16] Adamski W.: Elektroniczna postać dokumentacji konstrukcyjnej i technologicznej. IX Krajowa Konferencja Automatyzacji Procesów Przemysłowych Kozubnik wrzesień 1994, Zeszyt Naukowy Nr 115 Politechniki Śląskiej.

[17] Adamski W.: Systemy CAD/CAM w PZL, CadCamForum 4/94

Grzegorz Dymek, Krzysztof R. Nałęcki
Centrum Komputerowe Politechniki Śląskiej

Infrastruktura i usługi miejskiej sieci komputerowej środowiska Śląskiego

1. Wprowadzenie

Miejska Sieć Komputerowa środowiska Śląskiego ma strukturę bipolarną, z centrami w Gliwicach i Katowicach. Centra te są wciąż jeszcze stosunkowo słabo powiązane z przyczyn niezależnych od konstruktorów sieci (Politechnika oczekuje na zestawienie linii o przepustowości 2Mbps łączącej obie części sieci).

Powyższa struktura sieci wynika w głównej mierze z topografii regionu. Centrum gliwickie obejmuje Politechnikę Śląską, Instytuty i Zakłady Polskiej Akademii Nauk oraz dość liczną grupę tzw. branżowych Instytutów Naukowo-Badawczych. Wszystkie te jednostki działają w dziedzinach nauk technicznych. Centrum gliwickie będzie się rozbudowywać w kierunku Zabrza i wtedy obejmie filie i Kliniki Śląskiej Akademii Medycznej tam zlokalizowane jak również kolejne Instytuty Naukowo-Badawcze.

Centrum katowickie oparte jest o nietechniczne uczelnie wyższe regionu - Uniwersytet Śląski, Śląską Akademię Medyczną, Akademię Ekonomiczną, Akademią Muzyczną i Akademię Wychowania Fizycznego. Obejmuje instytucje naukowo-badawcze z terenu Katowic i Sosnowca, gdzie zlokalizowane są ponadto wydziały Uniwersytetu oraz filie Akademii Medycznej. Centrum to rozbudowywać się będzie właśnie w kierunku Sosnowca.

2. Historia budowy sieci.

Późną jesienią 1992 - z inicjatywy Politechniki Śląskiej - zawiązało się nieformalne porozumienie Rektorów śląskich wyższych uczelni i Dyrektorów kilku jednostek naukowo-badawczych, działających na tym terenie (Instytut Informatyki Teoretycznej i Stosowanej PAN, Główny Instytut Górnictwa, CMG KOMAG). W wyniku działań tego ciała powstał wniosek inwestycyjny do Komitetu Badań Naukowych o finansowanie realizacji Miejskiej Sieci Komputerowej środowiska Śląskiego, złożony przez Politechnikę Śląską w imieniu uczestników porozumienia. Projekt przewidywał ułożenie w latach 1993-1995 dwóch pierścieni FDDI w Katowicach i Gliwicach. Złożony wniosek został rozpatrzony pozytywnie i rozpoczęto pierwszy etap realizacji inwestycji, której koordynatorem stała się Politechnika Śląska.

W wyniku prac inwestycyjnych w latach 1993-1994 na terenie Gliwic powstała sieć kampusowa Politechniki oparta o standard FDDI oraz zakończono budowę pierwszego odcinka sieci miejskiej. Odcinek ten połączył (w standardzie Ethernet) sieć kampusową uczelni z Centrum Mechanizacji Górnictwa KOMAG, Instytutem Informatyki Teoretycznej i Stosowanej, Instytutem Inżynierii Chemicznej PAN[1] oraz międzymiastową centralą telefoniczną TP S.A. dzięki czemu możliwe stało się korzystanie z traktów cyfrowych łączących Gliwice z Katowicami. Dla zapewnienia dostępu do sieci pozostałym jednostkom z terenu Gliwic, zakupiono szesnastoportowy serwer komunikacyjny oraz modemy o szybkości 19200 bps. Inwestycja ta pozwoliła na uzyskanie dostępu do sieci (przez linie komutowane lub dzierżawione)

[1] Doprowadzenie światłowodu do tego punktu pozwoli przyłączyć jednostki zlokalizowane w pobliżu: Instytut Metali Nieżelaznych, Instytut Chemii Nieorganicznej oraz Zakład Karbochemii PAN.

Instytutowi Metalurgii Żelaza, Zakładowi Karbochemii PAN, Instytutowi Chemii Nieorganicznej, Instytutowi Onkologii, Instytutowi Metali Nieżelaznych. Kolosalne zapotrzebowanie ze strony jednostek na dostęp do sieci potwierdziło konieczność zapewnienia im bezpośredniego dostępu do sieci MAN przez linie światłowodowe i szybkie protokoły transmisji (FDDI, Ethernet 10Base-FL).

Reasumując w efekcie przeprowadzonych w roku 1994 inwestycji przyłączono do sieci światłowodowej 4 jednostki (Politechnika Śląska, Centrum Mechanizacji Górnictwa KOMAG, Instytut Informatyki Teoretycznej i Stosowanej PAN, Instytut Inżynierii Chemicznej PAN) zaś pozostałym jednostkom zapewniono tymczasowo możliwość korzystania z sieci poprzez linie telefoniczne. Niezależnie od tego Politechnika Śląska z własnych środków inwestycyjnych zakupiła i uruchomiła serwer sieciowy dużej mocy firmy Hewlett-Packard 9000/755 pracujący pod kontrolą systemu operacyjnego Unix. W okresie pierwszych 3 miesięcy eksploatacji tego komputera liczba zarejestrowanych w nim użytkowników przekroczyła 1000.

Prace nad siecią katowicką w latach 1993-1994 również koncentrowały się na utworzeniu sieci kampusowej Uniwersytetu Śląskiego oraz połączeniu jej z Głównym Instytutem Górnictwa oraz filiami Uniwersytetu w rejonie Placu Sejmu Śląskiego. Węzeł tam zlokalizowany stanowić będzie punkt wyjścia do dalszej rozbudowy sieci o jednostki położone w tym regionie (Biblioteka Śląska, Akademia Ekonomiczna, filia Politechniki Śląskiej, Akademia Muzyczna, Akademia Medyczna). Z kolei węzeł w Głównym Instytucie Górnictwa stanowić będzie punkt wyjścia do rozbudowy w kierunku północnym (Instytut Systemów Sterowania, EMAG). Sieć kampusowa Uniwersytetu Śląskiego oparta jest o standard Ethernet, zaś połączenia z Głównym Instytutem Górnictwa oraz filiami UŚ oparte zostały o FDDI.

Podobnie jak w Gliwicach mimo braku rozbudowanej sieci miejskiej z dostępu do Internetu korzystały (poprzez linie komutowane i dzierżawione) takie jednostki jak: Uniwersytet Śląski, Politechnika Śląska, Akademia Ekonomiczna, Instytut Ekologii Terenów Uprzemysłowionych, Główny Instytut Górnictwa, Centrum Elektryfikacji i Automatyzacji Górnictwa EMAG, Instytut Podstaw Inżynierii Środowiska PAN, Zakład Fizyki Ciała Stałego PAN, Centrum Chemii Polimerów PAN. Dla efektywniejszej obsługi tych użytkowników uruchomiono na Uniwersytecie Śląskim serwer sieciowy SUN SPARCserver 1000, do którego dostęp odbywa się przez dwa serwery komunikacyjne.

Reasumując sieć MAN w Katowicach łączy Uniwersytet Śląski wraz z jego filiami z Głównym Instytutem Górnictwa oraz otwiera możliwość dołączenia kolejnych jednostek w roku bieżącym.

3. Stan obecny

W trakcie realizacji prac, poprzez Radę Użytkowników akces do sieci miejskiej zgłosiły kolejne jednostki z terenu Śląska, w efekcie czego lista potencjalnych użytkowników sieci wzrosła do prawie 40 obejmując dodatkowo miasta Zabrze i Sosnowiec. Przyłączenie tych jednostek do sieci miejskiej realizowane będzie w 1995 i ewentualnie 1996 roku. W chwili obecnej lista Rady Użytkowników środowiska Śląskiego zawiera następujące jednostki akademickie i badawczo-naukowe:
Instytut Informatyki Teoretycznej i Stosowanej PAN, Instytut Onkologii, Instytut Chemii Nieorganicznej, Centrum Mechanizacji Górnictwa KOMAG, Instytut Metalurgii Żelaza, Instytut Metali Nieżelaznych, Politechnika Śląska, Instytut Materiałów Ogniotrwałych, Instytut Inżynierii Chemicznej PAN, Instytut Spawalnictwa, Zakład Karbochemii PAN, Uniwersytet Śląski, Akademia Ekonomiczna, Śląska Akademia Medyczna, Akademia Sztuk Pięknych w Krakowie - Filia w Katowicach, Instytut Ekologii Terenów Uprzemysłowionych, Główny Instytut Górnictwa, Instytut Systemów Sterowania, COBR Przemysłu Izolacji Budowlanej, Centrum Elektryfkacji i Automatyzacji Górnictwa, OBR Budownictwa Węglowego, OBR Przem. Urz. Klim.-Went. i Odp. "BAROWENT", Biblioteka Śląska, Wojewódzka Biblioteka Publiczna,

Pedagogiczna Biblioteka Wojewódzka, Muzeum Śląskie, Instytut Gospodarki Odpadami, Akademia Muzyczna, Akademia Wychownia Fizycznego, Instytut Medycyny Pracy i Zdrowia Środowiskowego, Instytut Techniki i Aparatury Medycznej, Instytut Podstaw Inżynierii Środowiska PAN, Zakład Fizyki Ciała Stałego PAN, Fundacja Rozwoju Kardiochirurgii, Centrum Chemii Polimerów PAN, Instytut Chemicznej Przeróbki Węgla.

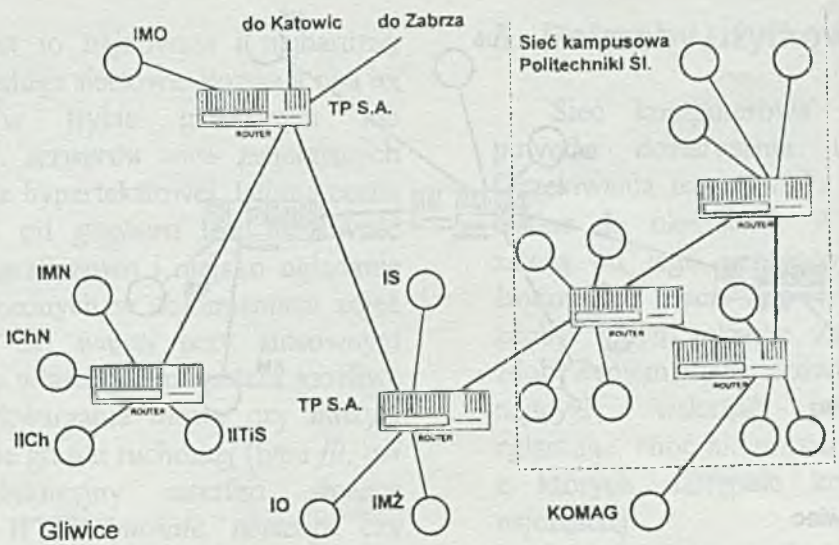
Z listy tej jedynie 6 jednostek włączonych jest bezpośrednio do sieci miejskiej, reszta zaś może uzyskać dostęp do sieci jedynie poprzez linie komutowane i dzierżawione. Jak z tego wynika kolejka oczekujących na włączenie do sieci miejskiej jest duża zaś lista jednostek żądających szybkiego i niezawodnego przyłącza do sieci ulega jak na razie wydłużeniu a nie skróceniu. Plany rozbudowy sieci w roku bieżącym przedstawiają rysunki na następnych stronach. Należy oczekiwać, że rok 1995 będzie pod tym względem przełomowy w efekcie czego uda się zrealizować przedstawione poniżej zamierzenia. Ponieważ w dalszym ciągu nie będzie możliwe zapewnienie połączenia wszystkim jednostką w regionie, przez to zadanie inwestycyjne budowy sieci będzie kontynuowane również w roku 1996. Podstawowym celem w roku bieżącym będzie przyłączenie do sieci miejskiej jak największej ilości jednostek, nawet kosztem zastosowania tańszych i mniej wydajnych standardów transmisji (np. Ethernet zamiast FDDI))

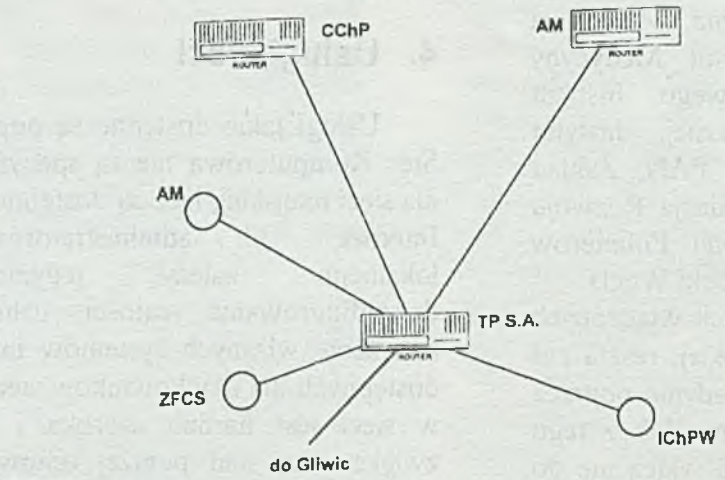
4. Usługi sieci

Usługi jakie dostępne są poprzez Miejską Sieć Komputerową nie są specyficzne jedynie dla sieci miejskiej, lecz są dostępne w całej sieci Internet. Od administratorów szczebla lokalnego zależy jedynie sposób dokonfigurowania całości usług czy też tworzenie własnych systemów informacyjnych dostępnych dla użytkowników sieci. Lista usług w sieci jest bardzo szeroka i wciąż ulega zwiększeniu, stąd poniżej omówione zostaną tylko te najbardziej powszechne i najczęściej wykorzystywane.

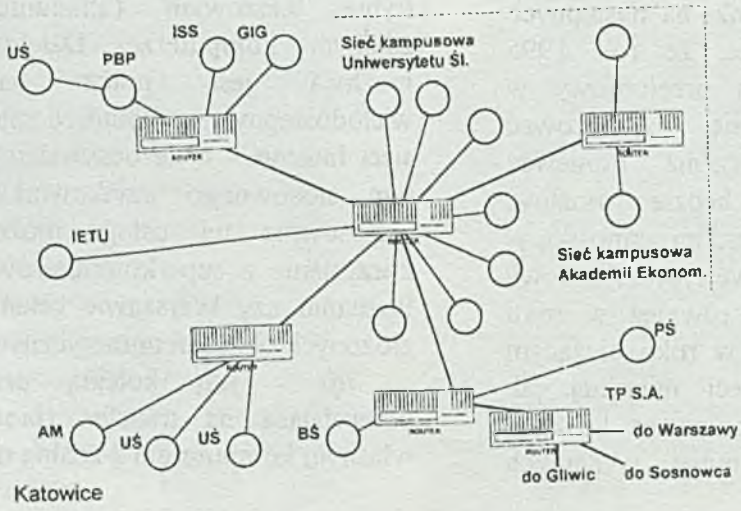
telnet - jest najbardziej elementarną aplikacją usługową, pozwalającą na pracę w trybie tekstowym (alfanumerycznym) na zdalnym komputerze. Dzięki tej aplikacji możliwa jest praca na dowolnym wielodostępnym komputerze znajdującym się w sieci Internet - o ile oczywiście posiada się na nim stosownego użytkownika. Przykładem zastosownia tej usługi może być zdalne korzystanie z superkomputerów w Krakowie, Poznaniu czy Warszawie celem dokonywania złożonych obliczeń numerycznych.

ftp - jest kolejną prostą aplikacją pozwalającą na transfer zbiorów pomiędzy własnym komputerem a zdalną maszyną.

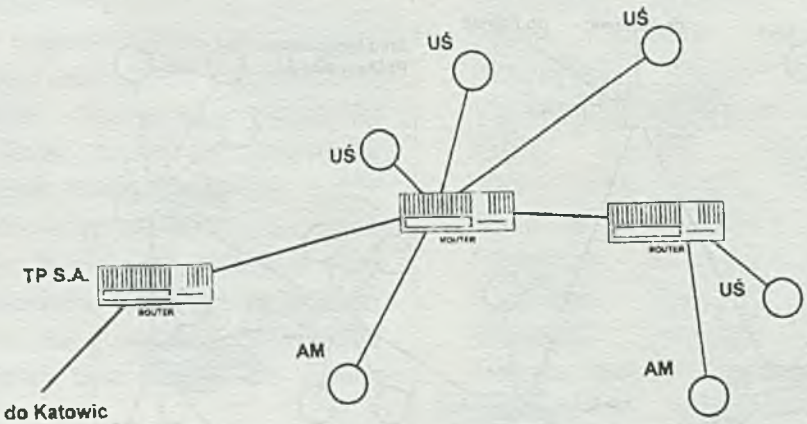




Zabrze



Katowice



Sosnowiec

Szeroki wachlarz opcji pozwala na transfer zarówno pojedynczych zbiorów jak i całych grup, zarówno w trybie interakcyjnym jak i wsadowym. Istotną cechą jest możliwość korzystania z tzw. użytkownika *anonymous* dzięki czemu możliwy jest transfer zbiorów z/do komputera, na którym nie posiada się własnego użytkownika.

mail - jest najczęściej wykorzystywaną aplikacją pozwalającą na przesyłanie pomiędzy użytkownikami sieci korespondencji. W rzeczywistości - w zależności od rodzaju systemu operacyjnego - mamy tu do czynienia z szeregiem różnych aplikacji poczty elektronicznej (*elm, pmail, z-mail*) bazujących na wspólnym protokole transmisji poczty *SMTP*. Wprowadzenie dodatkowych standardów wymiany danych takich jak *MIME* pozwala obecnie na transmisję nie tylko danych tekstowych ale również dźwięku czy grafiki, które mogą zostać odtworzone przez odbiorcę.

gopher - jest to szeroko rozpowszechniona aplikacja pozwalająca na dostęp (w trybie tekstowym) do komputerów, na których udostępniono tzw. *serwery gopher*, które są niczym innym jak prostymi bazami danych o strukturze drzewiastej. Usługa ta pozwala na przeglądanie oraz ściąganie danych (teksty, grafika, dźwięk) bez konieczności posiadania użytkownika na zdalnym serwerze. Informacje instalowane na *serwerach gopher* dotyczą zazwyczaj najczęściej zadawanych przez użytkowników pytań i żądanych do transmisji zbiorów.

WWW - jest to najnowsza i najbardziej rozwijająca się usługa sieciowa. Pozwala ona na przeglądanie (w trybie graficznym lub tekstowym) tzw. *serwerów www* zwierających dane o strukturze hypertekstowej. Istotną cechą różniącą *WWW* od *gophera* jest możliwość pracy w trybie graficznym i niejako oglądanie on-line zamieszczonych w dokumentach zdjęć czy rysunków. Co więcej przy stosownym skonfigurowaniu własnego komputera możliwe jest również odtwarzanie mowy czy muzyki oraz przeglądanie grafiki ruchomej (typu *fli, avi* czy *mpg*). Intuicyjny interfejs służący przeszukiwaniu *WWW* (*mosaic, netscape* czy też *lynx* w trybie tekstowym) powodują, że usługa ta cieszy się rosnącym powodzeniem, co skłania z kolei firmy do uruchamiania własnych *serwerów www* zawierających najnowsze dane o profilu firmy, oferowanych produktach czy też zamierzeniach inwestycyjnych.

irc - jest to niejako sieciowy odpowiednik CB-radio z tym, że poszczególne kanały przyporządkowane są do określonych tematów dyskusyjnych. Aplikacja pozwala na wymianę zdań w trybie on-line z grupą użytkowników włączonych do sieci na tym samym kanale.

listy dyskusyjne - jest to elektroniczny odpowiednik klubów korespondencyjnych gdzie można wypowiedzieć się na określony temat. Listy dystrybuowane są automatycznie (poprzez tzw. *listserver*) pomiędzy użytkowników znajdujących się na liście subskrypcyjnej danej listy dyskusyjnej.

Charakterystyczną cechą ostatnich miesięcy jest pojawianie się sieciowych aplikacji multimedialnych (takich jak mosaic czy netscape) pozwalających na pełne wykorzystanie możliwości komputerów PC czy stacji roboczych. Pojawiają się między innymi aplikacje pozwalające na bezpośrednią rozmowę poprzez sieć (netphone) czy też organizowanie wideokonferencji. Tendencja ta będzie dominować w najbliższym okresie przyczyniając się (dzięki łatwości użycia) do dalszego wzrostu popularności sieci Internet. Potwierdzają to zresztą nowe systemy operacyjne, których integralnymi elementami składowymi stają się multimedialne aplikacje sieciowe.

5. Potrzeby użytkowników

Sieć komputerowa ma sens tylko z powodu dostarczania usług użytkowych. Oczekiwania użytkownika są natomiast dość trudne do określenia. W znacznym stopniu zależą od jego przygotowania do pracy w środowisku sieciowym i doświadczeń, jakie zdobył w tym zakresie. Zmieniają się wraz ze zdobywaniem tych doświadczeń. Można co najwyżej wskazać potrzeby najczęściej zgłaszane, choć nie muszą to być wcale usługi, z których następnie korzystać się będzie najczęściej.

Usługą podstawową jest bez wątpienia dostęp do serwera lub serwerów sieci lokalnej, do

zgromadzonego tam oprogramowania i danych użytkowych. Na tym samym poziomie traktować trzeba dostęp do urządzeń zęwnętrznych - drukarki sieciowej, streamera, czasem napędu dysków optycznych.

Następny poziom, to usługi komputera o większej mocy obliczeniowej. Nie jest on na ogół dostępny w ramach sieci lokalnej, nie jest to istotne -- od tego jest właśnie sieć. Problem jest stosunkowo prosty dla zadań czysto obliczeniowych, komplikuje się, gdy dane mają być prezentowane graficznie a stopień komplikacji rośnie, gdy wymagana jest interakcja użytkownika z oprogramowaniem, realizowana w trybie graficznym ("okienka i mysz"). Wzrost trudności wynika z rozmiaru danych, które muszą być przesyłane z i do stacji roboczej użytkownika w odcinku czasu o tolerowalnej jeszcze długości. Bierny dostęp do danych (wyszukiwanie i prezentacja) zaczyna mieć również coraz częściej charakter wyżej przedstawiony.

Dostęp do sieci rozległych jest kolejnym, naturalnym etapem rozwoju zastosowań komputerów, następującym po uzyskaniu warunków do pracy grupowej w ramach lokalnej sieci komputerowej. Sieć rozległa umożliwia korzystanie z zasobów niedostępnych w ramach sieci lokalnej - zasobów informacyjnych (bazy danych), usług obliczeniowych o specyficznym profilu (komputery dużej mocy obliczeniowej) lub przekazywanie danych i dokumentów drogą elektroniczną (e-mail). Osobną klasę stanowią usługi kojarzone z siecią Internet - poczta elektroniczna, transport danych i wszystko co na tej bazie urosło - grupy dyskusyjne, elektroniczne gazety (z rynku krajowego wystarczy wymienić *Plotki, Pigułki* czy *Groch z Kapustą*) Archie, Gopher, WWW, WAIS, X.500.

Realizacja usług sieciowych rozpoczyna się w stacji użytkownika. Sieć może, co najwyżej wypełniać swoje funkcje lepiej lub gorzej -- wydajnie lub nie. Własności stacji użytkownika określone są przez sprzęt i oprogramowanie tam zainstalowane.

Przykładem stacji zorientowanej na pracę w sieci jest X-terminal. Bez pomocy odpowiedniego serwera jest on całkowicie "bezradny". Stacje robocze klasy HP lub Sun mogą pracować samodzielnie przez co w mniejszym stopniu obciążają sieć, jednak koszt stanowiska o dobrej wydajności jest dość wysoki. Najczęściej stacja konstruowana jest na bazie komputera personalnego, gdyż jest to rozwiązanie najtańsze i umożliwia eksploatację lokalną popularnego oprogramowania użytkowego. Warto zwrócić uwagę na fakt, że mocna konfiguracja sprzętowa komputera personalnego kosztowo zbliża się do najtańszym modeli stacji roboczych (te ostatnie mają jednak niższą wydajność).

Najpopularniejszy system operacyjny dla komputerów personalnych -- MS-DOS, do wersji 6.22 włącznie, nie dostarcza żadnych usług sieciowych. Warstwa sieciowa musi być konstruowana i konfigurowana przy użyciu dodatkowego oprogramowania. Na tym poziomie niewątpliwie dominuje sieciowy system operacyjny NetWare firmy Novell i związany z nim protokół IPX/SPX. NetWare może być uzupełniony o oprogramowanie wymiany danych z innymi sieciami, wykorzystującymi inne protokoły, w tym TCP/IP (wersja 4.xx ma tę własność wbudowaną).

Współczesne systemy operacyjne komputerów są (lub będą) wyposażane standardowo w mechanizmy komunikacji za pośrednictwem TCP/IP. Warunek ten był spełniany "od zawsze" przez systemy klasy Unix - dla nich TCP/IP stworzono i na nich Internet bazował w przeszłości. Obecnie dotyczy to już jednak wszystkich systemów operacyjnych.

Najnowsza wersja systemu IBM OS/2 3.0 (*Warp* - październik 1994) wyposażona jest w bazujące na mechanizmach TCP/IP oprogramowanie dostępu do Internetu, a w nim do *World-Wide Web* (WWW) - o nazwie *WEBExplorer*. MS Windows NT 3.5 (*Daytona*) zawiera wewnątrz mechanizmy protokołu TCP/IP. Przy tym Windows NT Server (w sprzedaży od 30 listopada 1994) jest wyposażony ponadto w mechanizmy aranżowania połączeń z siecią rozległą (*routing*) oraz obsługę do 255 zdalnych klientów z protokołem *IP*.

Zapowiadane w najbliższej przyszłości *Windows 95*, system przeznaczony dla **indywidualnych** komputerów będzie również

zaopatrzony w protokół TCP/IP. System ten, mimo że formalnie jeszcze nie istnieje, jest już przewidywany w wielu zastosowaniach, w tym np. oprogramowania do nawigacji w ramach sieci lokalnych i rozległych firmy *Novell*, o nazwie FERRET.

Warto zauważyć, że takie mechnaizmy jak *telnet*, *ftp* czy *ping* są poleceniami wewnętrznymi systemu operacyjnego MS-DOS 7.0, stanowiącego (o ile nie zmieni nazwy) podstawowy składnik *Windows 95*.

Obecnie dominującym systemem operacyjnym jest bez wątpienia system *MS Windows (Windows for Workgroups 3.11)*. Dla wersji tej dostępne są moduły programowe realizujące protokół TCP/IP zarówno w wersji 16-to bitowej jak i 32-u bitowej (zaczerpniętej z *Win'95*). Przykładem może być tu PC/TCP firmy FTP, które obok aplikacji typu *ftp* czy *telnet* oferuje również *e-mail, NFS* oraz dostęp do *WWW*.

Podstawowymi kryteriami przy kreowaniu stanowiska są bez wątpienia koszt realizacji i wydajność. Przy czym, koszt musi obejmować cenę sprzętu w konfiguracji gwarantującej szanse eksploatacji potrzebnego użytkownikowi oprogramowania a nie tylko systemu operacyjnego a wydajność mierzona powinna być w kategoriach eksploatowanego oprogramowania.

Wydajność w zasadniczym stopniu zależy od sposobu skonfigurowania systemu operacyjnego i oprogramowania wspomagającego, w tym sieciowego. Skonfigurowanie zaś systemu, szczególnie w obecnych inkarnacjach wymaga znacznej wiedzy (której brak często zastępują obiegowe mity czy legendy), ale jest konieczne. Zestawienie właściwego stosu protokołów, poprawnie współpracujących i pozostawiających jeszcze "trochę" pamięci dla oprogramowania użytkowego, do tego w warunkach ciągłych zmian plasterków z których to się robi (tzw. nowe wersje - *upgrade*) jest zadaniem trudnym.

Warto przy tym pamiętać, że wszystkie tzw. emulacje, inaczej mówiąc udawanie czegoś zupełnie innego są nadzwyczaj niewydajne. Przykłady - emulacja X-terminala w środowisku DOS lub Windows może na długo zrazić do standardu X-Windows. Emulacja DOS czy Windows w systemach klasy Unix (*SoftPC, Merge* i podobne) budzi zwykle rozbawienie obserwatorów.

Kolejnym ważnym składnikiem realizacji oczekiwań użytkownika w zakresie usług sieciowych jest modelowanie tych oczekiwań przez szkolenie. I musi to dotyczyć zarówno problematyki posługiwania się siecią i jej usługami w sposób poprawny jak i kulturalny. Przez szkolenie trzeba też wytworzyć odpowiednio liczną, a niezbędną kadrę administratorów niewielkich (a przez to, często występujących) sieci lokalnych.

Obserwując tendencje rozwojowe na rynku wydaje się, że potrzeby użytkowników w zakresie sieci są obecnie ukierunkowane na proste środowisko graficzne, które przeprowadzi go przez zawiłości adresowania w sieci, dziesiątki protokołów i standardów, różne typy danych i różne struktury systemów informacyjnych. Można przyjąć, że *mosaic* czy *netscape* przynajmniej częściowo spełniają te wymagania oferując w jednej aplikacji możliwość korzystania z hypertekstowych baz danych, przeglądania zasobów *gophera*, udziału w listach dyskusyjnych, przeglądania baz *news* i to wszystko z wykorzystaniem cech audio i video własnego komputera. Trzeba jednak pamiętać, że nawet najlepszy interfejs sieciowy nie zastąpi elementarnej wiedzy o sieci, jej strukturze i cechach funkcjonalnych. W przeciwnym bądź razie jeszcze bardzo często pojawiać się będzie pytanie: "*No dobrze wszystko to jest bardzo ładne ale po co jest ten Internet i gdzie się mogę nim połączyć ?*"

Leszek Bogusławski
Instytut Ochrony Środowiska, Warszawa

Bazy danych państwowego monitoringu środowiska

Stan środowiska jest bardzo zróżnicowany.
Poznanie go jest warunkiem ochrony środowiska przed wpływem współczesnej cywilizacji.

1. Państwowy monitoring środowiska

Wstęp

Ochrona środowiska przeżywa obecnie szybki rozwój, zarówno jako dyscyplina naukowa, jak i działalność praktyczna. Żywiołowo rozwijają się społeczne ruchy proekologiczne. Inwestuje się w system edukacji ekologicznej. Formułowane są programy ochrony środowiska ukierunkowane na rozwiązywanie problemów zarówno w skali lokalnej, jak i ogólnopolskiej. Opracowywanie i realizacja programów ochrony środowiska wymaga dokładnej znajomości przyczyn zmian stanu środowiska oraz możliwych skutków podejmowania działań. Funkcję kontrolera środowiska, skuteczności i efektywności realizowanych programów ochrony środowiska pełni system państwowego monitoringu środowiska (**pmś**).

1.1 Cele i zadania

Pojęcie "monitoringu", mimo braku polskiego odpowiednika, weszło do użycia w języku potocznym. Brak dokładnego tłumaczenia terminu powoduje różne jego interpretacje. Na ogół pod pojęciem monitoringu rozumie się pewien system obserwacji (pomiarów) ciągłych danego zjawiska, których wyniki są w sposób automatyczny przekazywane do „centrali" (przekazywanie to może być oczywiście wielostopniowe). W praktyce przy braku definicji systemu, procesu, czy zjawiska definiowanie ich polega na specyfikacji celów i zadań, jakie mają do spełnienia, bądź też właściwości, jakie posiadają. Tak właśnie zostanie przybliżone pojęcie monitoringu (na przykładzie **pmś**).

Celem państwowego monitoringu środowiska jest zwiększanie skuteczności działań na rzecz ochrony środowiska poprzez zbieranie, analizowanie i udostępnianie danych dotyczących stanu środowiska i zmian w nim zachodzących.

Do podstawowych zadań **pmś** należy:

- informowanie społeczeństwa o zagrożeniach ekologicznych w skali lokalnej, regionalnej krajowej i globalnej;
- dostarczanie informacji o aktualnym stanie i stopniu zanieczyszczenia poszczególnych komponentów środowiska dla oceny skuteczności realizacji programów ochrony środowiska w skali lokalnej, regionalnej, krajowej i globalnej;
- analizowanie procesów i zjawisk zachodzących w środowisku przyrodniczym;
- dostarczanie informacji o przewidywanych skutkach użytkowania środowiska w skali lokalnej, regionalnej, krajowej i globalnej.

1.2 Struktura organizacyjna państwowego monitoringu środowiska

Państwowy monitoring środowiska tworzą:

I. sieć krajowa (ogólnopolska) stacji i stanowisk pomiarowych obejmująca:

   A. stacje mierzące zanieczyszczenia środowiska w rejonach przygranicznych;

B. sieci wczesnego ostrzegania o skażeniach promieniotwórczych, związane z przeciwdziałaniem nadzwyczajnym zagrożeniom środowiska, szczególnie stacje służb pomiarów skażeń promieniotwórczych;

C. stacje pracujące według programów międzynarodowych (np. EMEP, HELCOM i inne);

D. stacje mierzące zanieczyszczenia środowiska pod kątem oceny skuteczności i efektywności wdrażania polityki ekologicznej państwa;

E. stacje charakteryzujące funkcjonowanie typowych dla Polski ekosystemów.

II. sieci regionalne (rejonowe) stacji i stanowisk pomiarowych - cel: realizacja zadań o znaczeniu regionalnym;

III. sieci lokalne stacji i stanowisk pomiarowych - cel: ocena oddziaływania podmiotów gospodarczych na środowisko.

Z punktu widzenia przedmiotu działania (podział tematyczny) **pmś** obejmuje podsystemy:

- monitoringu powietrza atmosferycznego, w tym hałasu i promieniowania niejonizującego;
- monitoring wód powierzchniowych, w tym Bałtyku;
- monitoring wód podziemnych;
- monitoring powierzchni ziemi, w tym gleb i odpadów;
- monitoring przyrody ożywionej;
- monitoring zintegrowany[1].

Państwowy monitoring środowiska obejmuje również podsystemy pomiarowo-kontrolne związane z przeciwdziałaniem nadzwyczajnym zagrożeniom środowiska, szczególnie podsystem monitoringu skażeń promieniotwórczych.

Z punktu widzenia kategorii, uzyskiwanych w ramach **pmś** informacji, można wyodrębnić następujące bloki:

- ***Emisja*** - dane o jakościowej i ilościowej charakterystyce zanieczyszczeń odprowadzanych do środowiska przez określone źródło zanieczyszczeń (wielkość emisji do atmosfery zanieczyszczeń gazowych i pyłowych, ilość i skład odprowadzanych ścieków, ilość i charakterystyka odpadów itd.);
- ***Imisja*** - informacje o zawartości określonych zanieczyszczeń w poszczególnych komponentach środowiska (stężenia zanieczyszczeń w powietrzu, stężenia zanieczyszczeń w wodach powierzchniowych i podziemnych, ilość i skład chemiczny odpadów zgromadzonych na składowiskach, zawartość metali ciężkich w glebach itd.);
- ***Zasoby naturalne i struktury przyrodnicze*** - informacje o strukturze geomorfologicznej kraju, o zasobach naturalnych i strukturach przyrodniczych (źródłem danych są wyniki prac naukowo-badawczych, ekspedycji naukowych i poszukiwawczych oraz systemy wykorzystujące teledetekcję);
- ***Warunki hydrometeorologiczne i klimatyczne*** - informacje o m.in. opadach atmosferycznych, prędkości i kierunku wiatru, temperaturze i jej pionowym rozkładzie, dyfuzji atmosferycznej, a także dane dotyczące stanu wód w rzekach, odpływu rocznego, przepływów itd.
- ***Prognozy*** - blok umożliwiający prognozowanie zmian środowiska w oparciu o dane zgromadzone w pozostałych blokach informacyjnych.

Podstawa funkcjonowania **pmś** jest baza pomiarowa: Wojewódzkich Inspektoratów Ochrony Środowiska, Wojewódzkich Stacji Sanitarno-Epidemiologicznych, Stacji Chemiczno-Rolniczych, jednostek naukowo-

[1] jego celem jest rejestracja i analiza krótko i długookresowych przemian zachodzących w systemach ekologicznych pod wpływem działalności człowieka. Relacje rejestrowane w podsystemie monitoringu zintegrowanego powinny umożliwić właściwą interpretację tendencji obserwowanych w podsystemach monitoringów branżowych

badawczych i dydaktycznych oraz innych laboratoriów.

2. System informatyczny państwowego monitoringu środowiska

2.1 Zadania systemy informatycznego

Realizcja zadań tak złożonego systemu, jakim jest **pmś**, wymagało podjęcia wielu różnych przedsięwzięć organizacyjnych, jak również informatycznych. Zadania jego systemu informatycznego wynikają bezpośrenio z funkcji do jakich został powołany. Uzupełniono je między innymi o elementy związane z funkcjonowaniem w niejednorodnym środowisku, a w szczególności przedstawiają się następująco:

- administrowanie państwowym monitoringiem środowiska;
- pozyskiwanie i gromadzenie danych pomiarowych o stanie środowiska;
- pozyskiwanie i gromadzenie danych przetworzonych o stanie środowiska;
- analizowanie, ocena, prognozowanie na podstawie danych;
- graficzne i kartograficzna prezentacja danych;
- tworzenie i wydawanie komunikatów, raportów i opracować dotyczących stanu środowiska;
- konwersja i transmisja danych z/do innych systemów.

Należy wyraźnie podkreślić trudności z konstrukcją tego systemu. Realizacja przedsięwzięcia informatycznego o takiej skali, uwarunkowana jest bowiem wieloma czynnikami. Świadomość ich istnienie była niezbędna do opracowania założeń. Do najważniejszych należały:

⇒ rozproszenie systemu na obszarze całego kraju. Wymagało to zaprojektowania precyzyjnego i niezawodnego systemu spływu informacji źródłowych, stworzenie węzłów pośrednich przetwarzających dane pomiarowe, zaprojektowania filtrów;

⇒ integracja danych pochodzących od różnych jednostek organizacyjnych, o różnej przynależności resortowej i administracyjnej, a zatem prawidłowe jego funkcjonowanie to wyraz dopracowanych podstaw prawnych;

⇒ bazy danych integrowane w systemie są bardzo zróżnicowane tematycznie, przy jednoczesnej konieczności zapewnienia ścisłych zależności i powiązań pomiędzy różnymi elementami systemu;

W ramach systemu informatycznego **pmś** będą docelowo występowały następujące bloki funkcjonalne:

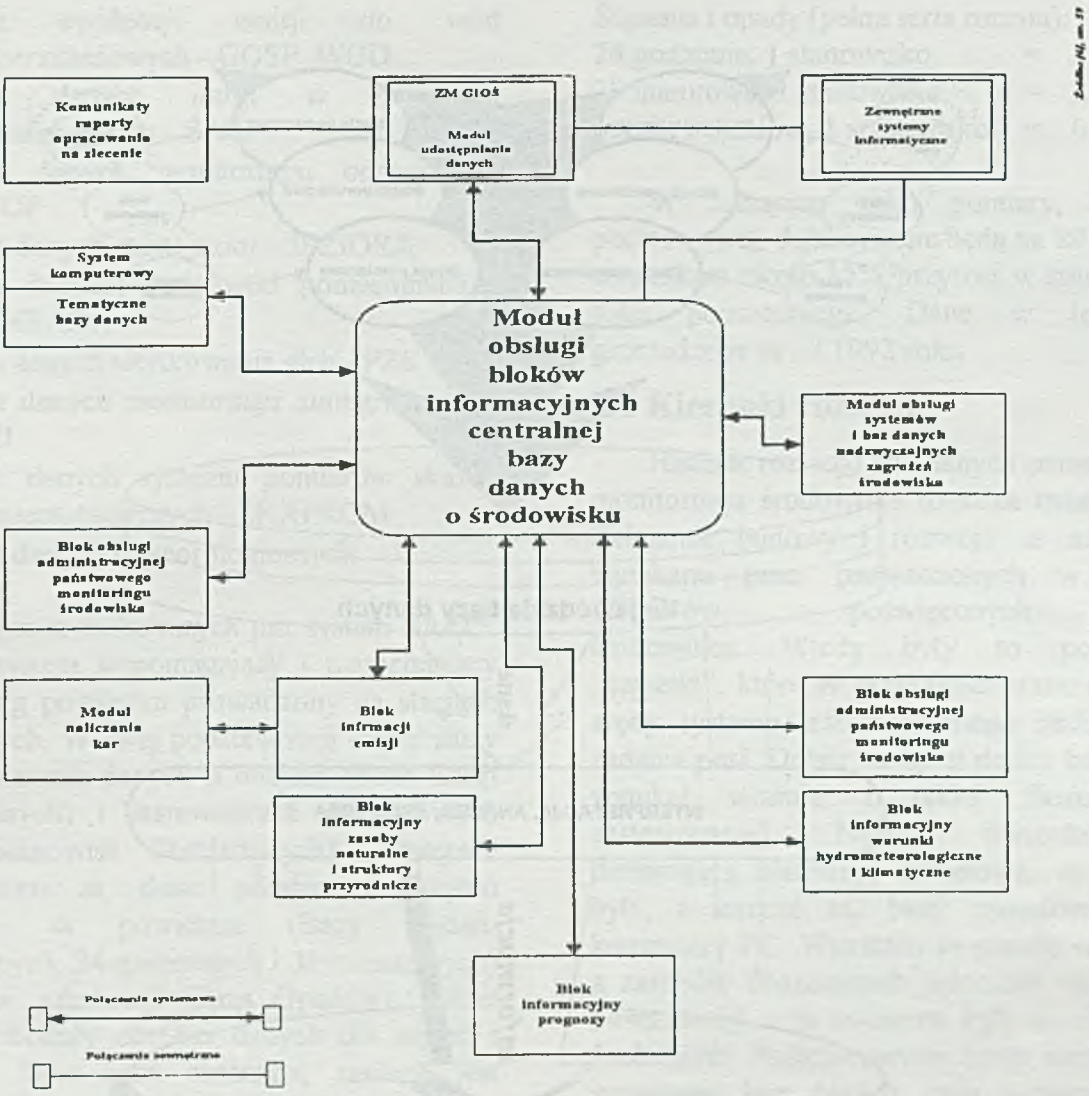
* **pozyskiwanie danych** (moduł zrealizowany w 1991 roku);
* **gromadzenie danych** (moduł zrealizowany w 1991 roku). Odpowiednie systemy zarządzania
* **prognozowanie** (moduł planowany do realizacji w latach 1994÷1997). Powinien zapewnić prognozowanie zanieczyszczeń środowiska na podstawie danych zawartych w bazach ewidencji emisji.
* **analizy i oceny stanu środowiska** (moduł wdrażany w 1994 roku). Docelowo: rutynowe przetwarzanie danych zgromadzonych w bazach tematycznych
* **prezentacja danych**[2] (moduł wdrażany w 1994 roku). W odniesieniu do danych przestrzennych najczęstszą formą prezentacji są mapy tematyczne. Formalnym standardem w tym wypadku jest system ArcInfo, w wersji na PC i stacje robocze.
* **konwersja i transmisja danych.** Z uwagi na różne pochodzenie danych do baz, jak również możliwość ich stosowania w różnych systemach konieczne jest opracowanie konwerterów formatów, kodów, itp. Wiąże się z tym także zapewnienie prawidłowej transmisja między systemami.

Współdziałanie poszczególnych bloków funkcjonalnych **pmś,** na przykładzie centralnej bazy danych[3], prezentuje schemat blokowy konstrukcji jej systemu.

---

2 w tym w formacie kartograficznym

3 identyczną konstrukcje mają wojewódzkie bazy danych

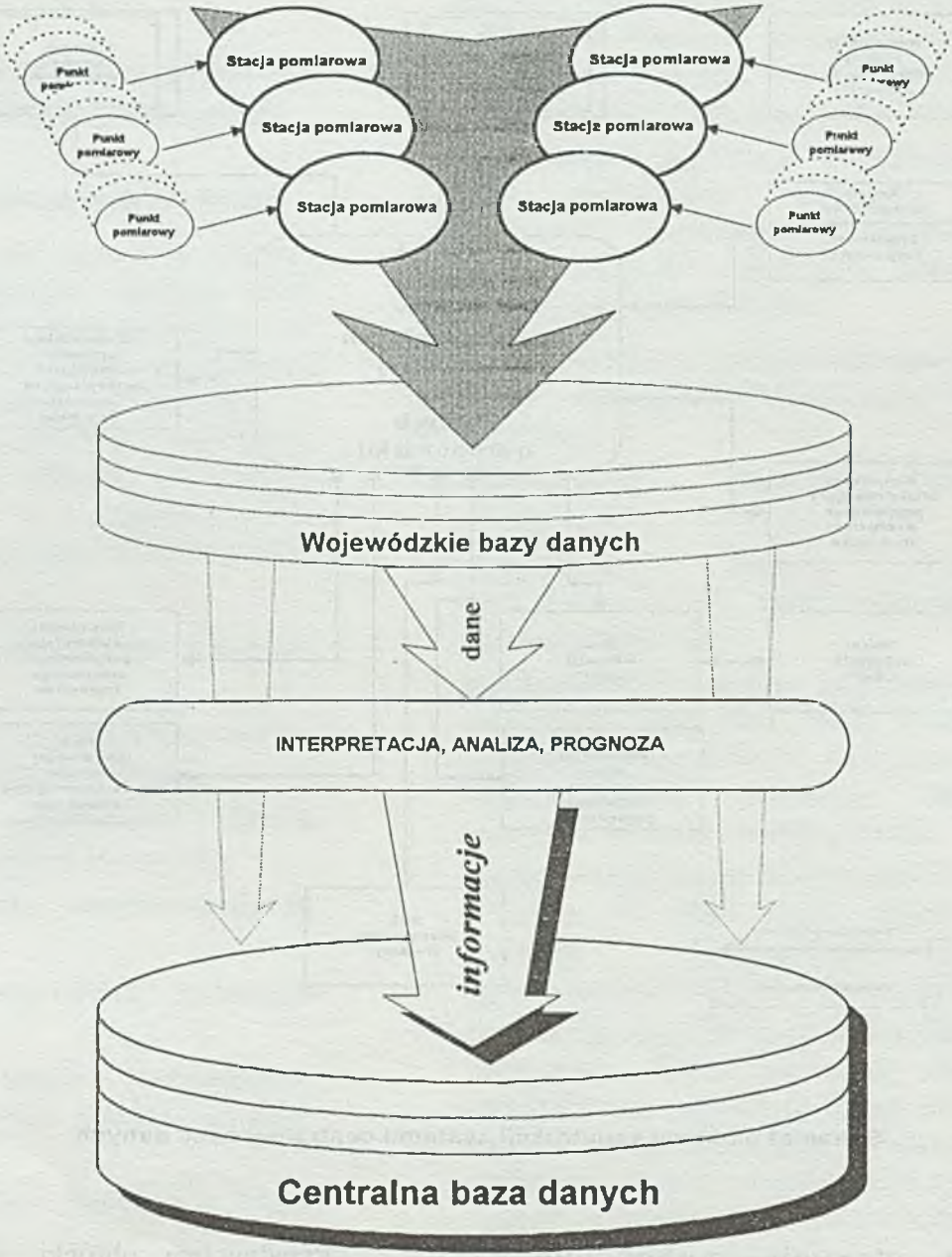


Schemat blokowy konstrukcji systemu centralnej bazy danych

3. Bazy danych państwowego monitoringu środowiska

System informatyczny **pmś** budowany jest w oparciu o istniejące bazy danych oraz przedmiotowe systemy zbierania i obsługi danych, podzielone tematycznie na poziomie wojewódzkim i krajowym. Podział ten odzwierciedla tematyczny podział państwowego monitoringu środowiska, przedstawiony wcześniej.

Przedmiotem obróbki, czyli materiałem źródłowym dla każdej z tych baz są dane pochodzące z różnych pomiarów, zgodnie z klasą tematyczną i obowiązującą w niej metodologią. Dane te są składowane na odpowiednich szczeblach, a następnie transportowane do kolejnych ogniw systemu. Po drodze odbywa się ich transformacja. Ideę tego procesu przedstawia poniższy schemat.



Schemat przepływu informacji pomiarowej w państwowym monitoringu środowiska

3.1 Zawartość informacyjna

Za stan każdej bazy danych, ich zawartość merytoryczną, w tym za ocenę jakości danych źródłowych odpowiadają instytuty nadzorujące podsystemy monitoringu (warstwa: interpretacja, analiza, prognoza w schemacie przepływu informacji pomiarowej). Podkreślić należy fakt, iż jednostki te są głównym ogniwem transformującym dane pomiarowe. Efektem tej części ich pracy są informacje w postaci analizy, prognoz, prezentacji graficznych, raportów, itd., które stanowią podstawę podejmowania decyzji przez poszczególne szczeble zarządzania ochroną środowiska.

Wykaz funkcjonujących, bądź opracowywanych, tematycznych baz danych **pmś** o zakresie krajowym lub wojewódzkim przedstawia się następująco:

→ baza ewidencji obiektów emitujących hałas - OPH

→ baza ewidencji emisji do wód powierzchniowych - GOSP_WOD
→ baza danych imisji w powietrzu atmosferycznym - SAZP
→ baza danych monitoringu odpadów - SIGOP
→ baza danych stanu jezior - JEZIORA
→ baza danych stanu wód podziemnych - MONBADA
→ baza danych użytkowania gleb - PZ8
→ baza danych monitoringu zintegrowanego - PZI
→ baza danych systemu pomiarów skażeń promieniotwórczych - SKAPROM
→ baza danych meteorologicznych

Jednym z ciekawszych jest system SAZP[4]. Jest to system wspomagający i uzupełniający monitoring powietrza prowadzony na stacjach badawczych. W swej podstawowej części służy on do zbierania danych o stacjach (Baza Stacji Pomiarowych) i stanowiskach pomiarowych (Baza Stanowisk Pomiarowych). Następnie gromadzone są dane pomiarowe: stężeń substancji w powietrzu (Bazy Stężeń: miesięcznych, 24-godzinnych i 30 minutowych) i opadów substancji (Baza Opadów). Pakiet posiada moduły obróbki danych dla stężeń i opadów. Jako baza centralna, zasilana jest danymi z baz wojewódzkich (SAZPW). Jest to zintegrowany pakiet obsługi baz danych z szerokimi możliwościami prezentacji danych w postaci graficznej, zarówno w postaci różnego typu wykresów, jak i map. System posiada również moduł do przeprowadzania analizy statystycznej danych według modeli empirycznych i teoretycznych. Jako uzupełnienie do powyższego systemu (imisji), wbudowano także podsystem emisji, który zawiera bazy danych przeznaczone do wprowadzania ważniejszych informacji o emitorach.

Rozmiary spływających danych niech obrazuje fakt, iż pełne dane stacji, stanowisk, pomiarów stężeń i opadów zajmują następującą ilość miejsca na dysku (dane szacunkowe):

1 stacja pomiarowa	=	0.3 KB
1 stanowisko pomiarowe	=	0.08 KB
Stężenia i opady (pełna seria roczna):		
24 godzinne, 1 stanowisko	=	1.7 KB
30 minutowe, 1 stanowisko	=	2.1 KB
pomiary opadów, 1 stanowisko	=	0.3 KB

W bieżącym roku pomiary, w sieci podstawowej, dokonywane będą na 89 stacjach, co stanowi około 15% przyrost w stosunku do roku poprzedniego. Dane w tej bazie gromadzone są od 1992 roku.

3.2 Kierunki rozwoju

Historia rozwoju baz danych państwowego monitoringu środowiska to kilka ostatnich lat. Założenie budowy i rozwoju są najczęściej wynikami prac, prowadzonych w ramach CPBR-ów, poświęconych ochronie środowiska. Wtedy były to pojedyncze „cegiełki", które w przyszłości stanowić miały zręby systemu informatycznego realizującego zadania **pmś**. Dobór narzędzi do ich konstrukcji wynikał właśnie z takiej filozofii (ich zastosowania). Należy stwierdzić, że dominującą platformą sprzętową, na których były, i jeszcze są, bazy posadowione, to komputery PC. Wynikało to przede wszystkim z zasobów finansowych jednostek, w których powstawały, a te związane były ze sferą tzw. *budżetówki*. Podstawowymi zatem narzędziami, systemami baz danych, były systemy DOS-owskie, takie jak: Clipper, dBase, FoxPro. Mylił by się jednak ten, kto oceniałby ten okres, jako „radosną twórczość" zespołów projektantów, programistów, itd. działających w oderwaniu od innych. W okresie tym powstały w Ministerstwie Ochrony Środowiska, Zasobów Naturalnych i Leśnictwa (MOŚ,ZNiL) założenia Zintegrowanego Systemu Informatycznego „Środowisko", jako wstęp do jego informatyzacji. Jednym z elementów był system informatyczny Państwowej Inspekcji Ochrony Środowiska[5] (PIOŚ). Podstawowym założeniem było doprowadzenie do stworzenia integracja danych uzyskiwanych w dziedzinie ochrony środowiska i gromadzonych w bazach danych systemu. Drugą z płaszczyzn była integracji systemów operacyjnych. Ogólnie

[4] w wersji dla instytucji nadzorujących SAZPK

[5] **pmś**, ze swoimi rozwiązaniami informatycznymi jest jednym z podstawowych elementów tego systemu.

wiadomo było, że kierunek, w którym dążyć będą globalne przedsięwzięcia informatyczne tej branży to systemy otwarte i komunikacja w oparciu o rozległe sieci komputerowe. I żeby się nie w dawać w zbędne dywagacje wybór padł na UNIX. W efekcie ujednolicono sposob wykonywania, bądź dalszego rozwoju, baz danych, finansowanych, w ramach **pmś,** przez PIOŚ. Narzędziem gwarantującym, w miarę gładkie, przejście od DOS do UNIX miał być FoxPro.

Jako medium transmisji danych wybrano sieć pakietową X.25. Jest ona jednym z najlepszych w kraju, ogólnie dostępnym środkiem do łączności między oddalonymi sieciami lokalnymi. Obecnie korzysta się z usług POLPAK-u jednego z operatorów oferujących tą usługę dla użytkownika spoza kręgów naukowo-akademickich. Z uwagi na to, że POLPAK jest siecią ogólnie dostępną, w przypadku konieczności podwyższenia ochrony danych (związanego z ich poufnością) najprawdopodobniej nie będzie to rozwiązanie docelowe. Spodziewać się należy odejścia w kierunku operatorów *stricte* z administracji państwowej. Przykładem może być CUPAK - sieć Centralnego Urzędu Planowania.

W bieżącym roku zdecydowano rozpocząć przygotowania do kolejnego etapu transformacji baz danych do systemu opartego na serwerze SQL. Ma to być przejście na docelowy system operacyjny, a co za tym idzie, wprowadzenie docelowego systemu zarządzania relacyjnymi bazami danych. Będzie to przedsięwzięcie pilotażowe[6], obejmujące wybrane inspektoraty wojewódzkie (Lublin, Łódź, Szczecin) oraz jednostki naukowo-badawcze (Państwowy Instytut Geologiczny, Instytut Meteorologii i Gospodarki Wodnej, Instytut Ekologii Terenów Uprzemysłowionych oraz Instytut Ochrony Środowiska). Dostarczone zostaną RISC-owe stacje robocze z systemem UNIX w wersji odpowiadającej normie systemu otwartego POSIX 1003.1. Jako system baz danych wybrano INFORMIX w wersji 7.0, dodatkowo dla instytutów wyposażony w komplet narzędzi do rozwoju aplikacji (między innymi: *4GL Development tools, VISION Development tools*).

Podsumowanie

System państwowego monitoringu środowiska, pełniący funkcję kontrolera środowiska, jest systemem złożonym. Jest to związane z jego rozległością, a jednocześnie koniecznością stałego funkcjonowania na różnych płaszczyznach. Podstawą działania systemu są tematyczne bazy danych, zawierające dane o stanie środowiska. Jakość i aktualność zgromadzonych w nich danych jest warunkiem koniecznym prawidłowej oceny stanu środowiska, a co za tym idzie właściwych decyzji podejmowanych na różnych szczeblach zarządzania ochroną środowiska. Bazy te ewoluowały, podobnie jak **pmś**. Wszystko wskazuje jednak na to, że przynajmniej na poziomie wojewódzkim i krajowym osiągnie się, w tym zakresie, zakładaną sprawność i skuteczność.

Bibliografia

1. Biblioteka monitoringu środowiska: „Program państwowego monitoringu środowiska". PIOŚ, Warszawa 1992
2. Instrukcja obsługi systemu analizy zanieczyszczenia powietrza SAZPK - 1 (wersja krajowa)
3. Biblioteka monitoringu środowiska: „Zanieczyszczenia powietrza w polsce w 1993 roku". PIOŚ, PIS, Warszawa 1994
4. Zarządzenie nr 2 Głównego Inspektora Ochrony Środowiska z dnia 20 stycznia 1995r w sprawie krajowej sieci stacji podstawowych monitoringu zanieczyszczenia powietrza.
5. „System informatyczny Państwowej Inspekcji Ochrony Środowiska". PIOŚ, GIOŚ, Warszawa 1994

[6] charakter polotażowy wynika wyłącznie z ograniczoności przedsięwzięcia, za sprawą kosztów jego realizacji.

Jerzy Gołuchowski
Stanisław Kędzierski
Krzysztof Kania
Tomasz Kajstura
Akademia Ekonomiczna w Katowicach

Modelowanie i realizacja procesów gospodarczych w rozproszonych systemach baz danych

1. Miejsce modelowania procesów gospodarczych w projektowaniu RSI

Tworzenie w sposób systematyczny i efektywny dużych rozproszonych systemów informatycznych (RSI), opartych na technologii baz danych, staje się coraz bardziej aktualnym i ważnym wyzwaniem dla informatyki. Prace nad systemami rozproszonymi prowadzone są od wielu lat. Osiągnięty poziom rozwoju sprzętu, oprogramowania, technologii przetwarzania danych i komunikacji stworzył możliwości budowy dużych, a przy tym wydajnych i przyjaznych, rozproszonych systemów informatycznych wspomagających organizację. Pożądaną cechą tych systemów jest to, aby pracowały w czasie rzeczywistym.

Zasadniczym założeniem leżącym u podstaw budowy dobrych systemów rozproszonych jest idea takiego systemu informatycznego, który funkcjonując jako jednolita całość, udostępniając niezbędne (rozproszone) zasoby, pozostawia rozproszenie zasobów niewidocznym dla programistów i użytkowników.

W klasycznym ujęciu przedmiotem rozproszenia są dane gromadzone w bazach danych (w rozproszonej bazie danych). Jednakże problematyka rozproszonych systemów informatycznych dotyka różnorodnych warstw systemu komputerowego. Systemom rozproszonym stawia się nierzadko wymaganie, aby pracowały w otwartym środowisku heterogenicznym.

Problemy przetwarzania rozproszonego stanowią jeden z najintensywniej, od wielu lat, rozwijanych obszarów zastosowań w zakresie sieci komputerowych, systemów operacyjnych i technologii baz danych.

Pomijając analizę różnorodnych aspektów warto zauważyć, że mówiąc obecnie o systemach rozproszonych bierze się pod uwagę:

- systemy komputerowe (sieciowe),
- rozproszone bazy danych,
- rozproszone obliczenia,
- przetwarzanie rozproszone,
- rozproszone rozwiązywanie problemów.

Wymienione kwestie wskazują na złożoność i kompleksowość tej problematyki.

Rozwiązywanie problemów złożoności i kompleksowości procesu budowy systemu rozproszonego możliwe jest poprzez wprowadzenie poziomów rozważań i warstw budowy (architektury) systemu rozproszonego. Na każdym z tych poziomów odmienne kwestie mają znaczenie kluczowe dla tworzenia dobrych systemów rozproszonych. Można zatem mówić o modelach systemu rozproszonego odpowiadającym różnym punktom widzenia.

Ze względu na dużą różnorodność koncepcji projektowych zrezygnowano z ich prezentacji poprzestając na zarysowaniu podejścia ANSA jako przykładowego sposobu opisu systemu rozproszonego, w którym RSI

przedstawia się w postaci pięciu modeli [BGZ93]:

- **model zadaniowy** dostarczający ogólnego pojęcia o systemie informacyjnym (aplikacji) oraz opisujący obiekty systemu pod kątem zadań i celów,
- **model informacyjny** dający podstawy do opisu niezbędnych dla systemu struktur danych (struktur informacyjnych), tzn. poszczególnych ich elementów, zasad ustalania zależności między nimi, ich współużytkowania i zasad dostępu,
- **model obliczeniowy** opisujący architekturę systemu (strukturę kodu) oraz narzędzia programistyczne, które powinny być dostępne dla twórców programów,
- **model inżynieryjny** przedstawiający podstawowe części kompilatora i systemu operacyjnego, niezbędne do realizacji obliczeń w środowisku heterogenicznym,
- **model technologiczny** opisujący system pod kątem powiązań z różnymi systemami operacyjnymi.

Rozwiązania problemów sprzętowych i oprogramowania można uważać w dużej mierze za zadowalające dla realizacji odpowiednich modeli RSI. Trudno jednakże uznać, że proponowane metodyki projektowania tych systemów pozwalają twórcom systemów rozproszonych uporać się z całą złożonością tworzenia modeli zadaniowych i informacyjnych, tak aby spełniały one oczekiwania użytkowników i odpowiadały wymaganiom inżynierii systemów. W literaturze przedmiotu spotkać można różnorodne próby opracowania właściwej metodyki, jednak te propozycje nadal nie pozwalają w zadowalającym stopniu uporać się ze złożonością problemów modelowania dziedziny i realizacją systemów umożliwiającą łatwą ich modyfikację.

Współczesne metodyki tworzenia są zorientowane na przetwarzanie transakcji. Podejście takie wynika z rozumienia SI w kategoriach manipulowania danymi w rozproszonych bazach danych. Takie ukierunkowanie myślenia projektantów RSI można uznać za zadowalające dla małych, stabilnych organizacji wymagających stosunkowo prostych aplikacji. W przypadku złożonych projektów informatyzacji dużych firm myślenie o działalności firmy w kategoriach transakcji i modelowanie jej w ten sposób jest zbyt daleko idącym uproszczeniem rzeczywistości i nie może być zadowalające.

W technologii baz danych konkurują ze sobą dwa podejścia do tworzenia systemów bazodanowych:

- podejście od danych,
- podejście od funkcji systemu.

Różne metodyki tworzenia systemów przyjmują jeden z tych sposobów patrzenia na rzeczywistość jako punkt wyjścia do analizy dziedziny przedmiotowej przeprowadzanej na potrzeby tworzenia RSI. Wynikiem wyboru jednego z podejść jest:

- albo ustalenie klas obiektów, ich atrybutów, powiązań między nimi oraz działań jakie wykonuje się na obiektach, traktowanych jako podstawa do ustalenia funkcji systemu
- albo ustalenie funkcji jakie powinien wykonywać system w toku przetwarzania, a na ich podstawie danych potrzebnych do realizacji tych funkcji.

Takie ujęcia wydają się nie w pełni zadowalające dla potrzeb modelowania dużych rozproszonych systemów informatycznych, takich jak np.: systemy wspomagające zarządzanie korporacją gospodarczą.

Organizacja gospodarcza zorientowana na rozwój jest systemem dynamicznym o zmiennej strukturze organizacyjnej, strategii działania, procesach gospodarczych, itp. Współczesne systemy informatyczne nie są dostosowane do takiej filozofii zarządzania firmą. Petryfikują one struktury organizacyjne i inne rozwiązania przyjęte w momencie projektowania i wdrażania systemu.

Podejście do tworzenia RSI, które wydaje się bardziej adekwatne do wymagań i filozofii zarządzania współczesną organizacją

gospodarczą, opiera się na uznaniu jako punktu wyjścia modelowanie procesów gospodarczych zachodzących w firmie. Przyjęcie takiego podejścia jest bardziej naturalne, ponieważ zarówno funkcje systemu jak i niezbędne dane, a także sposób rozproszenia i koordynacji zadań realizowanych przez RSI, wynikają z potrzeb realizacji procesów ujętych w postaci wzorców (instrukcji). Przedstawienie w sposób jawny procesów gospodarczych realizowanych w organizacji gospodarczej w modelu RSI możliwie jak na najwcześniejszym etapie tworzenia RSI oraz deklaratywna reprezentacja procesów na etapie realizacji systemu wydają się rozwiązaniami stwarzającymi lepsze perspektywy dla problemu rozwiązywania złożoności i kompleksowości tworzenia RSI.

Tradycyjne modelowanie RSI koncentrowało się niemal wyłącznie na analizie strumieni danych i transformacji danych. Modelowanie to brało pod uwagę jedynie dane organizacyjne oraz część procesów zachodzących w organizacji związanych z danymi. Procesy gospodarcze, mimo że były przedmiotem zainteresowania nie były ujęte w modelu systemu w sposób jawny. Modelując zatem systemy rozproszone trzeba wyjść poza przetwarzanie transakcji traktując je jako element procesów komunikacji i koordynacji prac (zadań) realizowanych przez użytkowników i systemy rozproszone. Skuteczna integracja tych zadań i kontrola zaawansowania ich realizacji (przebiegająca wielowątkowo) wymaga modelowania zarówno procesów manualnych jak i zautomatyzowanych.

Proponowane podejście umożliwia realizację wymagań, które musi uwzględniać dobra metodyka modelowania RSI:

1. specyfikację procesów już w toku analizy, możliwie jak najwcześniej,
2. rozwiązywanie problemów sprzężenia zwrotnego w modelu kaskadowym,
3. rozdzielenie modeli procesów od ich realizacji; procesy powinny być traktowane jako dane (zasoby).

Modele RSI muszą uwzględniać koncepcje zarządzania realizowane przez przedsiębiorstwo, odpowiadające wymaganiom zmiennego otoczenia gospodarczego. W konsekwencji muszą charakteryzować się tak pożądanymi cechami jak: elastyczność, transparencja, szybka i łatwa modyfikowalność. Uzyskać to można dzięki:

- idei obiektowości,
- deklaratywnej specyfikacji procesów,
- idei współbieżności procesów.

Jako punkt wyjścia proponowanej metodyki tworzenia rozproszonych systemów baz danych stawia się modelowanie procesów gospodarczych.

2. Zarys metodyki modelowania procesów gospodarczych

Modelowanie procesów wymaga uporządkowania bazy pojęciowej, tak aby tworzyła ona dobre podstawy do skutecznego i efektywnego modelowania systemów rozproszonych, podobnie jak udało się to zrobić w odniesieniu do struktur danych.

Model rozproszonego systemu informatycznego wymaga dwu ujęć [AuBr94]:

- ujęcia ekonomicznego, opisującego zarządzanie i reguły biznesowe, rozpatrywanej dziedziny problemowej. W ujęciu tym procesy manualne i automatyczne powinny być opisane przy użyciu tych samych środków formalnych. Ujęcie ekonomiczne jest przeznaczone głównie dla użytkowników końcowych, ale stanowi również specyfikację wymagań dla ujęcia technicznego.
- ujęcia technicznego, opisującego SI zgodnie z potrzebami automatycznego przetwarzania danych. Daje ono szczegółową, precyzyjną specyfikację elementów sprzętowych i programowych. To ujęcie musi spełniać wymagania ujęcia ekonomicznego.

Każde z tych ujęć wymaga rozpatrywania systemu na wielu poziomach szczegółowości

(abstrakcji) traktowanych jako kolejne modele rozproszonego systemu informatycznego. Każdy poziom jest opisywany przez:

- aspekt strukturalny (identyfikujący strukturę i powiązania elementów systemu),
- aspekt procesowy (dokumentujący działanie i funkcje elementów),
- aspekt informacyjny (dokumentujący informacje, które są pamiętane, generowane w systemie i wymieniane między elementami systemu).

W małych organizacjach procesy gospodarcze są stosunkowo proste do opisania i fizycznie mogą być realizowane przez niewielką liczbę agentów. W sporej liczbie przypadków jest to jedna osoba, ewentualnie dział zajmujący się realizacją od początku do końca określonego działania. Prowadzi to do utożsamiania osoby (lub niewielkiej grupy) realizującej działanie z samym procesem. Konsekwencją takiego podejścia jest projektowanie SI zorientowanych na stanowisko pracy użytkownika. Mając do czynienia z dużymi organizacjami o rozbudowanych procesach gospodarczych, punktem wyjścia powinna być analiza procesu i dopasowanie projektowanych procedur i zasobów (ich podziału, umiejscowienia) pod kątem efektywności jego wspomagania.

Przykładowo w małych bankach udzielenie kredytu indywidualnej osobie bywa zadaniem realizowanym w całości przez jedną osobę. Podobnie przyjęcie pracownika w małej firmie wymaga jedynie rozmowy z jej przyszłym bezpośrednim przełożonym. On to decyduje o przydatności pracownika dla siebie (a co za tym idzie i dla firmy). Często zatrudniany jest on w krótkiej perspektywie. W dużej firmie realizowana jest pewna strategia i przyjęcie pracownika wynika często z realizacji długookresowej polityki. Kariera pracownika jest planowana.

Procesy gospodarcze realizowane w przedsiębiorstwie są zazwyczaj opisywane w postaci instrukcji postępowania. Realizacja tych instrukcji zarówno w technologii manualnej jak i tradycyjnej technologii komputerowej (w postaci programów) jest niezadowalająca w wielu praktycznych sytuacjach. W przypadku prac realizowanych zwłaszcza przez duży zespół wielokrotnie zachodzi potrzeba modyfikacji wzorców procesów. Ponadto istnieje konieczność kontroli realizacji poszczególnych procesów. Zagadnienie to wymaga więc bardziej elastycznych narzędzi realizacji procesów, zwłaszcza w rozproszonych systemach informatycznych. W opracowaniu zostanie przedstawiony zarys takiego rozwiązania opartego na reprezentacji procesów za pomocą reguł.

3. W kierunku deklaratywnej reprezentacji procesów gospodarczych

Złożoność i wielkość procesów w dużych firmach prowadzi do decentralizacji i rozproszenia działań i zasobów. Z drugiej strony konieczne jest powiązanie zadań wykonywanych przez różnych agentów tak, aby była możliwa całościowa kontrola realizacji przebiegu procesów. Stan ten prowadzi do sprzeczności pomiędzy wymaganiami użytkowników i naturą wykonywania procesów a technologią informatyczną, która nie zapewnia odpowiedniej ich obsługi. W praktyce dochodzi do sytuacji, kiedy organizacyjnie następuje rozproszenie zasobów natomiast wspomagający organizację RSI jest jedynie formalnie zgodny z trybem działalności firmy. Zmieniająca się natura i sposób rozwiązywania zadań wymusza doskonalenie technologii i prowadzi do tworzenia systemów zarządzania danymi rozproszonymi, opartych na architekturze klient-serwer, wyposażonych w interfejsy graficzne, itp.

Tradycyjna realizacja procesów sprowadza się do ich reprezentacji w postaci programów. Od 1984 roku, czyli od artykułu [Morg84], w którym Morgenstern przedstawił koncepcję aktywnych baz danych trwają prace nad deklaratywną reprezentacją procesów. Znaczącą rolę w tej reprezentacji odgrywają reguły. Przybliżenie tej problematyki

praktykującym projektantom RSI staje się niezwykle istotną kwestią dla powodzenia dużych projektów informatycznych.

Rozwój teorii systemów baz danych doprowadził do powstania różnorodnych podejść do tworzenia (programowania) systemów informatycznych. Podejścia te w różnym stopniu zaspokajają potrzeby konstruktorów systemów rozproszonych. Można wyróżnić cztery zasadnicze podejścia (stylów) programowania systemów bazodanowych:

- imperatywne (proceduralne),
- aktywne,
- dedukcyjne,
- scenariuszowe,

Styl imperatywny. Procesy opisywane są za pomocą języków imperatywnych. Języki imperatywne to proceduralne języki ogólnego przeznaczenia nie zawierające instrukcji bazodanowych. Operacje na bazie danych wykonywane są za pomocą instrukcji z języka podstawowego lub poprzez wywołanie procedur pomocniczych. Jedynym trwałym typem danych jest plik. Języki wbudowane w bazy danych składają się z języków imperatywnych i języka wyszukiwania jak SQL np.: Codasylowe interfejsy do Cobolu, PL/I i Fortanu. Zapytania pisane są w języku podstawowym, prekompilator identyfikuje zapytanie i wywołuje język zapytań.

Styl aktywny. Ten styl stosowany jest w większości aktywnych baz danych korzystających z modelu *Zdarzenie - Warunek - Akcja*. W modelu tym reguła składa się z następujących elementów:

- zdarzenie; może być wewnętrzne (np.: wprowadzenie zmiany w podatku powoduje...) lub zewnętrzne (np.: godzina 7.00 powoduje...),
- warunek; występujący jako zbiór zapytań,
- akcja; jest programem zdefiniowanym przez użytkownika.

Jako reguły w aktywnych bazach danych wyrażane są warunki integralności i sterowanie danymi. Składnik akcyjny napisany jest w języku imperatywnym przy czym część sterująca /monitorująca decyzje o uaktywnieniu reguł/ napisana jest w wyższym stylu programowania aniżeli imperatywny.

Styl dedukcyjny. Styl ten wywodzi się z języków programowania w logice (np.: Prolog). Stosowany jest w luźnym lub ścisłym powiązaniu z językami baz danych. Styl dedukcyjny stosuje rekurencyjne przeszukiwanie baz danych dla wykonania procesu wnioskowania. W większości przypadków połączenie relacyjnych baz danych z Prologiem znacząco zmniejsza wielkość systemu nawet w przypadku prostych aktualizacji.

Styl scenariuszowy. W tym stylu programowania procesy implementowane są jako działania w rozwijających się sieciach Petriego zwanych scenariuszami. Scenariusz jest zbiorem aktywnych jednostek, z których każda posiada swój zbiór stanów wymieniających komunikaty.

Tabela 1 porównuje style programowania wskazując stopień spełnienia przez nie wymagań w stosunku do deklaratywnej (regułowej) reprezentacji procesów.

Tabela 1

Spełnianie wymagań przez stosowane style programowania baz danych

Wymaganie	Imperatywny	Aktywny	Dedukcyjny	Scenariuszowy
Przejrzystość struktur	-	o	-	o
Jednolitość	+/-	%	%	-
Poziom abstrakcji	-	o	o	o
Niezależność kontekstowa	-	o	o	-
Niezależność referencyjna	-	-	-	-
Wykonanie deterministyczne	o	%	%	-
Przestrzeganie zgodności	-	o	+	-
Redundancja aktualizacji	-	%	%	-

Oznaczenia: + w pełni spełnia wymagania
- nie spełnia wymagań
o częściowo spełnia wymagania
% tylko pewne implementacje spełniają wymagania

Źródło: [GaEt95]

Z nawet tak pobieżnego przeglądu zaprezentowanych stylów programowania baz danych wynikają wnioski wskazujące na pożądane dalsze kierunki prac nad metodologią tworzenia dużych rozproszonych systemów bazodanowych, wspomagających zarządzanie dużymi firmami, zwłaszcza rozproszonymi terytorialnie (geograficznie).

4. Regułowa reprezentacja procesów gospodarczych

Reguła może być reprezentowana w postaci:

- fragmentu programu nie będąc samodzielnym bytem,
- metaobiektu posiadającego określoną strukturę, nazwę i granice.

Przyjmując drugi sposób realizacji można dokonywać na regułach podstawowych operacji (takich jak: tworzenie, aktualizacja i usuwanie) charakterystycznych dla rekordów w bazie danych. W celu takiej reprezentacji reguły należy dokonać procesu dekompozycji reguły na składniki elementarne, które można przedstawić w postaci tablic relacyjnych.

Ogólnie reguły mogą być przedstawione za pomocą zdania postaci:

ON *wydarzenie* IF *warunek* THEN *rezultat*.

Struktura reguł wykazuje pewne podobieństwa do struktury zdań warunkowych używanych w proceduralnych językach programowania. *Wydarzenie* opisuje operację, która wyzwala regułę. Aby wykonana została *akcja (rezultat)* opisana przez regułę spełniony musi zostać *warunek*. *Rezultat* może być wartością, wagą, prawdopodobieństwem lub sekwencją operacji na bazie danych. Jeżeli *rezultat* zawiera operacje na bazie danych może wyzwalać następne reguły; reguły mogą też być wywoływane rekurencyjnie.

Zadania realizowane w aktywnych SZBD podzielić można na [CJL91]:

- *zadania zewnętrzne (ZZ),*
- *zadania zarządzania regułowego (ZZR).*

ZZ są zapytaniami i aktualizacjami użytkowników lub aplikacji. ZZR są wyzwalane przez reguły między innymi jako konsekwencje uruchomienia ZZ. ZZR również mogą wywoływać ZZR. Zbiór zadań będących konsekwencją wywołania ZZ składa się na pracę. Czas wykonania pracy można uważać za nowy miernik efektywności aktywnych baz danych.

Zadania składające się na pracę (proces) przyjmują w bazach danych postać transakcji. W wielu proponowanych rozwiązaniach użytkownik ma niewielki wpływ na określenie sposobu translacji ZZ i ZZR do postaci transakcji. Na przykład POSTGRES łączy ZZ i ZZR w jedną transakcję.

W rzeczywistości praca może składać się z jednej lub większej ilości transakcji, transakcja może być częścią najwyżej jednej pracy. Najprostszym przypadkiem jest powiązanie jeden-do-jeden zastosowane np. w POSTGRES-ie gdzie wszystkie zadania stanowią jedną transakcję. Drugim skrajnym podejściem jest reprezentacja każdego zadania pracy w postaci odrębnej transakcji. Gdy dopuszcza się różne sposoby odwzorowania zadań w postaci transakcji przedmiotem pojawia się problem powiązania pomiędzy odwzorowaniem a efektywnością wykonania pracy.

Rozkład pracy na mniejsze transakcje może następująco wpływać na działanie systemu:

- jeżeli uruchomienie i zamknięcie transakcji wiąże się z dużymi kosztami, wzrastająca liczba transakcji spowoduje gorszą realizację,
- rozbicie pracy na większą liczbę mniejszych transakcji spowoduje skrócenie średniego czasu dostępu do danych pomimo identycznej ilości dostępów do danych.

Przy rozważaniu reprezentacji prac należy wziąć pod uwagę oba powyższe czynniki. Należy także wziąć pod uwagę fakt, że sposób dostępu do danych może być różny w ZZ i ZZR (tylko do odczytu, aktualizacje) oraz wpływ różnego sposobu dostępu do danych przy różnych metodach odwzorowywania przy wzrastającej złożoności zadań.

Wydarzenia, *warunki* i *akcje* będące podstawowymi składnikami reguły mogą być implementowane do postaci transakcji w oparciu o różne modele. O sposobie implementacji ZZR i ZZ do postaci transakcji decyduje semantyka reguły oraz efektywność systemu.

Gdy do implementacji zadań zastosowany zostanie model zagnieżdżony konieczne jest rozpatrzenie następujących dwóch aspektów:

- modeli odwzorowania zadań w postaci transakcji,
- czasu wywołania zadań zarządzania regułowego. Zadania te mogą być wywoływane w momencie zaistnienia zdarzenia, bądź też dopiero po zakończeniu zadania wywołującego.

Wyróżnić można cztery modele odwzorowań zadań w transakcje:

1. ZZ i ZZR przyjmują postać jednej transakcji,
2. ZZ oraz zapytania sprawdzające czy spełnione są *warunki* reguł przyjmują postać jednej transakcji, *akcje* reguł przyjmują postać odrębnych transakcji,
3. ZZ jest pojedynczą transakcją, zapytania dotyczące warunków i odpowiednie akcje są grupowane w odrębne transakcje,
4. każde zadanie stanowi odrębną transakcję.

Procesy przebiegają w czasie. Ze względu na ich przebieg modele procesów gospodarczych mogą być zakwalifikowane do jednej z trzech grup:

- liniowe - w których kolejne zdarzenia podlegają porządkowi zgodnemu z upływem czasu,
- gałęziowe - umożliwiające predykcję i tworzenie alternatywnych scenariuszy działań,
- cykliczne - powtarzające się w czasie.

Dokładna analiza przebiegu procesów wymaga przechowywania danych z całego

okresu ich trwania. Niezależnie od wybranego modelu dane o przebiegających procesach należy w systemie przechowywać w taki sposób, aby móc prześledzić historię lub określić spodziewane przyszłe wartości. Wynika z tego, że atrybuty, których wartości opisują przebieg procesu muszą być przechowywane z zachowaniem ich czasowego kontekstu. Jest to warunek konieczny użycia mechanizmów zadawania pytań czasowych oraz czasowych mechanizmów wnioskujących. Przechowywanie wartości z różnych okresów czasu dotyczące tego samego atrybutu w tradycyjnych bazach danych nie jest ani wygodne, ani łatwe. Najodpowiedniejszą technologią w tym wypadku wydają się być czasowe (temporalne) bazy danych. Ich użycie pociąga jednak za sobą konieczność modyfikacji tradycyjnego modelu danych.

Jednym z najczęściej stosowanych modeli jest indeksowanie czasem poszczególnych zmiennych przez dodanie do wartości atrybutu (zmiennej) odpowiedniego rozszerzenia czasowego definiowanego jako trójka (tx, td, tv) gdzie:

tx - oznacza czas, wykonania transakcji zmieniającej wartość zmiennej (ang. transaction time)

td - oznacza czas podjęcia decyzji o zmianie wartości zmiennej co pozwala na określenie właściwej (rzeczywistej) kolejności zdarzeń (ang. decision time),

tv - oznacza czas w którym w rzeczywistym świecie nastąpiła lub nastąpi zmiana wartości zmiennej (ang. valid time).

Bazy danych wyposażone w mechanizmy obsługi danych czasowych: ich przechowywania, manipulowania, odzyskiwania są lepszym narzędziem odwzorowania rzeczywistości gospodarczej aniżeli tradycyjne- nieczasowe BD.

Ze względu na dynamiczny charakter procesów zachodzących w organizacji gospodarczej chcąc opisać je przy pomocy zbioru reguł należy jednocześnie z danymi zmodyfikować również reguły w taki sposób, aby możliwe było wykonywanie dzięki nim działań na danych dotyczących przeszłości i przyszłości. Wykonanie akcji związanej z uaktywnianą regułą często może nie tylko powodować zmiany w danych bieżących, lecz również aktualizować je wstecz lub wprzód. Czasowy aspekt wykonywania reguł uwidacznia się gdy:

- podjęta zostaje decyzja dotycząca przeszłości powodująca zmianę ustalonych wcześniej reguł,
- podjęta zostaje decyzja o zmianie wartości atrybutów w przeszłości (ang. retroactive update) lub przyszłości (ang. proactive update)
- do zbioru reguł wprowadzona zostaje reguła działająca w mocą wsteczną (ang. retroactive rule) lub działająca w przyszłość (ang. proactive rule).

Kolejnym czasowym aspektem jest odwoływanie się do czasu w treści warunków opisujących regułę.

Pamiętać należy także o tym, że moduł zarządzający zbiorem reguł również powinien być wyposażony w procedury umożliwiające przechowywanie i odtwarzanie historii zmian reguł, itp.

Każda z tych reguł może zawierać w sobie czas jako czynnik ograniczający ich stosowanie. Problemy pojawiają się wtedy, gdy na skutek popełnionych wcześniej błędów, pomyłki, zmiany decyzji, zmiany przepisów, itp, ulec musi zmianie warunek czasowy w regule. Taka zmiana może spowodować realizacje zupełnie innego ciągu zdarzeń.

Może to doprowadzić do obliczenia nowych wartości atrybutów, uaktywnienia innych reguł, zastosowania innych ograniczeń, itd. Zmiany wartości przeszłych przez reguły mogą być źródłem wielu nieznanych dotąd problemów obsługi baz danych. Jednym z nich jest problem pętli czasowych. Oznacza to, iż uaktywnienia reguły, może być spowodowane określonymi wartościami zmiennych, a wykonanie akcji związanej z regułą spowoduje iż znikną przesłanki do jej realizacji, lub wręcz spowodują deaktywację samej reguły jeśli

przesłanki były kombinacją wartości zmiennych i wartości czasowych.

5. Zakończenie

Analiza stosowanych metodyk tworzenia RSI pozwala stwierdzić, że podstawowym ich mankamentem jest nie tylko zbyt niski poziom abstrakcji ujmowania rzeczywistości, nie pozwalający uporać się ze złożonością tworzenia rozproszonego systemu informatycznego ale także problem właściwych założeń ontologicznych, tzn. takich, które ustalają co istnieje w modelowanej dziedzinie i co odwzorowuje się w postaci systemu informatycznego. Zagadnienie to wymaga pogłębionych badań.

W pracy przedstawiono zarys metodyki tworzenia RSI przyjmującej jako punkt wyjścia analizę procesów gospodarczych zachodzących w organizacji. Do realizacji procesów zaproponowano deklaratywny sposób ich reprezentacji jaki umożliwia wykorzystanie reguł, w tym reguł czasowych. Wydaje się, że zaproponowane podejście oparte na modelowaniu procesów gospodarczych może być wygodniejszym narzędziem tworzenia rozproszonych systemów informacyjnych.

Literatura:

[AuBr94] Aue A., Breu M.: Distributed information systems: an advanced methodology. IEEE Transactions on Software Engineering vol. 20 no. 8 1994 s. 594 - 605

[BGZ93] Bijałd M., Gwóźdź M., Zieliński K.: Rozproszone przetwarzanie w systemach otwartych. Model ANSA. Informatyka nr 3 1993 s. 17 - 22

[CJL91] Carey M., Jauhari R., Livny M.: On transaction boundaries in active databases: a performance perspective. IEEE Transactions on Knowledge and Data Engineering vol. 3 no. 3 1991 s. 320 - 329

[GaEt95] Gal A., Etzion O.: Maintaining data-driven rules in databases. Computer vol. 28 no. 1 1995 s. 28 - 38

[Morg84] Morgenstern M.: Constraint equations: declarative expression of constrains with automatic enforcement. w: Proceedings Very Large Data Base, Morgan Kaufmann, San Mateo 1984 s. 291 - 300

[PeLo94] Petrounias J., Loucopoulous P.: A rule-based approach for design and implementation of IS. w: Proceedings of 4th International Conference on Database Technology UK 1994

[SnAh86] Snodgrass R., Ahn J.: Temporal databases. Computer vol. 19 no. 9 1986 s. 35 - 42

Carl Erickson
Grand Valley State University
Tadeusz Wilusz
Akademia Ekonomiczna w Krakowie

USENET w procesie dydaktycznym

Streszczenie

W referacie omówiono pokrótce czym jest USENET, jakie są techniczno - organizacyjne uwarunkowania wykorzystania tej sieci w dydaktyce oraz przedstawiono najistotniejsze doświadczenia z wykorzystania sieci USENET w dydaktyce na uniwersytecie Grand Valley w Michigan (USA). Doświadczenia te dowodzą, że USENET, podobnie jak inne usługi dostępne w sieciach komputerowych, ma bardzo wiele do zaoferowania dydaktyce. Istniejące przeszkody na drodze do powszechnego korzystania z tej oferty mają zarówno charakter techniczny (braki w infrastrukturze technicznej) jak i organizacyjny. Do tej drugiej kategorii nader często należy zaliczyć luki w wiedzy osób odpowiedzialnych za proces dydaktyczny zarówno na temat rzeczywistych wymogów technicznych (często są wyolbrzymiane) jak i rzeczywistych, możliwych do osiągnięcia korzyści oferowanych dydaktyce przez sieci komputerowe (często niedoceniane). Przekazując praktyczne doświadczenia szkoły amerykańskiej autorzy liczą, że przyczynią się one w jakimś stopniu do zdynamizowania podobnych procesów w polskich szkołach i placówkach dydaktycznych.

1. Czym jest USENET

USENET w kategoriach usług sieciowych można traktować jako rozproszony biuletyn informacyjny. W Polsce znacznie szerzej znany jest biuletyn sieci BITNET o nazwie LISTSERV, Ma to przyczynę natury historycznej jako że w 1990 roku sieć BITNET jako pierwsza stała się powszechnie dostępna dla polskich użytkowników. Podstawowa różnica pomiędzy biuletynami USENET i LISTSERV polega na tym, że uczestnik listy dyskusyjnej LISTSERV otrzymuje wszystko co pojawia się w ramach grupy tematycznej, do której się zapisał, podczas gdy uczestnik forum USENET sam decyduje, które wiadomości go interesują. Jest to ewidentna zaleta. Innym poważnym atutem USENET jest to, że proces subskrypcji, stanowiący podstawowy mechanizm LISTSERV'a, został zastąpiony lokalnym, a zatem bardzo szybkim, procesem przeglądania spisu artykułów. Ma to szereg ewidentnych zalet zarówno od strony użytkownika jak i od strony administratora sieci. Z tego też powodu największe listy dyskusyjne LISTSERV'a są również dostępne w USENET (np. PACS-L oraz LIBREF-L).

Za punkt wyjścia do definicji usług sieciowych takich jak LISTSERV i USENET można również przyjąć użytkowników, a nie system komputerowy. Tak czyni Gene Spafford z Purdue University (jeden z autorytetów w tej dziedzinie), który definiuje USENET jako zbiór ludzi, którzy wymieniają informację z siecią USENET [6]. Przy takiej definicji interesującym jest, jak wielki jest to strumień informacji?

Przechodząc do charakterystyki ilościowej należy zacząć od konstatacji, że obecnie z sieci USENET korzysta przeszło 290 000 ośrodków. Podstawowe statystyki dostępne za pośrednictwem źródeł takich jak [5] mówią, że w marcu 1995 sieć USENET można było scharakteryzować przy pomocy danych zebranych w poniższej tabelce:

Dzienna liczba nowych artykułów:	114,300
Średnia objętość nowych artykułów dziennie:	365 megabajtów
Liczba użytkowników:	18,120,000
Liczba aktywnych grup dyskusyjnych z nielimitowaną dystrybucją na cały świat	1448
Aktywne grupy z limitowaną dystrybucją	2800
Lokalne grupy dyskusyjne	tysiące

W powyższych danych statystycznych liczba użytkowników określa ludzi którzy czytają, bądź wysyłają informację do sieci USENET. Zarówno cały INTERNET, jak i USENET, który jest jego częścią rozwijają się w zawrotnym tempie wskazującym, że właśnie na naszych oczach następuje radykalna zmiana technologii komunikowania się ludzi między sobą.

Z punktu widzenia zasobów informacyjnych USENET może być postrzegany jako hierarchia grup tematycznych. Tradycyjne, główne grupy tematyczne sieci USENET posiadające dystrybucję o zasięgu ogólnoświatowym to:

comp komputery i wszystko co się z nimi łączy

news siec USENET i oprogramowanie sieci tego typu

rec hobby i inne "rekreacyjne" tematy

sci nauki podstawowe

soc forum socjologiczne - dyskusje na temat różnych kultur

talk forum dyskusji na mocno kontrowersyjne tematy

misc wszelkie inne od w/w tematów

Inne, duże struktury tematyczne dostępne w sieci USENET:

bit dostęp do LISTSERV'u sieci BITNET

alt tzw. grupy alternatywne - tematy, które nie są dostępne we wszystkich ośrodkach

bionet tematy związane z naukami biologicznymi

Oprócz wyżej wymienionych, w sieci USENET możemy mieć do czynienia z lokalnymi i regionalnymi strukturami grup tematycznych, których jest literalnie tysiące.

Identyfikacja grup tematyczny oparta jest na tzw. notacji "kropkowej" (np., ***comp.graphics*** lub ***soc.culture.iranian***) która dobrze odzwierciedla odwzorowanie struktury grup tematycznych na hierarchiczną strukturę plików w sieciowym systemie operacyjnym. Każdy artykuł wysyłany do sieci ma jednoznaczny identyfikator wyznaczany na podstawie identyfikatora autora, ośrodka z którego pochodzi oraz numeru kolejnego w obrębie danego ośrodka. Dla użytkowników lokalnych ośrodka, z którego pochodzi artykuł jego numer kolejny jest wystarczającym identyfikatorem, jako że pozostałe części stanowią w takim przypadku wartości domyślne.

Z systemowego punktu widzenia, oprogramowanie obsługujące sieć USENET jest zaliczane do tzw. oprogramowania grupowego (groupware). Termin ten, bardzo źle brzmiący po polsku, oznacza oprogramowanie, które w środowisku sieciowym pozwala współpracować grupie ludzi, którzy z racji różnorodnych definicji zewnętrznych stanowią zespół pracujący nad tym samym zagadnieniem. Ze względu na relacje pomiędzy nadawcą a odbiorcą, oprogramowanie komunikacyjne tego typu można podzielić na trzy podstawowe kategorie:

1. jeden-do-jednego (poczta elektroniczna, "pogawędki" typu uniksowego TALK);
2. jeden-do-wielu (elektroniczne listy adresowe);

3. wielu-do-wielu (listy dyskusyjne, biuletyny informacyjne)

W świetle dotychczasowych rozważań, USENET z oczywistych powodów zalicza się do trzeciej z wymienionych wyżej kategorii.
Na zakończenie ogólnych rozważań o sieci USENET warto przytoczyć definicję zawartą w pracy [6]. Jej autor, definiuje czym jest USENET poprzez listę stwierdzeń na temat czym on nie jest . Dwa najważniejsze z nich, to:

1. USENET nie jest siecią ograniczoną do organizacji akademickich ani nie jest siecią opartą na konkretnym systemie (jak np. UNIX). Z USENET'u korzysta więcej ośrodków komercyjnych aniżeli szkolnych, czy akademickich. Równocześnie, oprogramowanie komunikacyjne konieczne do pracy z tą siecią jest dostępne praktycznie dla wszystkich liczących się platform systemowych takich jak DOS, MacOS, DEC VMS oraz różne systemy operacyjne dużych komputerów IBM.
2. USENET nie jest organizacją w tym sensie, że nie istnieje centralny ośrodek dyspozytorsko-koordynacyjny. Będąc z definicji systemem totalnie rozproszonym cała sieć składa się prawie wyłącznie z ośrodków - wolontariuszy, które je właśnie tworzą, ale dzięki temu gwarantuje dostęp każdemu zainteresowanemu, który ma dostęp do komputera przyłączonego do INTERNETU;

Ze stworzeniem tego typu sieci były rzecz oczywista problemy nie występujące w sieciach o mniej "anarchistycznej" koncepcji. Przykładowo, z racji założenia braku centralnego zarządzania siecią musiał zostać opracowany specjalny algorytm dystrybucji artykułów, pozwalający unikać sytuacji konfliktowych takich jak np. wielokrotne wysyłanie tego samego artykułu do danego ośrodka. Nie mniej jednak rosnąca popularność USENET'u dowodzi wyjątkowej trafności przyjętych założeń.

2. Warunki techniczno-organizacyjne korzystania z sieci USENET

W tym punkcie zajmiemy się krótką charakterystyką wymagań sprzętowo-programowych oraz organizacyjnych warunkujących korzystanie z możliwości sieci USENET.

Wymagania software'owe można najkrócej skwitować stwierdzeniem, że komputer, który ma zapewnić korzystanie z USENET'u musi być wyposażony w oprogramowanie umożliwiające korzystanie z usług sieci INTERNET. Stwierdzenie takie może jednakże okazać się niewystarczające, gdyż sieć INTERNET oferuje szeroką gamę usług oferowanych przez różne ośrodki (zwłaszcza komercyjne) w różnych zestawach. W związku z tym, może się okazać, że komputer prawidłowo skonfigurowany w aspekcie korzystania z poczty elektronicznej nie zapewni jego użytkownikowi dostępu do usługi INTERNET'u jaką jest USENET.

Podstawowym oprogramowaniem warunkującym korzystanie z sieci USENET jest oprogramowanie realizujące transfer informacji pomiędzy sąsiednimi ośrodkami sieci USENET. Pierwotnie było to oprogramowanie realizujące tzw. protokół UUCP (Unix to Unix CoPy), często w związku z tym nazywane "tradycyjnym" do komunikacji poprzez łącza telefoniczne [4,8]. Wraz z rozwojem sieci INTERNET bazującej na protokole TCP/IP protokół transportu wiadomości oznaczany symbolem NNTP (Network News Transport Protocol) mocno zyskał na popularności wypierając zasłużony UUCP. Sytuacja na dzień dzisiejszy wygląda tak, że protokołem wymiany wiadomości pomiędzy dużymi ośrodkami jest NNTP, podczas, gdy połączenie małych ośrodków lub komputerów osobistych z dużymi centrami INTERNET'u jest nadal realizowane albo w oparciu o protokół UUCP albo w oparciu o odmianę protokołu IP dla łączności modemowej oznaczanym jako SLIP (Serial Line Internet Protocol). Oprogramowanie realizujące protokół UUCP jest najczęściej oferowane nieodpłatnie przy zakupie systemu

operacyjnego UNIX w związku z tym, właśnie w środowisku maszyn uniksowych ten protokół nadal cieszy się popularnością tym bardziej, że de facto jest to zespół protokołów oferujący oprócz transportu wiadomości usługi takie jak poczta elektroniczna oraz transfer plików (FTP). Oprogramowanie dla NNTP jest łatwo dostępne w licznych archiwach oprogramowania dostępnych zdalnie za pośrednictwem sieci INTERNET .

Następny poziom niezbędnego oprogramowania to oprogramowanie zarządzające wiadomościami (tzw. news software). Najnowszą generacją oprogramowania tego typu jest pakiet o nazwie C News [7]. To oprogramowanie pełni funkcję systemu operacyjnego sieci USENET. Jest ono dostępne zarówno w archiwach programów ważniejszych węzłów sieci USENET jak i w archiwach ftp sieci INTERNET.

Ostatnią grupę programów stanowi oprogramowanie umożliwiające użytkownikowi dostęp do zasobów i usług sieci USENET. Popularnie, w slangu informatycznym , ta grupa programów jest określana mianem "czytników wiadomości" (news readers). Współczesne czytniki posiadają liczne udogodnienia zapewniające pamiętanie, które artykuły użytkownik przeczytał, śledzenie wątków dyskusji, wyszukiwanie poprzez tytuł lub autora, wysyłanie nowych artykułów, itd. W ostatnich latach powstała duża ilość oprogramowania tej kategorii, ale ciągle najpopularniejszymi czytnikami wiadomości są programy wymienione poniżej:

readnews	Oryginalny czytnik wiadomości sieci USENET
rn	Program opracowany w 1984, ciągle ulepszany. De facto standard w tej klasie oprogramowania
trn	Mutacja czytnika rn zapewniająca wygodniejszy dostęp do artykułów na ten sam temat.
xrn	Wersja rn dla środowiska X Windows systemu UNIX.

Większość aktualnie dostępnych wersji czytników ma wbudowaną obsługę protokołu NNTP co pozwala używać tych programów w trybie pracy klient-serwer. Ma to istotne znaczenie w dużych organizacjach gdyż eliminuje konieczność powielania plików z tymi samymi wiadomościami na komputerach uczestników sieci USENET. W praktyce wygląda to tak, że jeden komputer pełni funkcję serwera wiadomości a czytniki oparte na NNTP uruchamiane na komputerach osobistych i stacjach roboczych użytkowników zapewniają dostęp do wiadomości bez konieczności zbędnej w tej architekturze replikacji plików. Kody źródłowe rozmaitych czytników można skopiować z ogólnie dostępnych archiwów ftp sieci INTERNET.

Zabrzmi to pompatycznie, ale pierwszym krokiem na drodze prowadzącej do uczestnictwa w sieci USENET jest świadoma chęć dołączenia do istniejącej społeczności sieciowej jako jej pełnoprawna cząstka świadoma swoich praw i obowiązków. Dla szkół już przyłączonych do sieci INTERNET, dodatkowe koszty infrastruktury sieciowej są małe lub wręcz żadne. Ciągle typową infrastrukturę techniczną sieci komunikacyjnej stanowi łączność modemowa z typową szybkością transmisji 9600 bps. Szkoły nie mające dotychczas dostępu do INTERNET'u powinny zainteresować się istniejącymi zarówno w Stanach Zjednoczonych jak i w Polsce programami pomocy stworzonymi specjalnie dla tego celu. W USA jest to program fundacji National Science Foundation o nazwie "Connections to NSFNET". Granty otrzymane z tego programu pokrywają koszty przyłączenia szkoły do INTERNET'u poprzez właściwą sieć regionalną. Program ten ma głównie na celu pomoc szkołom podstawowym i średnim[3]. W Polsce podobne zadania ma program "INTERNET dla szkół", aczkolwiek jego zasobność póki co nie dorównuje amerykańskiemu odpowiednikowi.

Następnym zasobem niezbędnym do właściwego dołączenia szkoły do sieci USENET jest odpowiedniej klasy komputer przeznaczony do pełnienia funkcji węzła sieci

dla danej jednostki. Na dzień dzisiejszy powinien to być komputer minimum klasy stacji roboczej z minimum 16 MB pamięci operacyjnej oraz dużą przestrzenią wolnej pamięci dyskowej. Wielkość wolnej pamięci dyskowej decyduje o okresie przechowywania napływających wiadomości, a tym samym wyznacza czasokres dostępności informacji dla użytkowników sieci. Im mniejszy dysk, tym krótszy musi być okres przedawnienia dostępnej informacji tak, aby uniknąć przepełnienia dysku w trakcie napływania nowych wiadomości. Drastyczny spadek cen pamięci dyskowych pozwala postawić postulat udostępnienia minimum 1 GB przestrzeni dyskowej dla potrzeb USENET'u, co powinno pozwolić na dwutygodniowy okres przechowywania podstawowych hierarchii informacji. Jeżeli decydujemy się na pracę w trybie klient-serwer z czytnikami opartymi na protokole NNTP, to celowym staje się zalecenie uczynienia z tego komputera dedykowanego serwera. Niezbędnym uzupełnieniem całej konfiguracji przy coraz większym strumieniu informacji staje się co najmniej jeden modem o dużej szybkości transmisji, bądź, alternatywne, szybkie połączenie z siecią regionalną.

Po omówieniu podstawowych uwarunkowań sprzętowo-programowych czas na słów kilka o niezbędnych zasobach ludzkich. Bardzo często mogą one stanowić największy składnik ogółu kosztów, który przy planowaniu przedsięwzięcia jest najmniej widoczny. Tymczasem jest oczywistością, że ktoś musi skonfigurować i zainstalować system, a następnie administrować jego pracą. Funkcja ta wymaga technicznej bardzo dobrej znajomości wszelkiego rodzaju zagadnień związanych z sieciami komputerowymi tudzież umiejętności administrowania całym systemem. W idealnym przypadku szkoła powinna zatrudnić pracownika technicznego z odpowiednim przygotowaniem i wiedzą, który utrzymywałby funkcjonowanie połączenia z USENET jako usługę z której korzysta cała szkoła. Więcej szczegółowych informacji dotyczących warunków techniczno-organizacyjnych przyłączenia się do USENET znajduje się w [2].

3. USENET jako pomoc w nauczaniu

Obecnie w uniwersytecie Grand Valley State USENET jest dostępny na uniksowym serwerze o nazwie "river" praktycznie dla każdego zainteresowanego bez względu na to, czy jest to pracownik, czy też student. Jest to efektem całkiem niedawnej rozbudowy infrastruktury technicznej uczelnianej sieci komputerowej. Uprzednio, z powodów technicznych korzystanie z dostępu do sieci było przywilejem pracowników, studentów informatyki oraz studentów wyższych lat niektórych kierunków. Zainteresowani mogą za pośrednictwem łączności modemowej korzystać z zasobów sieci z poziomu swoich komputerów domowych. Jednym słowem, uniwersytet pokonał wszystkie przeszkody natury techniczno-organizacyjnej i zapewnia łatwy dostęp do sieci wszystkim zainteresowanym pracownikom jak i studentom.
Wykorzystanie dostępnych usług sieciowych w procesie dydaktycznych rozwija się w GVSU (Grand Valley State University) na dwóch płaszczyznach:

1. W warstwie organizacyjnej do asynchronicznej komunikacji ze studentami, oraz
2. W warstwie merytorycznej, dla osiągnięcia zakładanych celów dydaktycznych w ramach określonych przedmiotów,

Wykorzystanie sieci do komunikacja ze studentami w chwili obecnej staje się już powszechną codziennością, podczas gdy wykorzystanie zasobów informacyjnych dostępnych w sieci w ramach programu zajęć dydaktycznych nadal ma charakter eksperymentów podejmowanych indywidualnie przez prowadzących. Przytoczone tutaj doświadczenia pochodzą z kilkuletnich doświadczeń korzystania z sieci USENET w ramach zajęć z przedmiotów informatycznych prowadzonych przez Katedrę Informatyki i Systemów Informacyjnych (Computer Science and Information Systems Department) przez okres kilku ostatnich lat.

3.1 USENET jako kanał komuni-kacji ze studentami

Jedną z niezwykle użytecznych cech rozproszonej architektury USENET'u jest możliwość tworzenia lokalnych grup dyskusyjnych przez lokalnego administratora. Lokalne grupy dyskusyjne zwykle mają geograficznie ograniczony zasięg. Na przykład około dwunastu grup należących do hierarchii dyskusyjnej Michigan (mi) nie jest szerzej znana i wykorzystywana poza tym stanem. Posiadamy również hierarchię GVSU (gvsu), do której z łatwością możemy dodawać specjalizowane grupy dyskusyjne. Ogólna grupa GVSU nazywa się po prostu "gvsu" i jest ona forum dyskusyjnym dla każdego i na każdy temat. Obecnie znajduje się tam kilka grup dyskusyjnych odnośnie zajęć z informatyki i jedna grupa dyskutująca o Szekspirze (gvsu.english.shakespeare). W obrębie lokalnej hierarchii uczelnianej można zdefiniować odrębną grupę dyskusyjną dla każdego przedmiotu, czy nawet, dla każdej grupy studenckiej w ramach tego samego przedmiotu. Takie rozwiązanie definiuje bardzo wygodny i efektywny kanał publicznej, asynchronicznej komunikacji pomiędzy wszystkimi uczestnikami procesu dydaktycznego: student-do-studenta, student-do-nauczyciela, oraz nauczyciel-do-studenta.

Oto typowy zakres wykorzystania przez nauczyciela wyżej przedstawionych możliwości:

a. Ogłaszanie harmonogramów testów i zadań domowych na początku semestru,
b. Korygowanie błędów i pomyłek w sformułowaniu tematów prac studenckich odkrytych już po ich rozdaniu,
c. Odpowiedź na pytania zadane przez studentów podczas zajęć, na które prowadzący z rozmaitych powodów nie udzielił odpowiedzi w czasie zajęć,
d. Dodatkowe komentarze i wyjaśnienia odnośnie dyskusji mających miejsce na zajęciach,
e. Udostępnianie materiałów z innych grup dyskusyjnych, które według prowadzącego mogą mieć istotne znaczenie dla danego przedmiotu,
f. Działalność czysto informacyjna (możliwość ubiegania się o granty, oferty pracy, seminaria, stypendia, itp.),
g. Zadawanie prac domowych i projektów do samodzielnego opracowania zadań i projektów.

Studenci natomiast zwykle używali klasowych grup dyskusyjnych do następujących celów:

a. Zadawanie nauczycielowi pytań dotyczących projektów i innych prac do samodzielnego wykonania,
b. Dzielenie się z kolegami rozwiązaniami różnorodnych problemów związanych z użytkowaniem komputerów włączając w to osobiste doświadczenia z USENET'em,
c. Proszenie o pomoc w przypadku problemów z dostępem do komputerów,
d. Przekazywanie informacji o możliwościach pracy od swoich pracodawców.

3.2 Wykorzystanie USENET'u do realizacji programu nauczania przedmiotu

Korzyści jakie można osiągnąć poprzez wykorzystanie USENET'u do realizacji celów dydaktycznych przedmiotu można podzielić umownie na dwie zasadnicze kategorie: bezpośrednie jak i pośrednie. Bezpośrednie korzyści są związane z informacjami, które studenci uzyskują z poszczególnych grup tematycznych (newsgroups), związanych ściśle z tematyką danego przedmiotu. Na przykład **comp.arch** doskonale uzupełnia kurs architektury komputerów, zaś **comp.lang.c** jest cennym dodatkiem do kursu programowania w języku C. Korzyści pośrednie wynikają z uczestnictwa studentów w życiu codziennym globalnej "społeczności" użytkowników USENET.

Bezpośrednie korzyści są dość oczywiste, natomiast warto trochę uwagi poświęcić efektom, które należy zaklasyfikować do grupy korzyści pośrednich:

- *Poczucie przynależności do "społeczności sieciowej"*

  Studenci, tudzież pracownicy, zwłaszcza w przypadku małych szkół lub katedr uniwersyteckich szybko zyskują poczucie przynależności do

specyficznej społeczności, w której kryterium podziału jest wyznaczone wspólnotą zainteresowań i gotowością komunikowania się. W ten sposób powstaje wielostronna platforma wymiany informacji i doświadczeń pomiędzy ludźmi z przemysłu, uczelni i szkół oraz agend rządowych. Różnorodność punktów widzenia tych samych problemów w zależności od profesji ma bardzo niewymierny, ale trudny do przecenienia w sensie poznawczym, walor *dydaktyczny*,

- *USENET jako źródło wysokospecjalizowanej i aktualnej wiedzy*
  Informacje dostępne w USENET'cie (czy ogólniej, w jakichkolwiek komputerowych biuletynach informacyjnych) mogą się poszczycić atrybutem aktualności niedostępnym dla tradycyjnych - opartych na druku - technologiach. To samo dotyczy zresztą i łatwości dostępu. Jeśli do tego dodać, że wysokiej klasy specjaliści muszą korzystać z tej technologii aby ciągle nimi być, to w konkluzji mamy do czynienia z sytuacją, w której student ma zwielokrotnione szanse zapoznania się z faktycznym stanem w określonej dziedzinie, a nie z przekazem mniej lub bardziej historycznym.

- *Krytyczny stosunek do treści przekazu*
  Nie ma rzeczy idealnych więc i USENET posiada pewne wady. Jedną z nich jest jakby odwrotna strona medalu zalety jaką jest łatwość dostępu. Olbrzymia objętość dostępnych informacji generowana przez szybko rosnącą liczbę użytkowników, czy też uczestników sieci nie zawsze musi, i nie zawsze oznacza wysoką jakość tekstów publikowanych w sieci. Wręcz odwrotnie spory procent stanowi "szum informacyjny". Ale i to w aspekcie dydaktycznym okazuje się być zaletą, gdyż student musi posiąść umiejętność odróżniania w tej masie rzeczy wartościowych od rzeczy miałkich lub wręcz dezinformacji na zadany temat.

- *Wzrost zainteresowania przedmiotem*
  Jest to efekt konstatacji, że tematyka przedmiotu, poruszane w jego ramach problemy interesują szerokie grono ludzi na całym świecie, a nie są li tylko wymysłem prowadzącego. Studenci GVSU szczególnie mocno reagowali w ten sposób widząc, że problemy którymi zajmują w ramach danego przedmiotu są tymi samymi problemami, które muszą rozwiązywać ludzie z tzw. praktyki zawodowej.

3.3 USENET w opinii studentów

Wielu studentów podkreślało, że rozpoczynając pracę z USENET'em czuli się przytłoczeni ogromem dostępnej informacji. Z tego też powodu ich pierwsze prace oparte na wykorzystaniu USENET'u były naszpikowane wszelkiego autoramentu pytaniami związanymi z tym środowiskiem, na które nauczyciel musiał pracowicie odpowiadać. Nagrodą dla nauczyciela za ten dodatkowy wysiłek była satysfakcja zawodowa w końcowej fazie semestru kiedy mógł przysłuchiwać się w klasie merytorycznym dyskusjom studentów na wysokim poziomie osiągalnym w takiej skali tylko dzięki uprzedniemu uczestnictwu studentów w dyskusjach na forum USENET. Nawet najsłabsi studenci zgodnie podkreślali, że doświadczenia wyniesione z uczestnictwa w USENET w dużym stopniu ułatwiło im zadowalające opanowanie materiału objętego programem przedmiotu. Jest rzeczą oczywistą i zrozumiałą, że w dużej populacji studentów byli również studenci niezadowoleni z wprowadzanych do przedmiotu innowacji ale w zdecydowanej większości opinie studenckie, zawarte w rutynowej ewaluacji przedmiotu na koniec semestru, były pozytywne. Najbardziej typowe wypowiedzi studentów przytoczono niżej:

> *"Praca z USENET'em była dla mnie odkryciem nowego, fascynującego świata, w którym problem zgłoszony przez kogoś z*

najbliższego sąsiedztwa może być rozwiązany przez osobę na drugim końcu świata! Bardzo mnie ujęła nieformalność kontaktów ludzkich pod auspicjami USENET. Wszyscy mówią sobie po imieniu."

"Jestem absolutnie przekonany, że USENET jest idealnym środowiskiem do studiowania sieci komputerowych."

" Dwukrotnie na zajęciach była mowa o przełączaniu kontekstowym. Nie rozumiałam dokładnie dyskusji na ten temat ale muszę powiedzieć, że większość fachowej terminologii użytej w dyskusji na ten temat była mi znana."

"Zacząłem od listy skrótów i oznaczeń, których znaczenia nie znałem. To jest wspaniałe widzieć, że problemy o których mówimy w klasie są dyskutowane przez ludzi z całego świata."

4. Alternatywne formy komunikacji elektronicznej w procesie nauczania.

Grupy dyskusyjne USENET są jedną z kilku form komunikacji elektronicznej dostępnych obecnie i wykorzystywanych w GVSU w procesie dydaktycznym. Te inne, to oczywiście:

- *Poczta elektroniczna*
  Poczta elektroniczna może w zdecydowany sposób poprawić dostępność nauczycieli dla studentów. Ponieważ wysyłanie i czytanie poczty są czynnościami niezależnymi, student może przesłać pytanie w dowolnym czasie a nauczyciel także w dowolnym czasie może na nie odpowiedzieć. Innym dosyć ciekawym zjawiskiem są studenci milczący na zajęciach ale "gadatliwi elektronicznie".
- *Aliasy pocztowe grup studenckich*
  Aliasy pocztowe grup studenckich mogą być zainstalowane na serwerze river. Adres taki (np. cs452@river.it.gvsu.edu) umożliwia rozesłanie poczty wpływającej na niego do każdej osoby w grupie. Mechanizm ten tworzy dużą nadmiarowość, ale prawdopodobnie jest najpewniejszym środkiem kontaktu z poszczególnymi członkami grupy (ludzie częściej czytają pocztę elektroniczną niż korzystają z klasowej listy dyskusyjnej).
- *World Wide Webb (WWW)*
  WWW pozwala każdemu, kto tylko posiada swoje konto na riverze publikować informacje w Internecie. W 1995 roku został rozpoczęty eksperyment z dystrybuowaniem notatek kursowych i zadań w sieci za jego pośrednictwem. Jedna z głównych i oczywistych zalet WWW, pomijając rozsyłanie informacji jest fakt wspierania wielu formatów danych (tekst, grafika, dźwięk).
- *Ogólnie dostępne archiwa plików (FTP)*
  Sieć Novell w GVSU i host unixowy ogólnego przeznaczenia (river) pozwalają w GVSU na dostęp do określonych części sieciowego systemu plików każdemu użytkownikowi. Każdy z instruktorów może wykorzystać tę możliwość do umieszczania notatek z zajęć w wersji on-line. Studenci wiedzą, że po wykładzie są one dostępne w określonym miejscu archiwum plików ogólnie dostępnych.

5. Podsumowanie

USENET oferuje zarówno studentom jak i kadrze nauczającej stanie się cząstką ogólnoświatowej, sieciowej społeczności. Studenci uczą się krytycznego podejścia do lektury publikacji, uzyskują łatwy dostęp do najbardziej aktualnej technicznej informacji oraz komunikując się z ludźmi o podobnych zainteresowaniach weryfikują poprawność i kompletność swojej wiedzy z określonej dziedziny. Podsumowując przedstawiony w referacie szkic na temat doświadczeń GVSU z wykorzystywania możliwości sieci komputerowych w dydaktyce można dodać, że w zasadzie nie istnieje problem czy należy te

innowacje do procesu dydaktycznego wprowadzać. Istniejący faktycznie problem, to problem jak zrobić to najlepiej. Przy tym sformułowaniu szczególnej wartości dla zainteresowanych sieciami komputerowymi w dydaktyce szkół nabiera możliwość skorzystania z istniejących w tej materii doświadczeń praktycznych. I to ostatnie stwierdzenie najlepiej oddaje intencje niniejszego opracowania.

LITERATURA

[1] "Class Discussion by Computer: A Case Study", A. Michael Berman, 1992 SIGCSE Technical Symposium, March 1992.

[2] "How to become a USENET site", Jonathan Kamens, MIT Information Systems/Athena, monthly USENET posting in news.announce.newusers.

[3] "NSF907 - Connections to NSFNET Program Guidelines", National Science Foundation, July 26, 1990.

[4] "Managing UUCP and Usenet", Tim O'Reilly and Grace Todino, O'Reilly & Associates Inc. 10th Edition, 1990.

[5] " USENET READERSHIP SUMMARY REPORT", Brian Reid, DEC Network Systems Laboratory, monthly USENET posting in news.lists.

[6] "What is Usenet?", Gene Spafford, Purdue University, monthly USENET posting in news.announce.newusers.

[7] "USENET Software: History and Sources", Gene Spafford, Purdue University, monthly USENET posting in news.announce.newusers.

[8] "Using UUCP and Usenet", Grace Todino and Dale Dougherty, O'Reilly & Associates Inc., 1986.

[9] "List of Active Newsgroups", David C Lawrence, monthly USENET posting in news.lists

Edward Mazur
PKP - Centrum Informatyki Kolejnictwa

Sieć transmisji danych Polskich Kolei Państwowych KOLPAK

Infrastruktura telekomunikacyjna PKP

Sieć linii telekomunikacyjnych będących własnością Polskich Kolei Państwowych obejmuje swym zasięgiem cały obszar kraju. Wzdłuż tras kolejowych liczących ok. 23 tys. kilometrów zbudowane są tzw. szlakowe linie telekomunikacyjne. Około 15 tys. km tych linii to dalekosiężne linie kablowe, które łączą wszystkie ważne ośrodki państwowe i kolejowe, w tym stacje graniczne. Resztę stanowią linie napowietrzne, które ze względu na niską jakość realizowanych połączeń i dużą awaryjność sukcesywnie są zastępowane kablami.

Sieć kablowa przewodowa składa się głównie z telefonicznych kabli dalekosiężnych, symetrycznych, z żyłami miedzianymi w izolacji papierowo-powietrznej i powłokach ołowianych. W latach osiemdziesiątych zaczęto włączać do eksploatacji linie zbudowane z kabli o polietylenowej izolacji żył miedzianych i powłoce aluminiowej. Podstawowymi urządzeniami teletransmisyjnymi w zakresie telefonii wielokrotnej są urządzenia analogowe 12-krotne. Na początku roku 1994 została włączona do sieci telekomunikacyjnej PKP linia światłowodowa Północ - Południe, o długości ok. 1300 km. Przebiega ona od Koszalina do Zebrzydowic, przez Słupsk, Gdańsk, Bydgoszcz, Toruń, Warszawę, Radom, Kielce , Kraków i Katowice, z odgałęzieniem do Bielska Białej. Linię tę zbudowano z kabli optycznych (12- i 8-włóknowych) jednomodowych o długości fali 1300 nm i wyposażono w urządzenia teletransmisyjne systemu PDH o przepływności binarnej 140 Mbit/s. Na odcinkach linii o mniejszym natężeniu ruchu telefonicznego (Koszalin - Gdańsk, Czechowice - Bielsko Biała i Czechowice - Zebrzydowice) zastosowano system 34 Mbit/s. Planuje się, że w następnych latach zostaną zbudowane kolejne linie światłowodowe, które wraz z istniejącą linią Północ - Południe utworzą sieć przedstawioną na rys. 1. Będą zastosowane światłowody jednomodowe o długości fal 1300 nm i 1500 nm. Będą to z reguły kable 12 - włóknowe. Zastosowany będzie system teletransmisyjny SDH o przepływności binarnej 155 Mbit/s, zapewniający scentralizowany nadzór i programowe sterowanie siecią. W stosunkowo nielicznych przypadkach przewiduje się stosowanie radiolinii cyfrowych o przepływności 2 lub 2 x 2 Mbit/s. Do przekazywania danych pomiędzy węzłami pakietowej sieci Kolpak będą wykorzystywane łącza 2,048 Mbit/s i 64 kbit/s.

Na istniejących liniach telekomunikacyjnych utworzone zostały różnego rodzaju sieci łączności, związane z prowadzeniem ruchu kolejowego i zarządzaniem przedsiębiorstwem PKP, jak np. telefoniczne sieci dyspozytorskie, stacyjno - ruchowe, szlakowe itp. Jedną z takich sieci jest telefoniczna tzw. sieć ogólnoeksploatacyjna, która obejmuje cały teren przedsiębiorstwa PKP i umożliwia dowolnemu abonentowi uzyskanie połączenia z innymi abonentami, odpowiednio do przyznanych uprawnień. Z sieci tej korzysta ok. 120 tys. stanowisk pracy przyłączonych do ok. 1200 central. Użytkownikami są pracownicy wszystkich służb technicznych i administracyjnych PKP. Ze względu na potrzebę współpracy z klientami i przedsiębiorstwami niekolejowymi, centrale tej sieci mają połączenia z centralami telefonicznymi użytku publicznego. Za pośrednictwem sieci ogólnoeksploatacyjnej realizowane są także międzynarodowe

połączenia z innymi europejskimi zarządami kolejowymi.

W ramach modernizacji sieci telekomunikacyjnej PKP planuje się zainstalowanie w niedalekiej przyszłości 11 nowych wielousługowych (ISDN) głównych central cyfrowych (w Warszawie, Lublinie, Krakowie, Katowicach, Gdańsku, Wrocławiu, Poznaniu, Szczecinie, Łodzi, Bydgoszczy i Olsztynie).

Sprawne działanie urządzeń telekomunikacyjnych PKP zapewnia dobrze zorganizowana służba automatyki i telekomunikacji, której pracownicy działają na obszarze całego kraju.

Systemy informatyczne PKP jako użytkownicy sieci KOLPAK

Sieć Kolpak została utworzona dla potrzeb głównych systemów informatycznych PKP:

- systemu kierowania przewozami i zarządzania przedsiębiorstwem PKP (SKPZ), który w pierwszej fazie (1995-96) będzie zawierał ok. 4500 terminali, 33 komputery obszarowe i komputer centralny; system ten znajdujący się w początkowej fazie realizacji będzie generował duże strumienie ruchu, wymagał krótkich czasów transmisji oraz przyłączy do sieci, głównie synchronicznych o dużej przepustowości (do 2 Mbit/s),
- systemu rezerwacji miejsc i sprzedaży biletów (KURS '90), zawierającego komputer centralny i docelowo ok. 1000 terminali; system ten jest już eksploatowany i korzysta z sieci Kolpak (obecnie ok. 300 terminali); wymagane są krótkie czasy transmisji i przyłącza głównie synchroniczne o przepustowości 14,4 kbit/s,
- systemu poczty elektronicznej - początkowo 2000, a docelowo 20 000 użytkowników; system jest już eksploatowany (zainstalowanych jest ok. 400 skrytek) i pokrywa się częściowo z systemami SKPZ i KURS '90 - terminale tych systemów mogą być jednocześnie abonentami poczty elektronicznej; czas transmisji nie jest w tym przypadku parametrem krytycznym.

Sieć transmisji danych X.25 KOLPAK

Przyznanie kredytu z Banku Światowego oraz zniesienie embarga na nowoczesną technologię dało szansę utworzenia w krótkim czasie dla PKP nowoczesnej, niezawodnej i jednorodnej sieci transmisji danych.

Otwarty przetarg na dostawę i uruchomienie sieci wygrała firma SPRINT INTERNATIONAL z USA. Kontrakt został podpisany 10 listopada 1992 roku, a instalację rozpoczęto w marcu 1993 r.

Jako zasadę działania sieci przyjęto tzw. komutację pakietów według zaleceń CCITT X.25. Tego rodzaju sieci charakteryzuje wysoka niezawodność, łatwość utrzymania oraz bardzo duża wierność przekazywania informacji. Z tego względu znalazły one zastosowanie w większości europejskich zarządów kolejowych.

Wraz z siecią Kolpak zakupiono system poczty elektronicznej oraz modemy dla połączeń międzywęzłowych.

Topologia sieci

Szczegółowa topologia sieci Kolpak została opracowana w Centrum Naukowo - Technicznym Kolejnictwa. W topologii przewidziano:

- **10 węzłów głównych (WG) TP4954**, które są zainstalowane w ośmiu siedzibach Dyrekcji Okręgowych Kolei Państwowych (DOKP) oraz w Bydgoszczy, przy czym w Warszawie są dwa węzły główne; pozostałe siedziby DOKP to: Lublin, Kraków, Katowice, Gdańsk, Wrocław, Poznań, Szczecin,
- **75 węzłów regionalnych (WR) TP4944** - rozmieszczonych w miejscach zainstalowania komputerów obszarowych SKPZ oraz w innych miejscach koncentracji ruchu,
- **212 węzłów lokalnych (WL) TP8010-16**.

Układ połączeń między węzłami sieci Kolpak został zaprojektowany przy założeniu, że powinien on odpowiadać głównym strumieniom ruchu, zapewniać wystarczającą

niezawodność (możliwość tworzenia różnych dróg obejściowych) oraz uwzględniać istniejącą strukturę sieci telekomunikacyjnej PKP (rozmieszczenie ośrodków telekomunikacyjnych PKP i połączeń między nimi). Odpowiednio do tych założeń każdy węzeł główny ma połączenie z węzłem w Warszawie, a ponadto utworzone są połączenia między węzłami głównymi zamykające obwód wieloboku oraz połączenia skrośne Katowice - Gdańsk i Katowice - Lublin.

Węzły regionalne są przyłączone bądź bezpośrednio do węzłów głównych bądź co najwyżej tylko przez jeden inny węzeł regionalny. Węzły te mają co najmniej dwie drogi dojścia do węzłów głównych. Węzły lokalne są łączone w łańcuchy równoległe do przebiegu linii kolejowych (wzdłuż szlaków telekomunikacyjnych), co zapewnia im dostęp do wyższych poziomów sieci również co najmniej dwiema drogami. Dążyć się będzie, aby węzły lokalne miały bezpośrednie połączenia z węzłami regionalnymi, bądź głównymi. Omawiane zasady przedstawiono schematycznie na rys. 2.

Lokalizację wszystkich węzłów sieci Kolpak i połączenia między nimi przedstawiono na rys. 3.

W sieci Kolpak początkowa pojemność wyniesie około 9500 portów, wliczając w to porty dla połączeń międzywęzłowych. W miarę potrzeb będzie można rozbudowywać węzły regionalne i główne oraz dostawiać węzły lokalne i zmieniać ich konfigurację.

Parametry węzłów

Wszystkie węzły są wykonane z zastosowaniem technologii 32-bitowej. Węzły kategorii wyższych niż lokalne zapewniają nieprzerwane działanie. Oznacza to, że w przypadku uszkodzenia jakiegoś modułu w węźle, jego działanie przejmuje automatycznie inny moduł nadmiarowy, bez utraty informacji przekazywanych przez sieć. Węzły te mają budowę modularną, umożliwiającą zwiększenie liczby portów odpowiednio do wzrastających potrzeb. Pogląd na niezawodność struktury daje schemat blokowy węzła regionalnego przedstawiony na rys. 4.

Do portów sieci Kolpak można przyłączyć urządzenia działające zarówno w trybie synchronicznym, jak i asynchronicznym, a przyłącza mogą być zarówno trwałe, jak i komutowane.

Rodzina TP 4900

- **Własności programowe:**
  - komutacja pakietów w technice X.25 i "frame relay";
  - dynamiczna rekonfiguracja portów;
  - obsługa portów w standardach X.25, X.32 , X.28, frame relay, X.75, X.31.
- **Własności fizyczne**
  - szybkość transmisji na przyłączach 2,4 kbit/s do 2,048 Mbit/s;
  - przepustowość w pakietach (liczba 128-bajtowych pakietów na sekundę):
    TP 4944: 12 000,
    TP 4954: 12 000,
  - liczba portów o niskiej szybkości (do 64 kbit/s):
    TP 4944: do 96,
    TP 4954: do 800,
  - liczba portów E1:
    TP 4944: do 5,
    TP 4954: do 16,
  - "gorąca" wymiana kart;
  - nadmiar sprzętowy łącznie z redundancją 1:N w kartach liniowych, co daje możliwość automatycznego przełączenia na moduł zapasowy w sposób niezauważalny dla użytkownika;
  - lustrzana organizacja pamięci umożliwiająca działanie "fault tolerant" (odporność na błędy);
  - dysk twardy 40 MB.
- **Charakterystyka PAD**
  - szybkość transmisji na przyłączach: do 64 kbit/s;
  - kody znakowe: ASCII (asynchroniczne), EBCDIC (synchroniczne);
  - interfejsy liniowe: RS 232, V.35, X.21/V.11;
  - akceptowane protokoły: X.3/X.28/X.29, SDLC.

Rodzina TP 8000

W sieci Kolpak stosowane są z tej rodziny urządzenia o symbolu TP 8010-16.

Własności programowe i fizyczne:

- komutacja pakietów w technice X.25;
- możliwość jednoczesnej pracy w 6 różnych protokołach;
- szybkość na przyłączach synchronicznych do 64 kbit/s, a na asynchronicznych do 19,2 kbit/s;
- przepustowość do 400 pakietów na sekundę;
- dysk twardy 40 MB;
- liczba portów 16;
- akceptowane protokoły: X.25, X.32, X.3/X.28/X.29;
- kody znakowe: ASCII, EBCDIC;
- interface'y liniowe: RS232, X.21/V.11, V.35.

Centrum zarządzania

Bardzo ważną rolę w sieci Kolpak spełnia centrum zarządzania. Jego zadania można podzielić na pięć głównych funkcji:

- **zmiany konfiguracji:**

  Określenie konfiguracji software`owej każdego węzła sieci i zdalne przekazywanie tego oprogramowania do węzła, dokonywanie zmian konfiguracji (np. zmiany parametrów portów) bez przerywania pracy węzła;

- **diagnostyka i usuwanie uszkodzeń:**

  – Prezentacja graficzna stanu sieci z możliwością szczegółowego pokazywania fragmentów sieci, węzłów i ich modułów, śledzenie połączeń wirtualnych w sieci,

  – zgłaszanie meldunków alarmowych w przypadku uszkodzenia elementu, przeciążenia węzła lub łącza, prowadzenie statystyki uszkodzeń;

- **nadzorowanie sprawności usługowej:**

  prowadzenie statystyki obrazującej natężenie ruchu w sieci, obciążenia węzłów i łączy, długość kolejek;

- **taryfikacja:**

  gromadzenie danych niezbędnych do wystawiania rachunków za korzystanie z sieci;

- **zapewnienie bezpieczeństwa danych:**

  – przydzielanie uprawnień do korzystania z określonych zasobów sieci,

  – sprawdzanie, czy abonent żądający połączenia jest uprawniony do tego

  – połączenia i rejestracja żądań nieuprawnio-nych, centralne identyfikowanie użytkowników.

Przedstawione główne funkcje ośrodka zarządzania siecią oznaczają znaczne ułatwienie jej utrzymania. Przewiduje się, że w skali PKP do utrzymania sieci Kolpak będzie zatrudnionych około 90 osób. Z tego około połowa będzie stanowić obsługę ośrodka zarządzania. Węzły sieci pracują bezobsługowo.

W celu zapewnienia wymaganej niezawodności sieci zakupiono dwa ośrodki zarządzania. Platformę hardware'ową stanowi sprzęt DATA GENERAL. Podstawowy ośrodek zarządzania jest umieszczony przy węźle głównym w Centrum Informatyki Kolejnictwa, a ośrodek rezerwowy docelowo znajdzie się w Okręgowym Ośrodku Informatyki w Centr. DOKP (również w Warszawie) przy węźle regionalnym.

System poczty elektronicznej

Centralny system poczty elektronicznej, działający na komputerach TANDEM CLX 820, jest traktowany jako aplikacja technologiczna w sieci Kolpak. Umożliwia on przyjmowanie i nadawanie komunikatów pomiędzy dowolnymi parami następujących użytkowników:

- terminalami inteligentnymi,
- komputerami głównymi i regionalnymi
- dalekopisami,

oraz przyjmowanie i nadawanie komunikatów przez abonentów telefaksowych.

Możliwe jest przekazywanie komunikatów w trybie okólnikowym oraz przekazywanie zbiorów.

Będzie również oferowana usługa EDI w standardach EDIFACT. System poczty elektronicznej jest w stanie obsługiwać w chwili zainstalowania 2000 użytkowników. Istnieje możliwość docelowej rozbudowy do pojemności 20 000 użytkowników.

Połączenie z abonentami telefaksowymi będzie odbywać się przez interfejsy serwera poczty elektronicznej o automatycznej sieci telefonicznej PKP, a z abonentami dalekopisowymi - do automatycznej sieci telegraficznej.

Użytkownik ma do dyspozycji menu w językach polskim lub angielskim. Adresy są interpretowane według polskich liter. W systemie są dostępne zestawy znaków polskich i angielskich oraz jest możliwość wprowadzenia, w razie potrzeby, dodatkowych zestawów znaków.

Oprócz centralnego systemu poczty elektronicznej Sprint Mail na sieci Kolpak jest eksploatowany także system Microsoft Mail.

Łącza

Jak wspomniano, PKP mają własną sieć kabli telekomunikacyjnych, która może być użyta do tworzenia kanałów transmisji danych. W wyniku dotychczasowej eksploatacji stwierdzono, że bez specjalnego przygotowania, w istniejących analogowych kanałach telefonicznych możliwa jest szybkość transmisji 2,4 oraz 4,8, a niekiedy 9,6 kbit/s. Po dokonaniu odpowiednich korekcji parametrów łączy z reguły możliwa jest szybkość 9,6 oraz 14,4 i 28,8 kbit/s w wybranych kanałach. Z tego względu sieć Kolpak korzysta przede wszystkim z łączy telekomunikacyjnych PKP. Nie wyklucza się przy tym możliwości korzystania z łączy TP S.A. lub będących własnością innych przedsiębiorstw, z łączy radioliniowych oraz łączy satelitarnych, gdy wystąpi taka potrzeba i będzie to ekonomicznie uzasadnione.

W sieci Kolpak dla połączeń międzywęzłowych i przyłączy abonenckich są wykorzystywane przeważnie analogowe łącza trwałe typu telefonicznego. Podstawowa szybkość transmisji wynosi 14,4 kbit/s. W relacjach silnie obciążonych będzie stosowana szybkość 28,8 kbit/s oraz wiązki łączy.

W relacjach, w których są do dyspozycji łącza światłowodowe, są one wykorzystywane, a szybkość transmisji wynosi 64 kbit/s i będzie wynosiła 2,048 Mbit/s początkowo w technice "frame relay" (szybka transmisja danych według uproszczonego protokołu na łączach cyfrowych wysokiej jakości).

W dalszej przyszłości planowane jest zastosowanie urządzeń ATM (Asynchronous Transfer Mode), integrujących przekazywanie informacji (głos obraz, dane) z szybkościami 34, 155, 622 Mbit/s. W zakresie transmisji danych wymienione szybkie połączenia będą stosowane do przenoszenia dużych ładunków informacji pomiędzy węzłami głównymi sieci Kolpak, komputerami obszarowymi SKPZ, sieciami lokalnymi (LAN) instalowanymi w dużych ośrodkach administracyjnych PKP (np. Dyrekcja Generalna, dyrekcje okręgowe).

Przy połączeniach międzywęzłowych będzie także możliwość korzystania z łączy komutowanych, głównie w sytuacjach awaryjnych.

Przyłącza do komputerów głównych PKP oraz do komputerów obszarowych systemu SKPZ będą tworzone na łączach o przepustowości początkowo nie mniejszej niż 64 kbit/s. Docelowo planowana jest przepustowość 2 Mbit/s. Przyłącza trwałe dla terminali będą miały przepustowość 28,8 oraz 14,4 kbit/s

Modemy

Dla połączeń międzywęzłowych w sieci Kolpak jest zainstalowanych 1100 modemów Penril Datalink 14.4 o symbolu DLX V.32 M (RACAL RM 3222) zgodnych z zaleceniami V.32/V32 bis CCITT.

Modemy te umożliwiają tworzenie pętli lokalnych i zdalnych do testowania, zapewniają one wykonanie rozszerzonego zestawu rozkazów Hayesa, zmieniają szybkość transmisji w określonych zakresach, umożliwiają zwrotne wywołanie na łączach komutowanych, zapewniają korekcję błędów i

kompresję danych. Modemy międzywęzłowe zgłaszają swój status i status łącza do ośrodka zarządzania siecią.

Modemy dla przyłączy abonenckich są traktowane odrębnie. Dla systemu SKPZ ogłoszono przetarg, którego rozstrzygnięcie planowane w czerwcu br.

Przy udostępnianiu użytkownikom zasobów sieci Kolpak bezwzględnie będzie wymagać się, aby używane przez nich modemy miały homologację Ministerstwa Łączności RP. Dla pierwszych użytkowników kolejowych - systemu rezerwacji miejsc i systemu gospodarki materiałowej PKP - zakupiono odpowiednio modemy Penril DLX V. 32 M i modemy Swedex 2324 MP. Przewiduje się stosowanie modemów radiowych działających w paśmie 450 MHz.

Połączenie z innymi sieciami X.25

Połączenie z innymi sieciami transmisji danych (sieci klientów kolei, sieć pocztowa, sieci innych zarządów kolejowych) będzie zapewnione zgodnie z procedurami podanymi w zaleceniu X.75 CCITT lub przez gateway (interfejs) z zamianą adresów. Możliwość połączenia z systemami informatycznymi innych zarządów kolejowych będzie istniała także przez sieć EURADAT i sieć HERMES plus.

W marcu 1994 r. zostało zrealizowane połączenie z siecią transmisji danych kolei niemieckich. Jest to wstępny krok do włączenia sieci Kolpak w międzynarodową sieć kolei europejskich EURADAT/HERMES plus.

Przebieg realizacji

Wdrażaniem sieci Kolpak w Centrum Informatyki Kolejnictwa organizacyjnie zajmuje się Zakład Sieci Kolpak, w którym działają dwa piony: dział rozwoju i eksploatacji sieci X.25 oraz dział systemu poczty elektronicznej. Wdrażanie odbywa się przy ścisłej współpracy i bardzo dużym udziale służby automatyki i telekomunikacji PKP.

Instalowanie węzłów zakończono w październiku ub. r. W sieci Kolpak jest już eksploatowanych ponad 700 terminali (w tym około 300 terminali systemu rezerwacji miejsc i sprzedaży biletów KURS '90).

System poczty elektronicznej, jest jeszcze w trakcie uruchamiania. Obecnie działa system poczty centralnej wykorzystujący komputer TANDEM. Wykorzystywanych jest już około 400 skrytek - dla systemu gospodarki materiałowej i dla celów administracyjnych w służbie informatyki. Jest to traktowane jako testowanie systemu. Uruchomiony jest także moduł X.400 (współpraca z innymi systemami poczty elektronicznej).

Pozostałe moduły systemu poczty elektronicznej są w trakcie rozpoznawania:

- Remote User Agent (usługa poczty elektronicznej z wykorzystaniem terminali mikro-komputerowych, istnieje wersja angielska - wersja polska jest w trakcie opracowywania),
- EDIFACT (elektroniczne przekazywanie dokumentów),
- Global Sprint Fax (przyjmowanie i wysyłanie informacji telefaksowych),

Niezawodność sieci

Dotychczasowe doświadczenia eksploatacyjne wskazują, że sprzęt zainstalowany w sieci (węzły i modemy) jest niezawodny. Niezawodność połączeń jest zapewniona dzięki licznym drogom obejściowym oraz starannemu utrzymaniu traktów teletransmisyjnych.

Wszystkie węzły mają zapewnione podtrzymanie zasilania na czas co najmniej 25 minut przez zasilacze bezprzerwowe (UPS) z zerowym czasem przełączania (system on-line). Dla węzłów lokalnych zastosowano zasilacze typu UPS-1000 firmy MEDCOM, a dla węzłów regionalnych i głównych zasilacze DP 105 i DP 110 firmy SILCON.

Ze względu na znaczenie węzła w strukturze sieci i znaczenie dla użytkowników wprowadzono trzy kategorie zasilania. Do **pierwszej kategorii** należą wszystkie węzły główne oraz wybrane węzły regionalne. Mają one zapewnione:

a) zasilanie podstawowe (wydzielona linia zasilająca),

b) zasilanie rezerwowe z innej stacji energetycznej, z układem SZR wybierającym czynną sieć,

c) zasilanie awaryjne w postaci zespołu prądotwórczego z układem samoczynnego startu i o parametrach gwarantujących prawidłową współpracę z UPS.

Do **drugiej kategorii** należą pozostałe węzły regionalne i wybrane węzły lokalne. Mają one zapewnione zasilanie a), lecz z rozdzielni głównej obiektu i b) lub c).

Do **trzeciej kategorii** należą pozostałe węzły lokalne, które mają zasilanie a) według kategorii drugiej.

Stan prawny sieci i możliwości przyłączania abonentów niekolejowych

Decyzją ministra łączności z dnia 7 kwietnia 1994 r. sieć Kolpak uzyskała uprawnienia i obowiązki publicznej sieci pakietowej z wyróżnikiem międzynarodowym (DNIC) 2606.

Sieć jest jeszcze w trakcie budowy i obecni użytkownicy zewnętrzni mają świadomość, że eksploatacja ma charakter testów technicznych. Uzyskanie pełnej zdolności eksploatacyjnej jest przewidziane w II połowie 1995 r.

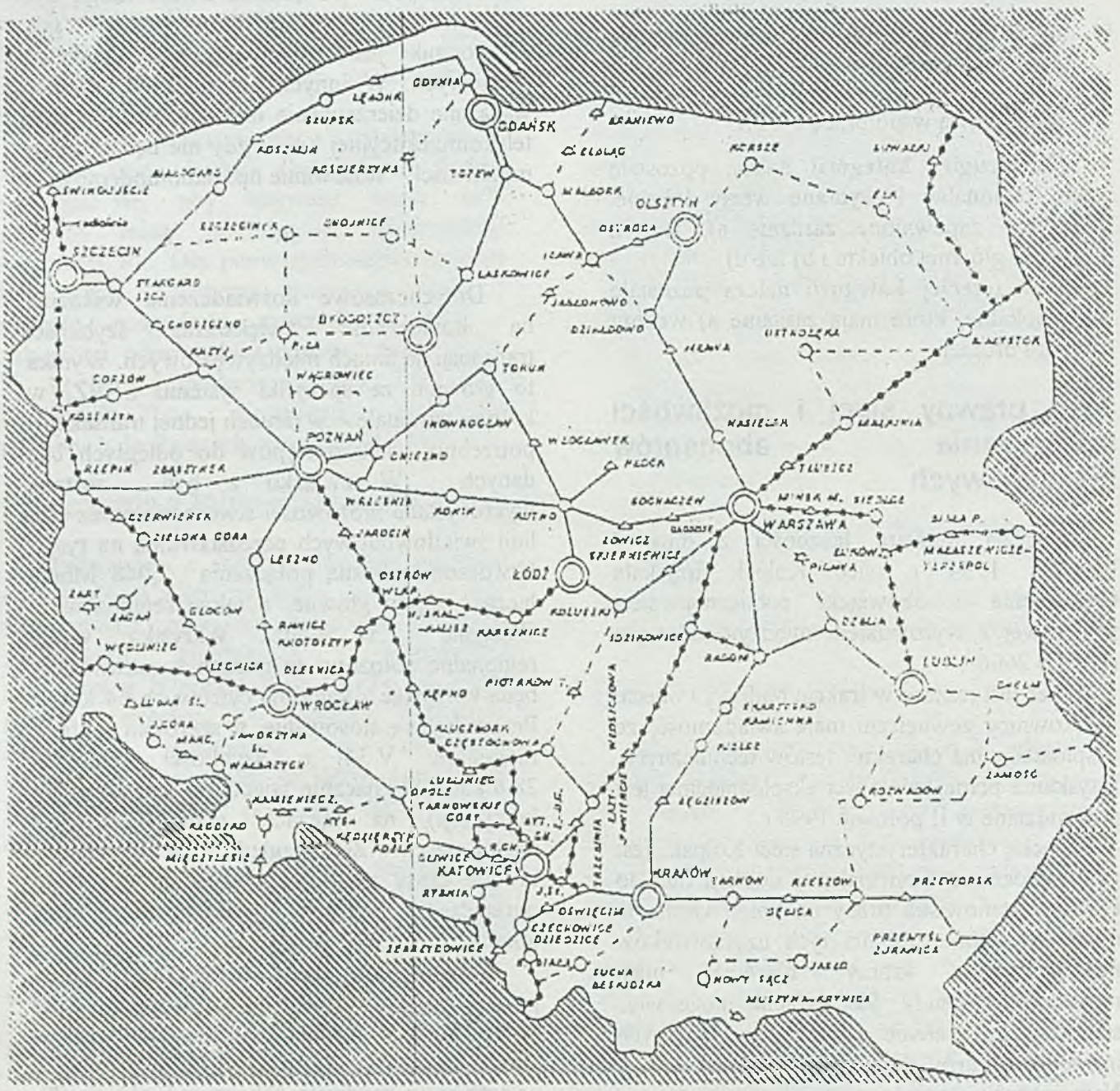
Cechą charakterystyczną sieci Kolpak jest jej rozproszenie terytorialne - dociera ona do każdego stanowiska pracy na kolei. Cecha ta może być atrakcyjna dla tych użytkowników niekolejowych, którzy również mają rozproszone agendy. Sieć Kolpak może więc stanowić uzupełnienie oferty transmisji danych innych operatorów działających w kraju.

Najprostszym sposobem przyłączenia się, w przypadku niedużych strumieni danych i wymaganej stosunkowo niskiej szybkości transmisji, jest korzystanie z portów komutowanych w kolejowych centralach telefonicznych. Połączenia trwałe mogą być realizowane stosunkowo łatwo, gdy użytkownik jest usytuowany w obiektach kolejowych. W innych przypadkach będzie wskazane dzierżawienie linii z publicznej sieci telekomunikacyjnej lub - gdy nie będzie takiej możliwości - stosowanie np. radiomodemów.

★ ★ ★

Dotychczasowe doświadczenia wskazują na konieczność zwiększenia szybkości transmisji na liniach międzywęzłowych. Wynika to głównie ze specyfiki systemu SKPZ, w którym terminale - w ramach jednej transakcji - potrzebują wielu dostępów do odległych baz danych. W związku z tym zostaną wykorzystane możliwości stworzone przez sieć linii światłowodowych przedstawioną na rys.1. Utworzone zostaną połączenia 2,048 Mbit/s łączące węzły główne, a także regionalne w Olsztynie i w Łodzi. Wszystkie węzły regionalne położone na trasach światłowodów będą korzystać z kanałów cyfrowych 64 kbit/s. Przewiduje się stosowanie w szerokim zakresie modemów V.34 o szybkości transmisji 28.8 kbit/s (i znacznie więcej po zastosowaniu kompresji) na łączach międzywęzłowych analogowych. Ze względu na wymagane krótkie czasy transmisji będzie się dążyć do ograniczania tranzytu ruchu przez węzły lokalne.

Planuje się także zwiększenie liczby portów synchronicznych w węzłach oraz doprowadzenie zasilania wszystkich węzłów lokalnych do drugiej kategorii (według podziału podanego w rozdziale "Niezawodność sieci").



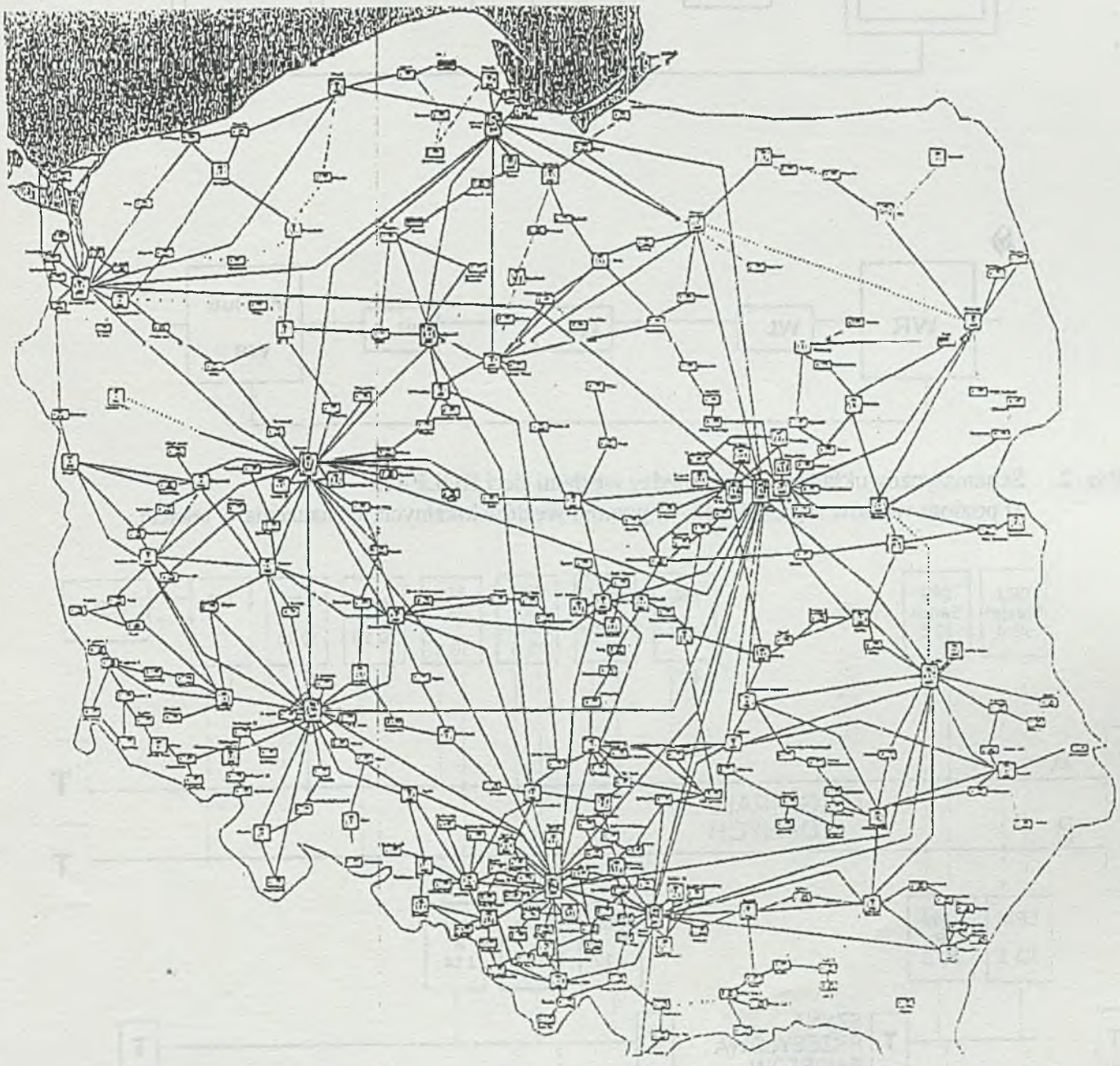
Rys. 1 Przewidywana budowa kablowych linii światłowodowych w sieci telekomunikacyjnej PKP.

Oznaczenia:

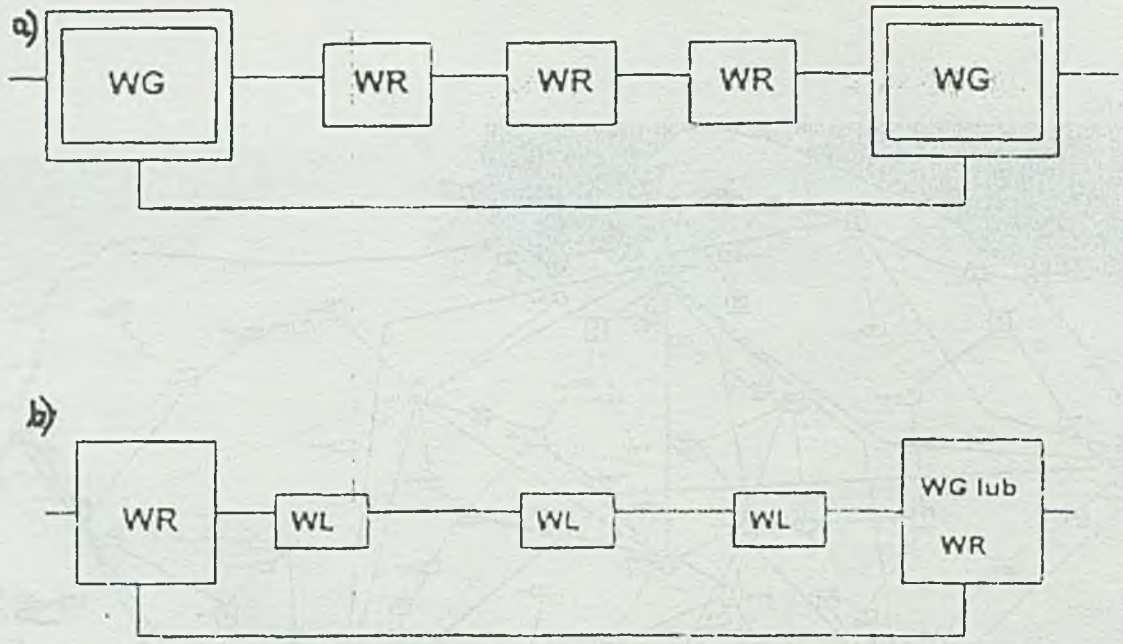
——— kablowe linie światłowodowe przewidziane do zbudowania do końca 1996 roku

..... kablowe linie światłowodowe przewidziane do zbudowania w latach 1997-2000

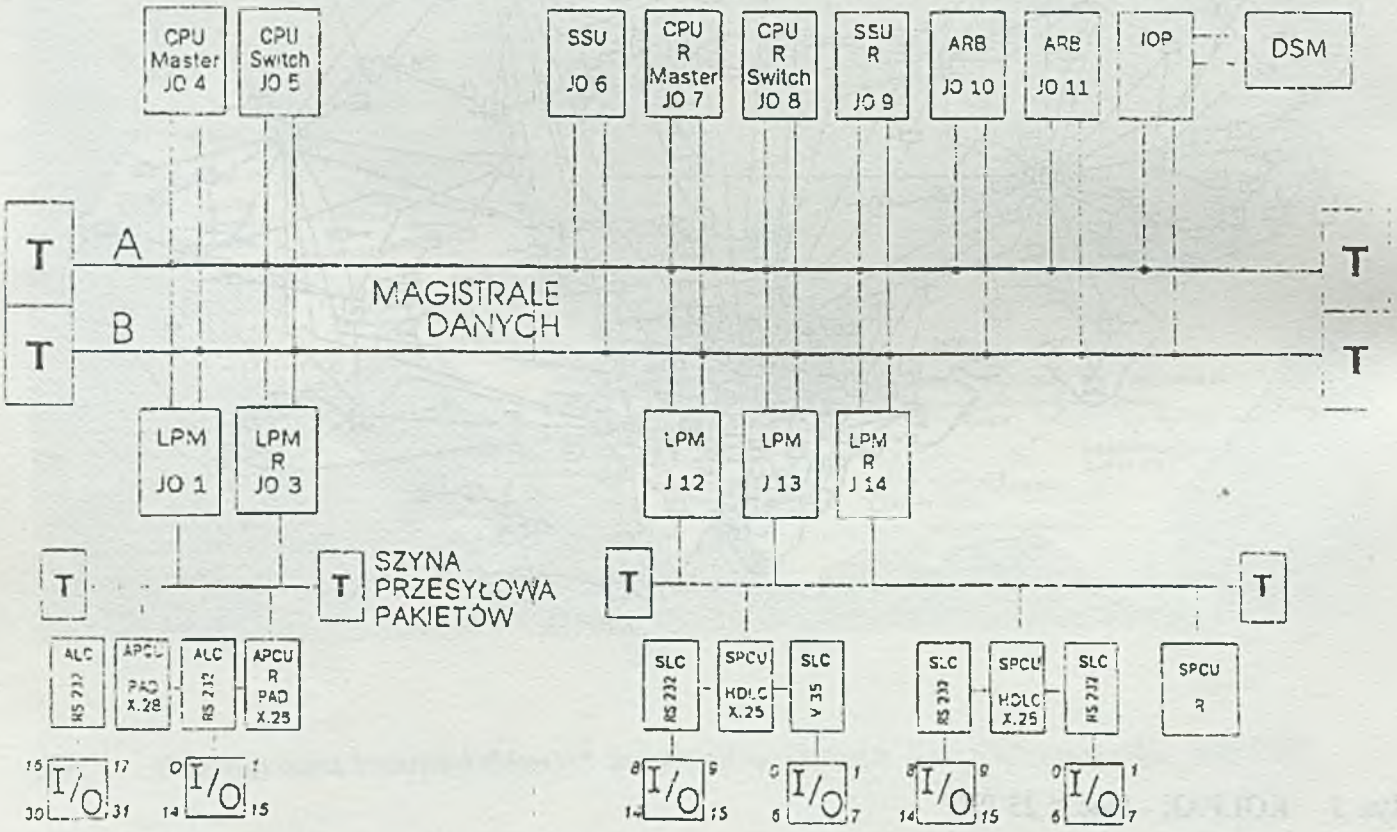
------ kablowe linie światłowodowe przewidziane do zbudowania w latach 2001-2004



Rys. 3 KOLPAK - Sieć X.25 PKP

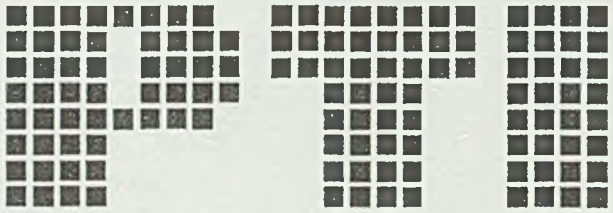


Rys. 2. Schematyczny układ połączeń między węzłami sieci KOLPAK:
a) poziom węzłów regionalnych, b) poziom węzłów lokalnych. Oznaczenia w tekście.



Rys. 4 Schemat blokowy węzła TP 4944.

Oznaczenia: CPU Master - procesor główny węzła, CPU Switch - procesor transmisyjny, SSU - pamięć wspólna węzła, ARB - arbitrator magistral danych, IOP - kontroler I/O, DSM - moduł twardego dysku, LPM - karta obsługi linii, APCU - moduł kontroli protokołu asynchronicznego, SPCU - moduł kontroli protokołu synchronicznego, ALC - moduł konwersji sygnałów protokołu asynchronicznego, SLC - moduł konwersji sygnałów protokołu synchronicznego, T - terminatory szyn, I/O - płytki z gniazdami portów, R - moduł zapasowy, Jxx - lokalizacja modułu w slocie xx kasety.



Wydawca:

Polskie Towarzystwo Informatyczne
Oddział Górnośląski
40-014 Katowice, ul. Mariacka 6
Tel./fax 0-3/153-81-02

Druk i oprawa:

Zakłady Mechaniczne "Bumar-Łabędy" S.A.
Centrum Informatyki
44-109 Gliwice, ul. Mechaników 9
Tel. 0-3/134-61-50 Fax 0-3/134-61-40

Redakcja i przygotowanie do druku:

Maria Rakowska, Czesław Uramowski

Projekt okładki:

Mirosława Minich